Organized by

Real Academia de Ingeniería

with the collaboration of
- Data Science and Engineering Consortium
- Project CASI-CAM-CM, Comunidad de Madrid

- DAMA Network, Ministerio de Economía y Competitividad
FOREWORD

The volume of registered data increases around 100% each year, and this produces more and more interest in Data Science and Engineering. Besides of advances to process massive data, the Big Data technologies, Machine Learning is providing much sophisticated algorithms to give answers to relevant questions, and Analytics offering insights into the interpretation of the problems that are being faced. Applications cover all the spectrum of activities, from Biology to Robotics, including Business, Entertainment, Energy, and Health, to mention a few.

The Spain Real Academia de Ingeniería is aware of this fast evolution, and, consequently, accepted with pleasure the suggestion of organizing a workshop on the subject made by three active clusters of research teams – The Data Science and Engineering Consortium, the CASI-CAM-CM project, and the DAMA network.

The workshop was celebrated at the premises of the Real Academia from June 14th to June 17th, 2016. Its sessions focused on two conceptual areas that are producing many research contributions, Deep Learning and Singular Problems –those opposing difficulties to conventional Machine Learning methods,— as well as on several selected real applications.

These proceedings include the available written versions of the invited lectures and contributions that were presented at the workshop. As the coordinator of that event, I declare my gratitude to the authors and to the editors, Dr. Harold Molina-Bulla and Irene Cordoba. I am sure that the result will be appreciated by all the readers.

Aníbal R. Figueiras Vidal
Academician, RAIing
Full Professor, UC3M
A Concise Tutorial on Deep Learning
Aníbal R. Figueiras-Vidal
GAMMA-L+/DTSC, Univ. Carlos III de Madrid, Spain

Abstract—A very brief overview of motivation and difficulties, origins, evolution, applications and present challenges and trends of Deep Learning is presented here. A carefully selected bibliography try to compensate the extreme concision of these pages.

Introduction

These pages present a very brief tutorial of Deep Learning (DL), or Deep Neural Networks (DNNs). Time constraints have imposed concision. Consequently, the only reasonable possibility is trying to get the interest of the readers. Yet this interest cannot be frustrated because the lack of enough information. So, the first is to mention more extensive tutorials: [1] [2] [3] [4] [5] serve for this purpose.

Contents of this paper are as follows. Section I refers to the motivation and the origins of DL. Its evolution is described in Section II under a structured perspective. The recently proposed algorithms and modifications that make possible a direct training of DL machines are visited in Section III. Section IV mentions some successful applications. Present trends and challenges appear in Section V. And the Conclusion is just a suggestion –or, to be precise, a recommendation.

I. Motivation and Origins

It is well known that shallow (one hidden layer) Multi-Layer Perceptrons (MLPs) have unlimited expressive capabilities, i.e., they can theoretically establish any correspondence between their inputs and their outputs. But, in the practice, the limited number of available training samples does not allow to reach this. So, a question naturally emerges: Do more layers contribute to improve shallow MLP performance? This remained without an answer along many years, because saturations of the MLP units make ineffective the application of the Back Propagation (BP) algorithm: Derivatives went out of control.

Yet some alternative designs early appeared. The first was the Group Method of Data Handling (GMDH) [6] [7], in which regression high-degree multi-variable (suboptimal) polynomials are built by constructing iteratively degree two, two-variable polynomials and selecting the “best” of them for the next iteration. Fukushima’s Neo-Cognitron [8] [9] was hand-tuned several layers architectures that weighted locally the signal under study, i.e., they were Convolutional Neural Nets (CNNs). The restricted form of CNNs make possible to train them with BP without great difficulties, as LeCun and his colleagues demonstrated [10] [11] [12].

CNNs were the only practical DL architecture for classification until the 2000s. Their structure was appropriate for some kind of problems, such as time series or images, but not necessarily for other cases.

II. The Evolution of DL

The next step in the progress of DL was the introduction of Representation Learning (RL): Layers are designed to solve a problem different from that to be addressed, and a shallow machine is put on the top representation layer and trained (usually, refining the pre-trained weights) for solving the real problem.

Although the first suggestion for applying the RL considered Auto-Encoding as the auxiliary task [13], the difficulties in the practical implementation of this idea –mainly due to the use of contractive layers– gave advantage to the proposals that are based on Boltzmann Machines (BMs) [14] [15], that represent probabilities as exponentials of “energy” functions of the NN weights, minimizing this energy by means of statistical sampling methods. The simplified BM architecture called Restricted BM (RBM) [16] originated the much celebrated Deep Belief Networks (DBNs) and computationally efficient algorithms to train them [17] [18].

On the other hand, the technique of noise learning (NL), born in [19], has allowed to construct deep expansive AEs, called (Stacked) Denoising AEs ((S)DAEs) [20] [21]. DAEs show similar characteristics to those of DBNs.

Both DBNs and DAEs constitute the family of RL machines. [22] is an excellent overview of them. It must be remarked that:

– RL machines extract, layer after layer, more and more high-level features of the training set, that can help to understand the key aspects of the problem under analysis and that can be useful for some concrete applications;

– It seems that RL produces a sort of “disentangling” of the sub-spaces in which samples of different classes lie, thus greatly simplifying classification problems.

In parallel to RL, CNNs have been much improved, and a new simplified training of a deep architecture has been proposed: The Deep Stacking Networks (DSNs) [23] [24].
Their conception is as follows: A first (shallow) MLP is trained, its output is added to the input variables to train a second MLP, whose output, together with that of the first machine, extends the inputs for training a third MLP, and so on. There are a number of different structures and algorithms to build DSNs, that can be found in [3], as well as details of several successful real world applications.

III. DIRECT TRAINING OF DNNs

The last years have seen the appearance of a series of algorithms—or tricks— that allow to directly train DNNs—including recursive architectures, DRNNs, that open an avenue to deal with non-stationary problems.–

This is the case of:

- Principled initialization methods, that take into account the variance of the different variables in the DNN architecture [25];
- Hessian-free optimization algorithms [26];
- Replacing sigmoidal units by one-sided saturations, thus alleviating the paralysis effects [27];
- Stochastic Gradient Descent (SGD), which works with mini-batches of training samples—thus reducing the effects of vanishing derivatives,—and that include adaptive [28] and momentum [29] versions;
- Batch normalization [30], which tries to further control the difficulties with the derivatives including additional location and scale parameters.

Further advances can be expected in a very near future; a part of them from combining some of these procedures.

IV. ON DL PRACTICAL APPLICATIONS

DL has demonstrated its usefulness from the very beginning: Ivakhnenko’s GMDH was successfully applied to several practical regression problems. And real world applications are so many that it is completely impossible to include here even a balanced brief review: There are cases in:

- Banking and finance
- Bioinformatics
- Business intelligence
- Energy
- Games
- Health
- Production industry
- Robotics
- Security and safety
- Signal processing
- Smart society

and this enumeration is by no means exhaustive. Then, before addressing some relevant examples, it is fair to mention [31], a structured recent bibliography in which the reader can find information on many practical applications.

Smart society applications are particularly relevant, because they provide new and socially important services: Activity detection and related tasks, emotion/sentiment analysis, collaborative filtering, recommendation systems, and more. But signal processing is probably the area in which more surprising and satisfactory results have been obtained: For example, in object recognition in images, with even superhuman performance.

Similar terms correspond to game playing: AlphaGo [32] is a very recent and famous example. Health is probably the field in which more hope is put on data science and engineering, and, although the processes need a delicate development, there are significative advances in diagnosis and prognosis, as well as in different aspects of management.

V. CHALLENGES AND TRENDS

There is a serious demand of a deeper knowledge and a better understanding of DL characteristics. First of all, more analytical visions are needed. [33] [34] established the unlimited expressive capacity of shallow MLPs, but there are only partial results for DL [35] [36] [37]. Criteria and rules to select architectures are lacking too, although there is a clear trend to deeper structures [38]. And more study must be dedicated to appreciate and solve DL limitations. For example, overfitting has been controlled by means of random eliminations during training, drop-out [39] and drop-connect [40], but it seems clear that qualitatively these are regularization procedures, and many others can be checked. In this sense, RL architectures offer the advantage of reducing the overfitting issues to the top classifier.

A second example is the fooling effects in DL machines [41]. Some of them are clearly due to the global approximation capacities of the MLP architectures, that make them—and, consequently, DNNs—sensitive to extrapolation errors. Again, RL designs can face this limitation working with the final classifier.

There is a general agreement on the interest of extending DL architectures for reinforcement learning formulations [42]. Another relatively recent trend is the design of Deep Recurrent NNs (DRNNs) [43] [44] [45] [46] [47], because the dynamic capabilities of recurrent algorithms. It can be said that DRNNs is a instance of a wider trend: To combine Deep, Diverse, Dynamic, and Distributed Learning (D4L).

Surprisingly, the possibility of obtaining performance improvements by combining Deep and Diverse Learning (D2L)
has deserved a moderate attention. Citations of a few published works that deal with this issue appear in [48]. Among them, the contributions of Schmidhuber and his colleagues in diversifying CNNs by means of different sizing of inputs or distortions of the samples [49] [50] offer record performances, and also [38] [51], that deal with diversifying CNNs in more conventional manners, a difficult task because the intrinsic instability of their basic elements (the convolutional branches), merit mention. Things are different –easier– when working with RL machines, as it is shown in [48] where binarization and label flipping are successfully applied to SDAEs, and in [52] [53], where pre-emphasis procedures are also applied. The performance of the best of these designs in solving MNIST and related problems are almost equivalent to those of the Multi-Column CNN [50], whose architecture is more appropriate for this kind of tasks.

Dynamic DL is a big challenge, because on-line adaptation is even more difficult than for shallow learners [54]. Maybe to adopt Extreme Learning Machine (ELM) based designs [55] can help, because ELMs are easy and quickly trainable. And there are not many works on distributed DL: [56] is an exception.

To close this section, a suggestion: DL machines can contribute –in combination with specific techniques [57] [58] [59]– to better solving a family of singular (imbalanced, cost-sensitive, and sample-dependent cost) problems that are very important because they frequently appear in a number of application fields: Alarms, diagnostic systems, finance, novelty and trend detection, business, and others.

VI. CONCLUDING REMARKS

From the above, it can be concluded that DL is a principal tool for data science and engineering, and, at the same time, that it is at its early development stages: Therefore, that much more work is needed. So, the advise of a senior researcher, specially for younger investigators, has to be: Evaluate the possibility of dedicating effort to this subject; in other case, do not forget DL machines as powerful instruments you can apply in your research and applications.

ACKNOWLEDGMENT

The good aspects of this paper come from my interaction with my colleagues José R. Dorronsoro, Alberto Suárez, Emilio Soria, and José-Luis Sancho-Gómez, my coworkers Marcelino Lázaro-Teja, Luis A. Azpícueta-Ruiz, Lorena Álvarez-Pérez, and Harold Molina-Bulla, and my graduate students Ricardo F. Alvear-Sandoval, Manuel Montoya-Catalá, Simón Roca-Sotelo, and Anas Ahammad. Defects and mistakes are mine.

This work has been partly supported by grant CASI-CAM-CM (S2013/ICE-2845), DGUI-Comunidad de Madrid, and grants DAMA (TIN2015-70308-REDT), and Macro-ADOBE (TEC 2015-67719-P), both from the Spain Ministry of Economy and Competitiveness.

REFERENCES

Exploring Convolutional Neural Network Image Embeddings

Dario Garcia-Gasulla
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center
SPAIN
Email: dariog@cs.upc.edu

Jonatan Moreno
Barcelona Supercomputing Center
SPAIN
Email: jonatan.moreno@bsc.es

Javier Béjar
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center
SPAIN
Email: bejar@cs.upc.edu

Ulises Cortés
Universitat Politècnica de Catalunya
Barcelona Supercomputing Center
SPAIN
Email: ia@cs.upc.edu

Abstract—Convolutional Neural Networks (CNN) are the most popular of deep network models due to their applicability and success in image processing. Although plenty of effort has been made in designing and training better discriminative CNNs, little is yet known about the internal features these models learn. Questions like, what specific knowledge is coded within CNN layers, and how can it be used for other purposes besides discrimination, remain to be answered. To advance in the resolution of these questions, in this work we extract features from CNN layers, building vector representations from CNN activations. The resultant vector embedding is used to represent first images and then known image classes. On those representations we perform several experiments: an unsupervised clustering process, an analysis of those clusters using the WordNet taxonomy, a study of the correlation between distances among vectors and distances in WordNet, and finally image vector arithmetics. Results indicate there are many hidden semantics within the embedding space to be exploited.

I. INTRODUCTION

Deep learning (DL) networks are representation learning techniques [LeCun et al., 2015], which build a very rich descriptive language by processing high-dimensional data. In the context of image data, Convolutional Neural networks (CNN) have shown remarkable performance, approaching human-level on object detection, segmentation and classification. Nevertheless, little is yet known about the nature of the models learnt by CNN, or how to exploit them. In this paper we seek to provide insight into this topic, by exploring the vector embedding space that originates from internal CNN activations.

II. RELATED WORK

Features learnt by CNNs have been explored through visualization techniques (see for instance [Zeiler and Fergus, 2014]). The pattern that a particular feature is encoding can be shown through such methods, for example by showing the input parameters which maximize its activation [Yosinski et al., 2015]. However, to explore the embedding space as a whole ones requires of more general methods. For example, [Donahue et al., 2013] explore the embeddings defined by layers of neurons found at various depths, and find that top layer features cluster input data based on the indoor or outdoor nature of the image. [Razavian et al., 2014] followed that same approach, using the first fully connected layer of a model to define the embedding space. Using that representation, authors explore attribute detection among other challenges, and find that support vector machines can be used to identify abstract classes such as is male or has glasses.

III. METHODOLOGY

A CNN trained with labelled images learns visual patterns for discriminating those labels. In a deep network there can be millions of those patterns, implemented as activation functions (e.g., ReLU) within the network features. Each feature within a deep network consequently provides a significant piece of visual information for the description of images, even if they are not maximally relevant for their discrimination (only the top layer features are). By considering all feature activation values for a given image, one is fact looking at everything the network sees within the image, as learnt from its training. Any visual semantics captured by the neural model will thus be found in those features values, values that we represent as a vector for their analysis.

The precision and specificity of a vector representation is bounded by the quality and variety of patterns found by the deep network; networks capable of discriminating more image classes with higher precision will provide richer image descriptions. To maximize both descriptive accuracy and detail we used the GoogLeNet architecture [Szegedy et al., 2014], a very deep CNN (22 layers) that won the ILSVRC14 visual recognition challenge [Russakovsky et al., 2015]. We used the pre-trained model available in the Caffe deep learning framework [Jia et al., 2014], trained with 1.2M images of the ImageNet test set for the task of discriminating the 1,000 ImageNet hierarchy categories.
The GoogLeNet model is composed by 9 Inception modules. We capture the output of these 9 modules and build a vector representation with their activation values. When an image is run through the trained network, these 9 different layers combined produce over 1 Million activations, expressing the presence and relevance of as many different visual patterns in the input image. In our vector-building process we treat all composing features as independent variables. Thus, our image high-dimensional, sparse vector representation is composed by over 1M continuous variables.

A. Image Classes and Vector Operations

After obtaining vectors for images, we perform an abstraction step to build vector representations of abstract classes, using the 1,000 classes images are labelled to (in the case of ImageNet). Any alternative labelling could be used. To build an image class vector we combine all the specific image vectors belonging to that class. As a result of this aggregation, we expect to obtain representative values of all variables for each class, reducing the variation found in specific images regarding brightness, context, scale etc.. The aggregated image class vector has the same size as an image vector (roughly 1M variables in the case of the GoogLeNet architecture), and is computed as the arithmetic mean of all images available for that class. At the end of this aggregation process we obtain 1,000 vectors, corresponding to the representations of each of the 1,000 leaf-node categories in the ImageNet hierarchy. After aggregation, class vector values are normalized to unit by layer to balance the evidence provided by each layer on a single class vector. The activations of each layer are normalized using the corresponding euclidean norm of that layer. Qualitatively, a class vector represents a list of the visual features commonly found in an abstract class (e.g., an elephant), as perceived by the deep learning network. The process is depicted in Figure 1.

IV. EXPERIMENTS

A. WordNet Correlation

To evaluate the consistency of the information captured by the proposed embedding space we use the labels of the represented classes. ImageNet labels are mapped to WordNet concepts, thus providing access to the lexical semantics implemented in WordNet. Since vector representations are supposed to capture visual semantics instead, a significant gap between both is to be expected. Nevertheless, WordNet remains the only source of validated knowledge available for evaluation.

Distances among image classes can be computed through WordNet measures, typically using the hypernym/hyponym lexical taxonomy [Pedersen et al., 2004]. At the same time we can compute image class distances in the vector-space, using the previously defined methodology. As a result we have, for every available image class, two sets of similarities with respect to the rest of image classes, similarities that can be reduced to a ranking. Spearman’s ρ provides a measure of correlation between two rankings, and is bounded between -1 and 1, with values close to either -1 or 1 indicating a strong correlation. We obtain a ρ value for every image class, by comparing its lexical and visual rankings. By considering the ρ values of all image classes we obtain a distribution of correlations, which indicates the level of semantic coherency between the WordNet taxonomy and the vector-space as a whole.

We consider six different WordNet distances to maximize consistency: three based on path length between concepts (Path, LCh and WuP) and three corpus-based focused on the specificity of a concept (Res, JCn and Lin) [Pedersen et al., 2004]. Additionally, we use two different corpus for the three corpus-based measures, the Brown Corpus, and the British National Corpus. Figure 2 shows the distribution of correlations between the vector embedding and each of the nine WordNet measures. The ρ values are mostly found between 0.4 and 0.6, indicating a strong correlation in a ranking with 999 elements. These results are consistent for all nine WordNet settings; the average ρ on the distributions
is 0.44 in the worse case (JCn bnc) and 0.49 in the best case (Res Brown & Res bnc). This results indicate that vector representations contain a large amount of semantic information also captured by WordNet. This is a particularly interesting, considering that WordNet does not capture visual semantics such as color pattern, proportion or context.

B. Clusters of Image Classes

To further analyze the semantics captured within the defined vector-space we perform a supervised analysis of clusters, using the WordNet hierarchy as ground truth; by knowing which image classes are hyponyms of the same synset, we can explore their distribution within the embedded space. To achieve visual results, we apply metric multi-dimensional scaling with two dimensions on the 1,000 image classes distance matrix. This method builds a two-dimensional mapping of the vector distances which respects the pairwise original similarities. We first use two synsets with many hyponyms within the ImageNet categories: dog (according to WordNet there are 118 specializations of dog in the image classes) and wheeled vehicle (with 44 specializations of wheeled vehicle in the image classes1). We highlight the location of the image classes belonging to each one of these two sets in the two-dimensional similarity mapping of Figure 3a.

At first sight, the two sets of highlighted images compose definable clusters. Although precision is not perfect, image classes belonging to the same WordNet category are clearly assembled together in the vector-space representation. In the case of dogs, this is relevant because of the wide variety of dogs computed, some of which have few visual features in common (e.g., Chihuahua, Husky, Poodle, Great Dane). According to these results, the visual features which are common on all dogs have more weight on the vector representation than variable features such as size, color or proportion. This is probably caused by the aggregation and normalization process, which reduces the importance within image classes of volatile properties. The cluster defined by wheeled vehicle image classes has a lower precision than that of dogs, probably because wheeled vehicles are more varied than dogs (e.g., Monocycle, Tank, Train). Nevertheless all but one wheeled vehicle are located on the same quadrant of the graph, indicating that there is a large and reliable set of features in the vector representation identifying this type of image classes. The one wheeled vehicle located outside of the middle-left quadrant, in the low-right part of Figure 3a, corresponds to snowmobile, a rather special type of wheeled vehicle which seems to be different to everything.

By looking at Figure 3a we notice a gap naturally splitting image classes into two sets. This separation is the only consistently sparse area visible at first sight in the graph. To explain this phenomenon we explored the most basic categorization in WordNet, separating ImageNet classes between living things, defined by WordNet as a living (or once living) entity and the rest. By painting the images belonging to living things we obtain the graph of Figure 3b. This graph shows how the separation found in the vector-space corresponds to this simple categorization with striking precision, unsupervisedly clustering images depending on whether they depict living things or not. The few mistakes done correspond to organisms with unique shapes and textures (e.g., lobster, baseball player, dragonfly) and things which are often depicted around living things (e.g., snorkel, dog sled). Other particular cases are rather controversial, as coral reef is not a living organism according to WordNet but in the vector-space it is clustered as such.

C. N-Clustering

Since we initially ignore the nature of the embedding space, we consider the existence of a variable number of clusters. Using spectral clustering, we identify k clusters, between 2 and 19, for each distance matrix. Each k-clustering provides an abstraction of image classes, but given the sub-symbolic nature of the features defining those classes (feature activations of a CNN) little can be said about their nature in symbolic terms. To try to characterize those clusters symbolically we use the WordNet hypernym/hyponym lexical taxonomy to identify the WordNet synset that better describes each cluster. In detail, since every image class is mapped to a WordNet synset, we can compute a F1 score for every pair of WordNet synset and cluster, considering that a synset S applies to an image class IC if the synset associated with IC is S or a hyponym of 1To these 44 classes we added the school bus, minibus and trolleybus image classes, which we consider to be wheeled vehicles.
Fig. 3. Scatter plot of image class vector similarities built through metric multi-dimensional scaling. On the left, black circles belong to images of dogs and dark grey circles belong to images of wheeled vehicles. On the right, black circles belong to images of living things.

### TABLE I
Distance matrix among 1,000 class vectors, considering nine different GoogLeNet layers (from inception/3a_output to inception/5b_output). Compute from 2 to 8 clusters through spectral clustering. For each clustering process and cluster identified, Table shows the WordNet synset with the best F1 measure. When relevant, second best synset by F1 measure is shown in parentheses. Clustering experiments are independent and their relation has not been explored.

<table>
<thead>
<tr>
<th></th>
<th>Living thing</th>
<th>Artifact</th>
<th>Conveyance</th>
<th>Artifact (clothing)</th>
<th>Instrumentality</th>
<th>Wheeled vehicle</th>
<th>Clothing</th>
<th>Craft</th>
<th>Structure</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

S. For example, given a 2-clustering, if 90% of image classes which are hyponyms of the dog synset can be found within cluster A, and 85% of image classes of cluster A are hyponyms of dog, the F1 measure of dog for cluster A will be 0.87. We use the synset obtaining the best F1 measure for each cluster as that cluster label. Notice two clusters may share the same label.

To illustrate the type of cluster labelling that we obtain, Table 1 shows the results when using all available layers from the GoogLeNet architecture. Briefly, the first 2-clustering separates between living things and non-living things. Means of transportation (i.e., conveyance) are separated next, at 3-clustering. Later on, these are further separated between crafts (i.e., ships and aircrafts) and wheeled vehicles. A cluster for birds and clothes is also reliably found. On the other hand, there is a cluster of mixed artifacts and instruments on all k-clusterings, likely due to the less coherent visual features of objects, and to the lack of specificity of WordNet synsets for characterizing items.

We evaluate the quality of the discovered sets numerically, using this labelling process. For each k-clustering, we compute its quality as the mean F1 score obtained on all its clusters. Results indicate that clustering quality decreases as the number of clusters increases, as shown in Figure 4. The best results are obtained for the 2-clustering, with an F1 score close to 0.9. To study the relevance of the various layers found in a CNN model, we separately consider the case of using all nine layers defined in §III, using only the 3a and 3b layers (layers found near the bottom of the CNN), using the 4a to 4e layers (layers found at the middle of the CNN), and a using the 5a and 5b
layers (those with maximum discriminative power). Results indicate that top layers (5a and 5b) provide the best clusters, although similar results are obtained when using all layers. Since top layers capture the most complex visual patterns these also provide the most descriptive power. Nevertheless, as shown in Figure 4, middle and low layers also convey a relevant amount of visual semantics, since one can also find high quality clusters using them.

D. Image Equations

To test the operability of the visual semantics captured in the vector-space, we now consider vector arithmetics by subtracting two image class representations. Given the resultant vector of such operation, as defined in the Vector Operations subsection, we then look for the closest image class through cosine similarity to solve equations of the form \(a - b \simeq c\). We start by considering image classes which can be understood as the concatenation (not overlapped) of two other classes. Afterwards we consider overlapped relations where two classes are strongly intertwined to produce a third.

We first consider concatenated images through church and mosque, two image classes located closest to one another in the vector-space. We performed subtraction operations on both directions, and found that the closest class to church - mosque was bell cote, an architecture element used to shelter bells typical of Christian churches and not found on mosques. On the opposite direction, we found the closest class to mosque - church was stupa, a hemispherical structure, often with a thin tower on top, typical of Buddhism. Although stupas do not belong to mosques, mosques are often hemispherical, and include thin towers. Features similar to stupas rarely found on Christian churches. These results can provide the answer to interesting questions such as why is this a church and not a mosque or how can I make this church look like a mosque.

An even clearer example of concatenated images is found in the equation horse cart - sorrel (sorrel is the only class of horse available). Since the wagon or cart classes are not available in ImageNet, the closest result turns out to be rickshaw, a two wheeled passenger cart pulled by one person. Visually speaking, a horse cart without a horse is almost indistinguishable from a rickshaw, more so when a person does not appear on or besides the rickshaw, as it often happens in our data. According to these results, the horse cart vector-representation can be decomposed into two independent vectors, corresponding to its two main composing entities: a horse and a wagon. These results indicate that the subtraction operator could be used analogously to any concatenated class to obtain isolated representations of the elements composing them.

We consider overlapped images to be a mix of more than
one visual entity which cannot be separated through physical cuts. To explore this case we computed the difference between brown bear and ice bear. The closest classes were kuvasz, Maltese dog, Sealyham terrier, white wolf, Old English sheepdog and Arctic fox in this order, all white coated mammals of varying size and proportion. These results indicate that the vector obtained from the ice bear minus brown bear subtraction resembles something similar to white fur entity, as being white is the main difference separating an ice bear from a brown bear, and also the main common feature found on all the classes closest to the subtraction. To further support this hypothesis, we perform a similar test by subtracting brown bear from giant panda. In this case the closest classes to the result were skunk, Angora rabbit, soccer ball and indri (a monkey). Remarkably, all four classes are characterized by having white and black color patterns, while being diverse in many other aspects (e.g., size, shape, texture). Thus, the vector resultant of giant panda - brown bear seems to represent an image class as complex as black and white spotted entity.

To analyze the consistency of the newly build vector black and white spotted entity, we subtract it from those classes which were found close to it: skunk, Angora rabbit, soccer ball and indri. This tests provides the following regularities: Panda is to Brown Bear, as Skunk is to Badger, as Angora Rabbit is to Persian Cat, as Indri is to Howler Monkey, and as Soccer ball is to Crash helmet. Consistently, when the vector of black and white spotted entity is removed from black and white spotted entities, we obtain elements which are close to it in many other aspects (e.g., shape, proportion, texture) but which have a different coloring pattern. This shows the existence of regularities within the vector-space, which can be used to perform multiple arithmetic and reasoning operations based on visual semantics.

ACKNOWLEDGMENT

This research was partially supported by the IBM/BSC Technology Center for Supercomputing (Joint Study Agreement, No. W156463).

REFERENCES


Partial Boosting of Deep Stacked Networks

Manuel Montoya-Catalá, Ricardo F. Alvear-Sandoval, and Aníbal R. Figueiras-Vidal

Abstract—Deep stacking is one of the most effective methods to design deep learners (DLs), and boosting the most celebrated ensemble building principle. Both are sequential procedures that train their new learners according to the outputs of the previously trained elements. This suggests that combining them—i.e., emphasis and injection—in appropriate forms can provide performance improvements.

Here, a first mode for such a combination is proposed. It applies a flexible and general emphasis form and injects to each learner the aggregated output of the previously trained units. This scheme is called Boosted and Aggregated Deep Stacked Networks (B-ADSNs).

Some experiments with a number of well-known benchmark databases serve to conclude that, if carefully designed, B-ADSNs are never worse than ADSNs (DSNs with aggregated injection), and that their performance is better in some cases: The conditions for it are analyzed and described.

Other possible combinations are suggested as future work with the conclusions of this contribution.

Index Terms—Deep learning, Deep Stacking Networks, boosting, aggregation.

I. INTRODUCTION

HERE has been an explosive advance of conceiving, designing, and applying Deep Neural Networks (DNNs) along the last two decades. These Learning Machines (LMs) maintain many classification records, showing superhuman performances in tasks for which persons were considered unbeatable. [1][2] are clear and complete overviews of their principles, origins, evolution and trends.

Deep Stacked Networks (DSNs) are DNNs that sequentially train NNs whose input includes the observations and the outputs of all previously trained units [3][4]. Monographic text [5] describes several DSN structures and algorithms, as well as their successful applications in audio processing, speech recognition, speech synthesis, and natural language processing among, others.

There is a clear parallelism between DNNs and the most celebrated machine ensemble construction algorithms, the widely studied boosting ensembles. The first practical versions of boosting appeared along the 1990s under the names of AdaBoost (AB) [6][7] and RealAdaBoost (RAB) [8][9]. Up to date, a huge number of modifications and extensions have been published, that the interested reader can find in [10].

Designing boosting ensembles consists of sequentially training weak learners paying more attention to those examples that offer more difficulties to be correctly classified by the aggregation of the precedent learners. An emphasis function depending on that difficulty serves to implement such an attentional intensity, as it weights the training objective to be minimized. We remark that the particular form of such an emphasis function has a secondary importance, their advantage depends on the particular problem to solve. In some sense, this is a deep procedure: It is not very different from DSN building, because the emphasis comes from the aggregated output of the previous learning elements. One of the nice surprises that boosting ensembles offer is their remarkable resistance to overfitting, while DSNs require some control.

DSNs are truly deep architectures, using the injection of all the previous outputs to construct each new unit, which does not need to be weak. One can interprete this injection as playing a role similar to the boosting emphasis. For DSNs, the final output is not an aggregation of the individual outputs of all the units but just the output of the last element: This element receives all the previous outputs.

The above analogy suggests an attractive possibility: Can a clever combination of boosting and deep stacking provide advantage? The answer is unclear: Consider, for example, that dealing with databases of small or moderate size will produce difficulties in training the deepest DSN units because the dimension of their inputs—observations plus injected inputs—becomes higher and higher. Of course, there are several alternatives for building such a kind of "combined" DNNs, as well as some critical issues, the main being how much weak or strong the learning units must be, and how to regulate the relative influences of emphasis and injection processes.

Here, we will propose a first method to combine boosting and DNNs, which we will call Boosted and Aggregated Deep Stacked Networks, B-ADSN. We will show some experimental performance results of these new DNNs, analysing also the characteristics of the architectures, that are selected by means of cross validation (CV). We say in advance that we will sequentially aggregate the outputs of the NN units for allowing an easy application of boosting, and that we will work with generalized and highly flexible analytical forms of the emphasis functions in order to control the effects of their combination with the injection mechanism.

The rest of this contribution is as follows. In Section II we present the architecture of the proposed B-ADSN. Section III is dedicated to the emphasis formulas, explaining the reasons for selecting some forms and the characteristics of the corresponding functions. The training procedures are described in Section IV. Section V shows experimental results for a number of traditional benchmark problems and discusses them, while Section VI is a more detailed analysis of convergence and sensitivity issues. The main conclusions of our study and some research directions that emerge from its ideas and experiments, close the paper.

This work has been partially supported by grants S-2013/ICE-2845, CASICAM-CM, Madrid Government, and TIN 2015-70308-REDT, DAMA, Spain Ministry of Economy and Competitiveness.

The authors are with the Signal Theory and Communications Department, University Carlos III of Madrid, Madrid, 28911 Spain (e-mail: mmontoya@tsc.uc3m.es, r.f.alvear.s@gmail.com, arfv@tsc.uc3m.es).
II. THE PROPOSED B-ADSN ARCHITECTURE

It is obvious that aggregating outputs allows the direct application of boosting mechanisms. We use the standard aggregation of RAB because it offers good results, and we normalize layer-by-layer the aggregated value before injecting it to the next layer, to avoid “dominance” effects. So, we have a first learner $l = 0$, and, for layer $l > 0, l = 1, ..., L$, an aggregated output:

$$f_l(x) = \sum_{l'=0}^{l} \gamma_l a_{l'}(x)$$  

where $\{a_l\}$ are the learner unit outputs, and $\{\gamma_l\}$ are the linear combination constants obtained from the weighted margin $\{r_l\}$

$$r_l = \sum_{n=1}^{N} t_n a_l(x(n)) e_l(x(n))$$  

using the RAB aggregation equation

$$\gamma_l = \frac{1}{2} \ln \left( \frac{1 + r_l}{1 - r_l} \right)$$

\{x(n), t_n\}, $n = 1, ..., N$ being the labeled examples and $\{e_l(\cdot)\}$ the emphasis functions that are applied to the observations $\{x(n)\}$ in forms we will see later.

$$\bar{f_l}(x^{(n)}) = f_l(x^{(n)}) / \max_n \{f_l(x^{(n)})\}$$

is the normalized aggregated output which we inject to the $(l + 1)$th B-ADSN learning unit.

The architecture of our B-ADSN is presented in Figure 1.

III. EMPHASIS FUNCTIONS

We need to limit the emphasis intensity in order to control the combination of boosting and outputs’ injection. There is also empirical evidence of the advantage of including a measure of the proximity to the border of each sample in the emphasis function [11][12]. So, we select emphasis functions that include a constant term, a term depending on the example error, and a term which considers proximity to the border. Using convex combinations to integrate these terms permits easy CV processes to select the combination parameters.

So, we will use a minor modification of the hybrid RAB equation [11][12], including the constant term leads to:

$$e_l(x^{(n)}) = \frac{\alpha}{N} + \frac{1 - \alpha}{Z_l} \left[ e_l(x^{(n)}) \right] \cdot \exp \left( \beta (t_n - \overline{f}(x^{(n)}) )^2 - (1 - \beta) \overline{f}^2(x^{(n)}) \right)$$

where $\alpha, \beta, 0 \leq \alpha, \beta \leq 1$ are the convex combination parameters and $Z_l$ a normalization constant for the second term –the sum of its values for all the samples.– $N$ and $Z_l$ serve to balance the magnitudes of the $\alpha$ and $1 - \alpha$ terms. $\beta$ is used to go from an emphasis including only proximity to the border to an emphasis fully based on the squared error.

In order to check if the specific functional form of the emphasis –i.e. the expressions for the error and the proximity term– has relevant effects, we will also employ a minor modification of a similar emphasis function which was successfully applied in [13] for boosting purposes and which was previously proposed as a pre-emphasis mechanism to improve the performance of Denoising Auto-Encoding (DAE) classifiers [14][15][16], with much favorable results. This form is

$$e_l^*(x^{(n)}) = \frac{\alpha}{N} + \frac{1 - \alpha}{Z_l} \left[ \beta (t_n - \overline{f}_{l-1}(x^{(n)}) )^2 / 4 \right. $$

$$\left. + (1 - \beta)(1 - \overline{f}_{l-1}(x^{(n)}) \right]$$

Note that $e_l(x^{(n)})$ (or $e_l^*(x^{(n)})$) is used not only in (3) but also to weight the error when training learner $l$. We will indicate the designs that use (5) and (6) as B1-ADSN and B2-ADSN, respectively. Obviously, $\alpha = 1$ produces an aggregated version of DSN. In a similar manner, $\alpha = 0$ imposes a non-moderated emphasis, whose direct combination with the injection mechanism out of control and can produce unreasonable results. Fixed extreme values of $\beta$ generate particular –less general and less flexible– forms of emphasis, looking only at the proximity to the border or at the error. In the experiments, it will appear that it is important to use general forms (5) or (6). It is also evident that cutting injections will originate boosting ensembles.

IV. TRAINING ALGORITHM

We will use the standard Back-Propagation (BP) algorithm in its sample-by-sample (online) version, weighting the square errors with the corresponding emphasis function. Other training algorithms that have been proposed for DSNs could have been employed, but here it is enough to follow the traditional way because we are just exploring possibilities of combining boosting and stacking.

To avoid the effects of paralysis situations, we follow the recommendations of [17]: The initial values for the weights (including biases) at each layer of the NN units are obtained from a $[-r, r]$ uniform distribution, with

$$r = \frac{\sqrt{6}}{\sqrt{N_{in} + N_{out}}}$$

$N_{in}, N_{out}$ being the number of inputs and output units of the weight layers, respectively. For both layers, the learning step is initially 0.01 and it linearly decreases to 0 with the number of training epochs, $N_{epochs}$, which is obtained by CV. This means that we do not need a stopping criterion. Although for some databases the convergence is very fast, more training epochs do not produce serious overfitting. All the learner units have the same number of hidden neurons, $H$, which is also selected by the CV process.

V. EXPERIMENTS AND THEIR RESULTS

We will consider here results for three well-known benchmark databases: Abalone, Image and Hepatitis [18]. We will
Fig. 1. Architecture of B-ADSN
denote these datasets by their three first letters from now on here. They are representative enough: These three databases have different characteristics: See Table 1.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>#Train</th>
<th>#Test</th>
<th>Dimension</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>C_1 / C_{-1}</td>
<td>C_1 / C_{-1}</td>
<td>(D)</td>
</tr>
<tr>
<td>Abalone</td>
<td>2507</td>
<td>1670</td>
<td>8</td>
</tr>
<tr>
<td></td>
<td>1238 / 1269</td>
<td>843 / 827</td>
<td></td>
</tr>
<tr>
<td>Image</td>
<td>1300</td>
<td>1010</td>
<td>18</td>
</tr>
<tr>
<td></td>
<td>736 / 564</td>
<td>584 / 426</td>
<td></td>
</tr>
<tr>
<td>Hepatitis</td>
<td>93</td>
<td>62</td>
<td>19</td>
</tr>
<tr>
<td></td>
<td>70 / 23</td>
<td>53 / 9</td>
<td></td>
</tr>
</tbody>
</table>

TABLE I
CHARACTERISTICS OF THE BENCHMARK PROBLEMS

The explored values of the non-trainable elements in the CV search were:

- \( H \): from 2 to 20 in unit steps for Hep, from 2 to 30 in 2 unit steps for Aba and Ima
- \( N_{ep} \): 25 50 100 150 200
- \( \alpha, \beta \): from 0 to 1 in 0.1 steps

Since the training set sizes are relatively small, we apply a 10 split \(
\times\) 20-fold CV. Average error rate \( \pm \) standard deviation for the resulting designs are obtained from 30 runs. Table 2 shows the experimental results, as well as the values that the CV selects for the non-trainable parameters, for B1-ADSN, B2-ADSN, ADSN (\( \alpha = 0 \)), and the boosting ensembles B1 (no injection) and B2 (no injection).

There are no relevant performance differences –nor in the values of the CV parameters– between both emphasis functions. We can conclude that the exact measures for emphasizing examples are secondary, a fact that other previous works have found.

All the three –constant, error dependent, and proximity dependent– terms appear in all the CV designs: The structure of the emphasis function, i.e., its flexibility, is important. And the selected values of \( \alpha \) for each problem are different: \( \alpha \) is high for Aba and Ima, but low for Hep. Thus, the appropriate “intensity” of emphasis is problem dependent. The values of \( \beta \) are relatively high in all these cases: But the proximity measure is also present, and we have checked that things become worse when \( \beta \) is fixed to its extreme values.

We must mention that low values of \( \alpha \) seem to reduce overfitting effects. We will return to this issue in the next section. The differences with respect to boosting algorithms are significant for Aba, not for Ima, and pure boosting is as good as B-ADSN for Hep. These results are reasonable, because we are working with small size databases: But note that there is some advantage in some cases.

The advantage with respect to the aggregated (pure) DSN is higher, but we cannot extract any general conclusion: It is plausible that the size of the training sets can limit the effectiveness of ADSN, and standard DSN algorithms are different from ours.

VI. SENSITIVITY AND CONVERGENCE ISSUES

We will first discuss the performance sensitivity of our designs with respect to the parameters of the emphasis function, \( \alpha \) and \( \beta \), and to the size of the hidden layer of the NN learners, \( H \). These are the true non-trainable parameters, and CV is the most usual method to establish their values. Performance varies smoothly with \( \alpha \) and \( \beta \); Figures 2 and 3 show the correct classification rate for Ima and Hep when \( \alpha \) and \( \beta \) take different values (the other design parameters being fixed to their CV values), databases whose \( \alpha \) CV values are clearly different. Note that we reverse the horizontal (parameter) scales for Ima to allow an easier visualization. The vertical scales reveal very low changes with them, and this means that CV is not a critical problem in order to get good designs. It can be observed in Figure 3 that \( \alpha = 0 \) and \( \beta = 0 \) (pure proximity emphasis) are a bad selection, as well as some improvement when \( \alpha \) goes from 0.9 to 1. And we remark also that the value of \( \beta \) is very relevant when \( \alpha \) is far from 1, as for Hep.

The error rates are rather smooth with respect to \( \alpha \) and \( \beta \), for moderate changes from \( \alpha \) and \( \beta \) CV values, although we repeat that there is some discontinuity for the extreme values of \( \alpha \), 0 and 1. This can be expected, because there are substantial changes when \( \alpha \) takes these extreme values: \( \alpha = 1 \) means...
that there is no emphasis at all, while $\alpha = 0$ means that the emphasis is pure, i.e. there is not regulation of the intensity. A practical consequence is that a smaller step size will be needed for CV of $\alpha$ if we visit these value regions.

Regarding the number of hidden neurons at each learner, $H$, it has been observed that the accuracy of the system does not greatly change for a wide range of values of $H$ around its CV selected value, and it changes in a smooth manner.

The accuracy vs $\alpha, \beta$, surfaces barely change in shape with $H$, except for its extreme values.

Let us now discuss the convergence of the training process. Figures 4 and 5 show the evolution of the correct classification rates for the training and the test sets when solving Aba and Hep problems, respectively. $\alpha$, $\beta$, and $H$ take their CV values. For the first database, Aba—which is bigger than Hep—from both the training and the test sets show a parallel evolution of the performance, which continuously increases until reaching saturation. On the contrary, these evolutions are opposite for Hep: The initial accuracy in the training set is high, and clearly decreases as we take their CV values.

<table>
<thead>
<tr>
<th></th>
<th>B1-ADSN</th>
<th>B2-ADSN</th>
<th>ADSN</th>
<th>B1</th>
<th>B2</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\alpha$/$\beta$/N$_{ep}$/H</td>
<td>$\alpha$/$\beta$/N$_{ep}$/H</td>
<td>$\alpha$/$\beta$/N$_{ep}$/H</td>
<td>$\alpha$/$\beta$/N$_{ep}$/H</td>
<td>$\alpha$/$\beta$/N$_{ep}$/H</td>
</tr>
<tr>
<td>aba</td>
<td>18.4 ± 0.2</td>
<td>18.5 ± 0.2</td>
<td>18.6 ± 0.2</td>
<td>19.1 ± 0.1</td>
<td>19.0 ± 0.1</td>
</tr>
<tr>
<td></td>
<td>(0.9,0.5,100,12)</td>
<td>(0.9,0.7,100,14)</td>
<td>(100,16)</td>
<td>(0.9,0.1,100,10)</td>
<td>(0.9,0.0,100,10)</td>
</tr>
<tr>
<td>ima</td>
<td>2.9 ± 0.3</td>
<td>2.9 ± 0.4</td>
<td>3.0 ± 0.3</td>
<td>3.2 ± 0.5</td>
<td>3.2 ± 0.2</td>
</tr>
<tr>
<td></td>
<td>(0.9,0.1,150,30)</td>
<td>(0.9,0.3,150,28)</td>
<td>(150,30)</td>
<td>(0.9,0.9,150,30)</td>
<td>(0.9,0.2,150,30)</td>
</tr>
<tr>
<td>hep</td>
<td>6.6 ± 0.0</td>
<td>6.7 ± 0.4</td>
<td>8.0 ± 0.4</td>
<td>6.6 ± 0.5</td>
<td>6.7 ± 0.5</td>
</tr>
<tr>
<td></td>
<td>(0.2,0.8,25,4)</td>
<td>(0.2,0.9,25,4)</td>
<td>(25,7)</td>
<td>(0.3,0.7,50,5)</td>
<td>(0.4,1,50,5)</td>
</tr>
</tbody>
</table>

% AVERAGE ERROR RATE ± STANDARD DEVIATION FOR THE CONSIDERED ARCHITECTURES - DEEP LEARNING ARCHITECTURES B1-ADSN, B2-ADSN including the restricted version ADSN, and the boosting ensembles B1 and B2. CV VALUES FOR $\alpha$, $\beta$, N$_{ep}$ AND H ARE ALSO INCLUDED.
Abalone is difficult (and big) enough to offer moderate quality results when using the first learning unit. This does not create generalization difficulties, because overfitting does not appear. Performance increases progressively, and, qualitatively, things remain as at the beginning, without overfitting effects. The CV degree of emphasis (do not forget that pure emphasis forms, such as those used in traditional boosting ensembles do not allow boosting control) is low: $\alpha = 0.9$ is the selected value for the constant component of the emphasis function.

Hep is an easier problem: The first learning unit can be strong enough to generate overfitting. And, then, to apply a relatively high degree of emphasis serves to reduce overfitting, increasing the correct classification rate for the test set: Note that the CV selected value for $\alpha$ is 0.2.

The above discussion is a first interpretation of why combining boosting and injection can be beneficial, but we recognize that the evidence is not enough to accept it unconditionally: More experimental work is needed. We remark also that the evolution of the convergence curves can be different for other values of $H$ –for some of these values, the optimal performance can appear earlier,– but we are considering just the best (CV) designs –so, the importance of these different forms of convergence is secondary.

More work is also needed for evaluating the effects of the training algorithm and parameters: We have applied BP with a robust initialization and a moderate search step, but there are many alternatives, both general and for DSN designs. We have also avoided the stopping difficulties by applying CV to the number of epochs: Of course it cannot be said that other procedures do not give better results. However, we must say that the saturation of $\alpha$ seems to be not a reasonable criterion: In some cases the role of boosting is minor, but, when it is relevant, performance improvements continue after the saturation –probably because boosting is fighting against overfitting.

VII. CONCLUSIONS AND FUTURE WORK

We have introduced an architecture whose conventional training serves to check experimentally that combining a flexible enough boosting algorithm with an aggregated form of Deep Stacked Networks can give performance advantages in solving moderate size classification problems. For this purpose, it is essential that the boosting emphasis includes a constant term –i.e. to have the possibility of regulating the influence of emphasis and stacking injection– which seems to be a parameter that can moderate the occasional propension to overfitting of these machines.

To improve the training algorithms, and also to explore these combinations in other contexts, such as with databases of bigger sizes, requires additional work. The potential role of boosting to limit overfitting effects is important enough to justify these research efforts.

We will finish saying that by no means the architecture we have introduced here is the only option to combine emphasis and stacking by using output(s) injection. As two examples of interest, let us mention, first, to inject fractions of all previous output(s) directly to the output activation of each learning unit. Second, to apply a unit-by-unit pre-emphasis to standard DSN forms -i.e., those that inject all the previous outputs. Both of them merit some attention.

VIII. ACKNOWLEDGMENT

We thank Dr. Alex Acero for some preliminary discussions on the subject.

REFERENCES

A Generalized Information Based Approach for Bayesian Optimization

Eduardo C. Garrido-Merchán
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11
28049, Madrid
eduardo.garrido@uam.es

Daniel Hernández-Lobato
Universidad Autónoma de Madrid
Francisco Tomás y Valiente 11
28049, Madrid
daniel.hernandez@uam.es

Abstract—This work presents PESMOC, Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints, an information-based strategy for the simultaneous optimization of multiple expensive-to-evaluate black-box functions under the presence of several constraints. PESMOC can hence be used to solve a wide range of optimization problems. Iteratively, PESMOC chooses an input location on which to evaluate the objective functions and the constraints so as to maximally reduce the entropy of the Pareto set of the corresponding optimization problem. The constraints considered in PESMOC are assumed to have similar properties to those of the objective functions in typical Bayesian optimization problems. That is, they do not have a known expression (which prevents gradient computation), their evaluation is considered to be very expensive, and the resulting observations may be corrupted by noise. These constraints arise in a plethora of expensive black-box optimization problems. We carry out synthetic experiments to illustrate the effectiveness of PESMOC, where we sample both the objectives and the constraints from a Gaussian process prior. The results obtained show that PESMOC is able to provide better recommendations with a smaller number of evaluations than a strategy based on random search.

I. INTRODUCTION

We consider the problem of simultaneously minimizing $K$ functions $f_1(x),...,f_K(x)$ subject to the non-negativity of $C$ constraint functions $c_1(x),...,c_C(x)$ over some bounded domain $X \subseteq \mathbb{R}^d$, where $d$ is the dimensionality of the input space. More precisely, the problem considered is:

$$\min_{x \in X} \quad f_1(x),...,f_K(x)$$

s.t. $$c_1(x) \geq 0,\ldots,c_C(x) \geq 0.$$  \hspace{1cm} (1)

This scenario is broader and more challenging than the one considered in traditional optimization scenarios, where there is a single-objective function and no constraints. In this setting a point $x \in X$ is feasible if $c_j(x) \geq 0$, for all $j = 1,\ldots,C$. We define the feasible space $F \subseteq X$ as the set of points that are feasible. Only the solutions contained in $F$ are considered valid. All potential solutions $x$ not found in $F$ are ignored.

Most of the times it is impossible to optimize all the objective functions at the same time. In particular, they may be conflicting between each other and may prefer different solutions $x \in F$. An example of this is finding good parameters for the control system of a four-legged robot in which we are interested in minimizing energy consumption and maximizing locomotion speed $[1]$. Most probably, maximizing locomotion speed will lead to an increase in the energy consumption and vice-versa. In spite of this, it is still possible to find a set of optimal points $X^*$ known as the Pareto set $[16]$. Let us define that $x$ dominates $x'$ if $f_k(x) \leq f_k(x')$ $\forall k$, with at least one inequality being strict. Then, the Pareto set is the subset of non-dominated points in $F$. Namely, $\forall x^* \in X^* \subseteq F$, $\forall x \in F \exists k \in 1,...,K$ such that $f_k(x^*) < f_k(x)$. Typically, given $X^*$ the final user may choose a point from this set according to their needs (locomotion speed vs. energy consumption). Importantly, the Pareto set is often infinite, so most strategies aim at finding a finite set to approximate $X^*$.

The constraints described in the first paragraph of this section also appear frequently in many optimization problems. For example, in the problem about the robot described before, without loss of generality, besides optimizing energy consumption and locomotion speed, we may have some constraints in the form that the amount of weight placed on a leg of the robot does not exceed a specific value, or similarly, that the maximum angle between the legs of the robot is below some other value for safety reasons.

Another example of the problems we are interested in can be found in the design process of a new type of low-calorie cookie $[3]$. In this case the design space may be the space of possible recipes and baking times. Here we may be interested in minimizing the number of calories per cookie and in maximizing tastiness. These are probably conflictive objectives. Such a problem can also be constrained in the sense that we may want to keep production costs below a particular level or we may want that the cookie is considered to be crispy for at least 90% of the population.

The optimization problems considered in this work also arise naturally in the process of tuning machine learning systems. Without loss of generality, we may have a deep neural network to be designed for some recognition task and we would like to find the architecture and training parameters to simultaneously maximize prediction accuracy and minimize prediction time. These objectives are conflictive because reducing prediction error will require bigger and deeper networks which will increase prediction time. Several constraints may also arise when trying to codify such network in a chip so that it can be included in a low energy consuming device.
mobile device. In this case we may want that the energy consumption of the corresponding chip is below a particular level. The same may happen with the area of the chip, which could be required to be below some particular threshold value.

In many problems of interest the cost of evaluating the objectives or the constraints is very high and the process may be contaminated by noise. Furthermore, there may be no closed form expressions for these function, which will make difficult any gradient computation. This is the case of the examples described before. Measuring locomotion speed may involve some experiment with the robot; measuring tastiness may involve some trials with some persons; measuring chip energy consumption may involve running a simulator; and so on. If this happens, we would like to account for the noise and to minimize the number of evaluations of the objectives and the constraints that is required to obtain the final approximation to the Pareto set. An approach that has shown promising results in such a setting consists in using Bayesian optimization techniques [11]. These techniques use a probabilistic model (typically a Gaussian process [13]) to describe the output of each function. At each iteration, they use the uncertainty in the probabilistic models to generate an acquisition function whose maximum indicates the most promising location on which to evaluate the objectives and the constraints to solve the optimization problem. After enough observations have been collected like this, the probabilistic models can be optimized to provide an estimate of the Pareto set of the original problem. Importantly, the acquisition function only depends on the uncertainty provided by the probabilistic models and not on the actual objectives or constraints. This means that it can be evaluated and optimized very quickly to identify the next evaluation point. By carefully choosing the points on which to evaluate the objectives and the constraints, Bayesian optimization methods find a good estimate of the solution of the original optimization problem with a small number of evaluations [2, 15].

In this paper we describe a strategy for constrained multi-objective optimization that is suited to the scenario described. For this, we extend previous work that uses information theory to build an acquisition function that can be used to optimize several objectives [6], and also previous work that uses information theory to build an acquisition function that can be used to optimize a single objective with several constraints [7]. The result is a strategy that incorporates the possibility of having several objectives and constraints, simultaneously. Such an strategy chooses the next point on which to evaluate the objectives and the constraints as the one that is expected to reduce the most the uncertainty about the Pareto set in the feasible space, measured in terms of Shannon’s differential entropy. The idea is that a smaller entropy implies that the the Pareto set, i.e., the solution to the optimization problem, is better-identified [5, 8, 17]. The proposed approach is called Predictive Entropy Search for Multi-objective Bayesian Optimization with Constraints (PESMOC).

A series of extensive experiments in which both the objectives and the constraints are sampled from a Gaussian process prior shows that the proposed strategy, PESMOC, has practical advantages over a random search strategy that chooses the points on which to evaluate the objectives and the constraints at random. In particular, PESMOC is able to provide recommendations for the Pareto set that are more accurate with a smaller number of evaluations.

II. PREDICTIVE ENTROPY SEARCH FOR MULTI-OBJECTIVE OPTIMIZATION WITH CONSTRAINTS

The proposed method maximizes the information gain about the Pareto set $\mathcal{X}^*$ over the feasible set $\mathcal{F}$. This method requires a probabilistic model for the unknown objectives and constraints. Let the set of objective functions $\{f_1, \ldots, f_K\}$ be denoted with $\mathbf{f}$, and the set of constraint functions $\{c_1, \ldots, c_C\}$ be denoted with $\mathbf{c}$. We assume that all these functions have been generated from independent Gaussian process (GP) priors [13]. We also assume observational noise that is i.i.d. Gaussian with zero mean. For simplicity, a coupled setting in which all objectives and constraints are evaluated at the same location in any given iteration is considered.

Let $\mathcal{D} = \{(x_n,y_n)\}_{n=1}^N$ denote all the observations up to step $N$, where $y_n$ is a $K + C$-dimensional vector with the values resulting from the evaluation of the $K$ objectives and the $C$ constraints at step $n$, and $x_n$ is a vector in input space representing the corresponding input location. The next evaluation point $x_{N+1}$ is chosen as the one that maximizes the expected reduction in the differential entropy $H(\cdot)$ of the posterior distribution over the Pareto set in the feasible space, $p(\mathcal{X}^*|\mathcal{D})$. The acquisition function of PESMOC is:

$$
\alpha(x) = H(\mathcal{X}^*|\mathcal{D}) - \mathbb{E}_x[H(\mathcal{X}^*|\mathcal{D} \cup \{(x,y)\})],
$$

where the expectation is taken with respect to the posterior distribution of the noisy evaluations of the objectives $\mathbf{f}$ and the constraints $\mathbf{c}$, at $x$. That is, $p(y|\mathcal{D},x) = \prod_{k=1}^K p(y_k|\mathcal{D},x) \prod_{j=1}^C p(y_{k+j}|\mathcal{D},x)$, under the assumption of independence among the GPs. In practice, the computation of Eq. (2), known as Entropy Search [5], is very difficult since it involves the entropy of a set of points of potentially infinite size. Houlsby et al. [9] and Hernández-Lobato et al. [8] describe an approach that makes the computation of that expression easier. More precisely, Eq. (2) represents the mutual information between $\mathcal{X}^*$ and $y$ given $\mathcal{D}$. Because the mutual information is symmetric, the roles of $\mathcal{X}^*$ and $y$ can be exchanged leading to the following simplified but equivalent expression:

$$
\alpha(x) = H(y|\mathcal{D},x) - \mathbb{E}_{\mathcal{X}^*}[H(y|\mathcal{D},x,\mathcal{X}^*)],
$$

where the expectation is now with respect to the posterior distribution for the Pareto set in the feasible space, $\mathcal{X}^*$, given the observed data, $\mathcal{D}$, and $H(y|\mathcal{D},x,\mathcal{X}^*)$ measures the entropy of $p(y|\mathcal{D},x,\mathcal{X}^*)$, i.e., the predictive distribution for the objectives and the constraints at $x$ given $\mathcal{D}$ and conditioned to $\mathcal{X}^*$ being the Pareto set in the feasible space. This alternative formulation significantly simplifies the evaluation of the acquisition function $\alpha(\cdot)$ because we no longer have to evaluate the entropy of $\mathcal{X}^*$. We note that the acquisition
function obtained in Eq. (3) favors the regions of the input space in which $X^*$ is more informative about $y$. These are also the regions in which $y$ is more informative about $X^*$.

The first term in the r.h.s. of Eq. (3) is straight-forward to evaluate. It is simply the entropy of the predictive distribution $p(y|D, x)$, which is a factorizing $K + C$-dimensional Gaussian distribution. That is,

$$H(y|D, x) = \frac{K + C}{2} \log(2\pi e) + \sum_{i=1}^{K} 0.5 \log(\nu_k^{PD}) + \sum_{i=1}^{C} \log(s_c^{PD}),$$

(4)

where $\nu_k^{PD}$ and $s_c^{PD}$ are the predictive variances of the objectives and the constraints, respectively. The difficulty comes from the evaluation of the second term in the r.h.s. of Eq. (3), which is intractable and has to be approximated. For this, we follow [6], [8] and approximate the expectation using a Monte Carlo estimate obtained by drawing samples of $X^*$ given $D$. This involves sampling several times the objective functions and the constraints from their posterior distribution $p(f, c|D)$. This is done as in [8]. Given a sample of the objectives and the constraints, we solve the corresponding optimization problem to find an estimate of $X^*$. For this, we use a grid search approach, although more efficient methods based on evolutionary strategies can be used in the case of high dimensional spaces. The Pareto set $X^*$ needs to be located in the feasible space. Thus, we discard all input grid locations in which the sampled constraints are strictly negative. The Pareto set is then simply obtained by returning all the non-dominated grid locations. Note that unlike the true objectives and constraints, the sampled functions can be evaluated very cheaply. Given a sample of $X^*$, the differential entropy of $p(y|D, x, X^*)$ is estimated using expectation propagation [10].

A. Using Expectation Propagation to Approximate the Conditional Predictive Distribution

We use expectation propagation (EP) [10] to approximate the entropy of the conditional predictive distribution $p(y|D, x, X^*)$. For this, the distribution $p(X^*|f, c)$ is considered first. We note that $X^*$ is the Pareto set in the feasible space $F$ if and only if $\forall x^* \in X^*$, $\forall x^* \in X$, $c_j(x^*) \geq 0$ for all $j = 1, \ldots, C$, and if $c_j(x) \geq 0$, for all $j = 1, \ldots, C$, then $k$ s.t. $f_k(x^*) \leq f_k(x)$. That is, the points in the Pareto set have to be feasible and have to be strictly better than any other feasible point in at least one of the objectives. Informally, the conditions just described can be translated into the following un-normalized distribution:

$$p(X^*|f, c) \propto \prod_{x^* \in X^*} \left( \prod_{j=1}^{C} \Phi_j(x^*) \left( \prod_{x^* \in X} \Omega(x^*, x^*) \right) \right),$$

(5)

where $\Phi_j(x^*) = \Theta(c_j(x^*))$ with $\Theta(\cdot)$ the Heaviside step function, and $\Omega(x^*, x^*)$ is defined as:

$$\Omega(x^*, x^*) = \prod_{j=1}^{C} \Theta(c_j(x^*)) \psi(x^*, x^*) + \left[ 1 - \prod_{j=1}^{C} \Theta(c_j(x^*)) \right] \cdot 1,$$

(6)

where $\psi(x^*, x^*)$ is defined as

$$\psi(x^*, x^*) = 1 - \prod_{k=1}^{K} \Theta(f_k(x^*) - f_k(x')).$$

(7)

The factor $\prod_{j=1}^{C} \Phi_j(x^*)$ in Eq. (5) guarantees that every point in the Pareto set belongs to the feasible space $F$ (otherwise $p(X^*|f, c)$ is equal to zero). The factors $\Omega(x^*, x^*)$ in Eq. (6) are explained as follows. The product $\prod_{j=1}^{C} \Theta(c_j(x^*))$ checks that the input location $x^*$ belongs to the feasible space. If the point is not feasible, we do nothing, i.e., we multiply everything by one. Otherwise, the input location $x^*$ has to be dominated by the Pareto point $x^*$. That is, $x^*$ has to be better than $x^*$ in at least one objective. That is precisely checked by Eq. (7).

Now we show how to compute the conditional predictive distribution $p(y|D, x, X^*)$. For simplicity, we consider a noiseless case in which we directly observe the actual objectives and constraints values. In that case, $p(y|x, f, c) = \prod_{k=1}^{K} \delta(y_k - f_k(x)) \prod_{j=1}^{C} \delta(y_j - c_j(x))$, where $\delta(\cdot)$ is a Dirac’s delta function. In the noisy case one only has to replace the delta functions with Gaussians with the corresponding noise variance. The unnormalized version of $p(y|D, x, X^*)$ is:

$$p(y|D, x, X^*) \propto \int p(y|x, f, c)p(X^*|f, c)p(f|D)p(c|D)dfdc\Omega,$$

(8)

where we have separated the factors $\Omega(\cdot, \cdot)$ that depend on the candidate point $x$ on which to compute the acquisition from the ones that do not depend on $x$. In order to approximate Eq. (5), using EP, $X^*$ is approximated with the set $X = \{x_n\}_{n=1}^{N} \cup X^* \cup \{x\}$. This set represents the union of the input locations where the objectives and constraints have been evaluated, the current Pareto set in the feasible space and the candidate point $x$ where the acquisition function will be evaluated. Then, all non-Gaussian factors in Eq. (5) are replaced by Gaussian factors whose parameters are found using EP [10]. Note that the only non-Gaussian factors are each $\Phi_j(\cdot)$ and each $\Omega(\cdot, \cdot)$.
Non-Gaussian factors are approximated with un-normalized Gaussians as follows. Each $\Phi_j(\cdot)$ factor is replaced by a one-dimensional un-normalized Gaussian distribution over $c_j(x^*)$. That is,
\[ \Phi_j(x^*) \approx \tilde{\Phi}_j(x^*) \propto \exp \left\{ -\frac{c_j(x^*)^2 \tilde{\epsilon}^2_{j}}{2} + c_j(x^*) \tilde{\mu}^2_{j} \right\} \right), \quad (9) \]
where $\tilde{\epsilon}^2_{j}$ and $\tilde{\mu}^2_{j}$ are natural parameters adjusted by EP. Each $\Omega(x', x^*)$ factor is replaced by a product of $C'$ one-dimensional un-normalized Gaussians and $K$ two-dimensional un-normalized Gaussians. In particular,
\[ \Omega(x', x^*) \approx \tilde{\Omega}(x', x^*) \propto \prod_{k=1}^{K} \exp \left\{ -\frac{1}{2} \tilde{v}^T_k \tilde{\nu}^\Omega_k \tilde{v}_k + (\tilde{m}^\Omega_k)^T \tilde{m}^\Omega_k \right\} \times \prod_{j=1}^{C'} \exp \left\{ -\frac{c_j(x^*)^2 \tilde{\epsilon}^2_{j}}{2} + c_j(x^*) \tilde{\mu}^2_{j} \right\} \right) \quad (10) \]
where we have defined $v_k$ as the vector $(f_k(x'), f_k(x^*))^T$. Furthermore, $\tilde{v}_k^\Omega$, $\tilde{m}_k^\Omega$, $\tilde{\epsilon}^2_{j}$ and $\tilde{\mu}^2_{j}$ are natural parameters adjusted by EP. Note also that $\tilde{v}_k^\Omega$ is a $2 \times 2$ matrix and $\tilde{m}_k^\Omega$ is a two-dimensional vector.

The factors, that do not depend on $x$ (the candidate location on which to compute the acquisition) are refined iteratively by EP until they do not change any more. These factors are reused on which to compute the acquisition) are refined iteratively by EP. The factors that depend on $x$ are refined only once by EP so that the acquisition function can be quickly evaluated. When the EP algorithm finishes, the Conditional Predictive Distribution $p(y|D, x, x^*)$ of the objectives and constraints at $x$ is approximated by the normalized Gaussian distribution that results from replacing in Eq. (8) each non-Gaussian factor by the corresponding Gaussian approximation. The Gaussian distribution is closed under the product operation, so, when all the factors exposed are replaced by Gaussians, the result is a Gaussian distribution.

**B. The PESMOC Acquisition Function**

After approximating the Conditional Predictive Distribution by a product of Gaussian distributions using the EP algorithm, we add the noise variances to the marginal variances. Then, the PESMOC acquisition function is given by the sum of the differences between the entropies before and after conditioning on the Pareto set. This, in combination with the expression shown in Eq. (3) gives:
\[ \alpha(x) \approx \sum_{j=1}^{C} \log s_j^PD(x) + \sum_{k=1}^{K} \log v_k^PD(x) + \frac{1}{M} \sum_{m=1}^{M} \left( \sum_{j=1}^{C} \log s_j^{CPD}(x|X^*_m) \right) + \sum_{k=1}^{K} \log v_k^{CPD}(x|X^*_m) \right) = \sum_{k=1}^{K} \alpha_k^P(x) + \sum_{j=1}^{C} \alpha_j^P(x), \quad (11) \]
where $M$ is the number of Monte Carlo samples, $\{X^*_m\}_{m=1}^{M}$, of the Pareto set over the feasible set. These samples are used to approximate the expectation in Eq. (11). Furthermore, $v_k^{PD}(x)$, $v_c^{PD}(x)$, $v_k^{CPD}(x|X^*_m)$ and $v_c^{CPD}(x|X^*_m)$ are the variances of the predictive distribution of the objectives and the constraints, respectively, before and after conditioning on the Pareto set.

We note that the acquisition function in Eq. (11) can be expressed as a sum across the objectives and the constraints. Intuitively, each term in this sum measures the reduction in the entropy of the Pareto set after an evaluation of the corresponding objective or constraint at the input location $x$. This allows for a decoupled evaluation setting in which each objective or constraint is evaluated at a different input location. Furthermore, it introduces a mechanism to identify which objective or constraint is expected to be more useful to evaluate. For this, we only have to individually maximize each of the $K + C$ acquisition functions in Eq. (11), i.e., $\alpha_k^P(\cdot)$, for $k = 1, \ldots, K$, and $\alpha_j^P(\cdot)$, for $j = 1, \ldots, C$. There is evidence in the literature that a decoupled evaluation setting improves over a coupled one, for the case of un-constrained multi-objective problems [6]. Similar improvements are expected in the constrained multi-objective case.

The computational cost of evaluating the acquisition function and the EP algorithm is $O(KCq^3)$, with $q = N + |X^*_m|$, with $N$ being the number of observations, $K$ the number of objectives and $C$ the number of constraints. The EP algorithm is run once per each sample of the Pareto set $X^*_m$. After this, it is possible to re-use again the factors that are independent of the candidate input location $x$. Thus, the complexity of computing the predictive variance is $O(KC|X^*_m|^3)$. In practice, the number of points in each Pareto set sample is set equal to 50, making $q$ to be just a few hundreds at most.

**III. EXPERIMENTS**

We carry out experiments to compare the performance of the proposed method, PESMOC, with that of a random search (RS) strategy. At each iteration, in RS the points on which to evaluate the objectives and the constraints are obtained by sampling from a uniform distribution over the input space. Thus, RS is expected to perform worse than PESMOC due to the fact that it does not use the model’s uncertainty to identify intelligently the next point on which to do evaluation. By contrast, PESMOC is expected to use that uncertainty to evaluate the objectives and the constraints only in those regions of the input space that are expected to be more useful. Both strategies, PESMOC and RS, have been implemented in the software for Bayesian optimization Spearmint [https://github.com/HIPS/Spearmint]. In our experiments we employ a Matérn covariance function in the GPs that are used model the objectives and the constraints. The hyper-parameters of the GPs (noise variance, length-scales and amplitudes) are approximately sampled from their posterior distribution using slice sampling [12]. We generate 10 samples for each hyper-parameter, and the acquisition function of PESMOC is averaged over these samples.
To compare PESMOC and RS, we generate 100 synthetic optimization problems obtained by sampling the objectives and the constraints from their respective GP prior. The input dimension of each problem is set equal to 3 and we consider 2 objectives and 2 constraints in each problem. After this, each strategy is run until 100 evaluations of the objectives and the constraints are done. We consider two scenarios: A first scenario in which the evaluations are noiseless, and a second scenario in which each evaluation (objectives and constraints) is contaminated with Gaussian noise with standard deviation equal to 0.1. At each iteration, each strategy, PESMOC and RS, outputs a recommendation in the form of a Pareto set obtained by optimizing the posterior means of the GPs. The performance criterion used to compare PESMOC with RS is the hyper-volume indicator, which is maximized by the actual Pareto set [18]. When the recommendation produced by a particular strategy contains a point which does not belong to the feasible space, we set the corresponding hyper-volume equal to zero. In practice, we report the logarithm of the hyper-volume and in the un-constrained setting we note that the feasible space \( \mathcal{F} \) is given by the box \([-10, 10] \times [-10, 10]\).

**Fig. 1** shows the average results obtained in the experiments described, alongside with the corresponding error bars. We observe that PESMOC is able to find better solutions to the optimization problems considered. More precisely, the solutions obtained by PESMOC are more accurate than those obtained by RS, since they have a hyper-volume that is closer to the hyper-volume of the actual Pareto set. Furthermore, they are obtained with a smaller number of evaluations of the objectives and the constraints. If the computational cost of evaluating the objectives and the constraints is very high, this represents a significant improvement.

To illustrate the usefulness of the acquisition function of PESMOC, we consider the following toy 2-dimensional optimization problem in the box \([-10, 10] \times [-10, 10]\):

\[
\begin{align*}
\min_{x,y} & \quad f_1(x,y) = xy, \quad f_2(x,y) = -yx \\
\text{s.t.} & \quad x \geq 0, y \geq 0 .
\end{align*}
\]

We evaluate PESMOC and RS in this problem and record the evaluations performed by each method at each iteration. **Fig. 1** shows the location of the evaluations done by PESMOC (top) and RS (bottom) in input space, after 20 evaluations of the objectives and the constraints. We observe that PESMOC quickly identifies the feasible space, i.e., the box \([0, 10] \times [0, 10]\), and focuses on evaluating the objectives and the constraints in that region. By contrast, RS explores the space more uniformly and, in consequence, evaluates the objectives and the constraints more frequently in regions that are infeasible. **Fig. 2** (top) also shows the level curves of the acquisition function computed by PESMOC. This function takes high values in regions inside \( \mathcal{F} \) and low values in regions outside \( \mathcal{F} \). These results explain why PESMOC is able to outperform RS in the previous experiments.

**IV. CONCLUSIONS AND FUTURE WORK**

We have described an information-based approach that can be used to address a wide range of Bayesian optimization problems, including multiple objectives and several constraints. Motivated by the lack of methods that are available solve these problems with an adequate exploration-exploitation balance, PESMOC has been presented. At each iteration, PESMOC evaluates the objective functions and the constraints at an input location that is expected to reduce the entropy of the posterior distribution of the Pareto set the most. We have illustrated with synthetic experiments the benefits of such an approach with respect to a random search strategy. The results obtained show that PESMOC is able to provide estimates of the Pareto set of the optimization problem that are more accurate with a smaller number of evaluations. This is very useful in practical situations in which the objectives and the constraints are very expensive to evaluate.

Future work will compare the proposed method, PESMOC, with the method described by Feliot *et al.* [3]. Such a method can, in principle, also consider multiple objectives and several constraints. However, it is based on the expected improvement of the hyper-volume, and in the un-constrained setting there is empirical evidence supporting that information-based approaches perform better [6]. Furthermore, the computation of the expected improvement of the hyper-volume is very expensive and it can only be done in the case of a small...
useful when we have access to a distributed system that can evaluate in parallel the objectives and the constraints at several candidate locations.

ACKNOWLEDGMENTS

The authors acknowledge the use of the facilities of Centro de Computación Científica (CCC) at Universidad Autónoma de Madrid, and financial support from the Spanish Plan Nacional I+D+i, Grants TIN2013-42351-P and TIN2015-70308-REDT, and from Comunidad de Madrid, Grant S2013/ICE-2845 CASI-CAM-CM.

REFERENCES


Bayesian Networks for Neuroscience

Pedro Larrañaga
Computational Intelligence Group
Artificial Intelligence Department
Technical University of Madrid
pedro.larranaga@fi.upm.es

Abstract—Bayesian networks are a type of probabilistic graphical models where the variables of a system are represented as nodes in a directed acyclic graph and the arcs between nodes are interpreted in terms of conditional independences among triplets of variables. This semantics, based on probability theory, in combination with the existence of algorithms for learning Bayesian networks from data and for making any kind of inference (reasoning) inside the network, allow to discover new scientific knowledge in domains where data-sets have been collected.

In this talk we will present two applications of Bayesian networks in the field of neuroscience. The first one aims to classify and name GABAergic cortical interneurons from their morphological variables and a labelling process carried out by several neuropathology experts. This turns the classification into a supervised learning problem where labels are probabilistic instead of crisp. The second application concerns to Parkinson’s disease. Multidimensional classification is used to transform patient perception of his illness into a generic measure of health for clinical and economical appraisal.

I. INTRODUCTION

Neuroscience deals with the structure, function, development, genetics, biochemistry, physiology, pharmacology, and pathology of the nervous system. Understanding the human brain is one of the greatest challenges facing 21st century science. If we can rise to the challenge, we can gain fundamental insights into what to be human means, develop new treatments for brain diseases, and build revolutionary new information and communications technologies. In Europe, the Human Brain Project aims to develop a large-scale computer simulation of the brain, whereas in the United States, the Brain Activity Map is working towards establishing a functional connectome of the entire brain, and the Allen Institute for Brain Science has embarked upon a 10-year project to understand the mouse visual cortex.

Modern neuroscience has been enormously productive but unsystematic. The data it produces describes different levels of biological organisation, in different areas of the brain, in different species, at different stages of development. We urgently need to integrate [9] and to analyse these data using machine learning methods [5] to show how parts fit together in a single multi-level system.

A Bayesian network (BN) [10] [14], is a compact representation of a probability distribution over a set of discrete variables. Variables represent the uncertain knowledge of a given domain and are depicted as the nodes of the network. The structure of a BN is a directed acyclic graph, where the arcs have a formal interpretation in terms of probabilistic conditional independence. The quantitative part of a BN is a collection of conditional probability tables, each attached to a node, expressing the probability of the variable at the node conditioned on its parents in the network. The joint probability distribution (JPD) over all variables is computed as the product of all these conditional probabilities dictated by the arcs. This distribution entails enough information to attribute a probability to any event expressed with the variables of the network. Moreover, there are efficient algorithms for computing any such probability without having to generate the underlying JPD (this would be unfeasible in many cases). BNs have enormously progressed over the last few decades leading to applications spanning all fields. BNs have been applied in a large number of neuroscience applications approaching problems of knowledge discovery, supervised classification and clustering with different types of input data: morphological, electrophysiological, -omics data and neuroimaging. See [1] for a review on this topic.

II. CLASSIFYING AND NAMING GABAERGIC INTERNEURONS

GABAergic interneurons of the cerebral cortex are key elements in many aspects of cortical function in both health and disease. Nevertheless, the classification of GABAergic interneurons is a difficult task and has been a topic of debate for a long time, since the pioneering work of Santiago Ramón y Cajal on the characterization and identification of interneurons. The difficulty stems from the high variability of these cells according to morphological, electro-physiological and molecular features. The scientific community lacks an accepted catalog of neuron names which makes it difficult to organize and share knowledge. There is some agreement on the set of morphological, molecular, and physiological features that can be used to distinguish among types of GABAergic interneurons. However, a comprehensive classification according to those features is difficult to perform in practice.

A recent experiment enabled 42 expert neuroscientists from all around the world to classify interneurons by visual inspection and according to pre-selected neuron names [7]. It showed that the experts agree on the morphological definitions of some of the pre-selected types while disagreeing on the definitions of others. In particular, some types seemed to overlap in terms of the cells that were assigned to them by the experts. In [7], the authors also showed that supervised classification models
can automatically categorize interneurons in accordance with the opinion of the majority of the experts.

During the talk we will present the main results achieved in [7] emphasizing the knowledge derived from the 42 Bayesian networks, each of one modeling, for each of the experts, the relationships among the six neuronaatomical variables collected in the experiment.

III. MAPPING PARKINSON’S DISEASE PATIENTS PERCEPTION TO A GENERAL HEALTH SCALE

The second application to be presented during the talk, is about the prediction of the European Quality of Life-5 Dimensions (EQ-5D) from the 39-item Parkinson’s Disease Questionnaire (PDQ-39). In fact, EQ-5D is a generic health-related quality of life (HRQoL) measure usable in general populations and in any disorder. It is considered a valid instrument and is recommended for evaluation of HRQoL in Parkinson’s disease (PD). EQ-5D contains five items, namely, mobility, self-care, usual activities, pain/discomfort, and anxiety/depression. Each item has three options of response: no problems, some problems and severe problems. However, PDQ-39 is a specific HRQoL instrument widely used in PD. It contains 93 questions each scoring on a five-point scale: never, occasionally, sometimes, often, and always. Using a real-world Parkinson’s disease data set containing 488 patients the challenge consists of developing a multi-dimensional classifier able to predict the five items of EQ-5D from the 39 items of PDQ-39 [4, 5].

Multi-dimensional Bayesian network classifiers (MBCs) [2], have been proposed to deal with multi-dimensional classification providing an accurate modeling of the probabilistic dependence relationships among all variables, the class variables included. In the talk we will compare the results provided by MBCs with other state of the art methods in this data set.

IV. CONCLUSION

These two examples on the application of Bayesian network methods as data-driven approaches for computational neuroscience show the possibilities of machine learning in modern neuroscience. As anticipated in [15], neuroscience is set to collect its own big data sets, but to exploit its full potential, there need to be ways to standardize, integrate, synthesize and analyze diverse types of data from different levels of analysis and across species.

ACKNOWLEDGMENT

This work has been partially supported by the Spanish Ministry of Economy and Competitiveness through the Cajal Blue Brain (C0800209; the Spanish partner of the Blue Brain initiative from EPFL) and TIN2013-41592-P projects, by the Regional Government of Madrid through the IDD13/ICE-2845-CASI-CAM-CM project, and by the European Union’s Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 604102 (Human Brain Project).

REFERENCES

Automatic classification of cortical interneuron morphologies

Bojan Mihaljević, Concha Bielza and Pedro Larrañaga
Escuela Técnica Superior de Ingenieros Informáticos
Universidad Politécnica de Madrid
Campus de Montegancedo s/n,
Boadilla del Monte 28660, Madrid, Spain
Email: bmihaljevic@fi.upm.es

Abstract—The classification of cortical interneurons is a topic of intense debate in neuroscience. Rather than a single, accepted classification, there exist a number of partially overlapping classification schemes. Given the increasing amounts of digital data available, a possible future solution would be a quantitative identification of interneuron classes, and subsequent automatic assignment of neurons to those classes. Here, we use a set of 219 quality morphology reconstructions of rat cortical interneurons to show that supervised classifiers can identify existing interneuron classes based on morphological data. We complement standard neuronal morphometrics with ad hoc features designed to capture the properties of known interneuron types. We consider a number of supervised classification algorithms and, in order to boost the prediction of under-represented classes, combine them with oversampling and binarization techniques. We show that five of seven classes present in our data set can be reliably distinguished.

I. INTRODUCTION

The neocortex allows mammalian brains to perform sophisticated cognitive tasks. The main obstacle towards untangling its complex design is the large diversity of inhibitory interneurons [1]. These cells, which constitute 20-30% of the neurons in the neocortex, are very diverse with regards to morphological, electro-physiological, molecular, and synaptic properties [2], [3], [4]. While most researchers consider that interneurons can be grouped into classes [5], a unique and comprehensive classification is lacking. Instead, there are only a handful of clearly identified cell classes, and a number of partially overlapping classification schemes, mainly based on a single type of properties (i.e., morphological, molecular, etc.) [6].

A possible solution to this problem might be a fully quantitative classification [7]. Laboratories are creating and sharing large amounts of digital data on morphological, electro-physiological, and molecular properties [8], [9], which could enable such an approach. Indeed, morphology alone can allow an expert to identify some of the known cell types [10], and morphometrics, i.e., quantitative properties of the morphology, can accurately predict some of those types [11]. With many interneuron morphologies available at the Neuromorpho [12] repository, it seems that automatic classification on the basis of morphology is a achievable goal.

Yet, the data at Neuromorpho present multiple problems. Morphology reconstruction can vary enormously from laboratory to laboratory, affecting the subsequently computed morphometrics [13], [14]. Important metadata, such as cortical layer, area, animal age and experimental condition are too often missing, and the morphologies are often of poor quality or incompletely reconstructed. Finally, even when the metadata is present, the differences across species and cortical areas [4] might render the problem difficult.

In this study, we use a set of homogeneous neurons, with high-quality reconstructions performed by a single laboratory, and contain the relevant metadata. The cells come from the rat somatosensory cortex and were carefully selected by [15]. Since the cells are according to the morphological classification of [1], we can use supervised classification in order to attempt to predict the neuronal type on the basis of morphometrics. This set of quality cells can indicate whether automatic classification is possible.

II. DATA

Our data consists of 219 hind-limb somatosensory neocortex interneurons from two-week-old Wistar rats. The cells were selected and used in the simulation of the cortical column by [15]. We used raw, unravelled morphologies, as used by the experts to classify them (see [15]). For each cell, we have a 3D digital reconstruction with the soma, and the dendritic and axonal branching properties. We know the layer which contains the soma, and have estimates of the thickness and width of the cortical layers and column, respectively. Finally, each cell is classified as one of seven interneuron types from the classification scheme of [1]. We have no data on synapse locations, nor molecular or electro-physiological properties.

A. Class labels

All of the seven interneuron types that are present as labels in our data are defined in [1]. Table I lists their full names and the acronyms used throughout this paper. The experts classified the cells according to their morphologies, but also their synaptic and molecular properties, which are not available in our reconstructed 3D cells.

B. Morphometrics

We are not given a matrix of predictor features but, corresponding to each row, a digitally reconstructed neuron as
a 3D tree structure, with the soma as root and the axon and dendrites stemming out of it. Multiple free (e.g., L-Measure [16]) and commercial (e.g., Neurolucida Explorer) software packages can compute morphometric features from these reconstructions. Such morphometrics include general properties, such as axonal height and width, and features of branching such as branching angles or branching tortuosity (see, e.g., [17]). In L-Measure, most features correspond to properties of the branching pattern and diameter.

L-Measure can compute 43 types of axonal and dendritic morphometrics. For each morphometric, it allows one to readily compute five descriptive statistics: the mean, the standard deviation, the minimum, the maximum, and the sum. We expanded this with the coefficient of variation, computed as the ratio of the standard deviation and the mean, and we computed the median for some of the features, via L-Measure’s histogram functionality. We omitted the morphometrics related to branch diameter, as they differ widely across laboratories [13], and fragmentation. We exclude features for which we a priori know that are not meaningful, such as the sums of angles, and we removed the minima statistics as they are often identical across neurons (e.g., minimum Euclidean distance from soma is 0).

We first computed those analyses across the whole axonal arbor. Since terminal branches can characterise certain morphological types [5], [1], we applied a subset of those analyses to the terminal branches. In addition, we included two dendritic features: the number of trees and the maximum branching order.

We wrote custom software to compute well-known morphometrics which are not available in L-Measure, such as the 2D axonal convex hull area, diameter, and shape.

Certain traits of specific interneuron types are not well captured by the generic L-Measure features. For example, the Martinotti cells’ axons reach up to layer 1 of the cortex. We wrote custom software to compute such ad hoc features.

Finally, we removed features with absolute correlation above 0.95. The aim was to improve the interpretability of the resulting models by reducing redundant information. For descriptive statistics of the same feature, we gave preference to the mean and the standard deviation when correlated with other statistics. We gave preference to whole-axon features over those confined to terminal branches, and to generic features over ad hoc ones. The final data set had 99 predictor features. Table II lists the L-measure features computed for the whole axon.

### III. Supervised Classification

Table III lists the supervised classifiers that we used and introduces their acronyms used from hereon. They range from linear models with no regularisation, such as LDA, to non-linear models with embedded feature selection, such as RF.

#### Table III. Supervised Classifiers.

<table>
<thead>
<tr>
<th>Name</th>
<th>Abbreviation</th>
<th>Ref.</th>
<th>R package</th>
</tr>
</thead>
<tbody>
<tr>
<td>Classification and regression trees</td>
<td>CART</td>
<td>[18]</td>
<td>e1071</td>
</tr>
<tr>
<td>Gaussian Naive Bayes</td>
<td>NB</td>
<td>[19]</td>
<td>e1071</td>
</tr>
<tr>
<td>k nearest neighbors</td>
<td>kNN</td>
<td>[20]</td>
<td>kknn</td>
</tr>
<tr>
<td>Linear discriminant analysis</td>
<td>LDA</td>
<td>[21]</td>
<td>MASS</td>
</tr>
<tr>
<td>Multinomial logistic regression</td>
<td>LR</td>
<td></td>
<td>nnet</td>
</tr>
<tr>
<td>Random forest</td>
<td>RF</td>
<td>[22]</td>
<td>randomForest</td>
</tr>
<tr>
<td>Support vector machines</td>
<td>SVM</td>
<td>[23]</td>
<td>e1071</td>
</tr>
</tbody>
</table>

#### A. Class imbalance

Our data set is highly imbalanced [24], with, for example, 5 times less ChC than MC instances (7 versus 41). It can be hard for a classifier to learn to distinguish examples of the under-represented classes. To evaluate the balance in learning the different classes, we complement prediction accuracy by measuring the geometric mean of per-class true positive rates (i.e., sensitivity),

\[
g = \sqrt[7]{\prod_{i=1}^{7} TPR_i},
\]

where \( TPR_i \) is the true positive rate for class \( c_i \), as proposed by [25]. (7 is the number of classes in our data set)
In attempt to improve the detection of under-represented classes, we oversample our training data (see [26]). As common with multi-class imbalanced data, we transform the problem into multiple binary classification problems before oversampling. We use one vs. one (OVO) [27] and one vs. all (OVA) binarization. On the binarized classification problems, we oversample with the SMOTE [28] method, creating synthetic instances of the minority class. We compare those with random oversampling [29] (ROS), i.e., exact copying of minority classes instances, on the multi-class (i.e., all-vs-all, A VA) setting, as well as the baseline A VA setting with no oversampling.

IV. EXPERIMENTAL SETTING AND SOFTWARE

We computed morphometrics with L-measure 5.2. All features are numeric and we centered them to zero mean and unit variance. This, along with the cross-correlation filtering, was the only data processing that we performed on all of our data, i.e., by taking into account data that will go into test folds in cross-validation.

We evaluated the classifiers with 5-fold stratified cross-validation. We oversampled the training data only, leaving the test data intact. In the A VA + ROS case, we followed [30] and oversampled each class until reaching 51 instances, the sample frequency of the majority class; this resulted in 39% of the final data set being copies of observed samples.

With SMOTE, we oversampled the minority class until either equaling the frequency of the majority class, or until artificial examples reached 30% of the data set. This limit was more often met in the OVA case, where the imbalance is stronger than in OVO.

We did all the data analysis and classification with R [31]. We used the mlr [32] package for the classifiers, their evaluation, binarization, and oversampling. For all algorithms we used the default parameters in their implementations, except for knn, where we used 5 neighbors instead of 7.

V. RESULTS

In the baseline A VA without oversampling setting, RF performs best, with NB also providing a good balance of G-mean and accuracy. With OVO binarization most classifiers perform similarly, while with OVA they degrade in both accuracy and G-mean.

ROS oversampling improves G-mean for RF, CART, kNN, and, especially, SVM, while slightly decreasing their accuracy. With OVO, RF yields the best overall G-mean, coupled at the same time with a high accuracy. OVA, again, provides the poorest results.

Overall, RF provides high accuracy and balance in predicting different classes, and the latter improves with oversampling. NB generally has high balance of predicting the different classes, but is not accurate. LDA and LR, the linear non-regularised models, are the worst performers.

Table V shows that the hardest classes for RF with OVO and SMOTE are BTC and ChC, two two least numerous types in the data set. The easiest type to distinguish is NBC.

<table>
<thead>
<tr>
<th>Method</th>
<th>BTA</th>
<th>TCA</th>
<th>BTH</th>
<th>THC</th>
<th>LBA</th>
<th>LBC</th>
<th>MC</th>
<th>NBC</th>
<th>SBC</th>
</tr>
</thead>
<tbody>
<tr>
<td>TPR</td>
<td>77</td>
<td>74</td>
<td>76</td>
<td>76</td>
<td>76</td>
<td>77</td>
<td>83</td>
<td>88</td>
<td>77</td>
</tr>
<tr>
<td>TNR</td>
<td>41</td>
<td>75</td>
<td>76</td>
<td>75</td>
<td>76</td>
<td>76</td>
<td>77</td>
<td>83</td>
<td>88</td>
</tr>
</tbody>
</table>

While twice as numerous as ChC, BTC, is the hardest class to predict. This is possibly because of the lack of dendritic morphometrics in our data, as this type as partially defined on the bases of its dendritic morphology.

VI. FUTURE WORK

A primary goal is to identify the features which distinguish the interneuron types. We plan to extend the present study with feature selection and heavily regularized models. The OVA setting might be used to identify per-class characteristic features.

Better results could have been achieved by adjusting the parameters of models such as the SVM by mean of grid search. Furthermore, morphometric features such as the topological properties [33] might provide useful predictive input. A number of extra ad hoc features can be readily defined on the basis of known class characteristics. Also, it would be useful to assess the relative importance of ad hoc and generic features.

Finally, interneuron classification by experts can be unreliable [10]. It would be useful to assess the amount of label noise in our data. This could be done by identifying outliers according to the most predictive features.
REFERENCES


Differential Analysis as a Data Science Tool for Cyber Security

James H. Jones, Jr.

Abstract—Cyber security is the art, science, and practice of protecting digital assets, and figuring out what happened when those protections fail - and eventually something always fails. Typically, we collect evidence related to such failures and ask an expert or experts to apply their knowledge and past experience to construct theories to explain the evidence and observed events, much like we might ask a detective to interpret a crime scene. However, the nature of the digital world is such that we are drowning in data and the possible mappings of cyber events to digital evidence is staggering, far beyond the capacity of a single human, or group of humans, to retain. Enter the computer and the analytic tools computing power makes available to us. In this paper, I will present a general class of technique termed differential analysis, and I will discuss our work using this and other data science techniques to solve a wide range of cyber security and digital forensics problems. Our research addresses the recovery and decay of deleted digital content on hard drives, mobile devices, and embedded systems, the recovery and interpretation of partial digital artifacts after attacks against drones and software defined radios, digital content remanence in cloud environments, and large-scale drive similarity measures for forensic triage.

Index Terms—cyber security, differential analysis, digital artifacts, digital forensics

I. INTRODUCTION

The practice of cyber security spans activities from prevention, through detection, and to response. Put another way, we try to prevent cyber attacks, but we know we won't be 100% successful so we put mechanisms in place to detect the attacks that do slip through and we prepare for the inevitable post-attack (or during-attack) investigation. Our research is focused on this investigation activity during or after an attack, typically termed triage when the attack is ongoing and forensics if conducted after the attack. Specifically, we are exploring methods to extract and analyze residual digital data in order to draw conclusions about past activities which may be attacks but could also include inadvertent activity, undocumented or unanticipated behavior, and component failures as well. To develop underlying theories supporting our methods, we treat digital systems as experimental subjects in a controlled environment. By varying carefully selected factors, we are able to establish the effects of those factors and create mappings between activity or actions and associated digital artifacts. In this paper, I describe our differential analysis approach, followed by discussions of several example applications drawn from our current research activities.

II. DIFFERENTIAL ANALYSIS

The question that differential analysis aims to answer is "what changed between system states A and B, and what caused those changes"? By measuring such changes as we vary different factors for identical test subjects, we are able to associate digital effects with different causal factors. In the context of our work, factors may include system properties as well as user or system actions, and our test subjects are physical digital devices or virtual machine instances of such devices. Outside of the digital domain, identical test subjects and the ability to recover prior states is rare. However, in the digital domain both are possible: we can create bit-for-bit copies of a digital system, and we can return to prior states using virtual machine snapshots or stored system images. Differential analysis for digital forensics was formally and generally described by Garfinkel et al [1] in 2012, but the idea had been applied to specific digital forensics and cyber security problems for years.

In our work, we create a baseline system or device image. Given a set of factors we wish to investigate, we then create multiple copies of this baseline and vary one factor in each image copy (as well as no changes in a control copy). We operate the systems under different scenarios and take snapshots of the systems at selected points in time. The resulting set of snapshots are then processed for differences, where differences may be file- or sector-based and may be based on changes in persistent storage as well as volatile memory. Differences identified in sequential image snapshots are then potentially associated with the actions taken between the image snapshots. We say "potentially" because digital systems typically have multiple concurrent processes running which may create changes as well. In our drone and SDR example applications that follow, we discuss one approach for establishing the actual causes of observed changes.
III. APPLICATIONS

A. Deleted Content Persistence

In most digital systems, deleting a file does not actually destroy the contents of the file. Rather, the file system removes references to the file, areas of the storage media containing the file contents are marked now as unallocated, and over time those contents are overwritten by new files. A physical analogy is that of a book, where the file system is the table of contents and a file is like a chapter. Deleting a file is like removing the chapter entry from the table of contents but leaving the chapter contents themselves intact. Over time, those contents will be erased and overwritten with new content (imagine paper that is erasable, like a whiteboard with sections that are not erased until they are needed for new material).

Our work in this area attempts to understand the factors affecting deleted file content persistence on different media and devices so that predictive and explanatory models may be constructed. We begin with a list of factors that might affect how much of a deleted file persists and for how long. These factors include free space on the storage media, fragmentation of the storage media, file system implementation, the type of physical media, file size, file type, file fragmentation, file location, and activity on the system after the file is deleted. From a common baseline image, we construct separate images representing different values for each of the factors of interest. We run these images as virtual machines under different usage scenarios and take snapshots of the storage media at various points in time. We then post-process these sequential images to track the contents of deleted files over time. In practice, this involves identifying files deleted between snapshots 0 and 1, then going back to snapshot 0 to record the sector locations used to store file contents and hashing the contents of each sector making up each file. We then hash those sectors in each subsequent image, recording any that change from the original content. Our analysis produces decay curves (see Figure 1) for each deleted file so that we can associate actions taken between snapshots with deleted file content persistence or destruction.

This work is ongoing for mainstream operating systems with magnetic hard disks and solid state drives, flash media, Internet of Things devices, and mobile devices. Models of deleted file persistence may be useful for privacy analysis, e.g., how long PII can be expected to remain on a given device or media, or forensic investigations, e.g., how much of a deleted file I should expect to find after some time period has elapsed and similarly what are the implications of not finding parts of a deleted file after some time has passed.

B. Inferring Past Activity from Partial Artifacts

Related to our work modeling the persistence of deleted digital content is work to establish partial digital artifacts associated with specific actions. For example, if an application is installed, used, then uninstalled, and the system continues to be used to include reboots, do fragments of the application and associated files remain and can they be used to infer the past activity of interest? Our first step in this work was to use differential analysis to establish the full file artifacts associated with various applications and user activity. We created a common baseline image as before, but for this work we executed the activity of interest while capturing sequential snapshots. Differential analysis then produced a set of files which were created, modified, or deleted when each action was performed. The created and modified files were decomposed into sectors representing all possible sector hashes from about 20,000 files. We implemented a whitelist variation of the first noise reduction approach and built a catalog of partial digital artifacts associated with 16 applications on three Windows systems. We proposed three approaches to deal with this noise. First, we created a blacklist of files that change frequently but are not useful for our purposes, such as $MFT and pagefile.sys on Windows systems. We proposed three approaches to deal with this noise. First, we created a blacklist of files that change frequently but are not useful for our purposes, such as $MFT and pagefile.sys. Second, we developed a collection method where each snapshot was replicated and two parallel instances were run forward, one on which we performed the activity of interest and the other on which we performed no activity. Differential analysis between the snapshot start and each of these subsequent images produced two sets of new and changed files, one associated with the activity of interest and one considered pure noise for our purposes. Artifacts occurring in both sets were removed from the activity interest set, essentially removing the noise. Third, we proposed to do differential sector analysis instead of differential file analysis.

We implemented a whitelist variation of the first noise reduction approach and built a catalog of partial digital artifacts associated with 16 applications on three Windows variants, then post-processed the catalog to remove low entropy and high frequency sectors since they are unlikely to have high inferential value. The resulting catalog contains about 8 million sector hashes from about 20,000 files. We matched the partial artifacts in the catalog to the final file content and used these matches to estimate the past activity of interest.

![Figure 1: Sample Deleted File Decay Curve](image)
snapshots from the M57 patents scenario data set [2]. This data set consists of scripted scenarios for four systems with 17 sequential snapshots taken over 25 days, providing at least partial ground truth for installed and uninstalled applications. We developed and applied a weighted measure to account for the background frequency of each matched sector and obtained F-scores of 0.89, 0.75, 0.75, and 0.44 for application detection on the four systems in the data set [3]. Of particular note is our successful detection of the Advanced Keylogger malware on the Pat system, where the malware was installed on day 16 of the scenario, used for two days then uninstalled, and the system snapshot we processed was taken seven days later after continued activity to include system reboots. Of 4,716 sectors from 23 files in the catalog for the Advanced Keylogger application, only 24 sectors from 8 files persisted to the end of the scenario, yet we were able to successfully identify the past installation of the Advanced Keylogger application by applying our weighted sector matching algorithm (probability of application, P(A), of 21.97%). As indicated by the F-scores we obtained, our approach exhibited good sensitivity and specificity for three of the four M57 patents scenario systems. The fourth system was a different Windows platform than the other three, which is likely responsible for at least part of the weaker performance.

Our approach is predicated on the assumption that subsequent instantiations of the same instance type from the same provider should be initially identical except for user and network interface configuration details. This assumption includes allocated and unallocated hard drive space, although different but random patterns in unallocated hard drive space would not indicate remanence or persistence across user instances. We implemented automated tools to instantiate multiple instances of the same type from the same provider and compare the contents of the hard drives in each instance. We compare both allocated and unallocated space by sector. We found differences in the unallocated space of concurrent and sequential same-type instances. The differences in some cases were not random data but rather matched sectors from other operating systems, which we had profiled and sector-hashed previously from the same provider.

D. Hard drive image similarity

Digital forensic investigators in some environments are faced with large quantities of digital media to potentially analyze with a limited number of qualified analysts. Quickly typing incoming media (drives and drive images) as interesting or not can help triage the incoming flood and allocate limited resources where they are most likely to find information of value. One approach to such drive typing is through drive similarity, where incoming media are assessed for similarity to previous media of interest. For example, an organization might have a collection of media from cyber criminals or terrorist organizations, and incoming media which is similar to one of these two groups might be analyzed with high priority.

In this work, we are developing digital media similarity measures for this purpose. Given a set of previously labeled media (e.g., cyber criminal, terrorist, benign user, etc.), we compute sector hashes for this media. The sector hashes are post-processed to remove low entropy and high frequency sectors, and to tag sectors hashes based on source, e.g., operating system files, application files, user files, or unallocated space. A new media of interest is also sector-hashed, and a similarity measure is computed based on a weighted Jaccard similarity index, where the weighting is based on sector frequency and source. In this manner, newly received media can be quickly compared to existing media of interest and, if appropriate, flagged for high priority analysis. This work is ongoing; currently we are running experiments with sample media and our initial weighted similarity measure.

IV. CONCLUSION

The volume of data available to digital forensic analysts is staggering, yet the nature of that evidence and the systems generating it present unique opportunities. Differential analysis has proven to be useful in various applications, especially as a tool for associating specific activity or software with residual digital artifacts. Additionally, digital forensics has not historically used probabilistic inference, yet such methods are proving useful when applied to the large volumes
of raw data available on a typical digital device combined with massive amounts of background data that can establish valid statistical profiles.

We expect future work to continue to leverage analytic techniques which treat digital systems as masses of raw data rather than complex systems to be modeled. Such an approach requires minimal prior knowledge as the techniques are statistical in nature vs. other approaches which are more knowledge-based. We expect our approaches to be more flexible and generalizable than knowledge-based approaches, and also to perform well for digital data of unknown or damaged origins.

ACKNOWLEDGMENT

This paper reflects past and ongoing efforts by current GMU PhD students Tahir Khan, Brad Snyder, and Myeong Lim.

REFERENCES


SEDG: a soft data-science approach for assessing consequences of natural disasters

J. Tinghuaro Rodríguez, Begoña Vitoriano, Javier Montero

Abstract—An adequate initial assessment of disaster consequences is crucial for decision-making in disaster and emergency management. However, such an initial assessment needs to be correct, but not necessarily fully precise, and thus it can be associated with a soft (or fuzzy) classification problem, in which the set of classes presents a relevant structure. Furthermore, as historical disaster data is available through emergencies-oriented databases such as EM-DAT, it is possible to pose such a classification problem into a supervised framework, thus enabling a data-science approach to be developed in order to produce reliable and flexible initial assessments of both disaster consequences and the needs of the affected populations.

Index Terms— Fuzzy rule based classification systems, disaster management, dissimilarity.

I. INTRODUCTION

Just after a disaster strikes somewhere in the world, international disaster relief agencies (as OCHA – UNDAC) and NGOs (as IFRC) start a decision process intended to reach a conclusion about the pertinence of a relief operation and about whether suitable conditions exist (or does not) to initiate it. Therefore, the decisions to be made in a first stage have a strategic nature, more concerned with assessing the degree of involvement of the organization in a possible response operation than with the specific content of such an operation. However, precisely because they determine the shape and guidelines of the actions to be done, strategic decisions have a major influence on the subsequent logistical and on terrain decision processes (see Fig. 1), which evaluate the amount of aid to be sent and how it will be delivered to the affected country and the suffering population (see [2]). As a consequence, strategic decision-making takes place in a highly time-pressured context, since any delay at this stage could affect the position of the organization in the international coalition delivering aid, thus affecting the organization’s prestige and reputation, and slow down the subsequent decision processes, thus delaying the reception of aid by the affected population.

Notice that this strategic decision-making process has to be flexible, in the sense that it has to be able to be carried out for every combination of disaster type and place, since international disaster relief agencies and NGOs are possibly specialized in covering some part of the relief tasks (as water sanitation, shelter and site management, health care, etc.) but are not specialized in response to, for instance, earthquakes in Haiti, floods in Pakistan or any other specific disaster scenario (despite the geo-strategic priorities an organization could have as a result of the interests of its donors, see [6]).

Figure 1: Hierarchy of decisions for disaster response.

As shown in Fig. 1, strategic decision-making (and thus all the subsequent decision processes) is strongly dependent on a correct assessment of the consequences of a disaster and the resulting needs of the affected population. However, the available information just after the impact of a disaster uses to be affected by different kinds of uncertainty. The first reports are usually pretty incomplete, if not directly confusing and contradictory. Moreover, when it exists, relevant information is usually expressed linguistically, and thus it could be vague and imprecise. In fact, due to the effects an adverse phenomenon has on the informational system of a region, a more or less exhaustive and precise picture of the situation could not be obtained until some days (or even weeks) after the moment of the strike. This poses a strong difficulty in the development of the mentioned strategic decision-making process, since in this context the urgency of the decisions to be made in order to relief the people affected by a disaster clashes with the need of a correct estimation of the consequences of such a disaster, i.e. of the needs of the affected populations.

In the spirit of [13], all this complexity suggests as a promising alternative the development and application of inference techniques enabling a fast, flexible and correct assessment of disasters’ consequences in the presence of uncertainty. Nevertheless, some constraints must be imposed on the nature of these techniques. For example, the procedures leading to such an assessment have to be understandable and interpretable by the decision makers, in order to guarantee the usability of such an inference tool. And, even more important, it is necessary to make realistic assumptions about the infrastructure and data requirements of a DSS to be used in organizations or countries where the operational infrastructure cannot supply highly sophisticated data. It has to be remarked

This research has been supported by the Government of Spain, grant TIN2012-32482, and by the Government of Madrid, Grant S2013/ICE-2845. All the three authors are with the Department of Statistics and Operational Research, Faculty of Mathematics, of the Complutense University of Madrid, 28040 Spain. Their emails are {jitrodrig,bvitoriano,javier_montero}@mat.ucm.es.
that this question is a critical issue when considering disaster management from the point of view of NGOs or developing countries (see [1],[3]).

The long term objective of the project containing this work is to develop and implement a general, standard decision support system (DSS) for disaster management, specifically designed to address a part of this complexity and help NGO’s decision makers involved in the design of humanitarian relief operations. Particularly, this work focuses on SEDD (the Spanish acronym for Disaster Diagnostic and Evaluation System), which is the part of such a global DSS concerned with the assessment of the consequences of disasters with the very first information available after the strike (see [12] for a description of other parts of such a global DSS being developed in this project). As discussed in [10],[11] SEDD constitutes one of the first proposals aiming to provide NGOs with disasters’ consequences evaluation procedures specifically designed for them.

This work is structured as follows: Section II states that the strategic assessment of disaster consequences can be understood as a particular instance of a classification problem. Section III discusses the structure of the set of classes of such a classification problem, a discussion that is extended in Section IV concerning to how the decision-oriented requirements poses further constraints on such a structure, which be modelled and introduced into the classification models through the dissimilarity structures described in Section V. The exposed methodology will then be illustrated in Section VI by means of some application examples, and finally some conclusions are shed in Section VII.

II. STRATEGIC DISASTER SEVERITY ASSESSMENT AS A CLASSIFICATION PROBLEM

As explained above, the strategic decision-making about the involvement of an NGO on a disaster response operation is strongly dependent on the initial assessment of the consequences of such a disaster. However, a fully precise numerical evaluation of disasters’ effects, as casualties, homeless people or the extension of the material damage, is unrealistic in such a decision context. In fact, even a more or less complete and precise description of these consequences is usually not available by the time in which such (urgent) strategic decisions have to be already taken.

This is mainly due to the uncertainty and the referred features of the available information just after a disaster strike, but also because of the imprecise nature of some of the relevant categories. For example, the notion of affected people shows such an imprecision, since it might be not always clear whether a person has been affected or not. As a consequence, the number of affected people is usually stated through an implicitly imprecise quantity, as happens when it is said that a disaster produced, for instance, 40,000 affected people.

Nevertheless, it is important to notice that a totally precise and exhaustive evaluation of consequences is not actually needed in order to perform the above described strategic decision-making. As pointed out above, strategic decisions determine the shape of an operation but not its specific contents. As the decision process in Fig. 1 develops, the decision-making process needs information to be more and more precise, since decisions become more and more concrete. In this sense, NGOs usually deliver experts on the affected location in order to be able to acquire such a more precise evaluation for its logistical and on terrain decisions. But the decision of acquiring such a further evaluation is a strategic decision that has to be taken in the first moments after the strike, when little information is available, i.e. on the basis of the initial reports of the disaster. However, such an initial assessment of consequences needs to be correct or accurate, but not necessarily fully precise.

For example, consider the estimation of the variable number of homeless people. This variable measures the number of people that become homeless as a consequence of a disaster. For NGOs, such a quantity constitutes a key indicator of the size of the efforts a potential relief operation should place in matter of temporary shelter and site management. This also provides an idea of the efforts to be placed in the water sanitation area, for instance. In practice, at a first stage, for an NGO decision maker it is not important at all to distinguish whether 50,000 or 70,000 people became homeless as a consequence of a disaster, since anyway such a number is going to be considered large and the strategic decisions in terms of both the subsequent actions and the size and nature of the required relief operation are going to be similar.

Therefore, in the first moments after a disaster strike, NGO decision makers assess disaster severity (and thus also the needs of a potential intervention) in a qualitative (rather than quantitative) way. In other words, their problem consists on evaluating, in a context of highly uncertain and imprecise information, the magnitude of the consequences of a disaster in relation with the relevant scenarios and decisions that can arise regarding the implementation of a relief operation. In such a context, it is even possible that an assessment stated in crisp and precise terms could result little trustworthy to those decision makers.

However, a linguistic description of the magnitude order of the consequences coming from a reliable source, for example stating that there are a lot of casualties or that buildings took a severe damage, will be much more trustable to decision makers, despite its implicit imprecision. This kind of linguistic information is enough relevant to elaborate a first perception of the disaster scenario, providing a base for the subsequent strategic decision making.

Thus, in order to obtain such an initial assessment giving rise to an adequate strategic decision making, we consider that instead of a numerical evaluation, it is rather more plausible and realistic to classify the severity of the consequences of a disaster in terms of the relevant scenarios for the NGO’s decision makers. Therefore, the abovementioned practical problem of evaluation of disaster consequences leads to a classification problem in which the classes are identified with the linguistic terms that describe those relevant scenarios, as no casualties or a lot of injured people. These linguistic labels or classes, assessing the magnitude of the different relevant consequences of a disaster, have to be assigned on the basis of the description or attributes of such a disaster given by the
first available information, as the type of disaster, its intensity and the features of the affected location (e.g. its vulnerability, see [5]).

III. STRUCTURE OF THE SET OF CLASSES

Consider now one of the variables that have to be linguistically evaluated in order to obtain such a first initial assessment. For instance, let us focus on the variable number of casualties (CAS). This variable estimates the number of people that were killed as a result of the strike of an adverse phenomenon. As just explained, in a first stage a fully precise estimation of such a number is not strictly necessary, but just a qualitative, linguistic assessment. In this way, for instance we can measure the magnitude of a disaster scenario, in terms of the casualties it produced, by means of the labels no casualties, very few, few, quite a lot and a lot of casualties. These labels represent the classes in which such a disaster scenario has to be classified in order to provide an initial assessment of the relevant consequence CAS.

The particular meaning of these labels has to be specified by means of intervals, or more generally, through fuzzy subsets of the range of the underlying numerical variable. Moreover, such meanings have to be related to the different scenarios that are relevant in terms of the decisions to be made. For example, each label can be associated with a different order of magnitude of the number of casualties, in an increasing way, as shown in Table 1.

<table>
<thead>
<tr>
<th>Class</th>
<th>Label</th>
<th>Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td>CAS1</td>
<td>No casualties</td>
<td>[0,10)</td>
</tr>
<tr>
<td>CAS2</td>
<td>Very few</td>
<td>[10,100)</td>
</tr>
<tr>
<td>CAS3</td>
<td>Few</td>
<td>[100,1000)</td>
</tr>
<tr>
<td>CAS4</td>
<td>Quite a lot</td>
<td>[1000,100000)</td>
</tr>
<tr>
<td>CAS5</td>
<td>A lot</td>
<td>[10,000,∞)</td>
</tr>
</tbody>
</table>

Table 1. Intervals associated to the linguistic labels defined for the variable number of casualties (CAS).

Notice that, as they are associated to different orders of magnitude of the consequences of a disaster, these labels can be considered as ordered from the lowest (no casualties) to the greatest (a lot of casualties) level of magnitude of such consequences. This is, the classes associated to the variable CAS (in the first column of Table 1) are linearly ordered, i.e. CASi < CASj whenever i < j.

This assumption of linearity on the effects of a disaster is quite general, since what decision makers try to assess in a first stage is precisely the order of magnitude or severity of these consequences, or equivalently the order of magnitude of the efforts that a relief operations should consider in order to adequately alleviate such consequences. As explained above, these orders of magnitude are not necessarily powers of ten of the underlying numerical variable, but they are rather associated to the different relevant scenarios that may arise, which can be also considered as ordered attending to the gravity of the humanitarian crisis taking place in each of them.

Thus, in this setting classes are not independent, unrelated items, but they conform a valuation structure, in which some relationships hold between the valuation states given by the classes. This is, following [4], in this context the set of classes presents a relevant structure. For example, we do not commit the same error when a disaster scenario having no casualties (CAS1) is assessed as one with very few casualties (CAS2), than when it is evaluated as having produced a lot of casualties (CAS5). Therefore, the abovementioned problem of disaster severity assessment can be understood as a classification problem with a structured set of classes, i.e., as a structured classification problem.

IV. CONTEXT REQUIREMENTS

As stated above, the notion of structure of the set of classes is introduced in order to capture some relevant relationships that hold between the concepts represented by the classes. These relationships are given by the features of each particular application context, which provides the specific meaning or semantics attributed to the classes. However, it is important to notice that these relationships not only depend on the semantics of the classes, but they also have to reflect the criteria and requirements of the decision context.

For instance, the set of classes introduced before for the variable CAS in principle fits into a linear structure, since classes are semantically associated to orders of magnitude of the consequences. As a consequence of this assumption, the classifier could be required to show a gradable, smooth behavior, in the sense that small variations on the attributes that describe a disaster scenario should not produce a large variation on the predicted consequences. Some classes are closer than others to a given class, and thus different error levels can be distinguished according to such a distance.

However, notice that the error committed by assessing a CAS1 scenario as a CAS5 one is also different from that committed when a CAS5 scenario is evaluated as verifying CAS1. In the first case, overestimation error is committed, while in the second case scenarios are underestimated. Though the two types of errors are relevant, notice that underestimation of disaster consequences could lead to much more dangerous situations than overestimation in terms of the prestige of an NGO and the relief of the affected population.

In this sense, overestimation of consequences may lead to an initial overreaction, but as soon as observers are deployed on terrain and further information is available the scenario can be reassessed and the decisions reconsidered without too many difficulties. However, when a disaster scenario is underestimated, it uses to attract less attention and to be considered as less important, which could lead to not properly ask for further information or even to ignore it in a first moment, thus potentially affecting the timing of the strategic decision-making stage (with the resultant delays on the subsequent logistical and operational decision phases) as well as the NGO’s reputation.

Consequently, NGO decision makers usually tend to avoid the risk of underestimation of disaster effects, for instance by carrying out a worst-case analysis of the scenarios under study. In this way, initial assessments of a disaster scenario
could be required to be developed under the assumption of avoiding underestimation risk. In this sense, such a decision-related requirement entails introducing a somehow asymmetric configuration in the linear structure of the classes, since different error levels are then attained depending on whether a disaster scenario under study is underestimated or overestimated. Therefore, as pointed out above, the structure of the set of classes has to capture both the relevant aspects of the semantics of the classes as well as the objectives and requirements related with the decision context in which the classifier is used.

V. DISSIMILARITY STRUCTURES

Notice that the assumption of asymmetry on the linear structure of the set of classes forces to look for more general structures than orders. In this work, we illustrate the usage of the notion of dissimilarity structure proposed in [7] to provide a formal definition of the structure of the set of classes in a structured classification problem. Recall that dissimilarity structures are based on the notion of semantic antagonism (also proposed in [7]), that provide a formal framework to model the opposition relationships between a set of concepts in which such an opposition is allowed to be asymmetric. Therefore, by adopting the notion of dissimilarity or antagonism instead of that of linear order, we somehow translate the semantic distance between two classes, coming from the linear ordering of the consequences, into the degree of opposition among them, which is however allowed to be asymmetric in order to reflect the requirement of underestimation risk avoidance.

Moreover, as we shall see in next section, dissimilarity structures provide an easy and effective method of introducing the relationships between the classes into the classification models, i.e. in the learning and reasoning processes of the classifiers. In this context, the opposition between classes represented in the dissimilarity structure enables to distinguish significant exceptions to a classification rule from simple, logical counterexamples, which leads to introduce a negative confidence degree of the rules. Such a negative degree together with the usual, positive confidence degree, constitute then a bipolar evidence pair for the evaluation of classification rules, which leads to a bipolar fuzzy rule-based classification framework (see [8] for further details).

Therefore, let us denote by $\zeta = \{C_1, \ldots, C_N\}$ the set of concepts or classes into consideration. Recall that a dissimilarity structure can be built upon this set by means of a dissimilarity matrix $\Delta = (d_{ij})_{1 \leq i, j \leq N}$, such that the value $d_{ij}$ represents a distance from a too small sample. Let us start by assuming that no opposition relationships hold between the classes, i.e. by taking $\Delta = 0$. In this case, it is $r_i(A) = 0$ for every premise $A$ and consequent $C_j$, so it is $t_i(A) = r_i(A)$ for all rules $A \Rightarrow C_j$. Therefore, this case corresponds to a non-structured classification framework, in which a usual non-bipolar fuzzy classifier, identical to that used by SEDD in [9], is obtained. As no structure is assumed on the set of classes, the resulting classifier treats all the classes as independent items. Consequently, the classifier will be biased towards the classes with more training examples.

VI. ILLUSTRATIVE EXAMPLES

Here we illustrate the effect of different dissimilarity structures on the assessment provided by SEDD. Particularly, in order to be able to produce a picture of the assessments obtained by using each dissimilarity matrix $\Delta$, we drop the population density $POP$ from the set of independent variables, which leaves the variables $MAG$ and $HDI$ as the only explanatory variables to be used. Similarly, in this example we will focus on just one consequence variable, the number of casualties $CAS$, and on one type of disaster, earthquakes. Therefore, it is $n=2$ and $N=5$ (the same classes as in Table 1 are used for the variable CAS). The training sample for these explanatory and dependent variables is shown in Fig. 3, and it is $m=386$. As the maximum number of premises is quite small ($5 \times 5 = 25$), we adopt a grid-based learning procedure, i.e. rules are built for all possible premises. However, a support threshold ($\delta=0.01$) is defined in order to avoid those rules built from a too small sample.

In order to illustrate and compare the effects of each matrix $\Delta$, the behavior of the resulting VAI classifiers is simulated in a dense mesh of points of the input space of the attributes $MAG$ and $HDI$, in such a way that a picture of the predictions and class boundaries produced by each dissimilarity structure is obtained. Also, two error measures are used in order to measure the performance of the different classifiers: 1) $%CC$ represents the rate of correct classifications obtained over the training sample, thus evaluating the predictive accuracy of each classifier; 2) to measure both the deviation of the predictions from the real classes and the risk of underestimation, the average cost $AVCOST$ of the predictions is computed over the training sample, where the cost of classifying a instance from the class $i$ in the class $j$ is given by the element $COST_{ij}$ of the matrix $COST = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 2 & 0 & 1 & 2 & 3 \\ 4 & 2 & 0 & 1 & 2 \\ 6 & 4 & 2 & 0 & 1 \\ 8 & 6 & 4 & 2 & 0 \end{pmatrix}$

A. No dissimilarity

Let us start by assuming that no opposition relationships hold between the classes, i.e. by taking $\Delta = 0$. In this case, it is $r_i(A) = 0$ for every premise $A$ and consequent $C_j$, so it is $t_i(A) = r_i(A)$ for all rules $A \Rightarrow C_j$. Therefore, this case corresponds to a non-structured classification framework, in which a usual non-bipolar fuzzy classifier, identical to that used by SEDD in [9], is obtained. As no structure is assumed on the set of classes, the resulting classifier treats all the classes as independent items. Consequently, the classifier will be biased towards the classes with more training examples.
This is clearly shown in Fig. 2, where the results of the simulation are depicted. Note that a huge part of the input space of the attributes is assigned to the lowest class \( \text{CAS}1= \text{no casualties} \), i.e. that with the highest proportion (54.4\%) of training patterns. This entails a great risk of underestimation of consequences. Furthermore, the behavior of the classifier is not smooth at all, since predictions present sharp variations. Table 2 presents the performance measures for this non-bipolar classifier. Though we will use these results for comparison with the rest of classifiers, it is important to remark that almost all the correct classification rate (%CC=54.15) is due to examples of class \( \text{CAS}1 \). In fact, notice that %CC is in this case almost equal to the proportion of examples from the class \( \text{CAS}1 \).

![Figure 2](image.png)

**Figure 2.** Simulation result of the classifier with \( \Delta = 0 \).

<table>
<thead>
<tr>
<th>%CC</th>
<th>AVCOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>54.15</td>
<td>1.453</td>
</tr>
</tbody>
</table>

**Table 2:** Performance measures of the non-bipolar classifier ( \( \Delta = 0 \)).

### B. Restricted asymmetric linear order

In order to reproduce the semantics of linear order associated with the classes, let us now introduce the dissimilarity matrix

\[
\Delta = \begin{pmatrix}
0 & 0.2 & 0.2 & 0.5 & 1 \\
0 & 0 & 0.2 & 0.5 & 1 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

This matrix produces a structure in which the higher classes are gradually more and more dissimilar to the two lowest classes, \( \text{CAS}1 \) and \( \text{CAS}2 \), but not symmetrically, i.e., the lower classes are not dissimilar to the higher ones. Moreover, classes \( \text{CAS}3-\text{CAS}5 \) are completely unrelated between them. Therefore, in this situation the classes \( \text{CAS}1 \) and \( \text{CAS}2 \) receive negative information from the higher ones in a progressive way, but not conversely, representing the linear structure of the classes together with the requirement of avoid underestimation risk. As a consequence, classes \( \text{CAS}1 \) and \( \text{CAS}2 \) will obtain a lower veracity degree in the presence of the higher classes, thus requiring more evidence for the former classes in order to be predicted. However, if it is estimated that such a risk can be disregarded for the higher classes (e.g. if \( \text{CAS}3 \) and higher scenarios are always further assessed), then it is possible to restrict the linear order structure to the first two classes and allow the classes to compete freely between them, similarly to what happened for all the classes when \( \Delta = 0 \). These are the assumptions behind the matrix \( \Delta_1 \) above. Compared to those of the non-bipolar classifier in Fig. 2, the simulation results now show a smoother behavior, in which the class \( \text{CAS}3 \) (in light blue in Fig. 3) appears in the transition zone between the lower and the upper classes. Notice that, in general, the upper classes obtain a greater portion of the input space than before. In fact, as shown in Table 3, this classifier obtains an average cost of 1.069, thus reducing the underestimation risk of the non-structured case. Furthermore, this classifier obtains a better classification rate (%CC=55.18) than the non-bipolar one, i.e. the consideration of a dissimilarity structure leads in this case to a more accurate classifier than without it.

![Figure 3](image.png)

**Figure 3.** Simulation result of the classifier with \( \Delta = \Delta_1 \).

<table>
<thead>
<tr>
<th>%CC</th>
<th>AVCOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>55.18</td>
<td>1.069</td>
</tr>
</tbody>
</table>

**Table 3:** Performance measures for the restricted asymmetric linear structure ( \( \Delta = \Delta_1 \)).

### C. Worst-scenario analysis

Finally, let us consider the requirement of a total avoidance of the underestimation risk. To this end, consider now the dissimilarity matrix

\[
\Delta_2 = \begin{pmatrix}
0 & 1 & 1 & 1 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & 0
\end{pmatrix}
\]

This leads to an structure in which each class is totally opposite to all the classes lower than it, but not conversely. Therefore, in this setting there is not a explicit linear structure, but just the assumption of a total asymmetry between the lower and the higher classes. Each class receives the positive confidence of all the higher classes as negative information. Thus, the lower the class, the harder it is for such class to be predicted. In this sense, matrix \( \Delta_2 \) fits to the requirement of performing a worst case analysis of the disaster scenario under study. Notice that in this setting, no class is guaranteed to be predicted unless it attains a positive confidence \( r^* > 2/3 \),...
except the highest, which is predicted whenever its confidence is bigger than 1/3. Consequently, the class CAS5 appears for the first time in the simulation results of this classifier, as shown in Fig. 4. Note also the improved smooth behavior of the classifier, producing a soft transition between classes. In fact, a straight line could be drawn in the input space passing through all the classes in order. More importantly, this classifier enable to distinguish a clear trend in the consequences, in such a way that worst consequences are associated with lower HDI values (and thus with a greater vulnerability) and greater intensities of earthquakes. This trend is logically expected, but notice that no one of the previous classifiers could express it so clearly. The performance measures of this classifier, shown in Table 4, presents a further reduction of the average cost (AVCOST=0.968) and thus of the underestimation risk, that could be even more important since the rate of correct classification (%CC=50.52) is lower than before (with the subsequent increment of non-zero costs).

<table>
<thead>
<tr>
<th>%CC</th>
<th>AVCOST</th>
</tr>
</thead>
<tbody>
<tr>
<td>50.52</td>
<td>0.968</td>
</tr>
</tbody>
</table>

Table 4: Performance measures for the worst case scenario analysis (Δ = Δ3).

![Figure 4](image)

Figure 4. Simulation result of the classifier with Δ = Δ3.

VII. CONCLUSION

When considering the production of an initial assessment of disaster consequences as a structured classification problem, the structure of the set of classes gets closely related with the semantic features and decision requirements of the disaster management context. For this reason, an adequate characterization of this structure is an important step towards the adaptation of a classifier to those features and requirements. In this paper, we have shown that the introduction of a dissimilarity operator over the set of classes enables the consideration of several structures and their introduction in the classifier’s learning and reasoning procedures. As a consequence, a significant improvement in the accuracy and adaptation of the classifiers to the decision requirements of the disaster management context is achieved.

REFERENCES

An architecture for risk management decisions in aviation safety at state level

David Ríos
ICMAT-CSIC, 28049, Madrid, Spain
Email: david.rios@icmat.es

César Alfaro
and Javier Gómez
Royal Academy of Sciences,
28004, Madrid, Spain
Email: cesar.alfaro@urjc.es
javier.gomez@urjc.es

Pablo Hernández-Coronado
and Francisco Bernal
Spanish Aviation Safety and Security Agency
28020, Madrid, Spain
Email: phernandez@seguridadaerea.es
fbernal@seguridadaerea.es

Abstract—Aviation is a key industrial sector for global development. Safety is essential for its healthy growth and sustainability. However its management is pervaded by simplistic methods based on risk matrices. We provide here a framework and architecture for risk management decisions in aviation safety at state level, which takes advantage of various heterogeneous databases available.

I. INTRODUCTION

Organizations involved in aviation have been dealing with the prevention of accidents from the early days of this industry. Since its creation in 1945, the International Civil Aviation Organization (ICAO) has focused interests in trying to make aviation the safest transportation mode. Since 2004, accident rates have remained relatively steady, with no significant improvement, averaging between 4 and 5 fatal accidents per 10 million flights.

The total elimination of aviation accidents and serious incidents is a desirable goal, but clearly unachievable. The idea of risk-free systems has evolved in recent years towards a perspective centered around safety management, aimed at supporting resource allocation processes in which a balance between “production” and “protection” is attained. In this regard, it is worth noting that one of the most widespread methods for risk management in AS is based on risk matrices. We might think that the almost ubiquitous presence of these tools makes them a de facto standard. However, this methodology is criticized in the AS community (see e.g., [1]). The most complete analysis of their weaknesses is available in [2], who considers that the use of low resolution (5x5, for example) risk matrices with non-coherent colour schemes and subjective inputs can easily lead to erroneous risk management decisions.

In this paper, we propose a novel and systematic methodology for risk management in AS, based on the principles of decision and risk analysis, and fully exploiting available data sources, and outline how we implement such methodology in an architecture called RIMAS. The paper just sketches the key ideas with technical details in several companion technical papers.

II. FRAMEWORK

This section provides a framework to support the identification of AS risks and the resource allocation to mitigate them at state level. Despite the high safety level in the aviation industry, occurrence继续 to emerge. In our case, 88 different occurrence types will be considered, ranging from bird strike to runway excursion. Five occurrence classes are proposed by [3] depending on their severity: Accident (1); Serious Incident (2); Major Incident (3); Significant Incident (4); and Occurrence without safety effect (5).

Safety occurrences entail consequences. Each organization must examine those of interest to them for risk management purposes. In our case, the incumbent organization decided to focus on the following eight consequences identified as most relevant in AS management at state level: Fatalities associated with the functioning of the aviation system; Minor wounded persons associated with the functioning of the system; Severe wounded persons; Delays caused by safety occurrences; Cancellations caused by safety occurrences; Maintenance and repair operations produced by safety occurrences; Destroyed aircrafts; Image loss due to negative perception of occurrences. All of the above consequences have natural attributes except the last one for which we used as proxy the number of accidents, as they are the occurrences that would tend to appear in the news.

As required by ICAO, each state must elaborate an AS plan which essentially leads to a resource allocation mechanism aimed at improving AS in the incumbent country. As usual in public policy, resources are limited and one must determine the best allocation taking into account various relevant constraints (economic, technical, logistic, legal, political,. . .). Thus, our aim is to establish a state-wide AS plan to minimise the above objectives.

Given the current configuration of the aviation system, and taking into account the current AS state, a change in the resources allocated to different types of occurrences may have a global impact over AS and, therefore, possibly on the distribution of: the occurrence rates, hoping to make them smaller and, therefore, make occurrences less frequent;  

1ICAO defines “occurrence” to indicate an accident or an incident.
and/or the proportions of occurrence classes, in an attempt to make the more severe occurrences less likely; and/or the consequences, reducing the associated impacts, if these were to occur. These are evaluated with the loss associated with that AS performance and, overall, with the expected loss of the corresponding safety policy. We shall try to minimise such expected loss, see [4].

The problem we face is illustrated in the generic influence diagram in Figure 1 where as usual, see [5], rectangular nodes represent decisions; the hexagonal node is a value node; circle nodes represent uncertainties; and, finally, doubly circular nodes represent deterministic nodes.

Here, \( n \) designates the number of operations over the incumbent planning period and \( k \) is the number of occurrences considered; \( \lambda_j \) represents the rate of the \( j \)-th occurrence; \( x_j \), is the number of \( j \)-th occurrences; \( p_j = (p^1_j, \ldots, p^5_j) \) and \( s_j = (s^1_j, \ldots, s^5_j) \) represent vectors that designate, respectively, the proportions and numbers of \( j \)-th occurrences in each severity class; \( l_j \) designates the loss associated with the \( j \)-th occurrence; and, finally, loss represents the global loss.

We associate with each safety policy a portfolio of countermeasures \( z = (z_1, z_2, \ldots, z_k) \), where \( z_j \) will represent the proportion of resources (inspection time, personnel, investment, …) allocated to the \( j \)-th type of occurrence. Then, the rate \( \lambda_j \) of the \( j \)-th occurrence will follow a distribution \( f(\lambda_j | z) = f(\lambda_j | z_j) \); and the split into the five occurrence classes \( p_j = (p^1_j, \ldots, p^5_j) \) will follow a distribution \( f(p_j | z) = f(p_j | z_j) \).

Once the plan \( z \) is implemented, and given the number \( n \) of operations:

- \( x_j \) occurrences of the \( j \)-th type will emerge, split into \((s^1_j, s^2_j, s^3_j, s^4_j, s^5_j)\) occurrences in the five classes, with \( x_j = \sum_{i=1}^5 s^i_j \).
- The \( g_j \)-th occurrence of type \( j \), designated \( g_{ij} \), results in \( n^g_{ij} \) fatalities, with distribution \( f(n_F | g_{ij}) \); \( n^h_{ij} \) and \( n^h_{ij} \) minor and serious injured, with distribution \( f(n_{Hi} | g_{ij}) \); and, expected delay, distribution \( f(t_{D} | g_{ij}) \); \( n^{2g}_{ij} \) cancellations, with distribution \( f(n_{Ci} | g_{ij}) \); and, finally, \( n^{3g}_{Rij} \) destructions or repairs, with distribution \( f(n_{Rij} | g_{ij}) \).

- Overall, these lead to: \( n_F = \sum_{j=1}^k \sum_{g_{ij}=1}^l n^g_{ij} \) fatalities; \( n_{Hi} = \sum_{j=1}^k \sum_{g_{ij}=1}^l n^h_{ij} \), \( i = 1, 2, \) minor and serious injured, respectively; \( t_D = \sum_{j=1}^k \sum_{g_{ij}=1}^l n^g_{ij} \), accumulated delay; \( n_{Ci} = \sum_{j=1}^k \sum_{g_{ij}=1}^l n^{2g}_{ij} \), cancellations; \( n_{Rij} = \sum_{j=1}^k \sum_{g_{ij}=1}^l n^{3g}_{Rij} \), repairs; and, finally, \( s^1 = \sum_{j=1}^k s^1_j \) accidents.

- We would then evaluate these consequences with the loss function \( l(n_F, (n_{Hi1}, n_{Hi2}), t_D, n_{Ci}, (n_{R}, n_{Rij}), s^1) \).

Then, for portfolio \( z \), the corresponding expected loss \( \psi(z) \) associated with the influence diagram, would have the form

\[
\psi(z) = E(l(n_F, (n_{Hi1}, n_{Hi2}), t_D, n_{Ci}, (n_{R}, n_{Rij}), s^1)|z),
\]

with expectation defined with respect to the probability model defined in Figure 1. Similarly, we could evaluate the expected loss contribution of the \( j \)-th occurrence by limiting the consequences to just such occurrence.

A. Risk maps for screening occurrences

A first use of the above model allows for the screening of occurrences on which to focus the greatest efforts in AS risk management. To avoid the problems associated with risk matrices mentioned above, we use risk maps in which the (continuous) \( X \) axis refers to the likelihood of aviation occurrences and the (continuous) \( Y \) axis conveys the severity of occurrences associated with such occurrences. Specifically, we shall here represent the expected number of occurrences per 100,000 annual operations (\( \psi \)) and the expected loss associated with such occurrence (\( \psi \)).

Occurrences on the “anti Pareto” frontier of the risk map would require special attention, since there are no worse occurrences in both relevant risk management dimensions. Similarly, occurrences with higher expected severities or occurrence rates may seem worthy of attention since they are more costly and frequent, respectively. In addition, risk maps from consecutive years are useful to identify occurrences that have worsened their risk level. For an occurrence characterized by \((x_t, y_t)\) in the \( t \)-th year map, we would have that if \( x_t \leq x_{t+1}, y_t \leq y_{t+1} \), it would seem to have worsened, as it tends to be more frequent and costly; if \( x_t \geq x_{t+1}, y_t \geq y_{t+1} \), the occurrence seems to have improved since it tends to be less frequent and costly; whereas if \( x_t \geq x_{t+1}, y_t \leq y_{t+1} \), or \( x_t \leq x_{t+1}, y_t \geq y_{t+1} \), it would depend on how both criteria are weighted. Then, to screen AS occurrences on which to focus risk management, once the risk maps for years \((t-1)\) and \( t \) are built, respectively called map\(_{t-1}\) and map\(_t\), respectively we propose to: 1) Identify the occurrences in the anti-Pareto frontier of map\(_t\); 2) Add some of the occurrences in map\(_{t-1}\) that might produce higher losses; 3) Add also some of the occurrences in map\(_t\) that might be more frequent; 4)
Add those occurrences that worsened from map$_{t-1}$ to map$_t$;

5) Finally, include also novel occurrences emerging that year.

### B. AS resource allocation. Stochastic version

More importantly, we may provide a coherent safety resource allocation procedure based on the above elements. For simplicity, assume that only one type of resource is included, for example, based on inspection, which is indeed the main resource available to the organization at hand. Consider that a fraction $z_j$ of inspection time is allocated to address the $j$-th occurrence, with $z_j \geq 0$, $j = 1, \ldots, k$ and $\sum_{j=1}^{k} z_j = 1$.

Similarly, there may be additional constraints such as:

- **Minimum inspection level for each occurrence.** The organisation might require the inspection of at least a fraction $z_{min} \geq 0$ to address each occurrence. This would be formulated

$$z_j \geq z_{min}, j = 1, \ldots, k.$$  

- **Maximum inspection level for each occurrence.** The organisation could also establish a maximum level $z_{max} \geq 0$ to address each occurrence. Formally, we represent this constraint through

$$z_j \leq z_{max}, j = 1, \ldots, k.$$  

Then, we associate with each policy $z$ its expected loss $\psi(z)$ as in (1) and aim at solving

$$\min \psi(z)$$

s.t.

$$\sum_{j=1}^{k} z_j = 1,$$

$$z_j \geq z_{min}, j = 1, \ldots, k,$$

$$z_j \leq z_{max}, j = 1, \ldots, k.$$  

The optimal solution would be $(z^1_j, \ldots, z^*_{j})$, where $z^*_{j}$ would be the inspection fraction allocated to address the $j$-th occurrence, $j = 1, \ldots, k$. In order to compute the expected loss for a given policy, we would typically use a Monte Carlo approximation to (1) at a few portfolios and approximate its surface with a regression meta-model, optimising it subject to the above constraints.

The solution proposed in Section II-B may be excessively expensive from a computational point of view. A more affordable approach would use a deterministic version of the risk management problem based on, for example, the expected values of the relevant random variables.

### III. IMPLEMENTATION

We present some ideas about modelling the required elements in the nodes of the influence diagram in Figure [2].

#### A. Predicting the number of occurrences

We outline the class of models used to predict the number of occurrences of each type in a given period, typically, a year or a month. Note that AS planning is performed annually, but some of the occurrences present a seasonal (monthly) pattern. We focus on the case in which the occurrence rate is given as number of occurrences per 100,000 operations.

We use models in which both the number of operations and the occurrence rate evolve dynamically, as in Figure [2]. For such purpose, we combine in a novel way several standard models. Specifically, we use a Dynamic Linear Model (DLM), see [5], to predict the number of operations (upper block); a Poisson model to predict the number of occurrences given the rate and number of operations (midblock); and, finally, a DLM to predict the evolution of the occurrence rate (lower block). With this class of models, we are able to deal with the effects we have found in the evolution of rates for all occurrence types, mainly the possible presence of seasonal and trend components.

![Figure 2: Model to predict the number of occurrences](image)

To predict with these models we apply a particle filter, see [7].

#### B. Prediction of occurrence classes

Conditional on the number $x_t$ of occurrences, we must predict the corresponding occurrence classes which, as mentioned in Section II, are five. Let $p = (p^1, p^2, p^3, p^4, p^5)$ be a vector designating the proportion of occurrences of each class with $p^i \geq 0$, $\sum_{i=1}^{5} p^i = 1$. Let $s_t = (s^1_t, s^2_t, s^3_t, s^4_t, s^5_t)$ be a vector with the number of occurrences of each class with $s^i_t \geq 0$ and $\sum_{i=1}^{5} s^i_t = x_t$. Then, we use the multinomial-Dirichlet model.

#### C. Prediction of consequences

We must predict also the eight consequences for the different types of occurrences and five severity classes. The kind of issues we need to address is, for example, assuming that there
has been a bird strike occurrence of severity 2, forecast the number of minor injuries produced. In some cases, we need to make a distinction between the type of aircraft involved, for example, to predict more adequately, the number of fatalities. We use the following classification: T1, general aviation, aerial works, or business aviation, with less than 19 passengers; T2, regional flights (< 100 seats); T3, continental flights (< 200 seats); T4, intercontinental flights (> 200 seats). Bayesian forecasting models have been built in all cases.

D. Loss function

As for the loss function used to assess an AS plan, we use the concepts of measurable multi-attribute value function (8) and relative risk aversion (9) to obtain a utility function. First, we aggregate the consequences through a measurable value function. We then assume that the regulator has constant risk aversion with respect to \( v \).

IV. RIMAS: IMPLEMENTING THE FRAMEWORK

The framework has been implemented in R in a system supporting its application. The architecture includes modules to support various stages mainly:

- Forecasting models for the numbers of various (88) types of occurrences.
- Forecasting models for the occurrence severity classes.
- Forecasting models for the consequence.
- The construction of a multiattribute utility function to assess such consequences of occurrences.
- Screen riskier occurrences and assigning resources optimally to mitigate aviation hazards.

Connections to various heterogeneous required databases have been produced.

V. CONCLUSION

In striking contrast with the technological sophistication achieved in the aviation system from the aeronautical engineering perspective, risk management in AS is pervaded by unsophisticated methods evolving around the concept of risk matrix, with its potential pitfalls.

We propose a methodology for risk management in AS based on sound principles of risk and decision analysis [10]. We provide two versions of the general model, stochastic and deterministic, to be implemented depending on the level of accuracy required and the available computational resources. The methodology is useful in defining the countermeasures that allow us to manage the resources referred to in [11], minimizing the risks associated with AS, taking into account various constraints (economic, technical, logistic, . . . ) over such resources, and taking advantage of several heterogeneous data sources.

A decision support system implementation of the proposed methodology has been produced.
Geometric models for video surveillance in road environments: vehicle tailgating detection

Eduardo Pla-Sacristán, Iván González-Díaz and Fernando Díaz-de-María
Department of Signal Theory and Communications
Universidad Carlos III, Leganés (Madrid), Spain

Abstract—Traffic accidents constitute one of the main causes of death in many countries. Despite the current efforts devoted to mitigate the effects of road incidents, there are still some variables affecting this problem which are not yet under control or regulation. Spain, for instance, still lacks official regulations about especially risky driving behaviors, such as tailgating. In many cases, the rationale behind is that these behaviors are hard or expensive to detect reliably, thus limiting the extent of the automatic detection systems. This paper proposes a method to identify certain elements in road scenarios, define geometric models that allow computing quantitative measures of the scene and, consequently, detecting offending driving behaviors. In this work, we have focused on the particular case of study of tailgating detection. However, the proposed geometric models might become the basis of many other useful applications.

Index Terms—Computer vision, video analytics, traffic surveillance, geometric models, tailgating detection.

I. INTRODUCTION

Sin ce the widespread growth of automobile use worldwide during the XX century, road safety has become a key issue for modern societies. Many strategies have been proposed to address this problem, such as regulated driving licenses, restrictive signaling and, more recently, speed monitoring through radar systems [1]. The advances of technology have allowed traffic regulation entities to incorporate new preventive systems to real scenarios.

Among the broad family of technologies applied to road environments, computer vision systems have emerged as interesting solutions for certain problems due to their reduced cost and easy deployment. Video analytics and anomaly detection systems have been successfully implemented to properly detect some events, like traffic congestion, accidents or other anomalies that may occur in road scenarios [2]. Detection is usually achieved by clustering or classification of object trajectories in videos [3,4]. However, other lines of research also play a relevant role in anomaly detection, such as measuring microscopic traffic variables in real-time [5]. Anomaly detection systems have also been improved by machine learning techniques [6]. Computer vision systems are used, as well, for road segmentation purposes, with the intention of saving computational costs in subsequent video analysis [7].

However, the use of computer vision is still limited when accurate distance or speed measures need to be computed, mainly due to certain limitations caused by lack of camera calibration, the camera viewpoint, occlusions or visual artifacts caused by variable illumination. Hence, radar techniques are still predominant to compute quantitative measures in road environments, as they provide reliable data in terms of speed or spatial location, often at the expense of a notably higher price. Even more, other driving behaviors (reckless driving, running stops/red lights, etc.) that clearly affect road safety cannot be supervised with radar systems and, as a result, they are neither regulated nor monitored.

This paper proposes a methodology to apply computer vision techniques to those scenarios that require computing quantitative measures related to general traffic or specific vehicles. For the sake of usefulness, the developed system will adapt to any road environment with little human interaction, thus being robust under many realistic circumstances: CCTV operators controlling traffic cameras may often change their viewpoint, external factors, such as wind or heavy rain, may also affect the position of the camera, etc. Our proposal would adapt to this unpredictable changes minimizing user’s interaction.

In this paper, we have focused on one particular case of interest: vehicle tailgating. Although it represents one of the more frequent causes of traffic accidents [8], most countries do not present a clear regulation on this issue. Institutions in Spain, for instance, present recommendations on the distance considered to be safe in different situations [9], but no official regulations exist. However, much of the proposed processing pipeline could be easily applied to detect other offending driving behaviors without too much effort.

The remainder of this paper is organized as follows: in Section II we describe the method proposed to obtain the geometric models in which the proposed system is based and explain the case for which it has been tested (tailgating detection). In Section III, we describe the experiments that have been conducted so far, and the preliminary results obtained. Finally, Section IV provides a brief summary of the
II. PROPOSED METHOD

The proposed system, depicted in Fig. 1, is based on the analysis of a video flow. The scene is analyzed frame by frame and a convenient set of features are extracted from it. The main goal is to avoid the need of human interaction with the system, by means of an automatic adaptation to new scenarios and road environments. As shown in Fig. 1, the processing pipeline is composed of two main blocks: the first one is in charge of obtaining the geometric models associated with the road environment. For that end, elements of interest in the road are detected and classified to support the subsequent determination of a geometric model. The computed geometric models will later allow the second block to obtain and interpret data from the scene in order to calculate accurate measures. In the case study of this paper, values for an estimation of distance between vehicles and their speeds are obtained and processed to detect tailgating events.

It is worth noting that the current system for geometric modeling and tailgating detection builds over a previous system for video analytics in road environments. Hence, some blocks are also available providing useful information such as road segmentation masks, background models for the scene, foreground masks indicating vehicles or even tracks associated with moving vehicles. In the remainder of this document, we will refer data or information coming from this system as external.

A. Geometric models for road environments

The objective of this first block is to automatically compute geometric models based on the analyzed scene so that quantitative data can be obtained, regardless of the camera point of view, height, orientation or any other affecting parameter. Road scenes are usually monitored with large field of views cameras which allow operators to visualize the whole scene.

In order to compute the geometric models, we first need to identify some elements in the scene that will be used to calibrate the camera. No external elements (such as marks) are added to the scene, thus any element involved in the automatic calibration has to be intrinsic to the road. Moreover, the real dimensions of those elements have to be known a priori, as otherwise any measures obtained from the analysis would be inaccurate.

This paper uses discontinuous lines that divide lanes as the central elements to determine the geometric models, and the border lines that form the rest of the road markings as necessary supporting elements. The length of the discontinuous lines of the road is regulated and, therefore, can be known a priori [10]. This allows the system to calibrate the vertical dimension of the image. To calibrate the horizontal dimension, lane width is the chosen parameter, as this is also regulated [11]. However, note that the horizontal calibration serves mainly visualization purposes, as the relevant measures are taken along the vertical dimension.

The objective is to determine the geometric models to define a stable scene. Thus, a new model is only needed if the point of view changes, i.e. the operator changes the rotation angle or zoom of the camera, etc. From this premises, it is easy to realize that this first block will not need to analyze every frame of scene, but just the first one.

1. Road elements detection

The first step of the process, as shown in Fig. 1, is to detect the structuring elements of the road, namely, the set of white lines that define the limits of the road. For that end, an external input is used that consists of a background image of the analyzed scene (Fig. 2.a). Given the background image, a gradient-based analysis is carried out to detect the road lines that gives more importance to the gradient orientation in those
parts of the image for which the gradient magnitude is more significant. A weighted histogram with the orientations is obtained and those objects that do not follow the main orientation of the image are discarded. Hence, it is more effective detecting those edges corresponding to road lines.

2. - Road lines classification

Once a binary image identifying the objects of the image has been obtained (Fig. 2.b), the next step is to determine which of these objects are discontinuous lines (DL), border lines (BL) or none of them. For this end, we have developed two alternative approaches:

- **Automatic method**: this system automatically decides which objects are DL, which are BL, and which are discarded from further processing, thus avoiding any human interaction. The process is based on restrictions on the detected candidates. In particular, constraints are applied to characteristics such as area, orientation, eccentricity and location, so those objects matching the expected criteria are marked as DL or BL, respectively.

- **Semi-automatic method**: This method requires some degree of human interaction. The automatic system will first identify candidate elements and, then, the operator is in charge of labeling them as DL, BL or others. Note that the user will only have to choose from an already processed selection of objects so that no drawing or precise definition of areas is required from the user.

As a result of any of these methods, an annotated image is obtained where DL and BL are identified. Fig. 2.c shows a visual example of the different labels over the original image. Each one of the detected DL will be then analyzed separately, defining one sub-scene per discontinuous line.

3. - Key points identification

In order to support quantitative measures, it is necessary to transform the image from its original camera viewpoint to a normalized top-view (see Fig. 2.e) in which real and stable measures can be taken. This is done through a homography. In order to estimate the geometric transform relating both views, we need to identify at least four points in the original view whose coordinates are known in the top view. In our case, and as it is shown in Fig. 2.e, the proposed system uses a six-candidate-point method in an attempt to make the transformation more precise. However, it is noteworthy that not all of them have to be available to compute the transformation, thus providing a robust solution against visual artifacts.

Since DL becomes the central element of our algorithm, the two points in its extremes should be always available. Starting from these points, the other four can be obtained, if possible, from the nearby objects of the scene (BLs, or other DLs in case of multi-lane motorways). For that end, two orthogonal lines are drawn from the first two points to connect the analyzed DL with nearby BLs or DLs extensions. Fig. 2.d shows an example in which the six points are found in a DL and two BLs.

We would like to note that our approach is just an approximation and would be perfect only if the central DL and the nearby lines were parallel which, in general, is not completely true due to the projective view. However, it is well known that for planar surfaces (as the road) of small area with respect to their distance with the camera center, projective transformations can be successfully approximated by affine transformations which do not break the parallelism between lines. Under this hypothesis, the use of orthogonal lines from the DL to the nearby lines (DL or BL) will be precise enough.

It should be mentioned, however, that we have developed a method to add robustness to the point detection. The system is based on the RANSAC (RANdom SAmple Consensus) method [12]. The method here developed is not random, but it chooses the optimal combination of detected points in order to obtain the highest possible quality for the homography. Thanks to this method, we are able to discard possible outliers that may affect negatively to the resulting transform.

4. - Homography estimation

Once the points are identified, they are mapped to their equivalent points in the top view, whose coordinates are known a priori (we assume we know the length of DLs and the width of the road lanes). Given this set of matches, a
transformation matrix $H$ that relates the two views can be computed.

This transformation matrix defines the homography between the original view of the scene and the top view of the current sub-scene. In order to estimate this homography, the Direct Linear Transformation (DLT) is used [13]. The desired $H$ is a 3x3 Homography matrix such that:

$$1 \times Hx_i = 0$$ (1)

where $x_i = (x_i, y_i, 1)^T$ and $x_i' = (x_i', y_i', w_i')^T$ are the homogeneous coordinates of the key points in the original view and the normalized top view, respectively. From the eq. (1) it is easy to notice that vectors $x_i'$ and $Hx_i$, which correspond to a matched pair of key points (original view and top view), must be parallel, i.e. they must only differ in magnitude by a factor $w_i'$, no matter its value. Given the set of points $(X, X')$, the components of the matrix $H$ can be obtained by transforming the eq. (1) into an homogeneous linear system represented as a matrix-vector product, and applying Singular Value Decomposition (SVD) to it. The interested reader is referred to [13] for a detailed description of this method.

This process is applied to every DL detected in the image, so that the output of this block is a set of homography matrices that relate the normalized and original views of the sub-scenes associated with local areas in the road.

The following steps of the processing pipeline will extract data from the vehicles that pass through each of the analyzed sub-scenes. This is done in the second block of the system.

### B. Driving infraction detection subsystem: Tailgating

After the first block, the system is able to identify the relevant parts of the scene and transform them to a normalized domain (top view of the sub-scene). The goal of this second block (bottom row in Fig. 1.) is to compute real measures associated with the vehicles in the scene. In this work, we intend to measure two variables: the speed of the vehicles passing through the regions of interest (sub-scene), and the distance between adjacent vehicles. These two measures will become the basis on which building up the system for tailgating detection. However, it is important to notice that the first block of the system (geometric models subsystem) would be also useful for getting multiple statistics about the state of traffic in the scene.

#### 1.- Frame by frame transformation

As opposed to the first block, the second block analyses the scene through the video, this is, frame by frame. Hence, it is important to highlight that the process described in this section is repeated for each frame of the video. The input video was previously stabilized by the external system to compensate for vibrations caused mainly by the wind. Our system also uses a second input coming from the external system, which corresponds with a background model (road image without vehicles and other moving elements, as shown in Fig. 2.a) that is updated at each frame.

This first module transforms the regions of interest (sub-scenes) using the corresponding homography matrices obtained in the first block. The transformation is performed in both the original video and the background video, yielding the sub-scene with and without vehicles. This is illustrated in Fig. 2.g.

#### 2.- Background subtraction

This module performs a simple image subtraction between the current top view and its corresponding background. This difference is then thresholded to obtain a binary mask representing the foreground (FG) of the current sub-scene (Fig 2.h Left). This FG sub-scene gives us an idea of the present vehicles in the region of interest at the time corresponding with the analyzed frame. However, the system does not have a notion of each vehicle as an independent object. Instead, it considers just two types of regions (vehicle area and no-vehicle area).

#### 3.- Object alignment

The next step aims to identify individual vehicles in the scene. To achieve this, another input is gathered from the external system: the tracks of the vehicles in the road. Each track is a sequence of spatial locations and shapes (encoded as bounding boxes) that define the trajectory of a vehicle in the scene. The tracks are obtained online so that, at each frame we can get the bounding boxes associated with every vehicle in the frame. However, let us note that the goal of the tracking subsystem is to obtain an approximate location of a vehicle, lacking the precision needed to measure speeds or distances between vehicles.

This information about trackers is then paired with the foreground masks in the current sub-scene, so that each connected component is aligned with one vehicle in the tracker database. The output of this module is an image in which every vehicle present on the region of interest is identified (Fig. 2.h Right).

#### 4.- Event specification

In the particular case of tailgating, for those pairs of vehicles moving in the same lane, the system pursues measuring the distance separating them as well as their speed. To this end, this module has been in turn decomposed into three basic sub-modules:

- **Event detection**: it detects when two or more vehicles are present in the top view of the current sub-scene and evaluates whether they are driving in the same lane or not. If the system detects at least two vehicles in the same lane within the area of the sub-scene (Fig. 2.h), an event is created.

- **Event measurement**: once the event is detected, this sub-module measures the considered magnitudes. For the case of tailgating, these are the distance between both vehicles and the speed at which the rear vehicle is moving (it only takes into account the speed of the vehicle behind, as it is...
In order to obtain such measurements, the system establishes key points that define each of them. In the case of the distance, the calculation is based on the two points belonging to adjacent vehicles that are closer to each other. This is, the top of the vehicle at the bottom of the image and the bottom of the vehicle at the top of the image. As the domain in which the event is detected is normalized, we can neglect the horizontal coordinate and simply perform a subtraction between the vertical coordinates of each point.

It is worth mentioning that the distance between vehicles at a given moment does not require information from previous frames, so it is only measured when an event is detected. In contrast, the speed of the vehicles is obtained for every vehicle passing through a sub-scene, even if the vehicle does not need any further processing (there is no other vehicle in the same lane). This is done due to the need of at least two frames to compute the speed. Hence, the system requires measuring and updating the speed of every vehicle passing through a sub-scene. For this, it takes note of the front point of each vehicle, i.e. the most forward point belonging to the analyzed vehicle, and the frame in which is detected. From this set of points, it later estimates a motion vector that will be used to calculate the speed.

- **Event classification**: it takes as input the values that define the event (in this case, the speed and the distance), and it estimates the danger of the situation. Depending on this danger factor, the system decides if tailgating exists.

The specific factor used to decide on the danger is the **time of impact** $t_i$:

$$
t_i = \frac{d_v}{v_r} \tag{2}
$$

where $v_r$ is defined as the speed of the rear vehicle and $d_v$ is the distance between adjacent vehicles. The time of impact represents the time needed by the rear vehicle to impact the front one in case of a sudden stop. Assuming that both vehicles need the same time to stop once the brake pedal is pressed, then, time of impact is the time that the driver of the rear vehicle has to react to a sudden break of the previous vehicle.

5.- **System output**

The final system output is a set of pairs of vehicles driving in the same lane (Fig. 2.i) with their corresponding times of impact $t_i$. By setting up an absolute threshold over $t_i$, one could determine whether or not tailgating is taking place. Unfortunately, due to the lack of official regulation, this threshold is not specified and only recommendations can be found in the literature.

In Fig. 3 we show an example of the final output of the system, identifying involved vehicles, and measuring speed and distances.

### III. EXPERIMENTS AND RESULTS

A. **Dataset**

We have assessed the performance of the proposed method with different videos recorded by the CCTV system of the Spanish Traffic management Administration (DGT). The videos show real scenarios with varying traffic situations. As it has been stated before, this work corresponds with an ongoing research, and so the results are still preliminary.

The current dataset consists of seven locations. For each location, we have videos between 8 a.m. and 17 p.m. During this period of time, each camera changes its point of view at least once. As a result, we have obtained as much as 80 different background images in which the geometrical models can be calculated. Some examples are shown in Fig. 4.a.

B. **A comparison of methods for road lines classification**

A comparative study of the considered methods was performed. The results, despite being preliminary, clearly indicated that the semi-automatic method works better than the automatic method. This occurs under the assumption of human error absence. This means that, when the user properly indicates the system which objects are DL and BL out of the presented ones, the system will work better in semi-automatic mode than in automatic mode. This improvement is visually illustrated in Fig. 4 for different examples. In the images in the center (b) we see the results of the automatic method, whereas the right-hand images (c) show those of the semi-automatic method.
Table 1 shows a relation between the number of sub-scenes found in total (in the whole sample space) for each method. From a strictly objective point of view, we can confirm at first glance that the performance of the semi-automatic method is better, as the total number of homographies found with this method is almost twice the number found with the other. A subjective study was also conducted to evaluate the quality of the obtained transformations, yielding positive results for both methods, but again better for the semi-automatic method. In this subjective study, it was first observed if the detection was correct. As we can see, the automatic method usually has troubles with discontinuous lines that are not completely contained on the scene, and are therefore useless, which the user can immediately discard using the semi-automatic method. The automation of this kind of discrimination is reserved for future lines of work.

<table>
<thead>
<tr>
<th></th>
<th>Automatic</th>
<th>Semi-automatic</th>
</tr>
</thead>
<tbody>
<tr>
<td>Total</td>
<td>79</td>
<td>142</td>
</tr>
<tr>
<td>Acceptable</td>
<td>45</td>
<td>109</td>
</tr>
<tr>
<td>Excellent</td>
<td>11</td>
<td>30</td>
</tr>
<tr>
<td>Defective</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td>Incorrect</td>
<td>20</td>
<td>0</td>
</tr>
</tbody>
</table>

Table 1. Total number of homographies calculated.

C. Evaluation of Tailgating detection

Concerning the second block of the system, it is important to note that, currently, we do not have an objective method to evaluate its performance as we lack ground truth information about vehicles speeds and distances. In the forthcoming months we plan to use alternative technologies, such as radar, which have shown great performance to measure distances and speeds, and generate an annotated dataset.

In any case, a direct observation of the results over various traffic sequences has allowed us to check that the system successfully detect those cases where tailgating is very clear.

IV. DISCUSSION

In this paper we have developed a system capable of determining general geometric models to transform traffic scenes to a top view where taking measures is feasible, regardless of the scene under analysis. This allows us to detect offensive driving behaviors that currently lack a proper preventive system.

The data obtained from the tailgating detection system is very promising, as this driving behavior clearly presents a threat to road safety, and yet it has not been regulated. This is partly due to the impossibility to monitor it and sanction it. With this automatic system, the possibility of accurate and affordable monitoring of tailgating behavior is closer. In this paper, a study case for tailgating detection has been proposed, but future lines of investigation could open a wide variety of possibilities. The system could be oriented to monitor many different driving behaviors, such as reckless driving, changing two lanes at once, overtaking vehicles on the right, etc.

REFERENCES


The Evolution of Healthcare: First Steps Towards Cognitive Healthcare

Agusti Solanas

Abstract—Technological advances affect almost every aspect of our daily lives. The healthcare sector is not an exception and it has been deeply transformed as a result of the adoption of information and communication technologies. Since the generalization of electronic health (e-health) in the beginning of this millennium, healthcare has rapidly evolved by embracing new technological advances. The adoption of mobile and ubiquitous technology led to the appearance of mobile health (m-health). Later, the spread of smart solutions in context-aware environments gave birth to the paradigm of smart health (s-health) and, currently, the omnipresent connectivity provided by the Internet of Things and the tremendous computational and storage capabilities, that allow Big Data Analysis and Advanced Artificial Intelligence, open the door to the ultimate healthcare breakthrough: Cognitive Healthcare.

In this article we briefly describe the evolution of healthcare from classic healthcare to smart health, and we introduce the new paradigm of Cognitive Healthcare understood as a highly distributed and interconnected model of healthcare that surpasses individuals' capabilities and places healthcare in a new collaborative plane in which humans and machines interact in order to achieve unprecedented levels of healthcare excellence.

Index Terms—Cognitive healthcare, Electronic health, Mobile health, Smart health.

I. INTRODUCTION

Our daily activities are deeply influenced by information and communication technology (ICT). Transportation and logistics, education and learning, sustainable management of resources, government and citizens’ participation, and healthcare are just a few examples of areas that have changed profoundly due to the use and generalisation of ICT.

It is apparent that ICT has transformed, and still reshapes, systems (and systems-of-systems) affecting a number of global processes and people worldwide. However, a not so obvious influence of technology emerges from its decisive role in modifying behaviours and individual processes (both physically and mentally). In what follows, we concentrate on three behaviours/processes that explain the line of reasoning that ended up with the emergence of the Cognitive Healthcare concept, namely the way people learn (i.e. learning), the way people live (i.e. living), and the way people take care of their health (i.e. healthcare).

As previously stated, there is no doubt that technology affects a plethora of areas, but in this article we only focus on learning, living and healthcare since those are the ones that interweave to create the fabric of Cognitive Healthcare.

With regard to learning, the wide adoption of technology motivated Siemens to propose the concept of connectivism in [1] as an evolution of previous learning paradigms like behaviourism, cognitivism, and constructivism. In opposition to those approaches, connectivism takes into account the impact associated with the emergence of technology. In this line, we sustain that the process of learning (of a human being or a machine) is no longer centred on the individual entity, who gathers information to generate knowledge, but on a globally interwoven system, which provides access to knowledge and allows its sharing.

Therefore, technology has changed the way we learn, but it has also deeply influenced the way we live. We have witnessed a population shift towards cities and the urbanisation process accelerates worldwide. Cities are crowded and the use of ICT has been understood as the most promising mean to cope with the challenges associated with a very densely gathered population. As a result of the adoption of technology within cities, the concept of smart city appeared. The main aim of smart cities is to improve sustainability, e-participation, economy, quality of life, and so on. To do so, they rely on ICT so as to gather data and convert it into information that help local governments to make better decisions [2]. Combining the principles of connectivism and smart cities appears the idea of cognitive city, proposed by Mostashari et al. [3], who claim that these are cities that learn, adapt their behaviour, and are able to sense, understand and respond to changes in their environment. Note that the main difference between a smart city and a cognitive city is the ability of the latter to learn. It is in this learning process where we find the contact point with the connectivism paradigm proposed by Siemens.

Last but not least, we pay special attention to healthcare and the way it has changed as a result of the adoption of ICT. It is widely known that healthcare systems in developed countries are struggling to manage a growing and ageing population with high percentages of chronic illnesses. With this main goal in mind, ICTs have been put in place to reduce costs and improve the quality of life of patients and that of their relatives and caregivers. Hence, classic healthcare has been transformed into electronic health (e-health) and the adoption of mobile

A. Solanas is the Head of the Smart Health Research Group in the Department of Computer Engineering and Mathematics of the Rovira i Virgili University, Tarragona, Catalonia, Spain. http://smarthealthresearch.com (e-mail: agusti.solanas@urv.cat · agusti.solanas@gmail.com).
technology has given birth to mobile health (m-health). Quite recently the synergy between smart cities and e- and m-health led to the appearance of smart health (s-health), which aims at fusing the context-aware capabilities of smart cities with the ubiquity potential of m-health. Figure 1 shows an overview of these learning, living and healthcare paradigms, their relations, and the decisive influence of technology.

From a global perspective, we think that there is little doubt about the profound transformations that ICTs have induced in our daily lives. They have changed how we learn, how and where we live, and how we take care of our health. If we have a look into these areas (i.e. learning, living, and healthcare) individually we might see great changes. Notwithstanding, if we approach them from a global interwoven perspective the change that we will observe is even more important and it will have a greater impact on the society as a whole.

A. Contribution and plan of the article

In this article we introduce a novel paradigm of healthcare that we like to call cognitive healthcare (a.k.a. c-health / c-healthcare). This new healthcare paradigm is the result of the evolution and transformation of learning theories through connectivism, the advent of smart and cognitive cities, and the appearance and consolidation of smart health.

The rest of the article is organised as follows: in section two we briefly summarise some background and concepts that interweave to create the idea of cognitive healthcare. Next, in Section three we propose a definition for the new concept of cognitive healthcare and describe its rationale, its main actors and their interactions.

Since this is an introductory article, whose main goal is to introduce, discuss and spread the concept of cognitive healthcare, we do not include experimental results or technical analyses of real solutions and applications of the concept. On the contrary, we prefer to provide the reader with open ideas and food for thought. Hence, in Section four we point out to a non-comprehensive list of open challenges and future opportunities for the cognitive healthcare paradigm that we think that will be studied and developed in the years to come.

II. Background overview

This section briefly discusses the fundamental concepts that led to the idea of cognitive healthcare.

A. Learning Theories before & after Information Technology

Through history, learning theories have been influenced by philosophy, psychology, social structures, and scientific and technologic advances. As society changed, so learning theories did. Before the generalisation of information technology, knowledge grew quite slowly and learning paradigms like behaviourism, cognitivism and constructivism made perfect sense. However, nowadays, the so-called half-life of knowledge (i.e. the time between the acquisition of knowledge and its obsolescence) is shrinking rapidly. Thus, a new way to understand learning processes is necessary.

For the sake of brevity and assuming the risk of crossing the boundaries of oversimplification, we could say that behaviourism (in parallel to objectivism) states that reality resides outside the learner and that he/she acquires knowledge by experience. Following the same assumption, cognitivism (in parallel to pragmatism) goes one step forward and says that knowledge is acquired by the learner through experience and reasoning. On the contrary, constructivism (in parallel to interpretivism) assumes that reality is internal to the learner and, thus, knowledge is constructed by the very learner.

These learning theories have their pros and cons, and respond to several thinking trends. However, they do not consider the influence of technology and, hence, they might be considered outdated. On the contrary, connectivism is a learning theory that explains how ICT generates new ways for people to learn and share their knowledge. In the words of Siemens [1]:

“Connectivism is the integration of principles explored by chaos, network, and complexity and self-organization theories. (...) Learning (defined as actionable knowledge) can reside outside of ourselves (within an organization or a database), is focused on connecting specialized information sets, and the connections that enable us to learn more are more important than our current state of knowing”.

In this sense, connectivism supports the idea of on-line, distributed, and community-based learning (e.g. massive open online courses (MOOC)). All in all, connectivism suggests that the very act of learning is no longer an interior/exterior process to gather or create knowledge but a multilayer, connected, and inherently dynamic way to understand the very concept of knowledge and its sharing.

B. Smart and Cognitive Cities

Smart Cities are, according to the definition of Caragliu et al. [4], those that invest in human and social capital and in traditional (transport) and modern (ICT) communication infrastructures that fuel sustainable economic growth and a high quality of life, with a wise management of natural resources, through participatory governance. This definition was augmented by Pérez et al. in [5] by considering the
fundamental right to privacy in order to avert the Big Brother effect. The definition reads as follows:

“Smart Cities are cities strongly founded on ICT that invest in human and social capital to improve the quality of life of their citizens by fostering economic growth, participatory governance, wise management of resources, sustainability, and efficient mobility, whilst they guarantee the privacy and security of their citizens”.

After the definition of Caragliu, the concept of cognitive city was defined by Mostashari et al. [3] as:

“(a city) that learns and adapts its behaviour based on past experiences and is able to sense, understand and respond to changes in its environment”.

The logical flow of actions within a cognitive city are: (i) sense internal/external changes, (ii) perceive the overall picture that these changes imply, (iii) relate the new situation to past experiences and find potential responses, (iv) plan various alternatives in response to the change, (v) choose the best course of action, (vi) take action by adjusting resources to various alternatives in response to the change, (vii) choose the picture that these changes imply, (iii) relate the new situation sense internal/external changes, (ii) perceive the overall behaviour of the action taken and learn from its impact.

In essence, cognitive cities differ from other ICT-based cities in that cognition implies additional properties such as the existence of learning, memory creation and experience retrieval.

C. Electronic, Mobile and Smart Health

As we have stated in the introduction, our conception of healthcare has deeply changed in recent years due to the wide use of ICT. Figure 2 represents a conceptual scheme of the relations between the diverse paradigms of health that are next summarised. Complex systems comprised by computers, sensors, databases and many other devices give shape to what is known today as electronic health (e-health). Eysenbach et al., in [6], defined e-health as:

“... an emerging field in the intersection of medical informatics, public health and business, referring to health services and information delivered or enhanced through the Internet and related technologies.”

Although e-health is no longer an emerging field, it is important to emphasise the fact that it is definitely multidisciplinary and has helped (and still contributes) to improve healthcare processes (both from an economical and assistive perspective).

Moreover, the adoption of mobile technology (e.g. smartphones, tablets, RFID, etc.) within the healthcare sector gave birth to the concept of mobile health (m-health) that could be defined in the words of Istepanian et al. [7] as:

“... emerging mobile communications and network technologies for healthcare systems”.

Like in the definition of e-health, we think that m-health can no longer be considered an emerging field at all, and it should be regarded as a consolidated area of research and application. Clearly, it augments the capabilities of indoor monitoring provided by e-health, and allows the continuous evaluation of inpatients and outpatients. We can find many examples of m-health solutions: from the remote monitoring of elderly with mild cognitive impairments (MCI) to mobile apps to foster medication adherence and healthy habits.

Finally, as an evolution of e-health and m-health, recent advances in smart cities inspired the concept of Smart Health (s-health) proposed by Solanas et al. [8], which is defined as:

“the provision of health services by using the context-aware network and sensing infrastructure of smart cities.”

It is worth noting that smart health, as a concept, goes far beyond the boundaries of smart cities and can be understood within any smart environment with context-awareness (e.g. smart houses). However, a smart city is the perfect environment for the idea of smart health to flourish. Specially because a smart city is a system-of-systems with multiple interactions that can be improved, tuned and augmented to provide better services.

III. THE COGNITIVE HEALTHCARE CONCEPT

The study and combination of the concepts summarised in the previous section drove us to the conception of the idea of Cognitive Healthcare (c-health), which we like to define as follows:

“healthcare services provided within a context-aware, cognitive environment, where the underlying systems, infrastructures, and intelligent beings are able to adapt to changes, learn, and make decisions based on experience, collaboration and accessible and actionable knowledge.”

We briefly stated before that the main difference between smart health and cognitive healthcare relies on the capacity of the latter to learn from experience and integrate decision-making processes to choose amongst self-generated plans and
analyse the outcomes. Thus, the cognitive healthcare paradigm implies the existence of learning strategies, knowledge management and decision making strategies in a complex system comprising dynamic resources, machines and intelligent beings.

A. Rationale

Healthcare systems have been continuously adapting to new needs and technologies. The concept of s-Health [8] implies a paradigm change and assumes the exploitation of multiple information flows governed by the context. In essence, it is assumed that citizens receive data from the city and make decisions based on these data but, also, citizens send data to the city that can act accordingly. It is widely known that the environment affects humans and justifies the attention that is paid to it. For example, in [9], Perera et al. show that the early exposure to polycyclic aromatic hydrocarbons (PAH) might lead to Attention Deficit Hyperactivity Disorder (ADHD) and Riaño and Solanas in [10] explore the relation between environmental factors and diseases like Chronic Obstructive Pulmonary Disease. All in all, contextual data, collected in Smart Cities, is a rich source of information that could be used to improve population's health.

Cognitive healthcare is built upon the smart health concept by adding further capabilities. It takes advantage of the concept of knowledge, acquired through a connectivist approach, and considers that healthcare knowledge is no longer in the hands of doctors and hospitals only, but distributed amongst multiple parties (e.g. patients, relatives, databases, clinical guidelines, machines, etc.) and there is a need for sharing it and for deciding which knowledge is important in each situation.

B. Main actors and their relations

In a Cognitive Healthcare system, we distinguish three main kinds of actors:

- **Individuals**: Citizens/users/patients are essential elements of the system, since they are the main reason that justifies the existence of the concept. We assume that individuals are active and/or passive (depending on the situation).
  - **Active individuals** contribute with their data and collaborate with others. Also, they initiate processes that revert on their own health. For example, a runner that asks for suggestions about the healthier route to take considering the current state of air pollution.
  - **Passive individuals** receive information from the other actors and decide whether to modify their behaviour accordingly, but do not initiate processes neither contribute with data or knowledge. For example, citizens subscribed to an alert service that receive an informative message when allergens (e.g. pollen) are detected in the city.

- **Collectives**: Those are other citizens/entities that collaborate. In this category we could find a wide variety of cases. For example, if we consider human beings, we might find a group of nurses, doctors, professional caregivers, etc. If we consider “things”, we might find nursing homes, intermediate care facilities, pharmacies, etc. They all contribute by sharing knowledge in the form of documents, databases and experience.

- **Cognitive Environments/Cities**: The infrastructure of a cognitive environment (e.g. a cognitive city) includes sensing capabilities to gather contextual data (like in the s-health model), and also learning processes and reasoning. Although the infrastructure might include human beings, it must also contain intelligent machines that adapt to changing conditions, create plans and make decisions based on previous experience and knowledge. For example, the city infrastructure might include an expert system that monitors traffic conditions and correlates them with humidity and wind speed data to predict the concentration of pollutants in different quarters of the city. This way, it could help citizens to make decisions during their commuting time, and can also modify traffic lights and speed limits to cope with high levels of pollution. Moreover, it could improve and suggest routes for ambulances to reduce the time to reach an accident in real-time and using its previous experience.

Figure 3 shows an illustration/scheme of the Cognitive Healthcare environment, where the three main actors are depicted. It is paramount to note that in this scheme, knowledge is no longer individual only. On the contrary, all possible forms of knowledge (e.g. documents, databases, personal experiences, standards, clinical guidelines, timetables, recommendations, etc.) are shared and the actors of the system have their own ways of learning. For example, an individual user might learn from experiences shared by other citizens, and an expert system in the cognitive city might learn from mobility data shared by individual users.

Every actor of the system is able to learn individually thanks to the interaction with the others and their experiences. The learning capability does no longer resides in human beings only but also in machines. Moreover, the cognitive healthcare system, understood as the sum of all the actors and their relations, learns and the overall knowledge of the system grows. Thus, this global knowledge increase benefits all the actors and helps them all to make better decisions for their
health. In the spirit of connectivism, the capacity of the cognitive healthcare system to learn and know more is more important than the current state of knowledge of the system. Hence, the interaction and the capability to gather knowledge from multiple interwoven parties in a dynamic and reactive-proactive way constitute the core of the concept of cognitive healthcare.

IV. OPEN CHALLENGES AND CONCLUSIONS

The cognitive healthcare paradigm opens the door to formidable difficulties in a variety of fields. The following is an indicative, non-comprehensive list of research challenges that remain open and might be studied in the near future:

- **The human factor:** The interaction between machines and human beings to achieve common goals is a great challenge. Human beings have to interact with cognitive environments and exchange knowledge with them. Information flows become multidirectional and knowledge moves from humans-to-humans, from machines-to-humans and from machines-to-machines. It has been shown that this is not a trivial issue. For example, in the case of patients monitored in control environments [11] Curtis et al. showed that human factors are very relevant and they emphasise the importance of user acceptance and expectations of multiple stakeholders.

- **Transfer learning:** Like humans, machines need time, examples (experience) and training to learn. However, it has been shown that leveraging previous knowledge from similar domains might help to improve learning rates. Hence, when dealing with multiple machines learning in different domains it becomes very important to optimise the procedures for transfer learning from previously trained machines into newly deployed devices. By doing so, machines can “inherit” knowledge acquired by other machines and minimise cold-start-related issues. However, this is a very tough problem when applied to healthcare because although healthcare domains might be similar they are very complex and transfer knowledge from one to another might require extensive experimentation.

- **Dynamic sensing:** Currently, most sensors deployed in cities and context-aware environments are static. This implies that the infrastructure can hardly adapt to changes in the environment (e.g. population increase in certain areas, demonstrations, accidents). It is necessary to develop and deploy dynamic sensors systems able to relocate themselves depending on the needs and events in the environment. Autonomous robots and more specifically drones are good candidates for this job since they can be relocated depending on current needs and based on the experience previously obtained by the city. In the healthcare context, the city is able to learn and transfer knowledge to dynamic sensors (e.g. drones). Next, those dynamic sensors relocate themselves to better and more precisely collect data (e.g. about pollution in streets with dense traffic). With these data, intelligent algorithms can suggest alternative routes to citizens so that they can avoid polluted areas and reduce their impact on their health condition.

- **Advances in Artificial Intelligence:** As a general thought, we might say that there are many areas of Artificial Intelligence that must keep advancing so as to make cognitive healthcare a reality. Some fundamental areas that we do not discuss for the sake of brevity are: Multi-agent systems, Decision support systems, Pattern recognition, Knowledge representation, Complex networks, problems management and conflicts resolution.

- **Advances in Communications:** Although great advances have been done in wireless communications technologies (e.g. LTE), the huge amount of devices that will be connected (i.e. following the idea of the Internet of Things) and exchanging information pose a formidable challenge for communications in all layers of the OSI architecture from the physical to the application layer.

In this article we have introduced the concept of Cognitive Healthcare as a new paradigm of healthcare in cognitive, context-aware environments with learning, decision making and multi-party collaboration capabilities. Clearly, this is an emerging concept that is not a reality yet. However, governments, scientists, engineers, and healthcare stakeholders have a great opportunity to embrace the concept now and make it real in the years to come for their benefit and that of the society as a whole.

**REFERENCES**


Active Sensing in Human Activity Recognition

Alfredo Nazábal, Antonio Artés-Rodríguez, Senior Member, IEEE

Abstract—This paper considers the problem of energy consumption reduction of inertial sensors in a Human Activity Recognition (HAR) system. We implement a HAR system using Hidden Markov Models and evaluate the entropy of the posterior of the activities to decide when to perform the acquisition of new data. This problem is intractable in general, so we develop and implement three different algorithms to find numerically these data acquisition events. We compare the performance of these algorithms in two HAR databases and show that we obtain a significant reduction on the number of observations, thus reducing the energy consumption, while maintaining the performance of the system.

Index Terms—Active Sensing, Human Activity Recognition, Hidden Markov Models, Maximum entropy, Energy consumption

1 INTRODUCTION

Human Activity Recognition (HAR) is a field that has grown considerably in the last years, illustrating its huge influence in many modern social applications including ambulatory monitoring of elderly patients, human behaviour characterization or camera surveillance applications [1]. For example, in a monitoring application the knowledge of the activities being performed is critical to evaluate the context in which the patients are being monitored. The target activities of these applications range from simple activities such as walking, sitting or standing to more complex activities such as eating, sleeping or brushing your teeth.

Wearable sensors are a common approach for long term monitoring HAR systems [2], [3]. The most popular ones are inertial sensors, either used by themselves or combined with other multiple sensors, as the systems used in applications like Parkinson symptoms detection [4] or gesture recognition [5]. Furthermore, smart-phones are gaining popularity in HAR systems as they include several built-in sensors that provide information of the daily activities of its user [6].

Sensor-based activity recognition systems present several inherent problems, like the position of the sensors in the body, the noisy nature of the input observations or the energy consumption of the devices. The energy consumption problem in smart-phones is an important topic [7], as the HAR system shares its resources with the rest of the applications in the device and the embedded sensors are primary sources of power consumption. During short periods of time, it is reasonable to use all the information provided by the built-in sensors, however, when continuous monitoring is considered, the optimization of the battery and memory of the devices used is of critical importance. In this paper, we study how to optimize jointly the energy consumption of the sensors and the performance of the HAR system.

The main approach to reduce this energy consumption is to minimize the data acquisition of the sensors. In [8] the authors propose a framework that uses a hierarchical sensor management strategy to recognize user activities as well as to detect activity transitions and decide which sensors to use at any given time. Instead of the ad hoc approximation considered in [8], we follow a systematic approach to decide when to perform the data acquisition. In this work, we consider a HAR system based on Hidden Markov Models (HMMs). At a sampling rate of tens of Hertz, the posterior probability of the activities given the observations is concentrated on a single activity during most of the time. Accordingly, the uncertainty of the performed activity is low, as well as the entropy of the posterior probability. When no observations are available, the posterior probability tends to the stationary distribution of the HMM, and the entropy of the posterior probability increases. We propose exploiting this fact to define an active sensing strategy: to stop acquiring observations when the entropy is low, and to estimate the time instant when the entropy reaches a certain threshold and we need to start acquiring new samples. This is a reasonable assumption in a long term activity recognition system, where some activities are performed for long periods of time and only a few data samples are needed to recognize the activities.

The rest of the paper is organized as follows. Section 2 introduces the problem and the notation used in this work. Section 3 presents a preliminary result using the properties of the entropy and the transition matrix. In Section 4 we propose three algorithms to solve our problem numerically. Section 5 shows results of our algorithms in two HAR databases and Section 6 concludes this paper.

2 PROBLEM STATEMENT

We define $X = \{x_1, x_2, \ldots, x_N\}$ as a sequence of $N$ vector observations of dimension $D$, $x_n \in \mathbb{R}^D$. We use a HMM (Figure 1) to model $X$, where we define $S = \{s_1, s_2, \ldots, s_M\}$ as the sequence of hidden states that explain the data, with $s_n \in \{1, \ldots, M\}$ and $M$ being the
number of possible states. A HMM is characterized by three parameters [9], the initial probabilities distribution \( \pi = p(s_1) \), the transition matrix \( T \in \mathbb{R}^{M \times M} \), with elements \( t_{ij} = p(s_{n+1} = j | s_n = i) \) where \( \sum_i t_{ij} = 1 \), and the observations probabilities \( p(x_n | s_n) \).

In a HMM the posterior of the state \( s_n \) given all the data until time instant \( n \), \( x_{1:n} \), is defined by the forward step of the Forward-Backward algorithm. We will denote this posterior as \( \alpha_n = p(s_n | x_{1:n}) \) with elements \( \alpha_n(i) = p(s_n = i | x_{1:n}) \).

Furthermore, this result is easily extended for any discrete time duration \( n_0 \) in the future, where we only need to multiply \( n_0 \) times the transition matrix by the posterior of \( s_n \):

\[
\alpha_{n+n_0} = \sum_{s_n} p(s_{n+n_0}, s_{n:n+n_0-1} | x_{1:n}) = T^{n_0} \alpha_n
\]

Equation (1) represents our knowledge at a time instant \( n \) over the future. By computing the entropy of this posterior we obtain a measure of the uncertainty of the future activities depending on \( n_0 \). Our main problem consists in finding the value \( n_0 \) where the entropy of (1) exceeds a certain threshold \( H_0 \) where we will include new sensor observations in our model to reduce the entropy. Mathematically this means finding the value of \( n_0 \) such as

\[
H(\alpha_{n+n_0}) < H_0
\]

where for any random variable \( X \) with probabilities \( p(x_i) \), \( H(X) = -\sum_i |x_i| p(x_i) \log p(x_i) \) is the entropy of \( X \) and \( |x| \) is the dimensionality of the vector \( x \). Also, we can express this entropy in matrix form using (1)

\[
H(\alpha_{n+n_0}) = -(T^{n_0} \alpha_n)^T \log(T^{n_0} \alpha_n)
\]

Solving (2) for \( n_0 \) is intractable, so we need to resort to some numerical approximations.

### 3 Activity Independent Approximation

In a HMM, the transition matrix has a limiting distribution or stationary state \( \mathbf{p} \).

\[
\lim_{n \to \infty} (T^n) = P
\]

where all rows of \( P \) are the limiting distribution \( \mathbf{p} \), and \( \mathbf{p} = P \nu \) where \( \nu \) is any probability vector. The existence of this limiting distribution implies that the entropy of \( p(s_{n+n_0} | x_{1:n}) \) must converge in \( n \to \infty \) with value

\[
H(\mathbf{p}) = \lim_{n \to \infty} H(\alpha_{n+n_0}) = -\mathbf{p}^T \log(\mathbf{p})
\]

This means that as we look further in the future, the entropy of the posterior only depends on the structure of the transition matrix and not in the posterior of the activities. A naive approximation to this problem consists in finding the value of \( n_0 \) where the system reaches the limiting distribution.

We can represent \( T \) by its eigendecomposition \( T = U \Lambda U^{-1} \), where \( U \) is the matrix of eigenvectors of \( T \) and \( \Lambda \) is a diagonal matrix of its eigenvalues \( \{\lambda_1, \cdots, \lambda_M\} \), such as \( |\lambda_1| > |\lambda_2| > \cdots > |\lambda_M| \). Every transition matrix has, at least, an eigenvector associated to the eigenvalue 1 and the largest absolute value of all its eigenvalues is also \( |\lambda_1| = 1 \). This eigenvalue is directly related to the limiting distribution \( \mathbf{p} \) of the transition matrix. Furthermore, if we express the \( n \)-power of the transition matrix as \( T^n = U \Lambda^n U^{-1} \), we can observe that the eigenvectors remain unaltered and we only need to compute the \( n \)-power of its eigenvalues. As \( \lambda_1 = 1 \) and all the other values in \( \Lambda \) are \( 0 \leq |\lambda_i| < 1 \), our naive approximation consists in finding the maximum value of \( n_0 \) that satisfies \( |\lambda_2|^{n_0} < \epsilon \), being \( \epsilon \) the tolerance, as all the other values of \( \Lambda \) decrease to zero faster

\[
n_0^\ast = \left\lfloor \frac{\log(\epsilon)}{\log(|\lambda_2|)} \right\rfloor
\]

(3)

Figure 2 shows an example of \( H(\alpha_{n+n_0}) \) as a function of \( n_0 \) for a transition matrix with \( M = 10 \) states. For any transition matrix, this function is non monotonically increasing in general, with several local maxima in the interval \([0, n^\ast]\) that are greater than the value of the entropy in the limiting distribution. When \( n_0 > n^\ast \) the entropy converges to the entropy of the limiting distribution \( H(\mathbf{p}) \). When \( n_0 = 0 \), we obtain the entropy of the posterior of \( s_n, H(\alpha_n) \). Choosing the value \( n_0^\ast \) as the time instant when we sample new data is a naive approximation, as there exists in general an interval of values in \([0, n^\ast]\) where the uncertainty is greater than \( H(\mathbf{p}) \). We need to select \( n_0 \) in the interval where the entropy is already large, and (2) holds.

### 4 Active Sensing Strategy

#### 4.1 Threshold method

A direct approach to solve the data acquisition problem consists in finding numerically the value of \( n_0 \) that satisfies (2). In this case, we need to select an appropriate value of \( H_0 \). If we choose a small \( H_0 \) the uncertainty will be too small and if \( H_0 \) is too large we will never reach the value, as the entropy is bounded above. A practical choice is to select \( H_0 \) in terms of the entropy of the limiting distribution, \( H_0 = a H(\mathbf{p}) \), where \( a \in [0, 1] \) is a parameter that controls the distance to the entropy of the limiting distribution \( H(\mathbf{p}) \). Thus our first approximation consists in finding the maximum value of \( n_0 \) such as (2) holds,

\[
H(\alpha_{n+n_0}) < a H(\mathbf{p})
\]
The entropy at is the slope of the tangent line, that corresponds to the \( y \) where \( y \) is the continuous time variable. If we consider that the probability of a particular state remaining unaltered only depends on its rate, then our problem consist in finding the value of \( t \) where this probability is some small value \( \epsilon \), that is

\[
e^{rt^*} = \epsilon \Rightarrow t^* = \frac{\log \epsilon}{r_i}, \forall i
\]

We can see that this approach is similar to (3), however, the value \( t^* \) selected depends on the particular state of the system rather than in the whole transition matrix. We will obtain a different \( t^* \) for every state \( s_n = i \).

5 Experiment results

We test the algorithms presented in Section 4 using a HAR database created using APDM Opal [11] wearable inertial sensors.\(^1\) This database contains the measurements from eight different people with two sensors placed on the waist and ankle. We assume that only one of the sensors was available during each of the experiments, evaluating our algorithms in two different settings. All the sequences contain a combination of five different activities: running, walking, standing, sitting and lying (in no particular order) under semi-naturalistic conditions in an indoor environment during a minimum of 20 minutes.

We train a HMM classifier with a Gaussian mixture observation model for each sensor using the standard Baum-Welch algorithm [12]. We configure our HMMs using the structure described in [13], assigning three states per activity, standing, sitting, and lying (in no particular order) under semi-naturalistic conditions in an indoor environment during a minimum of 20 minutes.

We represent the precision loss of all of the methods in each of the settings and the average precision loss of the algorithms described in Section 4 to obtain the next time instant \( n_0 \) where the sensors need to acquire a new window of observations.

In the threshold algorithm, we test three different values of \( \alpha = \{0.7, 0.8, 0.9\} \). In the line intersection algorithm, we stop our algorithm when the value of the slope of the line \( y_2 \) is less than 0.1, 0.01 and 0.001. In the rate matrix algorithm, we choose our \( n_0 \) as the value when the exponential of the rate is 0.1, 0.01 and 0.001 respectively.

Table 1 shows the average observations’ reduction in each of the settings and the average precision loss of the system due to this sample reduction. In particular, in Figure 3 we represent the precision loss of all of the methods in terms of the number of observations used in the waist

\(^1\) The dataset is available at http://www.tsc.uc3m.es/dataproxy/har/databases.zip
As we reduce the number of samples used in the system we can observe a decrease in performance. Under the same conditions however, the loss in precision is not heavily influenced by the reduction in window size. It is more important to update the model with new observations when the entropy increases than to acquire large windows of observations, as the entropy is practically zero during these windows. The best model in terms of precision loss is the rate matrix algorithm. In the rate matrix algorithm, we choose the next observation window. The number of observations used is in general larger than in the other models. The rate matrix algorithm reduces the number of observations, as the entropy is practically zero during these windows. The best model in terms of precision loss is the rate matrix algorithm. The number of observations available. In blue we represent the true activity.

There exists several differences in the entropy evolution between the different algorithms. In general, when we are performing a static activity, the entropy of the forward step increases more slowly than when we are performing dynamic activities. This effect can be observed in more detail in figures 6, 7 and 8.

In the threshold algorithm the entropy increases until a certain fixed threshold, where the algorithm acquires a new observation window. The number of windows acquired in each activity differs considerably.

In the line intersection algorithm, we choose the next window size. In green we represent the activity estimation when observations are available, and in red the activity estimation when no observation are available. In blue we represent the true activity.

Figures 4 and 5 show an example of the effects of the proposed algorithms over the acquired samples for each activity using one of the sequences. The number of samples required when we are performing a static activity like sitting is much less than when we perform a dynamic activity like walking, and also the distance between two consecutive data acquisitions is much larger. Furthermore, we can observe in Figure 5 that the performance of the algorithm in terms of the estimated activities while using all the observations is not affected by the data reduction.

As we reduce the number of samples used in the system we can observe a decrease in performance. Under the same conditions however, the loss in precision is not heavily influenced by the reduction in window size. It is more important to update the model with new observations when the entropy increases than to acquire large windows of observations, as the entropy is practically zero during these windows. The best model in terms of precision loss is the line intersection algorithm, though the number of samples used is in general larger than in the other models. The rate matrix algorithm reduces the number of observations more abruptly, but suffers the most in performance. The performance of the threshold algorithm falls between the others.

Figures 4 and 5 show an example of the effects of the proposed algorithms over the acquired samples for each activity using one of the sequences. The number of samples required when we are performing a static activity like sitting is much less than when we perform a dynamic activity like walking, and also the distance between two consecutive data acquisitions is much larger. Furthermore,
we decrease the parameter of the model, as the entropy of the limiting distribution is constant for a specific transition matrix. Also, the number of observation windows used in this model is larger in general, leading to shorter periods of time where there is no data acquisition and thus to a reduced precision loss.

In the rate matrix algorithm we use the diagonal elements of the rate matrix to decide our new data window instead of the entropy of the forward step. However, in Figure 8 we observe that the results in terms of data samples needed are similar to the threshold algorithm, while the precision loss is larger. This is a consequence of considering only the effect of one of the states in each data window by using only its component in the rate matrix, while the other algorithms in contrast, compute the entropy of the forward step using the information of the whole transition matrix.

6 CONCLUSIONS
We have proposed a systematic approach to solve the problem of energy consumption reduction in HAR systems with inertial sensors. Our work shows that using the maximum entropy of the posterior of the activities we can reduce the number of observations while maintaining the performance of the recognition system. We have implemented and evaluated three different methods to deal with the data acquisition problem where we have shown the importance of updating the model with new observations when the entropy increases rather than acquiring large windows of observations. In the best case, we reduce the number of observation used to 24% while only decreasing the precision by 0.018. We have shown how the acquisition system depends strongly on the activities being performed, needing less observations while performing static activities than when we are performing dynamic activities.

A future line of this work is to implement these algorithms when we consider a combination of sensors in the HAR system. We need to study how to select the sensor or a combination of sensors to sample data from. Furthermore, an implementation of these algorithms in a real time smart-
phone system will provide a numerical evaluation of the energy consumption reduction.

REFERENCES


Abstract—The main existent tool to monitor chemical environments in a continuous mode is gas sensor arrays, which have been popularized as electronic noses (enoses). To design and validate these monitoring systems, it is necessary to make use of machine learning techniques to deal with large amounts of heterogeneous data and extract useful information from them. Therefore, enose data present several challenges for each of the steps involved in the design of a machine learning system. Some of the machine learning tasks involved in this area of research include generation of operational patterns, detection anomalies, or classification and discrimination of events. In this work, we will review some of the machine learning approaches adopted in the literature for enose data analysis, and their application to three different tasks: single gas classification under tightly-controlled operating conditions, gas binary mixtures classification in a wind tunnel with two independent gas sources, and human activity monitoring in a NASA spacecraft cabin simulator.

I. INTRODUCTION

Gas sensors can detect changes in temperature, humidity, air pressure, and, obviously, human presence. Ogawa and Togawa [1] show that they are able to identify specific events in a home like waking up, start to cook, have breakfast, go out, come home, have supper, and go to bed. These observations hinted at the use of chemical traces as a manner to monitor homes [2], but the possibilities of environment monitoring are immense since these sensors can be used to monitor many other places such as hospitals or offices. In any of these applications, the use of gas sensor arrays requires gas sensor calibration. The calibration of a gas sensor array consists on establishing the functional relationship between measured values, analytical quantities and/or analytical identification. Traditionally, calibration includes first, the selection of the functional form of a computational model; second, the estimation of the corresponding model parameters based on a training dataset; and third, the model validation [3]. The resulting computational model is then used to predict the analyte amount/class of new measurements. Therefore, gas sensor calibration naturally translates to classification and regression problems in machine learning, and the sequence of steps involved in the design of a machine learning system must be taken, namely: data collection, features and model (classification/regression) selection, and training and evaluation [4]. In this work, we will focus on machine learning methods used for enose calibration in four different applications (I.1, I.2, II.1, III.1) that make use of three sources of data (I-III):

I. Electronic nose data under tightly-controlled operating conditions [5]. An array of 16 metal-oxide (MOX) gas sensors is exposed to six different volatile organic compounds at different concentration levels under tightly-controlled operating conditions. An extensive dataset was collected over a period of three years (13,910 measurements). For each measurement, a 128-component vector is processed from the sensors’ responses to extract steady-state and transient features (Section III).

1. Active sensor calibration for the discrimination of three gases [6]. This work investigates the optimal experiment selection to calibrate a gas sensor array to get the maximal possible performance in the classification of three different gases. It will be described in more detail in Section II.

2. Sensor calibration for the discrimination of six gases [7]. This work proposes a new pointwise Fisher consistent multiclass hinge loss function, which is used to efficiently calibrate an electronic nose and classify six different gases.

II. Electronic nose data for turbulent gas mixtures [8]. In order to reproduce more realistic environments, an electronic nose composed of 8 chemo-resistive gas sensors was exposed to turbulent gas mixtures generated naturally in a wind tunnel with two independent gas sources that generate two gas plumes. The sensor array was exposed to binary mixtures of ethylene with either methane or carbon monoxide. Volatiles were released at four different rates to induce different concentration levels in the vicinity of the sensor array. Each measurement is defined by 8 time series corresponding to MOX sensors’ conductivity together with temperature and humidity information.

1. Sensor calibration for ethylene discrimination in binary gas mixtures [9]. This work proposes a sensor calibration methodology to detect ethylene in a turbulent and changing background composed of methane or carbon monoxide on air (binary classification).

III. Electronic nose data from a NASA spacecraft cabin simulator. An electronic nose was installed on a NASA spacecraft cabin simulator and operated continuously while different volunteers (15 females and 47 males) were performed different daily activities (physical exercise on fitness equipment, defrosting frozen dinners, eating
dinner, and hygienic activities). This dataset has sensor readings for four weeks: two weeks used for training and two weeks used for testing. Data are captured at 0.05Hz and each pattern represents one-hour time series. Therefore, the dataset is formed by 672 patterns. Each measurement (pattern) is defined in a 4,320 dimensional space associated with 24 180-dimensional time series corresponding to 24 MOX sensors’ responses in one hour.

1. Sensor calibration for human activity monitoring

This paper performs enose calibration to predict the number of people in the room (regression).

In the following sections, we will revisit some of the machine learning approaches adopted in these applications and involving different steps of a machine learning system. In section II, the data collection procedure is analyzed to demonstrate the impact that external control parameters (gas concentrations) can have on the performance of a calibrated system. Section III will describe different techniques for feature selection and extraction used to build the attribute space that will serve as input to the classification/regression model. Section IV will focus on presenting some of the classification and regression models used for sensor calibration. Finally, Section V will point out some further work directions.

II. DATA COLLECTION: OPTIMAL EXPERIMENT SELECTION

A training dataset needs to be collected to perform the calibration of an analytical system. The challenge facing engineers building sensor arrays for event discrimination is that data is not available until the particular experiment is conducted. Moreover, each experiment carries a significant overhead that has no guarantee of positively impacting calibration quality. For example, each of the 13,910 experiments in the chemical gas sensor array dataset described in [5] requires 20 minutes of sensor array exposure time to collect just one example and, depending on the experiment, it may also carry health risks to those in the lab. Additionally, systems based on metal-oxide gas sensors are dynamic systems that need to be periodically recalibrated [11]–[13]. These difficulties are present in many applications including space travel [10], [12], environmental monitoring of public spaces [1], and industrial leak detection [14], making it fertile ground for the machine learning community.

In order to alleviate the experimental cost and reduce the frequency of the recalibrations, a methodology to select the best training examples to calibrate the system without modifying the configuration of the sensor array is proposed in [6]. This approach differs from other approaches in the literature focused on modulating sensors’ parameters such as frequencies, operating voltages, or temperature in response to environmental changes or application needs [15], [16]. In [6], the focus is on sensor networks where one has has to calibrate a device in controlled conditions and then deploy it for real operation. Given an analyte discrimination problem in which every training sample (measurement) is defined by a class label (gas type) and a control parameter \( c \) (gas concentration), the authors analyze which sampling distribution must follow the control parameter in the next batch of experiments in order to obtain the best calibration of the sensor array at each time, so the classifier is trained sooner rather than later. Note that the concentration level of each recording is not explicitly provided to the classifier. The active sampling strategy focuses on gas concentration selection since it is known that classification performance is more heavily impacted by concentration selection than gas label selection [17], [18]. They assume a canonical sampling probability distribution over the space of gas concentrations \( p^*(c) \propto \exp(-\kappa c) \) with \( \kappa \) the rate of the sampling distribution. This distribution allows biasing the sampling towards lower (\( \kappa > 0 \)) to higher (\( \kappa < 0 \)) concentrations, and it also recovers the uniform (uninform) distribution (\( \kappa = 0 \)). The proposed active sampling strategy works at batch level, and a fixed value for \( \kappa \) is set for each batch of size \( B \). Then, given the best sampling strategy in the preceding batches and given the sampling distribution \( p^*(c) \), the algorithm selects the optimal \( \kappa \) value for the following batch experiments as that minimizing the cross-validation error of an Inhibitory Support Vector Machine classifier (ISVM) [19] defined by the cost parameter \( C \) and the inverse of the kernel width \( \gamma \). This classifier is described in more detail in Section IV. Under this framework, the probability of sampling at concentration \( c \) at the \( m \)-th batch given the sequence of the optimal \( \kappa_i \) up to the \( m \)-th batch, \( \{\kappa_1, \kappa_2, \ldots, \kappa_m\} \), is given by the following equation

\[
p_{m}(c|\kappa_1, \kappa_2, \ldots, \kappa_m) = \frac{1}{m} \sum_{i=1}^{m} \frac{\exp(-\kappa_i c)}{\sum_{c' \in C} \exp(-\kappa_i c')},
\]

where \( C \) is the space of feasible values for the gas concentrations. Each addend corresponds to the probability at level \( c \) for one batch by having normalized the sampling distribution \( \exp(-\kappa_i c) \) so that \( \sum_{c' \in C} p(c) = 1 \). Finally, the factor \( \frac{1}{m} \) comes from the normalization of the joint probability \( p_m(c|\kappa_1, \kappa_2, \ldots, \kappa_m) \) so that \( \sum_{c' \in C} p_m(c'|\kappa_1, \kappa_2, \ldots, \kappa_m) = 1 \). The proposed algorithm is shown in Algorithm 1. The algorithm is tested in a subset of the enose Dataset I described in Section I (Application I.1.).

The final dataset has 1,800 128-dimensional patterns of three distinct pure gaseous substances, namely Ethanol, Ethylene and Acetaldehyde, each dosed at concentration values ranging from 2.5 \( \mu \text{mol/mol} \) (ppm) to 300 \( \mu \text{mol/mol} \) (ppm). The uniform distribution of the conditional probabilities \( P_{\text{gas}|c} \) in this dataset ensures uninformative (random) sampling on the label. The grid of \( \kappa \) values ranged from \(-0.03\) to \(0.03\) with a stepwise resolution of 0.001. The results reported in this work show that the active sampling strategy described in Algorithm 1 can only improve an uninform (random) selection of samples. The best experimental configuration yields a mean classification error of 0.5% as it will be shown in Section IV. Additionally, Fig. 1 shows the classification error in Algorithm 1 with \( \kappa = 0 \) as a function of the range of concentrations used for training and testing the ISVM model. As expected, the performance of the model is better when...
the range of concentrations in testing samples is the same as that used for training the classifier. According to Fig. 1(a), the lowest error rates for evaluations in a narrow range of concentrations (around the diagonal). On the contrary, Fig. 1(b) shows that the lowest error rates for evaluations of recordings in a range of concentrations not used in training correspond to scenarios when the sensor is calibrated using the widest range of concentrations. Furthermore, not including low concentrations during calibration significantly worsens the classifier’s performance; this is not a surprising result as it is generally believed that gases at higher concentrations are easier to classify than those at lower concentrations.

Algorithm 1 Active sampling algorithm where the main input parameters are the entire dataset \( U \), the rate of the sampling distribution at the current stage \( \kappa \), the sequence of the optimal \( \{\kappa_1, \kappa_2, \ldots, \kappa_m\} \) values for the previous stages, and the batch size \( B \). The output is the average error rate, \( \mu \) and the standard deviation, \( \sigma \), for \( N_{CV} \) draws using an ISVM as classifier. In all the runs, \( N_{CV} \) is set to 100.

**Inputs:** \( U, \kappa, \{\kappa_1, \kappa_2, \ldots, \kappa_m\}, B, N_{CV} \).

**Outputs:** \( \mu, \sigma \).

For \( C \leftarrow \{0, 1, 1, 10, 100, 1000, 5000, 10000, 50000\} \) do

For \( \gamma \leftarrow \{0.005, 0.01, 0.05, 0.1, 0.1, 0.5, 1.0, 10.0\} \) do

For \( t \leftarrow 1 \ldots N_{CV} \) do

\( S \leftarrow \text{sample} (m + 1) \times B \) points according to Eq. 1.

\( T \leftarrow \{1, \ldots, 100\} \setminus S \)

Train the ISVM \((C, \gamma)\) classifier on \( S \).

\( \tilde{v}_T \leftarrow \text{test error on } T \)

End for

End for

End for

\( \mu \leftarrow \min_{C, \gamma} \tilde{v}_T \)

\( \sigma \leftarrow \min_{C, \gamma} \tilde{v}_T \)

Finally, it should be also remarked that this idea of optimally selecting the next experiment to calibrate the sensor array is in line with the active sampling algorithms proposed in machine learning [20]. For a more detailed discussion about the proposed algorithm, the experimental results, and the relation between this work and active learning techniques in machine learning, the reader is referred to [6].

### III. Feature Extraction and Selection

A critical point for the success of a machine learning system is the correct representation of the input space of the machine learning model. When working with electronic noses based on metal-oxide gas sensors, the sensor response is given by the conductivity across the active layer of each sensor [21]. The interaction processes between the sensor and the analyte identity and/or concentration dosage define the response profile, which allows machine learning algorithms to perform different tasks such as sensor calibration for gas discrimination or activity monitoring. Different strategies can be found in the literature to represent the input space data. Some of them directly use the raw time series provided by the sensors and solve a multidimensional classification/regression problem capable of dealing with the temporal structure of the data [22]. Other approaches downsample the time series to obtain a multivariate representation of the data, and then apply a classification/regression algorithm that does not take into account the temporal structure such as Support Vector Machines with a Gaussian Kernel. This is the approach adopted for sensor calibration in gas binary mixtures (Application II.1) and human monitoring applications (Application III.1) presented in Section I. Finally, new features can be generated based on the sensors’ response time series. For example, the sensor calibration problem under tightly-controlled operating conditions introduced in Section I and used in Section II generates eight features for each sensor: two steady-state features and six features reflecting the sensor dynamics. Let \( r[k] \) be the sensor resistance time profile, \( k \) the discrete time indexing in the recording interval \([0, T] \), and \( T \) the duration of the measurement, the steady-state feature \( \Delta R \) is given by the difference of the maximal resistance change and the baseline, \( \Delta R = \max_k r[k] - \min_k r[k] \). Its normalized version \( ||\Delta R|| = \frac{\Delta R}{\max_k r[k]} \) is also useful for gas discrimination. On the other hand, six features based on the exponential moving average \( (ema) \) reflect the sensor dynamics of the increasing/decaying transient portion of the sensor responses [23]. The \( ema_\alpha \) transform evaluates the rising/decaying portions of the sensor resistance by considering the maximum/minimum values of a first-order digital filter \( y[k] = (1 - \alpha) y[k - 1] + \alpha r[k - 1] r[k - 1] \), with \( y[0] = 0 \) and the smoothing parameter \( \alpha \in [0, 1] \). Different values of \( \alpha \) generate different features with information about the transient response, so the six features are generated by setting \( \alpha = \{0.1, 0.01, 0.001\} \) for both the rising and the decaying stages.

Besides the importance of setting a representative input space, another common issue in electronic nose data is that MOX gas sensors are characterized by high correlation in their response [24]. This redundancy can be used to extend...
the lifetime of sensor arrays [25], [26] and to correct sensor drift [13], [27]. As an example, Fig. 2 shows the Pearson correlation for the 10 × 10 pairs of sensors utilizing the complete time series captured for each sensor in Dataset II (Section I). From Fig. 2, we can conclude that the sensors TGS2600 and TGS2602 are highly correlated among them, as well as the sensors TGS2620, TGS2612, TGS2611, and TGS2610. Therefore, if some of these sensors are not considered, the information provided by the rest of the sensor array is expected to be similar to the complete sensor array. That is why, feature selection is a common step in the design of a chemical detection system [28]–[32]. One of the methods used for selecting the optimal subset of sensors is the Quadratic Programming Feature Selection (QPFS) algorithm [33], a multivariate filter technique that takes into account redundancy among features and relevance of each feature with respect to the task to perform. This algorithm is especially suitable for the problem here considered given its computational efficiency and outstanding performance in highly redundant databases. In particular, the QPFS feature selection method in a M-dimensional space consists of minimizing a multivariate quadratic function subjected to linear constraints as follows

$$\begin{align*}
\min_{w} & \quad \frac{1}{2} (1 - \alpha) w^T Q w - \alpha F^T w \\
\text{s.t.} & \quad w_i \geq 0 \quad \text{for} \quad i = 1, 2, \ldots, M, \\
& \quad \|w\|_1 = 1, 
\end{align*}$$

(2)

where \(w\) is an M-dimensional vector, \(Q\) is a symmetric positive semidefinite matrix in \(\mathbb{R}^{M \times M}\) with non-negative entries, and \(F\) is a vector in \(\mathbb{R}^M\) with non-negative entries. \(Q\) represents the similarity among variables (redundancy), and \(F\) measures the similarity of the features with the target (relevance). The components of the solution vector \(w\) represent the normalized positive weight of each feature. Thus, the goal of Eqs. 2-4 is to select those features that provide a good trade-off between relevance and redundancy. Finally, the real parameter \(\alpha \in [0, 1]\) enables to overweight the linear and the quadratic term in Eq. 2. In other words, \(\alpha\) regulates the trade-off between relevance and redundancy. As an example, Pashami et al. adapt the QPFS algorithm to select a subset of sensors for detecting changes in the activity of a distant gas source from the response of an array of metal-oxide gas sensors deployed in an open sampling system. They use the Pearson correlation to quantify redundancy among features (matrix \(Q\)), and they use a measure based on the Fisher Index to determine the relevance of each sensor (vector \(F\)). Their results show that selecting sensors with QPFS allows obtaining detection rates comparable with those corresponding to the best single sensor, while providing lower detection delays than the single sensor. For more details, the reader is referred to [29], [33].

IV. SENSOR CALIBRATION AS A SUPERVISED LEARNING PROBLEM

Envose technology offers immense possibilities for environmental monitoring applications, but its proper calibration is crucial to guarantee the applicability of the sensing technology to the task that it is designed for. As stated in Section I, gas sensor calibration can be easily identified with classification and regression problems in machine learning, and during these years a large variety of calibration techniques has been investigated for chemical detection systems, including artificial neural networks, linear discriminants, multilayer perceptrons, k-NN classifiers, partial least square regressors, and more recently, Support Vector Machines [15], [34]–[37]. This section will review the classification/regression models and results in sensor calibration for the four enose applications described in Section I. The regression problem associated with the prediction of the number of people in a NASA spacecraft cabin simulator (Application III.1.) is addressed by applying the well-known Support Vector Regression (SVR) model with a RBF kernel [38]. The multiclass classification problems in Table I use Inhibitory Support Vector Machines (ISVMs) [19] and ε-Support Vector Machines, an extension of ISVMs. ISVMs have shown good performance in calibrating sensor arrays [6], [7], [9], [39] and their goal is to provide a simple algorithm for multiclass classification by directly integrating the concept of inhibition into the SVM formalism. However, the use of the ISVM classifier in sensor calibration settings, such as the one described in Algorithm 1, arises a fundamental question of whether the successive inclusion of training points leads to the optimal classifier. This point implies guaranteeing the Bayes consistency of the ISVM model, which can be addressed by analyzing the pointwise Fisher consistency (or classification calibration) of the classifier as this property states necessary and sufficient conditions to have Bayes consistency when a classifier minimizes a surrogate loss function [40]. ISVM authors show the consistency of the ISVM model for problems with two or three classes [19], such as the
active sensor calibration procedure (Application I.1) and the classification calibration procedure for binary gas mixtures (Application II.1). However, when working with more than three classes, the ISVM model cannot guarantee the Bayes consistency of the classifier. That is why the $\lambda$-SVM model [7], an extension of ISVM, is proposed as a universally pointwise Fisher consistent multiclass classifier. The $\lambda$-SVM model is defined by a real parameter $\lambda$ representing the margin of the positive points of a given class. The margin is set to 1 for points belonging to other classes. The ISVM classifier is a particular case of $\lambda$-SVM by setting $\lambda = 1$. Formally, given a training set of $N$ patterns, $\{x_i\}_{i=1}^N$, in which each point $x_i$ belongs to a known class $y_i \in [1, L][N]$, the $\lambda$-SVM objective function is defined as follows,

$$\min_{\omega} \frac{1}{2} \|\omega\|^2 + C \sum_{i=1}^N \sum_{j=1}^L \eta_{ij}$$

s.t. $\eta_{ij} \geq 0$

$$-1 - (\lambda - 1) \eta_{ij} + \frac{1}{2} f_j(x_i) y_{ij} + \eta_{ij} \geq 0,$$  

where $\omega$ is the concatenation of the hyperplanes of each class, $\omega = [w_1, w_2, \ldots, w_L]$, $\{\eta_{ij}\}$ are the slack variables that provide room to handle the noisy data, and $y_{ij}$ takes the value 1 if the pattern $x_i$ belongs to class $j$ (i.e. $y_i = j$) and $-1$ otherwise. The cost parameter $C \in [0, \infty)$ establishes a trade-off between the two objectives of the model: maximizing the margin while classifying correctly as many training patterns as possible. The key difference between $\lambda$-SVMs and standard SVMs relies on $\lambda$-SVM's decision function, which includes an inhibitory term regulated by a scalar parameter $\mu$. $\lambda$-SVM’s decision function associated with the $j$-th class and the input pattern $x_i$ is defined as $f_j(x_i) = \langle w_j, \Phi(x_i) \rangle - \mu \sum_{k=1}^L \langle w_k, \Phi(x_i) \rangle$. It can be shown that the optimal value for $\mu$ is $\frac{1}{L}$, which can be directly obtained by minimizing the Lagrangian in Eqs. 5-7 [7], [19]. The classification of a data point $\tilde{x}$ is determined by the maximum of the evaluation function for each class: $y(\tilde{x}) = \arg \max_j f_j(\tilde{x})$. Finally, the function $\Phi$ is a map from the input space to a higher dimensional space where the optimal hyperplanes, $w_i$, are calculated. In the results here presented, the mapping function $\Phi$ is that associated with the RBF kernel with compact support. It can be shown that the multiclass classification function defined in Eqs. 5-7 is pointwise Fisher consistent for $\lambda \in (-\infty, 0) \cup ((L - 2)/2, (L - 1)]$ [7]. This is the first multiclass hinge-loss function capable of giving unlimited weight to the positive examples without breaking the classification calibration property, which is shown to be beneficial in terms of classification accuracy and training times. In fact, the optimal $\lambda$ value for the sensors calibration problem (Application I.2) is obtained in a classification calibrated scenario with $\lambda = 10, 100$). For more detailed description and analysis of the $\lambda$-SVM method, the reader is referred to [7] and references therein.

A summary of data characteristics and sensor calibration results is shown in Table I. Column Feature Type indicates whether the input space for the classification/regression model is defined either by the time series captured by the sensors or by an aggregate of synthetic features (see Section III). Column C/R shows if the sensor calibration problem is a classification problem (C) or a regression (R) problem. In parenthesis, the number of classes in the classification problem is shown. Column ML Model indicates the machine learning model used for sensor calibration. Column Err. shows the classification error in the case of classification problems, and the mean relative error for the regression problem. Column Ref. lists the papers related to the corresponding dataset and application, in which a detailed description of the experimental setup for each application can be found. According to the results shown in Table I, machine learning techniques applied to electronic nose data analysis allow the design and development of effective chemical detection systems.

V. FURTHER WORK

The works presented in this review show the capability of machine learning techniques to properly calibrate sensors and monitor environments, but they are mainly obtained in controlled environments where data are correctly labelled. Future lines of research in machine learning applied to electronic nose data analysis should be focused on extending existing algorithms or designing new ones to overcome some of the challenges arising from the deployment of sensor arrays in open and non-controlled environments. This scenario brings up several challenges for the machine learning community as many of the most popular machine learning algorithms cannot be directly applied. Some of these difficulties include: (i) high correlation between gas signals and environmental variables such as humidity and temperature, which makes it difficult to extract the chemical information captured by the sensors; (ii) lack of labeled data or the presence of noise in the labels, which makes it difficult to adopt machine learning strategies without the supervision of an expert; (iii) existence of drift when sensors are working for long periods of time, which compromises the applicability of the machine learning algorithms that assume distribution stationarity.

ACKNOWLEDGMENT

IR-L acknowledges partial support by Spain’s grants TIN2013-42351-P (MINECO), S2013/ICE-2845 CASE-CAM (Comunidad de Madrid), and TIN2015-70308-REDT. IR-L and RH acknowledge the partial support by 3² Convocatoria de Proyectos de Cooperacion Interuniversitaria UAM–Banco Santander EEUU.

REFERENCES


TABLE I
DESCRIPTION AND RESULTS OF MACHINE LEARNING MODELS FOR ELECTRONIC NOSE APPLICATIONS.

<table>
<thead>
<tr>
<th>Application</th>
<th>#samples</th>
<th>#attributes</th>
<th>Feature Type</th>
<th>C/R</th>
<th>ML Model</th>
<th>Err.</th>
<th>Ref.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active sensor calibration (Application I.1.)</td>
<td>1,800</td>
<td>128</td>
<td>Synthetic</td>
<td>C(3)</td>
<td>ISVM</td>
<td>0.50%</td>
<td>[5], [6]</td>
</tr>
<tr>
<td>Sensor calibration (Application I.2.)</td>
<td>13,910</td>
<td>128</td>
<td>Synthetic</td>
<td>C(6)</td>
<td>λ-SVM</td>
<td>0.35%</td>
<td>[5], [7]</td>
</tr>
<tr>
<td>Sensor calibration (Application II.1.)</td>
<td>180</td>
<td>23,688</td>
<td>Time series</td>
<td>C(2)</td>
<td>ISVM</td>
<td>2.11%</td>
<td>[8], [9]</td>
</tr>
<tr>
<td>NASA human activity monitoring (Application III.1.)</td>
<td>672</td>
<td>4,320</td>
<td>Time series</td>
<td>R</td>
<td>SVR</td>
<td>0.32%</td>
<td>[10]</td>
</tr>
</tbody>
</table>


Decoding for Neural Prostheses with Deep Architectures

Henrique Dantas and V John Mathews
School of Electrical Engineering and Computer Science
Oregon State University
Corvallis, OR 97331, USA

Abstract—This paper presents recent methods for decoding electromyogram and electrocorticogram signals to interpret human intent and control prosthetic arms. Specifically, we demonstrate that deep learning architectures can perform this task better than currently available methods. The system is trained using a modification of a Dataset Aggregation algorithm. In addition, we demonstrate that auto-encoder architectures are useful for feature selection in neural decoding applications.

I. INTRODUCTION

Limb amputation profoundly affects the activities of daily living. Although a substantial disability, individuals who have lost a limb retain the underlying neural circuitry and much of the ability to sense and control movements of their missing limb. That is, they have the ability to sense artificially evoked percepts and to control artificial limbs in manners similar to how they controlled their limbs prior to limb loss.

A number of methods are available in the literature for neural decoding of human intent. Some of these methods are nonlinear [1]–[4], but most are linear. Decoding algorithms employing linear generative models relating neural signals to the actual intended movements fall into broad categories of Wiener filters [5], [6], population vectors [7], [8], probabilistic methods [9], [10], and recursive Bayesian decoders such as Kalman filters [11]–[14].

In recent years, deep learning systems have become very popular because of their ability to drastically improve baselines performance [15], [16]. Reinforcement learning [17]–[20] has achieved state-of-the-art results for many hard problems. In the neural decoding field, a solution based on solving the Bellman’s Equation using temporal difference with kernels was proposed in [21].

The goal of this paper is to explore the usefulness of deep learning architectures for decoding electromyogram (EMG) and electrocorticogram (ECoG) signals to interpret human intent and control prosthetic arms. Specifically, we demonstrate that deep learning architectures can perform this task better than currently available methods. The system is trained using a modification of a no-regret-based algorithm called Dataset Aggregation (DAgger) [22]. In addition, we demonstrate that auto-encoder architectures are useful for feature selection in neural decoding applications.

II. NEURAL SIGNALS AVAILABLE FOR DECODING

For a brief overview of motor system physiology in the context of neural prostheses, the reader is referred to [14]. The most basic neural signal available for decoding is the spike event, the extracellular measurement of an action potential. Such signals may be recorded using an electrode array, for example, the Utah Electrode Array [23], implanted in the cortex or in peripheral nerves. The electrocorticogram (ECoG), recorded at the cortical surface, strikes a balance between invasive measurement of spike events recorded from highly invasive and complicated implants in the cortex and electroencephalogram (EEG), signals measured from the scalp, in terms of invasiveness and spatial resolution. Although technically EMG is not a neural signal, we they are used in commercially available, state-of-the-art myoelectric prostheses. Because of space limitations, we limit the experimental results presented in the paper to ECoG and EMG signals.

III. PROBLEM FORMULATION

It is common to model the motor tasks using a Markov decision process [17], [18]. The goal is to learn a policy \( \pi_\theta(u_k|s_k) \), where \( s_k \) is the \( k^{th} \) state and \( u_k \) is an action for the \( k^{th} \) state. Furthermore, \( \pi_\theta(u_k|s_k) \) can be interpreted as the probability of the system taking an action \( u_k \) given that the system state is \( s_k \).

For our problem, we define the state as the union of the most recent \( H \) instances of the measured neural
Workshop on Advances and Applications of Data Science & Engineering

In particular, we only need the policy \( \pi \) can be trained to learn the parameters of the system works, convolutional neural networks and many more architectures such as multilayer perceptron (MLP) networks, which is described in (5). During training, the kinematic data represents the desired hand movements. In the experiments done on an amputee subject, the kinematic data was obtained from a virtual hand during training. During testing, this data was replaced by estimates of hand kinematics by the system model.

\[
s_k = [Z_k, ..., Z_{k-H}] \bigcup X_k, ..., X_{k-H} \tag{1}
\]

\[
u_k = [X_{k+1}] \tag{2}
\]

respectively. We assume that the state \( s_k \) evolves according to a Markov property such that

\[
p(s_{k+1}|s_k, ..., s_1) = p(s_{k+1}|s_k) \tag{3}
\]

For a given trajectory \( \tau = \bigcup_{i=1}^{H} (s_i, u_i) \), where \( H \) is the horizon, it is possible to write \( p(\tau) \) as

\[
p_{\theta}(\tau) = p(s_1) \prod_{i=1}^{H-1} p(s_{i+1}|s_i, u_i) \pi_{\theta}(u_i|s_i) \tag{4}
\]

For a given model, we learn the parameters of the model by maximizing the cost function

\[
J(\theta) = \log(p_{\theta}(\tau)) \tag{5}
\]

The gradient of \( J(\theta) \) is given by

\[
\nabla_{\theta} J(\theta) = \sum_{i=1}^{H-1} \nabla_{\theta} [\log \pi_{\theta}(u_i|s_i)] \tag{6}
\]

This results means that, no knowledge of the system dynamic \( p(s_{i+1}|s_i, u_i) \) is needed to maximize the \( p(\tau) \). In particular, we only need the policy \( \pi_{\theta}(u_i|s_i) \) [18].

If \( \pi_{\theta}(u_i|s_i) \) is assumed to have the form

\[
\pi_{\theta}(u_i|s_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2} [u_i - \phi_{\theta}(s_i)][u_i - \phi_{\theta}(s_i)]^T} \tag{7}
\]

\( \nabla_{\theta} J(\theta) \) can be written as

\[
\nabla_{\theta} J(\theta) = -\sum_{i=1}^{H-1} [\phi(s_i) - u_i][\nabla_{\theta}\phi_{\theta}(s_i)]^T \tag{8}
\]

Equation (8) is the equivalent of the gradient of a neural network trained with back propagation with the mean-square-error as cost function. Consequently, architectures such as multilayer perceptron (MLP) networks, convolutional neural networks and many more can be trained to learn the parameters of the system model \( \phi_{\theta}(\cdot) \). The system parameters can be updated using

\[
\theta = \theta + \alpha \nabla_{\theta} J(\theta) \tag{9}
\]

where \( \alpha \) can be interpreted as the learning rate.

IV. System

A. System Architecture

Deep learning architectures have became very popular in recent years. Probably the most cited deep learning architecture is the convolutional neural network (CNN) [16], [17], [24]. A schematic block diagram of the general architecture used in this work is shown in Figure 1. The parameters of the network are fitted during training to maximize the objective function described in (5). During training, the kinematic data represents the desired hand movements. In the experiments done on an amputee subject, the kinematic data was obtained from a virtual hand during training. During testing, this data was replaced by estimates of hand kinematics by the system model.

Figure 1. A high level system representation of the hand state decoder based on a deep learning architecture.

B. Training

Training large and complex deep learning systems is challenging in many situations. In most instances, the network has a large number of parameters, and learning a general model without over-fitting requires very large training sets. This is especially so for reinforcement training [17], [25], [26] and to lesser extent, for imitation learning [27], [28].
In this work, we use a modified version of the Dataset Aggregation (DAgger) approach [22] to train the deep learning architecture. DAgger provides a compromise between the complexity of training and the overall performance. In the first iteration the system is trained using back propagation using the neural data and intended movement. In subsequent iterations, the trained system will run in the original training set and data will be augmented based on the decisions made by the system and the known intended movement. The modified DAgger approach is described using a pseudo-code in Algorithm 1. During each iteration, a small amount of pseudo-random noise is added to the neural signal to mitigate problems with over-fitting the model.

Initialize $D \leftarrow (S, U)$;
Train policy ($\pi_\theta$) with $D$
for $i = 1$ to $N$
do
Get $\hat{x}_k$ from performing sequence with $Z_k$
Make $\hat{S} = [Z_k + \text{noise}, ..., Z_{k-H_t} + \text{noise}]$
Do $X_k, ..., X_{k-H_2}$
Make $U = U$
Agregate datasets $= D \leftarrow D \cup (\hat{S}, \hat{U})$
Train the Policy ($\pi_\theta$) with $D$
end

**Algorithm 1:** Modified DAgger Algorithm

### C. Feature Selection

Feature selection plays an important role for the neural prosthetic decoders [13], [14]. Feature selection allows simplification of the systems and avoids over-fitting. Often, it leads to a better understanding of the problem and results in shorter training times.

An elegant way to perform feature selection is the usage of auto encoders and auto decoders [29]. This approach is an application of standard fully-connected neural network in an auto-associative way, and is schematically shown in Figure 2. The system tries to recover the input in the last layer, after the data is passed through one or multiple bottlenecks. The presence of the bottlenecks simultaneously allows the dimensionality reduction in the data and selects the features that are most relevant to the recovery process.

This system is trained with standard backpropagation algorithms.

This auto-encoder architecture has some interesting features when compared with other feature selection methods such Principal Component Analysis (PCA) [30]. Although, both methods are trained in an unsupervised manner, PCA assumes linearity in the data set, while auto-encoders do not make this assumption.

### V. EXPERIMENTAL RESULTS

#### A. Experiment Setup

1) **EMG Signals:** The results presented here are for a single amputee subject. After receiving approvals from the Institutional Review Board and informed consent, the subject was implanted with 32 EMG electrodes to acquire intra-muscular EMG data. The tasks were performed in a virtual environment (Musculoskeletal Modeling Software [31]). The program modeled a virtual hand with 12 degrees of freedom (DoF), divided as follows: flexion and extension of each digit (5 DoFs), adduction and abduction of all except for the third digit (4 DoF), and wrist roll, pitch and yaw (3 DoF).

During training, the subject was instructed to track the movement of a simulated hand with his phantom limb while the the EMG signals were recorded. The instructed movement followed a semi-sinusoidal path at a velocity deemed comfortable by the subject. Only movements of a single DoF was instructed during each training trial and multiple ($\leq 11$) training trials of each movement were performed. To train multi-DoF decoding methods, movement trials for different DoF were concatenated.

2) **ECoG Signals:** After receiving approval from the institutional review board and informed consent from the patient, two 16-channels non-penetrating arrays (PMT Corporation, Chanhassen, MN) were implanted in a patient undergoing surgery to treat epilepsy. The arrays were implanted underneath a standard clinical electrocorticographic (ECoG) grid. The electrodes were placed in the arm and hand areas of the motor
cortex. The patient was instructed to perform simple tasks such as reaching using a mouse in a tablet (20 cm x 20 cm). The patient moved the mouse from the bottom center of the table to the upper left or upper right corner of the table. All neural channels and the 2 position data (x and y coordinates) were recorded at a sampling rate of 30 kHz.

B. Results and Discussion

In this work, we compared the performance of three different decoders, a deep learning architecture and two different realizations of Kalman decoders. A block diagram of the deep learning system is shown in Figure 1. The branch processing the neural data contained 4 convolutional layers with 64 filters of size 7 coefficients each. Each convolutional layer contained an activation function layer and a dropout layer to avoid over-fitting. More details of this architecture can be found in [32]. The first layer above contained a max pooling phase [33]. Following the convolutional layers, was a fully connected layer with 256 hidden nodes with an activation function and a dropout layer. In the kinematic branch, a fully connected layer with 256 hidden nodes was used with an activation function and a dropout layer. Finally, the two branches were merged using a fully connected layer with 256 hidden nodes, followed by an activation function and a final fully connected layer with the number of DoFs to be decoded as hidden nodes. All activation functions involved hyperbolic tangent functions.

Two variations of Kalman decoders were used here. The first one is a traditional Kalman Filter [34] with a linear generative model and was reported in [11]. The system used signals from the neural channels as the observation vectors and the number of elements in the internal state vector was DoFs to be decoded. The second variant of Kalman Filter [13] used a nonlinear transformation of the observation vector used by the linear model above. Details of the system is available in [13], and employed a polynomial transformation.

The conventional multi-layer perceptron network with 3 layers of 256 hidden nodes was also tested. However, the system performed substantially poorer than the other systems, and only one set of results are included for this case.

The auto encoder structure employed is shown in Figure 2. This architecture contained 3 fully connected layers with hyperbolic tangent activation functions. The number of nodes in the third layer corresponded to the number of features used by the decoder. This part comprised the auto-encoder whose outputs became the input features for the decoder. The overall system consisting of three additional layers was trained to reconstruct the original neural signals. As a comparison, we also employed principal component analysis (PCA) of the neural data and used the principal components of the data matrix as the features for the decoding task.

Figure 3 compares the effect of applying the DAgger algorithm for both the convolutional neural network and the standard multi-layer perceptron network. We observe that the deep learning architecture converged to a model with relatively small decoding error outside the training data, whereas the conventional system did not. Consequently, only the deep learning architecture is used in the comparisons below.

Figure 4 shows the output of proposed decoder based on the deep learning architectures and the Kalman Filter with quadratic nonlinearities as proposed by Dantas et al. [13] for EMG signals. We can see from this figure that, in general, the deep learning architecture performs much better than the Kalman filter in this example. The results are better for movements of the thumb, the index finger and the middle finger, at least in part because of the better access to the muscles controlling these fingers in the amputee subject’s arm, and partly because it very hard to control the ring and little fingers separately from each other.

Figure 5 and 6 show the comparison of the capabilities of PCA and auto-encoders for feature selection for the deep learning architecture of this paper and the two
Kalman Filters for EMG and ECoG signals, respectively. The auto-encoders and PCA performed more or less similarly in these experiments, even though the performance of the auto-encoder was slightly better with the EMG signals. Another important observation is that the deep learning architecture exhibited relatively stable performance as the number of features increased. Both Kalman filters displayed more erratic behavior with increasing number of features.

VI. CONCLUSION

This paper demonstrated that deep learning architectures can be used to implement efficient neural prosthetic decoders. Furthermore, the DAgger algorithm applied to train the system improved the system performance over traditional methods. The auto-encoder approach employed for feature selection is very effective in reducing the dimensionality and complexity of the decoder algorithms. Additional studies on how to implement regret-based reinforcement learning algorithms, and how to update the systems parameters on line are underway at this time.

REFERENCES


A guideline for a public-private partnership on urban big data sharing

Daniel Sarasa Funes
Polytechnic University of Madrid (UPM)
daniel@openyourcity.com

ABSTRACT

This paper explores the construction of public-private partnerships on the subject of sharing big data in cities. It considers data as an strategic asset whose exploitation benefits are not sufficiently permeating our cities neither in the form of better local jobs, new scientific knowledge or well-informed urban operations. It analyzes the barriers and inhibitors of such a sharing agreement between key urban players, especially privacy concerns and cooperation dynamics, and goes over the potential advantages that mixing big data sources in the urban context could have.

The work presents and develops a set of implementation principles for the system, including agents and roles, building blocks, governance and the new kind of professional profiles needed, building a case for a stronger public action in the field of big (urban) data. We argue in favor of the City Hall as the best positioned institution to take the leadership flag in this endeavor, which can be pursued by combining a soft regulation strategy with the activation of other facilitation tools. We also highlight the contour of data-driven government alongside possible success metrics.

Finally, we list the conclusions of our work as a set of guidelines for those cities interested in progressing in this idea, signaling as future lines of research both the study of optimum locations for its practical implementation and the detailed study of the business model and design of a viable prototype.

1. INTRODUCTION

We live in the age of cities and big data. Privatization, out-contracting and the booming Internet have resulted in distributed 'de facto' city operations and management schemes. Today, digital businesses like Google, Amazon or telcos, the so-called 'shared economy' companies or traditional businesses like banks or utilities (through their new smart metering units), sense and know partially how the city operates, although its opacity prevent city halls to new smart metering units), sense and know partially how the city operates, although its opacity prevent city halls to

1.1 Knowledge cities vs. 'data-driven' cities

Urban operation, as a result of the multiplicity of agents involved and overlapping layers, and due to urban growth, is an increasingly complex task. But, at the same time, with city activity operating twenty four hours a day its importance can not be sufficiently highlighted. The conclusion that, as more and more people move to cities, there is no better way to improve people's life than improving life in cities, is pretty straightforward. As urban concentration grows, the multiple risks that our planet and societies face at the environmental, social or economic spheres can be better addressed by adopting more sustainable, more innovative and better-informed urban policies.

Nowadays, those policies can not ignore the potential offered by the set of processes and technologies grouped under the generic 'big data' buzz-word. Big data, alongside other emerging technologies like, smartphones, Internet of Things (IoT) devices and machine (and deep) learning may have a substantial impact in how cities are understood, planned and managed. IoT, for example, is expanding exponentially the amount and diversity of the data collected. Although its acronym relates to things, IoT devices and networks increasingly harvest information about us, people, through a wide variety of sensors that track (with or without our consent) our daily actions. Smartphones and apps complement IoT devices by further extending the personal data that we release, including opinions, habits, relationships and health. Big data glues all that unstructured information together extracting the relevant traits of individuals, either considered as customers, voters, or fully recognized citizens. Finally, machine and deep learning automate the consumption of that information and many of the subsequent actions or decisions that happen in the organizations that hold and/or analyze the data.

The novelty is that these emerging technologies may join to provide the deepest level of comprehension so far about the physical and human systems and subsystems that form the city, that this can be achieved in real time and that it may be possible to better forecast its short and long term evolution. The nature of the difficulties that stand in the way of such a decisive jump in urban practice are more organizational than technical and require soft-skilled, multi-threaded and highly creative professionals rather than pure technological capacities. Professionals capable of grasping, for example, the subtle implications that arise when we blindly add machine and deep learning to big data and we apply the resulting combination to automated decision making processes in the urban context. By doing so, we can advance significantly towards an old futuristic idea that has been lately renewed: the data-driven government, although we might as well question if we can afford the price of losing equity and human understanding of human needs along the way.

As stated before, cities are probably our most precious tool to tackle and eventually fix the main problems of mankind: environment, democracy, economy or decent life conditions. But, paradoxically, and at the same time, they showcase the crudest representations of those very
same problems: pollution, corruption, unemployment, inequalities, poverty, isolation, etc. Hence, the question that quickly arises when deepening in the relationships between big data and cities is up to what extent big data contributes to solve those crucial challenges without creating new problems. Or, formulated differently, if a city would improve at all if we were able to scientifically test all decisions, and by how much.

1.2 The inflationary phase of big data

If we take an historical angle, after the 'big bang' explosion of data, the new 'data-verse' is experiencing a quick inflationary phase. This inflationary expansion, as shown later, is everything but homogeneous through organizations, sectors or processes.

In terms of organizations and sectors, the Internet businesses are clearly leading the way, fueled, first, by the inherent use of big data related technologies, some of which are even powering the development of the general concept of big data itself, and, second, by the highly competitive and innovative markets in which those companies operate. Obviously, a business logic works here, and the use of big data that those Internet giants make is driven by their needs to gain a competitive advantage. To illustrate this, let us take the case of a company like Google, which bases its whole business operation on big data, investing a huge quantity of resources in the development of the technology. Just as an example, at the base of big data storage and retrieving techniques, we find MapReduce, which is a Google patent [1]. In addition, known Google's businesses are based on data that users release, either openly (as in Google's search engine or Google Maps) or privately (as in Gmail).

There is another interesting set of businesses, grouped under the label of the 'shared economy' which is worth analyzing, since some of these new Internet companies are disrupting local economies in areas such as transportation or accommodation while somehow managing to surf or bypass local regulations. Uber is a visible example of a new paradox. The serious blow that it afflicts to the community of local cab drivers not only affects self-employment in the city but also leads to a reduction in the overall local tax collection. To rub a little salt into the wound, their systems may very well use, for instance, the data about road outages that the city hall releases in open data formats for routing optimization purposes, while the company locks the vast amount of valuable information gathered through their daily trips around the city.

1.3 The 'data sharing value gap'

Uber, like many other companies, are free to use government data available from the numerous open data initiatives that many public administrations, at local, regional, national or continental levels are implementing, being such reuse one of the goals of those open data policies, under the assumption that open data has the potential to create new business opportunities. The European Union [2] estimates that opening government data could add up to 40 billion euros to the European economy. Given that the Europe’s GDP is 24% of the world’s GDP, we could estimate the impact worldwide of opening government data at ~200 billion euros. Although this is just a rough estimation, it is perfectly valid for our purpose of pointing out the distance between the value that can be unlocked by considering opening just public data (let’s call this public data value) and what can be obtained by unlocking the potential of collaborative data policies between the private and the public sector.

A report from McKinsey Global Institute [3] estimates that a joint policy between private and public sectors to cooperate on opening their respective data silos could unlock between $3 Trillion and $5 Trillion a year in additional value across just seven business domains. Let’s call this figure public-private data value. Now, even if those figures are rough estimations, we see that the difference between public data value and public-private data value is in the range of several trillion dollars.

![Figure 1. Data sharing value gap.](image)

Narrowing what is shown in figure 1 as the data sharing value gap is therefore the main purpose of our work, since, as shown earlier, there is a clear potential to increase the economic and social dividends of big data by including the private sector at the core of the open data policies. This will imply not just considering companies as data consumers or providers of data analytics capabilities. Private companies need to be recognized as the main data producers, with (if not equal) comparable rights and responsibilities regarding open data than those of the public sector. In the absence of city, state, or nation-wide regulations about data sharing schemes, this paper explores an organizational and conversational approach to a mutual relationship between key city players (public and private) that allows sharing ‘at some extent’ the knowledge that big data brings in ways that favor public interests while protecting individual privacy and legitimate business assets.

1.4 Data as a public good

We will deal in subsequent sections in more detail with individual privacy; let us just make now the general consideration that, in a digital world, privacy does not exist any more for the vast majority of us. Even in the absence of malicious or accidental information disclosure,
at least a dozen of big corporations, and therefore hundreds of people working on them, have access to the records that precisely form our daily lives. Aggregated over millions of other users and thoroughly analyzed, that information constitutes a secret goldmine.

The term data mining was originally coined some decades ago to illustrate the nature of the processes that deal with retrieving relevant information out of large data bases. In the age of big data, and taking into account the analysis and projections about the value that unlocking the full potential of data can bring to the world economy, the term data mining acquires a new relevance, conveying a clue about the strategic place that data holds for our economies. Considering data as a strategic asset leads naturally to develop a normative framework that would recognize data as an essential public good whose exploitation rights and mechanisms have to be revised.

2. URBAN BIG DATA

2.1 Big data as a relative concept

There are several definitions of big data [4]. The most common big data definition encompasses three main features: volume, velocity and variety (the 3V rule). Lately, it has been added a 'fourth V' (value), highlighting the importance that industry concedes to big data. It would be pointless to give absolute magnitudes for those characteristics, since innovation speed these days would out-date any value by the time this work is published.

A logistic operator with terabytes of supply chain records may claim to be dealing with big data. But then, how about the trillions of stock exchange operations performed in real time by an army of autonomous algorithms operating globally? From an organizational perspective, which is the one this paper addresses, absolute values mean very little. Therefore, a complementary definition depicts big data as 'datasets whose size is beyond the ability of typical database software tools to capture, store, manage, and analyze.' For the purpose of our research, centered in the potential, barriers and challenges that big data implies for the main urban players, we will extend that technical view to the organizational dimension, and provide an alternative definition of big data as

'Big data whose volume, velocity and variety establishes new challenges for an organization or business, by the opening of new prospects but also by requiring new efforts and skills for its treatment, setting it out of its comfort zone at many levels.'

This definition deals with the original three V's (volume, velocity and variety) of big data as relative magnitudes linked to the level of maturity that organizations present in the field of data analytics, incorporates the fourth V (value) through the opening of new business opportunities, and expands the requirement to innovate that big data brings to the whole organization. As a result of this definition, understanding the level of progress, expectations and needs of every stakeholder pertaining to our data-sharing system with regards to the use of big data will necessarily be a key aspect of study when building a practical implementation of such a system.

2.2 Key processes of urban big data

The technical literature about big data is broad and deep. Let us just picture a simplified building block diagram of the processes involved in big data treatment, as depicted in [5]:

```
<table>
<thead>
<tr>
<th>Collection/Recording</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extraction/Cleaning</td>
</tr>
<tr>
<td>Integration/Aggregation/Representation</td>
</tr>
<tr>
<td>Analysis/modeling</td>
</tr>
<tr>
<td>Interpretation</td>
</tr>
</tbody>
</table>
```

*Figure 2. Canonical processes of Big Data*

We will not discuss the convenience of this standard model for intra-organization purposes, where big data is produced, collected, stored and interpreted within the boundaries of an organization. However, we will argue in this research that it falls short to grasp the subtle and specific demands of the urban milieu. Hyper-urbanization, ultra-technification and skyrocketing innovation have joint forces to keep [...] 

7. FUTURE WORK

After setting in this paper the convenience and principles that may guide an urban big data sharing system, future work should deepen into practical implementation issues and into the business model.

A second line of work would conduct a thorough analysis of worldwide innovation hubs in order to identify those cities that better meet the conditions for implementing this system: strong public leadership, a participative and collaborative business ecosystem, a thriving civic community, as well as an open and advanced vision with regards to data, suitable facilities and access to funding opportunities.

REFERENCES:


[28] If Mayors Ruled the World. Benjamin Barber. 2013


[33] Thinking, fast and slow. Daniel Kahneman. 2011


CONVERSATIONS:

The following persons had the patience and time to help in this research by means of fruitful conversations:

Elena Alfaro, BBVA Data Analytics
Carlos Alcéan Alcalde, Zaragoza City Hall
José Carlos Arnal Losilla, Zaragoza City of Knowledge Foundation
Claudio Feijóo, Polytechnic University of Madrid (UPM)
Dennis Frenchmann, Leventhal Professor of Urban Design and Planning - M.I.T.
Nigel Jacob, Mayor's Office of New Urban Mechanics, City of Boston
Michael Joroff, M.I.T.
Ana Jiménez Train, Zaragoza City Hall
Marta de Miguel Esponera, University of Zaragoza  
Enrique Morgades, CIRCE (University of Zaragoza)  
Kevin O'Malley, Bristol City Council.  
José Manuel Páez, Polytechnic University of Madrid (UPM)  

ACKNOWLEDGEMENTS:  

This research was possible thanks of the support of Real Colegio Complutense, a joint institution between several Spanish universities and Harvard University at Cambridge, MA (U.S.A.). Special thanks to the Sergio Ramos (Polytechnic University of Madrid – UPM) as academic supervisor and José Carlos Arnal (Zaragoza City of Knowledge Foundation) for their thorough revisions.
Enhancing *Smart Mobility*: some Artificial Intelligence solutions


Knowledge Engineering and Machine Learning Group
Universitat Politècnica de Catalunya-BarcelonaTech
Barcelona, Spain

Email: {jvazquez,salvarez,igomez,loliva,jatejeda,dariog,vcodina}@cs.upc.edu

Abstract—*Smart Cities* is a recent trend on the intensive use of Information and Communication Technologies (ICT) in urban and metropolitan areas with the aim to improve social and political capacities, increase efficiency of processes and support social, cultural and urban development. There are two main approaches for Smart Cities and the role that ICT may play: 1) ICT solutions to ease the gathering and integration of vast amounts of data to optimize the management and control of the city resources and infrastructures on the basis of pre-defined metrics; and 2) ICT solutions to empower citizens with new ways to improve their usage of resources and knowledge, to foster creativity and to make them participate in the government of the city, with the aim to change the way the city behaves by changing their citizens' behaviour. In this paper we show the potential of Artificial Intelligence techniques for augmenting ICT solutions not only to increase the cities competitiveness but also to support the active participation of citizens in those processes. As example we will describe our experience on the usage of Artificial Intelligence techniques to support Smart Mobility in the context of the SUPERHUB Project.

I. INTRODUCTION

*Smart Cities* is a recently coined term that usually refers to the smart management of the cities’ socio-economic and environmental capital through the use of Information and Communication Technologies (ICT for short). It is important to note that, in this case, the usage of the word smart applied to these technological solutions is independent from the use of (or lack of) any Artificial Intelligence approach on the systems; these technologies are said to be smart in the sense that they provide ways to enable and empower social, cultural and urban development, improving social and political capacities and/or efficiency. It is also worth noting that there is already some abuse on the term, converting any technological solution that eases information gathering, analysis, management or sharing as a Smart City solution, without careful analysis of its real impact on the cities efficiency or capacity, and on the users empowerment.

There are two main views for Smart Cities and the role that ICT may play. In the classical, top-down view a reduced set of relevant actors (governors, major city infrastructures) manage and control the cities resources in smart way, and efficiency is the result of this top-down management. This approach makes use of large communication infrastructures gathering vast amounts of data from city sensors and other city actors as input to ICT solutions that support infrastructure managers and policy makers in the efficient management of resources, activities and knowledge. In this view, the aim tends to be the optimization of one or several pre-defined performance metrics (i.e. Key Performance Indicators or KPI’s). There is also an alternative, bottom-up approach where citizens are the main actors, and cities become smarter by making their citizens behave in a smarter way. This approach takes advantage not only on the existence of sensors, Internet of Things components and communication networks but also on the increasingly widespread knowledge that citizens have on ICT and the social use of these technologies by citizens on their everyday lifes. The aim on these citizen-centric approaches is not only the improvement of the cities (by measurable KPI’s) but the impovement of their citizens (in terms of access to technology, improving the quality of the information they receive to take well-informed decisions, providing tools to foster creativity and entrepreneurship, or putting governance in the hands of the citizens).

Mobility is an important aspect in Smart Cities, as it is one of the main challenges for urban planners. Now-a-days it is still difficult for policy makers and transport operators to 1) know the state of the city in (near) real-time, and 2) achieve proximity with the end-user of such city services, especially with regards to communicating with the citizen and receiving proper feedback. Many Smart City solutions for mobility have focused on empowering the user with the ability to decide the best way to move through the urban or metropolitan transport network, by creating journey planners. But most of the existing journey planners only provide a few options to let users customize, to some extent, how the journey should look like. The reality, however, is more complex and diverse – different users might prefer different routes and/or combinations of modes and these may, in addition, depend on the users context (e.g., a shopping trip, travelling with kids or going back home) as well as on the environmental context: traffic, weather conditions, events in the city and so on. Specifying all of the circumstances affecting what the user perceives as the ideal plan, however, would be overwhelming. The use of personalization in the field of journey planning is mainly focused on tourism [1], although there are other domains where routes are also recommended, e.g., for...
sporting/leisure purposes [2], [3]. For instance, [2] produces journey hiking walkways according to a set of milestones or personalize journey plans by choosing from pre-defined routes which better fit users preferences. Afterwards, the selected routes are enhanced by means of adding Points Of Interest (POI) -relevant geographical features that may be relevant to the user. In general, these approaches can be considered as aimed at providing journey plans to users on closed domains where the set of potential journey plans is already defined, and those plans are built based on a recommended set of POIs (e.g., touristic routes), meaning that first the POIs are obtained and routes designed to visit them. Other approaches try to overcome this closed domain constraint. While some solutions successfully generate this kind of routes, they do not incorporate users’ preferences or just a small set of generic preferences are taken into account (e.g., departure time or cost) [4].

Many ICT solutions for Smart Mobility see the user only as a service consumer, overlooking the role citizens can play to perceive and build an up-to-date model of the city status. Thanks to smartphones, users that move in a city can potentially generate automatic data that may be hard to obtain otherwise: location, movement flow, average trip times, and so on. Moreover, network problems and incidents that affect mobility services are often documented by users in Social Networks at the same time or, in some cases, even before those are detected by the mobility operators and communicated through official channels. This phenomenon has been referred to as humans as sensors [5]. Sensing the city through mobile humans potentially provides sensor coverage where events are taking place, and takes advantage of the citizen’s capacity to analyze the situation and evaluate its impact at the individual and even the collective level. Some proposals have recently appeared to exploit data generated by users to sense the cities’ status, either from the individual sensing perspective, closer to multi-agent based systems [6] or from big data techniques applied to social network streams by combining statistical analysis with semantic interpretation [7]. However, such proposals work at a granularity level that is either too high or too low: in the former approach there is, in principle, no centralised mechanism to maintain global aggregations; in the latter, systems do not take advantage of the end-user terminals and their potential dispersion in time and space. In this paper we present a system that can work at both levels, leveraging global aggregations with local awareness to have the best possible picture of the context of the city whenever possible.

In this paper we present some of the results of the EU SUPERHUB project [8], describing how, by the usage of some Artificial Intelligence techniques (knowledge representation, machine learning, natural language processing, sentiment analysis, semantic inference, information diffusion models and knowledge mining techniques), interactions of users in certain social networks -i.e.: Twitter and Foursquare- are combined with knowledge of the city, city events calendars and geospatial data to learn local behavioural patterns in the city, capture those deviations that may reflect events that affect mobility in the city and then provide better travel recommendations, taking advantage of a richer city model and a more accurate user profiling.

II. THE SUPERHUB PROJECT

SUPERHUB [8] was a project co-funded by the European Commission that aimed at realizing a new services mobility framework supporting an integrated and eco-efficient use of multi-modal mobility systems in an urban setting. The main goal of the project is to provide an open platform able to consider in real time various mobility offers and to provide a set of mobility services able to address user needs, promote user participation, and foster environmental friendly and energy-efficient behavioural changes.

To achieve this SUPERHUB developed:
1) a persuasive engine based on captology principles to facilitate the voluntary adoption of environmentally-friendly multi-mobility habits,
2) novel methods and tools for real-time reasoning on large data streams coming from heterogeneous sources,
3) new algorithms and protocols for inferring traffic conditions from mobile users by coupling data from mobile operator networks with information coming from both GPS based mobile phones and social network streams,
4) mechanisms for dynamic matchmaking of resources, and
5) a journey planner with the goal of best fulfilling user mobility needs and preferences while minimizing negative environmental impact.

The concept of the project built on the notion that citizens are not just mere users of mobility services, but represent an active component and a resource for policy-makers willing to improve sustainable mobility in smart cities. In SUPERHUB, users play an active, consumer and producer role (prosumer), acting, on the one hand, as providers of mobility offers, such as in car-pooling negotiations, and on the other hand, as consumers of mobility resources, such as taking the bus, using a public parking, or renting a bike).

The SUPERHUB travel planner employs the state of the art user modelling and recommender systems techniques – by observing the past users’ choices and the context in which these choices were made, SUPERHUB gradually learns the model that accurately reflects the multifaceted nature of each users preferences and constraints. During a journey plan search, the user model is used, in addition to the automatically acquired contextual information, to guide the search process in order to provide highly-tailored journey plan recommendations that best reflect the unique needs and situational context of each user. Technically, the personalization process relies on a mixture of contextual content-based filtering plus the use of semantics applied to contextual data, thus being able to assess different contexts, even if these have not been experienced by the user yet.

In addition, journey plans are further personalized by means of opportunistic recommendation. The SUPERHUB opportunistic recommender enhances journey plans by adding points
of interest that might be interesting for the user during the journey. For example, if the user is going back home and she has to wait to take the train, the recommender may suggest having a coffee or doing some small shopping. In some circumstances, the recommender may even suggest alternative destinations (e.g. a different cinema or supermarket) if the original destination is difficult to reach in a current traffic.

Another issue is aggregating heterogeneous situational data in a common model when such data is specified at different levels of abstraction. For instance, a subset of the SUPERHUB sensor adaptors belong to the field of web-based social networks, with content typically expressed in plain text. Some intermediate steps (based in Natural Language Processing (NLP) semantic extraction techniques) must be followed in order to extract a structured concept representation of the meaning of the raw data. More generally, data has to be normalised, so that values used in the representation of the data belong to the same abstract data type ranges; and aggregated, so that data that comes from potentially unreliable sources can be contrasted and therefore reinforced or discarded. Expert knowledge in the form of input models, such as the model of the city, designed by mobility experts, is converted into rules that will be used to infer a first abstraction of knowledge.

Both data homogenisation and the inference of the state of mobility in a city are made by the Semantic Interpreter. The core of this component is a reasoning engine based on Clojure, which is able to infer knowledge from both situational and historical data. Via the Semantic Interpreter, raw data is filtered, normalised and interpreted into high-level concepts. Such concepts can be merged and analysed to generate derivative concepts that are not explicit in the sensor data but implicit in the aggregation of large instances of it. The analysis relies in applying expert knowledge and information about the city and is based in semantic methods rather than statistical ones. The obtained knowledge is represented in the form of RDF triples, which are then sent through a subscription-based mechanism to other SUPERHUB components that will use such knowledge for several purposes, such as managing mobility high-level contexts to generate more fine-grained user mobility profiles, or being able to understand normality with respect to policy fulfillment and thus derive and predict unexpected situations.

III. FROM CITY SENSING TO CITY INTERPRETATION

In a real-world setting, the SUPERHUB journey planner has to be ready to receive high amounts of journey plan requests and deliver multimodal recommendations that best fit a wide range of criteria, including the user preferences. However, the evaluation of such criteria is continuously dependent on factors that occur in the external world, what we call the context of the request. But the context, given the same external conditions, is always city-dependent. For instance, a request for a journey plan made in January in a day that it is snowing defines different contexts if the user is in Helsinki (where snowing is considered a normal situation and public transport is prepared) or if the user is in Barcelona (where snowing is considered an extraordinary case that has numerous, unpredictable consequences on the transport networks). Therefore, weather sensor data may be useful to detect a context, but not sufficient by itself.

In SUPERHUB, situational data is retrieved not only from sensors such as weather or traffic sensors, but also from the social networks. Twitter (microblogging), Instagram (microblogging) and Foursquare (location-based) data are combined in order to automatically model the normal situations in the city and detect abnormal ones. Based on user’s geolocated posts, the Semantic Interpreter is able to identify: which are the main points of interests, how the people move along the city on their daily basis or before/after an event occurs (trajectories), and specific mobility user profiles (e.g. tourists). Besides that, other data sources are cross referenced with social networks data in order to filter noisy information, provide reliability and
add explanations about events and validate the results of the detection system.

In the case of tweets, text content related to mobility concepts are aggregated and analysed in time windows in order to find located trends [9]. In information diffusion, reliability of data obtained is a big concern, and our solution is to filter bots and spammers by analysing the topology of the individual social networks of the users [10], and by calculating their influence on other users [11]. Instagram geo-localized activity is also aggregated, taking advantage that, in relevant events, people tend to report it both in a textual form (tweets describing an accident between a car and a bus, blocking the street) and in a graphical form (photos on Instagram showing the wrecked vehicles). Where similar activity trends occur in the same geo-localised area in a similar time window, the reliability score for that potential event increases. The Foursquare API adds to the reliability score of the aggregated data by contrasting the detected trends against area-based collections of check-ins in the same time windows.

One interesting usage of all this city model that is built off-line from social network data is for the run-time detection of disruptive (un-expected) events. Disruptive events and incidents in the mobility city field are abnormal situations that have a negative impact on the city mobility, for instance: traffic or metro service. In order to maintain a reliable mobility city behaviour, it is necessary to detect and predict these kind of events. Basically, a normal situation in the city can be modelled under a spatio-temporal context; which can be defined as any information that characterize a situation. It means that a day of a week, weather conditions, city zone (geolocation) and the type of road could be part of this context. For instance: traffic could be different on Monday at 7:00 am compared to Saturday at the same time: between two different weather conditions (rainy or sunny); or the type of road (highway or boulevard). However, to monitor an entire city with several type of sensors for discovering disruptive events and incidents may not be affordable. Alternatively, a human as a sensor approach brings a new opportunity to minimize the cost and reveal information that cannot be extracted directly from others sensors. Users can inform and provide context about events having real impact on them in real-time. The identification of recurrent events allow the classification of what is a normal, expected event and what it is not. Figure 1 shows the SUPERHUB City Sensor application, which shows the discovered unexpected events into a city map. This visualization can be useful for mobility observers to get an early warning of a disruptive event forming. For instance, a meaningful increase of activity in social networks about a non-scheduled meeting somewhere in the city center to protest against government policies, which starts being scattered across the city but then ‘moves’ into a geo-localised area in the map, suggesting that people are gathering in that area of the city.

V. FROM JOURNEY PATH PLANNING TO CONTEXT-AWARE TRAVEL RECOMMENDATION

One of the aims in SUPERHUB was to generate journey plans taking into consideration both a rich user behavioural model and an up-to-date context model. Users select a destination and plans are designed accordingly, using different modes of transport, including car-pooling. The Intelligent Mobility Recommender is composed by three main components: the IMR Manager, which manages mobility requests and keeps user behavioural profiles; the Journey Plan Recommender (JPR), which incorporates the user preference to support journey plan design; and the Opportunistic Recommender (OR), which provides opportunistic recommendations.

In SUPERHUB, Journey Plans are dynamically generated by the Journey Plan Recommender with regards to different criteria (among them the user preference). The generation of Journey Plans is done iteratively, searching for suitable Journey Legs and using critics to reduce search space. Finally, once the set of Journey Plans have been found, the Opportunistic Recommender selects POIs along the route that may be appealing and may represent opportunities for the user to improve his experience. Semantic relations and interpretations of items are used to enhance the OR results, to obtain semantically similar POIs which are described in SUPERHUB’s City Model. Then, the selected POIs are added to the Journey Plan to enhance it. Given that the domain of POIs is not as large as Journey Plans it is feasible to consider collaborative-filtering approach, which has proven to be effective and requires less information about items (venues in this case). Nevertheless, collaborative-filtering can suffer from data sparsity, thus we use Matrix Factorization to find latent factors and reduce sparsity.

Both the Journey Plan Recommender and Opportunistic Recommender contribute to build and maintain a user model and make use of contextual data to perform context-aware
recommendations. When context matters, as is the case of mobility domain (e.g., when it is raining people prefer to avoid walking or riding a bike), it is sensible to use a user model that has been learnt with feedback acquired in the same context as the target user because only that feedback is relevant for the prediction. This allows improving results and dynamically adapt to the real situation in the city and to user’s needs, goals and current context (e.g., in a rush, going to work, with children). Both recommenders use context to pre-filter the learned user models and exploit contextual user models which are built taking into account the contextual information obtained from situational data.

Both approaches, using contextual information and semantics, provide novelty to the state of the art beyond the use of recommenders in the mobility domain. The closest approach to ours is the one by [2] which incorporates the user preference to support journey plan design. SUPERHUB differs from that approach that routes are not taken from a pre-defined set, but are dynamically generated according to contextual situation in the city. Thus, [2] adds a set of POIs to enhance the routes which is similar to SUPERHUB’s approach for opportunistic recommendation, although it does not take into consideration contextual information.

VI. CONCLUSIONS

In this paper we have outlined the usage of a combination of Artificial Intelligence techniques (knowledge engineering, machine learning, sentiment analysis, semantic inference, information diffusion) to enhance mobility in the cities, by empowering users not only with means to plan their routes through the transport network smartly but also to act as sources of event information that can help build a better snapshot of the city status.

On one hand SUPERHUB’s Semantic Interpreter receives interactions of users in certain social networks and combines them with city-specific knowledge, city events calendars and geospatial data. Data is analysed in two stages, off-line and on-line, in order to learn behavioural patterns that can be used to then capture those deviations that may reflect events that affect urban mobility. On the other hand SUPERHUB’s Intelligent Mobility Recommender, by having a strong focus on the use of semantically-enhanced contextual information, provides better route recommendations more suitable for the current user needs and situation.

The Semantic Interpreter was tested by extensive runs that observed and interpreted not only the city status for Barcelona, Milan and Helsinki (the three trials cities for the SUPERHUB project evaluation), but also other cities such as Sao Paulo, Mexico City, London and Tokio. Depending on the periodicity of the aggregation set in the configuration of the Semantic Interpreter instance, the delay between the start of the event in the city and the instant at which it is detected by the SUPERHUB platform may vary, but we have empirically proven it can be as low as 5 minutes. The Intelligent Mobility Recommender was tested and validated with real users in the three trials cities as part of the project’s final validation.

ACKNOWLEDGEMENT

The work described in this paper has been supported by the EU SUPERHUB project (ICT-FP7-289067).

REFERENCES


Prediction of building temperatures for energy optimization

Pablo Rodríguez-Mier*, Marc Fresquet*, Manuel Mucientes*, Alberto Bugarín*

*Centro Singular de Investigación en Tecnoloxías da Información (CITIUS), Universidade de Santiago de Compostela
Santiago de Compostela, Spain
Email: pablo.rodriguez.mier@usc.es, marc.fresquet@rai.usc.es, manuel.mucientes@usc.es, alberto.bugarin.diz@usc.es

Abstract—The reduction of energy consumption in buildings is one of the goals to improve energy efficiency. One way to achieve energy savings in buildings is to develop intelligent control strategies for heating systems that are able to reduce power consumption without affecting the thermal comfort. An intelligent control system must be able to predict the temperature of the building in order to manage the heating system. In this paper, we present three rule-based models that are able to predict the indoor temperature. The models have been learned with FRULER, a genetic fuzzy system that generates accurate and simple knowledge bases. Our approach has been validated with real data from a residential college showing errors lower than $0.50^\circ$C in the prediction of the temperatures.

I. INTRODUCTION

Buildings account for 40% of the total energy consumption in the EU, according to European Directive 2010/31/EU on energy efficiency in buildings. Since the expansion this sector is currently experiencing will cause an inevitable rise of that percentage, it seems clear that the reduction of energy consumption and the use of energy from renewable sources in the building sector will play a key role in the measures taken to reduce emissions of greenhouse gases.

One way to achieve energy savings in buildings is by reducing the total working hours of heating systems. However, a decrease in the total usage may lead to important decreases of indoor temperatures that can affect thermal comfort. In order to prevent this, automatic heating control systems must predict the future indoor temperature for a particular control policy in order to find the best strategy that minimizes power consumptions while keeping thermal comfort.

Current methods for indoor temperature prediction [2] are mostly based on physical model simulations [12] and black-box machine learning methods [6], [13], [1], [11]. Physical models describe the building behaviour by solving theoretical equations that describe to a certain precision the different dynamics and interactions between the variables. Although these methods are very powerful to simulate the different dynamics of a building, especially when there is no real data available, in general these methods are: 1) very time-consuming since they require many simulation hours, which prevents their application for predicting temperatures in small temporal windows; and 2) complex to formulate, since it is very difficult to produce a detailed model of a complex building, especially when there are many unknown factors that can affect the temperature dynamics. On the other hand, machine learning models can overcome some of these limitations by learning the behaviour from real data. However, current techniques, which are mostly black-box models based on neural networks, are hard to interpret and thus the interaction of the different variables of the building remains unknown.

In this sense, the generation of accurate and interpretable models for indoor temperature prediction is fundamental for 1) modelling the energy-building behaviour and 2) discovering which are the most relevant variables that affect the indoor building temperature and are related to power consumption. Within this context, initiatives such as the EU LIFE-OPERE project [4], where this research is framed, have started. OPERE has among its goals the setting of efficient management systems in energy networks, both thermal and electrical, in existing installations with large energy consumption.

In this paper, a rule-based regression model for indoor temperature prediction is proposed. The aim is to build and validate prediction simple and accurate models. To do so, we have modelled the indoor temperatures of a number of of a residential college using the FRULER Genetic Fuzzy System (GFS) [9]. The knowledge bases learned by FRULER include TSK fuzzy rules that accurately predict the temperature dynamics from a set of different predictors that can be measured both inside and outside the building.

II. FRULER: FUZZY RULE LEARNING THROUGH EVOLUTION FOR REGRESSION

FRULER (Fuzzy RUle Learning through Evolution for Regression) [9] is a novel GFS that obtains accurate and simple linguistic TSK-1 fuzzy rule base models for regression problems. FRULER (Fig. 1) is composed of a new instance selection method for regression, a novel multi-granularity fuzzy discretization of the input variables, and an evolutionary algorithm that uses a fast and scalable method with Elastic Net regularization to generate accurate and simple TSK-1 fuzzy rules.
The object of the instance selection module is to reduce the variance of the models, focusing the generated rules on the representative examples. The instance selection method for regression is an improvement of the CCISR (Class Conditional Instance Selection for Regression) algorithm [8], which is an adaptation for regression of the instance selection method for classification CCIS (Class Conditional Instance Selection) [5].

The instance selection process is based on a relation called class conditional nearest neighbor (ccnn) [5], defined on pairs of points from a labeled training set as follows: for a given class $c$, ccnn associates to instance $a$ its nearest neighbor computed among only those instances (excluding $a$) in class $c$. This relation, therefore, describes proximity information conditioned to a class label. In regression problems, the outputs are real values instead of labels and, therefore, they must be discretized in order to use the ccnn relation. FRULER uses Kernel Density Estimation (KDE) with a gaussian kernel in order to estimate the probability density function of the output variable ($y$) in a non-parametric way. Once the probability density function is obtained, the local minimum determines the split points, and, therefore, which labels/classes are used for the ccnn relation. Thus, each instance is associated with one of the labels obtained by this process, and the instance selection method can follow the CCIS procedure.

In a multi-granularity proposal, each granularity has a different fuzzy partition. The generation of the fuzzy linguistic labels can be divided into two stages. First, the variable must be discretized to obtain a set of split points $C^g$ for each granularity $g$. Then, given the split points, the fuzzy labels can be defined for each granularity. In a top-down approach, the split points are searched iteratively, i.e., only a new split point is added at each step, obtaining two new intervals. Therefore, the approach followed by FRULER aims to preserve interpretability between contiguous granularities: adding a new label to the previous granularity and modifying the flanks of the adjacent labels. In regression problems (TSK-1 in our case), the discretization process must search for the split point that minimizes the error when a linear model is applied to each of the resulting intervals.

In order to select the maximum number of split points for a variable, we used the well-known Bayesian Information Criterion (BIC). FRULER measures the error for BIC with a least squares model fitted for each interval of the discretization, and evaluates the complexity of the model as the number of inner splits and parameters fitted by the regression of each interval. After the discretization of a variable for each granularity, FRULER applies the method proposed in [3] in order to get the multi-granularity fuzzy partitions.

The evolutionary algorithm learns a linguistic TSK model. The integration of the evolutionary algorithm with the preprocessing stage is as follows (Fig. 1):

- First, the instance selection process is executed over the training examples $E_{tra}$ in order to obtain a subset of representative examples $E_S$.
- Then, the multi-granularity fuzzy discretization process obtains the fuzzy partitions for each input variable.
- Finally, the evolutionary algorithm searches for the best data base configuration using the obtained fuzzy partitions, generates the entire linguistic TSK rule base using $E_S$ and evaluates the different rule bases using $E_{tra}$.
1) Chromosome Codification: The chromosome is codified with a double coding scheme \((C = C_1 + C_2)\). \(C_1\) represents the granularity of each input variable. \(C_2\) represents the lateral displacements of the split points of the input variables fuzzy partitions.

2) Initialization: The initial pool of individuals is generated by a combination of two initialization procedures. A half of the individuals are generated with the same random granularity for each variable, while the other half is created with a different random granularity for each variable. The lateral displacements are initialized to 0 in all cases.

3) TSK Rule Base Generation: FRULER uses the Wang & Mendel algorithm to create the antecedent part of the rule base for each individual. The consequent part of the rules is learned using the Elastic Net method \([14]\) in order to obtain the coefficients of the degree 1 polynomial for each rule. Elastic Net linearly combines the \(\ell_1\) (Lasso regularization) and \(\ell_2\) (Ridge regularization) penalties of the Lasso and Ridge methods, minimizing the following equation:

\[
\hat{\beta} = \arg \min_\beta \|Y - X \cdot \beta\|_2^2 + \lambda \cdot \alpha \cdot \|\beta\|_1 + \lambda \cdot (1 - \alpha) \cdot \|\beta\|_2 \tag{1}
\]

where \(\beta\) is the coefficients vector, \(Y\) is the outputs vector, \(X\) is the inputs matrix, \(\lambda\) is the regularization parameter and \(\alpha\) represents the trade-off between \(\ell_1\) and \(\ell_2\) penalization. In order to solve the minimization problem of Elastic Net (Eq. 1), we used Stochastic Gradient Descent (SGD).

The rule base is generated using only those examples in \(E_s\). In this manner, those examples that are not representative are not taken into account, the method avoids the generation of too specific rules, and reduces the time needed to create the rule base.

4) Evaluation: The fitness function is:

\[
\text{fitness} = \text{MSE}(E_{tra}) = \frac{1}{2 \cdot |E|} \sum_{i=1}^{|E|} (F(x^i) - y^i)^2 \tag{2}
\]

where \(E_{tra}\) is the full training dataset and \(F(x^i)\) is the output obtained by the knowledge base for input \(x^i\). Using all the examples for evaluation can be seen, in some way, as a validation process, as the rule base was constructed with a subset of them \((E_s)\).

5) Selection and Replacement: The selection is performed by a binary tournament. On the other hand, the replacement method joins the previous and current populations, and selects the \(N\) best individuals as the new population.

6) Crossover and Mutation: FRULER has two crossover operations: one-point crossover for exchanging the \(C_1\) parts (it also exchanges the corresponding \(C_2\) genes) and, when the \(C_1\) parts are equal, the parent-centric BLX (PCBLX) is used to crossover the \(C_2\) part.

The mutation (with probability \(p_{mut}\)) applies two possible operations with equal probability to a randomly selected gene of the \(C_1\) part: i) decreasing the granularity by 1 or ii) increasing the granularity to a more specific granularity —all the granularities have the same chance. In order to calculate the new lateral displacements in the corresponding \(C_2\) part, the displacements of the previous granularity are taken into account.

7) Local Search: After the replacement, all the new individuals go to a local search process. This stage generates \(n_{ls}\) new \(C_1\) parts with equal or less granularity for each variable, and the \(C_2\) part is generated randomly. If there is a solution that obtains better fitness, then it replaces the original individual.

8) Restart and Stopping Criteria: The restart mechanism uses the incest prevention threshold \(L\) as a trigger. At each iteration, \(L\) is decreased in a value that depends on if there are no new individuals or the best individual does not change. When \(L\) reaches 0, the population is restarted. When the restart criterion is fulfilled twice or the number of evaluations reaches a threshold, the algorithm stops.

III. INDOOR TEMPERATURE PREDICTION

The main goal of the OPERE project \([4]\) is to implement efficient management systems in both thermal and electrical energy grids in existing installations with large energy consumption. To achieve this goal, in this work we propose a method that automatically learns an accurate and interpretable non-linear model using FRULER. The learned models predict the indoor temperature dynamics of an existing building in order to find a better heating control that minimizes the energy consumption without sacrificing thermal comfort. Concretely, we focus this study on the residential facilities of Monte da Condesa, a building located at the University of Santiago de Compostela.

Monte da Condesa comprises a set of centers that act as separate buildings, but nevertheless maintain thermal interaction through their conditioning circuits connected to a common cogeneration plant. The building is about 25,000 m\(^2\) and reached in 2013 a total power consumption of 5,747 MWh. The set of all centers is supervised by a SCADA system that has more than 469 variables (inputs and outputs) that are associated with signals from the primary heating circuits and power consumption. Signals are collected in two different ways: synchronous (sync) and asynchronous (async). Synchronous signals are sequentially sampled at a fixed interval of 10 s, whereas asynchronous signals are registered by detecting a change of a value above an established threshold. These signals include information about the indoor temperature of each floor, the outside temperature, the water temperature of the pump water heating systems, plus many other low level variables. In order to predict the indoor temperatures, we focus on the variables that may directly affect the temperature dynamics. These variables are represented in Fig. 2, which shows a high-level representation of the building. \(T_0\) corresponds with the indoor temperature sensors of the building. In total, there are 6 different sensors \((T_0, \ldots, T_6)\), one for each floor, which are the objective variables we want to predict. \(T_{flow1}\) and \(T_{flow2}\) refer to the temperature of the pumped water of the two heating systems installed in Monte da Condesa. \(T_{flow1}\) corresponds with the pump water heating system that feeds both floors 0 and 1, whereas \(T_{flow2}\) feeds the remaining floors.
Fig. 3 shows an example of $T_{\text{flow1}}$ and $T_{\text{in}}^0$ between 22-02-2016 and 24-02-2016.

In addition to these SCADA variables, we also obtained the humidity $H_r$ and solar radiation power $P$ from Santiago-EOAS, a Meteogalicia [7] weather station situated approximately 100 meters from the reference building. Synchronous measures were downsampled to 1 h bins and asynchronous measures were converted into time series by applying linear interpolation and 1 h resampling. To summarize, the selected signals, sampled at 1 h interval ($t$) are:

- $T_{\text{in}}^n(t)$: indoor temperature at $t$ of floor $n$ ($^\circ$C, async).
- $T_{\text{out}}(t)$: outside temperature at $t$ ($^\circ$C, async).
- $T_{\text{flow1}}(t)$: water temperature of the first heating system (1) at $t$ ($^\circ$C, async).
- $T_{\text{flow2}}(t)$: water temperature of the second heating system (2) at $t$ ($^\circ$C, async).
- $H_r(t)$: relative humidity (% sync, Meteogalicia).
- $P(t)$: global solar radiation power ($W/m^2$, sync, Meteogalicia).

$T_{\text{out}}^d$, calculated as $T_{\text{out}}^d(t + k) = T_{\text{out}}(t + k) - T_{\text{out}}(t)$, includes information about the trend of the outside temperature in the given interval.

Note that, for the sake of clarity, in the following we will refer to $T_{\text{flow}}$ instead of $T_{\text{flow1}}$ and $T_{\text{flow2}}$, where $T_{\text{flow}} = T_{\text{flow1}}$ and $T_{\text{flow}} = T_{\text{flow2}}$ for $n \in [0, 1]$ and $T_{\text{flow}} = T_{\text{flow2}}$ for $n \in [2, 5]$. $T_{\text{flow}}$ is used to calculate other variables, such as the boiler operating percentage ($%r$) and the time since the boiler stopped working ($t_{\text{stop}}$):

- $%r$: it is calculated, from $T_{\text{flow}}$, as the percentage the boiler is working in a time interval. It is assumed that the boiler is not working when the temperature is decreasing.
- $t_{\text{stop}}$: this variable represents the time from the moment the boiler stopped working to the prediction time. If the boiler is still working, $t_{\text{stop}} = 0$.

All these features were used to construct different rule-based regression models $F$ with FRULER to predict each variable response $T_{\text{in}}^n(t + k)$, $n \in [0, 5]$ for different values of $k$ (hours ahead in time), where $T_{\text{in}}^n$ is the predicted indoor temperature on floor $n$ at instant $t + k$. $T_{\text{out}}(t)$, $T_{\text{out}}^d$, $H_r(t + k)$, and $P(t + k)$ are used in all models. Note that the values of relative humidity and global radiation are set for the prediction time. We propose three different models (Fig. 4):

- **Model 1**: $T_{\text{flow}}$ is hourly-averaged, so that four predictors ($k = 4$) are considered ($T_{\text{in}}^1, \ldots, T_{\text{in}}^5$):
  
  $$\hat{T}_{\text{in}}^n(t + k)_{M1} = F(T_{\text{flow}}(t), T_{\text{in}}^n(t), T_{\text{out}}(t), P(t + k), H_r(t + k), T_{\text{out}}^d(t + k), T_{\text{in}}^1, \ldots, T_{\text{in}}^5)$$

- **Model 2**: every 10 minutes, a value of the boiler operation is calculated from the related flow temperature. Thus, $\{\%r_0, \ldots, \%r_3\}$ are the hourly-averaged result of this binary response:
  
  $$\hat{T}_{\text{in}}^n(t + k)_{M2} = F(T_{\text{flow}}(t), T_{\text{in}}^n(t), T_{\text{out}}(t), P(t + k), H_r(t + k), T_{\text{out}}^d(t + k), \%r_0, \ldots, \%r_3)$$

- **Model 3**: this model intends to reduce the number of input variables. Therefore, the boiler operating percentage ($%r$) represents the working time for the full interval. Despite the loss of information, $t_{\text{stop}}$ is introduced to give a better comprehension of the boiler behaviour:
  
  $$\hat{T}_{\text{in}}^n(t + k)_{M3} = F(T_{\text{flow}}(t), T_{\text{in}}^n(t), T_{\text{out}}(t), P(t + k), H_r(t + k), T_{\text{out}}^d(t + k), \%r; t_{\text{stop}})$$

**IV. EXPERIMENTS AND RESULTS**

**A. Experimental setup**

FRULER was designed to keep the number of parameters as low as possible. For the instance selection technique, no parameters are needed. In the multi-granularity fuzzy discretization, the fuzziness parameter used for the generation of the fuzzy intervals from the split points was 1, i.e., the highest fuzziness value. For the evolutionary algorithm, the values of the parameters were: population size $= 61$, maximum...
Figure 4: Graphical representation of the tested models.

\[ T_{in} \quad T_{out} \quad T_{out}^d \quad Hr \quad P + \]

Model 1
\[ t \quad t \quad t + 4 \quad t + 4 \]
\[ T_{f0} \quad T_{f1} \quad T_{f2} \quad T_{f3} \]

Model 2
\[ \%r_0 \quad \%r_1 \quad \%r_2 \quad \%r_3 \]

Model 3
\[ \%r \quad t_{stop} \]

(a) Average number of rules (#Rules) and test error in °C (Test Error) for the compared models.

Table I: Experimental results of the three models.

<table>
<thead>
<tr>
<th>Floor</th>
<th>#Rules</th>
<th>Test Error</th>
<th>#Rules</th>
<th>Test Error</th>
<th>#Rules</th>
<th>Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>P0</td>
<td>9.1</td>
<td>0.46</td>
<td>5.1</td>
<td>0.50</td>
<td>11.1</td>
<td>0.54</td>
</tr>
<tr>
<td>P1</td>
<td>8.5</td>
<td>0.35</td>
<td>6.0</td>
<td>0.32</td>
<td>7.6</td>
<td>0.32</td>
</tr>
<tr>
<td>P2</td>
<td>6.2</td>
<td>0.47</td>
<td>4.6</td>
<td>0.48</td>
<td>8.4</td>
<td>0.47</td>
</tr>
<tr>
<td>P3</td>
<td>5.6</td>
<td>0.56</td>
<td>5.2</td>
<td>0.54</td>
<td>4.8</td>
<td>0.54</td>
</tr>
<tr>
<td>P4</td>
<td>5.8</td>
<td>0.43</td>
<td>4.8</td>
<td>0.41</td>
<td>6.5</td>
<td>0.40</td>
</tr>
<tr>
<td>P5</td>
<td>7.5</td>
<td>0.30</td>
<td>7.0</td>
<td>0.28</td>
<td>5.9</td>
<td>0.28</td>
</tr>
</tbody>
</table>

(b) Friedman Test.

number of evaluations = 100,000, \( p_{cross} = 1.0, p_{mut} = 0.2, \) and \( n_{ts} = 5. \) For the generation of the TSK fuzzy rule bases, the weight of the tradeoff between \( \ell_1 \) and \( \ell_2 \) regularizations on the Elastic Net is \( \alpha = 0.95, \) and the regularization parameter \( \lambda \) was obtained from a grid search in the interval \( [1, 1E − 10]. \) \( \eta^0 \) was obtained halving the initial value (0.1) until the result worsens.

A 5-fold cross validation was used in all the experiments. Moreover, 6 trials (with different seeds for the random number generation) of FRULER were executed for each 5-fold cross validation. Thus, a total of 30 runs were obtained for each model. The results shown in the next section are the mean values over all the runs. Data was recorded from 25-01-2016 to 20-05-2016 (2,754 h).

B. Results

Table Ia shows the average results of FRULER for the three learned models. For each model and floor, the table displays the number of rules of the learned knowledge base, and the test error measured in °C. These indicators allow to compare both the simplicity and the accuracy of the learned models. The values with the best accuracy —lowest error— and best number of rules —lowest value— in Table I are marked in bold.

Model 2 gets the lowest number of rules in four of the six floors. Nonetheless, the accuracy is similar for all models. In order to check whether there are significant differences among the models, we applied the Friedman statistical test, that computes the ranking of the results of the algorithms. The application of the test, using the STAC platform [10], rejects the null hypothesis which states that the results of all the algorithms are equivalent with a given confidence significance level (\( \alpha = 0.05 \)). Table Ib shows the ranking for the test error and the p-value of the test, which indicates that the differences among the models are not statistically significant.

V. CONCLUSIONS

In this paper we presented three different models for indoor temperature prediction using the FRULER Genetic Fuzzy System to generate the knowledge base, made up of TSK fuzzy rules. The models have been learned from data recorded at Monte da Condesa Residential College during 2,754 hours and from several sensors. The models can predict the future indoor temperature for each floor of the building with an average error in the range 0.28-0.50 °C. The learned models will be used in the near future in the LIFE-Opere EU project [4] for planning efficient heating control strategies, predicting the indoor temperature in order to guarantee that the global power consumption of the heating system is reduced without sacrificing thermal comfort.

REFERENCES


Abstract — With the rise of sports analytics in basketball, teams are focused on implementing mathematically “efficient” play. Relatively new statistics have allowed NBA teams to model and describe phenomenon within the game accurately, describe keys to success, and improve overall scoring efficiency. This project aims to model the jump shot performance of the NBA regular season by utilizing the player tracking statistics and machine learning techniques. Through the process stepwise logistic regression, the most significant features to that affect shot performance will be described.

Keywords — NBA; Logistic Regression; Stepwise Regression; Goodness of Fit;

I. INTRODUCTION

Sports analytics in basketball have grown in popularity within the past couple years [5]. The NBA has launched its own site (http://www.nba.com/stats) dedicated to statistical analysis of players and teams in 2013 [13]. The site filled with live player tracking stats enabled its fans to “digest it the way they wanted to” and opened the door for more people to uncover interesting mathematical truths about the game [5]. The goal of the project is to build a model of predicting NBA Jump Shot Performance using machine learning techniques and the newly available data [4].

Success in basketball, like many other sports, is taught through its fundamentals. As basics like jump shooting and dribbling are honed, advanced team strategies become the pathway to success for most teams. Ball movement, separation from defenders, and shot selection are some examples of common themes in high level basketball coaching staff philosophies. By quantifying and trying to observe these characteristics of the game, an approximate model can be built describing the performance of jump shots in the NBA. Since the game of basketball is a highly complex system, this project hopes to simplify its goals by only focusing on pure jump shots. Layups, dunks, and bank shots (shots that bounce off the backboard) are excluded from this analysis. By adequately predicting the probability of a jump shot going in, the model also aims to assess which features of the game are most important in jump shot performance.

II. RELATED WORK/BACKGROUND

Although the field of sports analytics has grown [14], there are not many professional articles published on the topic of jump shot regression. The major source of papers published on NBA statistical modeling come from the MIT SLOAN Sports Analytics Conference [11]. In this paper, we look at using logistic regression for jump shot prediction. This problem is similar to that found in medical disease regression. Here, we estimate the probability of a basketball jump shot being made based on the statistics of the game. The response or outcome is binary but has an underlying probability that governs the result. Similarly, in medical disease regression, a patient either has the disease or not and the goal is to predict the probability of a patient contracting a disease.

The simplest regression technique is the standard linear least squares approach. Standard linear regression models are not appropriate for determining probabilistic outcome for two main reasons [10]. These models do not contain bounds, therefore creating the possibility of an outcome with a probability higher than 1 or than 0. Additionally, linear regression models assume equal variance across features which is fundamentally incorrect for this problem. By adding a modification to the standard regression model, logistic regression allows for predictive probabilistic outcomes [6]. Logistic regression is found under the topic of generalized linear models. The generalized linear model is a regressive solution using a linear combination of features along with a link function, an exponential equation in this case. The set of functions used by logistic regression,

\[ \pi(x) = \frac{e^{g(x)}}{1 + e^{g(x)}} \]

\[ g(x) = \beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N \]

are extensions of the familiar linear regression models. Additionally, they only output values ranging from 0 to 1 and can be generalized for multiple independent variables.

Medical studies use logistic regression to build predictive models of contracting diseases or symptoms. One example would be modeling the odds of developing angina pectoris (chest pain) based on cholesterol, sex, smoking, age, body mass index, and heart rate [10]. The maximum likelihood estimated coefficients in the model are found using iterative reweighted least squares methods [6]. After an estimated model
In order to build a successful model, a researcher needs to evaluate and compare the goodness of fit (GOF) of different solutions [8]. The two common statistics that assess GOF in logistic regression are the deviance and the Pearson chi-square statistic [8]. Both statistics use the calculation of residuals and their distributions cannot be easily interpreted under the assumptions of this problem [8]. Both statistics have degrees of freedom (DOF) approximately equal to \( n - (p + 1) \) [8]. The statistics themselves can only be compared to their DOF [6]. For example, one model of the angina study resulted in the summary statistics listed below [8].

\[
D = 293.1 \\
\chi^2 = 283.7 \\
DOF = 279
\]

**Figure 3. Angina Study GOF Summary Statistics**

Since the deviance and Pearson chi-square statistic were relatively close to the degrees of the freedom, a researcher could say that the model has an adequate goodness of fit [9]. However, when adding a third GOF statistic, the Hosmer-Lemeshow statistic, \( C = 22.1 \), the model is questioned [8].

The Hosmer-Lemeshow (HL) statistic, \( C \), is a decile based GOF statistic that groups similar predictive values [9]. The predicted probabilities are sorted and divided into 10 groups [6]. Then the expected number of positive and negative outcomes are compared to the observed [6]. The grouping of similar probabilities creates a summary statistic that has a chi-square distribution with degrees of freedom equal to the number of groups minus two [6]. Since we know the distribution of the summary statistic, we can test the goodness of fit by assessing the p-value of C [6]. This interpretation of p-value is different than previously mentioned. In this case, the p-value of the HL statistic should be high to claim that the estimated coefficients appropriately describe the outcome [6]. If it is less than \( \alpha \), then the researcher can assume that is highly likely that the estimated coefficients can adequately describe the outcomes [6]. In the example above, where \( C = 22.1 \) and the DOF=10-2=8, a p value of 0.005 is obtained. Since the p-value is small, the model is considered inadequate [8]. Therefore, a combination of deviance, Pearson chi-square statistics and a decile based summary statistic are necessary when assessing GOF in logistic regression [8]. Additionally, it is beneficial to analyze the individual residuals calculated during the process for greater insight into GOF [8].
III. PREPROCESSING DATA/PRELIMINARY ANALYTICS

Organizing and preprocessing the large amount of data was a daunting task. In order to pull the data from the NBA site, a Python script using JavaScript requests was developed. This project was my first time using Python, which created a large learning curve and ate up a lot of time. After learning some basics of the language, I was able to modify some data mining scripts found online and saved off all the jump shot information for 2014-15 NBA regular season. The data for this project was found across multiple JavaScript data structures and required some organizing and preprocessing for regression [13].

The available features for regression analysis are listed below. Most of the features used in this project were found in the data structures available. However, some new features were developed based on knowledge of the game. These features are denoted with an asterisk.

- Time Remaining (Period, Minutes, Seconds)
- Shot Distance (ft)
- X and Y positions on court when jump shot is attempted (ft)
- Shot Clock (s)
- Average Player Speed (3 seconds before shot) (ft/s)*
- Average Ball Speed (5 seconds before shot) (ft/s)*
- 5 Defender Distances when a shot is attempted (ft)*
- Team (Categorical Feature)
- Shot Type (Categorical Feature 2 Pt vs 3 Pt)

Due to the erratic nature of the data structures, the full dataset had to be trimmed to valid jump shots. Invalid observations occurred because of data drops, time mismatches, edge cases, out of court positions, and many more errors. Once trimmed to valid jump shots, the dataset was comprised of 20077 jump shots across the 2014-15 NBA regular season.

An exhaustive analysis on each feature was conducted, and the highlights of each analysis are shown here. By looking at some of the feature histograms, we can see some limitations of the data provided along with statistical anomalies. For example, the recording of Shot Distance was always rounded to the nearest foot as seen in Figure 4 below. The discrete nature of this feature might have an effect on the regression analysis. Disregarding the discrete behavior, the Shot Distance feature seems to have a multimodal distribution with three spikes. The first spike occurs around 4 ft (near the basket), the second at 18 ft (mid-range 2 pointers), and the last occurring at 24 ft (3 point line). One can infer that although we limited our analysis to only pure jump shots, there are still sub categories that are not well described within our feature space.

Another similar statistical anomaly found in the data was the large occurrence of integer Shot Clock values. Although not as sparse as the Shot Distance feature, clear spikes can be found in the histogram at every second (see Figure 5 below). The large spike at 24 seconds corresponds to the shot clock running out which forces the players to take a jump shot or surrender the ball. Otherwise the shot clock feature seems to have a somewhat normal distribution.

Fortunately, no other anomalies were found in the feature observations. Due to the large number of observations, many of the features themselves seemed to tend towards well known distributions. For example, the Defender Distances along with Ball and Player Speed features looked similar to a chi-square distribution with 2 degrees of freedom. This is somewhat expected due to the sum of squares nature of the distance calculation.
Looking at the X position histogram in Figure 7, the distribution is multimodal and is symmetric about the half court mark. Although this is expected from what we saw within the Shot Distance histogram, this feature is inappropriate for a generalized linear regression. No linear relationship can separate the two sides of the court correctly since made and missed shots lie intertwined throughout the distribution. Therefore, a nonlinear transformation must be applied to this feature when used in regression analysis. By plotting the X and Y positions of each jump shot on a scatterplot and distinguishing between made and missed shots (Figure 8), there is a clear relationship between the probability of a shot going in and the position on the court. However, due to the limitation of the regression technique, the model might not be able to capture this phenomenon within the features.

Building a feature space for regression analysis requires somewhat uncorrelated variables [6]. Although the ideal analysis occurs with orthogonal features, this scenario is highly unrealistic. Figure 9 below shows the correlation matrix of the features available for regression. Most of the features seem to have low cross-correlation scores, but many of the newly developed features have cross correlation scores higher than 0.6. This is expected due to the fact that the five defender distances are directly related in terms of defensive spacing. It is very rare for a team to bunch their defenders on the floor, except in double team or help situations. Since most of the features provided for the regression are uncorrelated, the algorithms are expected to behave normally and we can continue to assume independence.
The second model described utilized the stepwise algorithm to rank the significance of the variables based on an exhaustive iterative search. The table below shows the summary statistics of the stepwise model only containing linear terms. The variables on the left hand side are ordered by importance or when the algorithm added it to the model. The two alphas used to train this algorithm were 0.15 and 0.2, which are looser common values used in stepwise regression [6]. Although the Pearson chi-square statistic is relatively close to the DOF, the deviance and HL statistic suggest that this model is still a poor fit. However, both models have stated similar variables contribute to fitting the underlying probabilities. Interestingly, the stepwise algorithm chose to include the 5th Defender Distance. This occurred because of the looser constraint on significance (α=0.15) and suggests that the maximum spacing between defenders may help improve the model. Additionally, the stepwise algorithm included the Time Remaining as the 4th most important variable, while the full model chose to discard it.

<table>
<thead>
<tr>
<th>Variable</th>
<th>β</th>
<th>σ</th>
<th>P Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-0.3083</td>
<td>0.1113</td>
<td>0.0056</td>
</tr>
<tr>
<td>Period</td>
<td>-0.0756</td>
<td>0.0542</td>
<td>0.1636</td>
</tr>
<tr>
<td>Time Remaining</td>
<td>0.0034</td>
<td>0.0044</td>
<td>0.4417</td>
</tr>
<tr>
<td>Shot Distance</td>
<td>0.0186</td>
<td>0.0042</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>X Position</td>
<td>0.0003</td>
<td>0.0005</td>
<td>0.4885</td>
</tr>
<tr>
<td>Y Position</td>
<td>-0.0001</td>
<td>0.0011</td>
<td>0.9100</td>
</tr>
<tr>
<td>Shot Clock</td>
<td>-0.0057</td>
<td>0.0028</td>
<td>0.0447</td>
</tr>
<tr>
<td>Player Speed</td>
<td>-0.2960</td>
<td>0.1261</td>
<td>0.0189</td>
</tr>
<tr>
<td>Ball Speed</td>
<td>0.3325</td>
<td>0.1112</td>
<td>0.0028</td>
</tr>
<tr>
<td>1st Def. Distance</td>
<td>0.0590</td>
<td>0.0055</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>2nd Def. Distance</td>
<td>0.0002</td>
<td>0.0055</td>
<td>0.9649</td>
</tr>
<tr>
<td>3rd Def. Distance</td>
<td>-0.0040</td>
<td>0.0060</td>
<td>0.5066</td>
</tr>
<tr>
<td>4th Def. Distance</td>
<td>0.0023</td>
<td>0.0058</td>
<td>0.6873</td>
</tr>
<tr>
<td>5th Def. Distance</td>
<td>-0.0049</td>
<td>0.0041</td>
<td>0.2298</td>
</tr>
<tr>
<td>Team</td>
<td>0.0002</td>
<td>0.0017</td>
<td>0.9260</td>
</tr>
<tr>
<td>Shot Type</td>
<td>-0.3588</td>
<td>0.0492</td>
<td>&lt; 0.0001</td>
</tr>
</tbody>
</table>

D = 25544.8, $\chi^2 = 20056.6$, DOF = 20068
C=2515.1, P = 0

Figure 11. Stepwise Logistic Model Summary

The last model utilized the stepwise algorithm but also included interactions between variables and quadratic terms. The hope of adding complexity to the model would create a better fit for the underlying probabilities. The table below describes the summary statistics of the model generated. The variable names are listed using Wilkinson notation. A colon '::' denotes multiplicative interaction and 'i^2' denotes squaring the variable. Similar to the second model, the variables are listed in order of significance. Within this model, the amount of predictors used dropped from 8 to 5 but still includes 8 terms. The added use of interactive and quadratic terms suggest that the available features were not sufficient or required additional preprocessing. The features used in this model reiterate the importance of the variables found in previous models. 1st Defender Distance, Shot Type, Shot Clock, and Ball Speed are clear indicators of modeling the underlying probabilities. However, they do not fully approximate the function since the
deviance, Pearson chi-square and Hosmer-Lemeshow statistic still suggest that this model has a poor goodness of fit.

<table>
<thead>
<tr>
<th>Variable</th>
<th>Estimate Coefficient (β_i)</th>
<th>Estimated Standard Error (σ)</th>
<th>P Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intercept</td>
<td>-1.5896</td>
<td>0.1262</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>1st Defender Distance</td>
<td>0.1252</td>
<td>0.0173</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>1st Defender Distance^2</td>
<td>-0.0061</td>
<td>0.0008</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Shot Type</td>
<td>-0.2557</td>
<td>0.0366</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Shot Clock</td>
<td>0.11004</td>
<td>0.0110</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Shot Clock^2</td>
<td>-0.0048</td>
<td>0.0004</td>
<td>&lt; 0.0001</td>
</tr>
<tr>
<td>Ball Speed</td>
<td>-0.2644</td>
<td>0.2213</td>
<td>0.2320</td>
</tr>
<tr>
<td>Ball Speed: 1st Defender Distance</td>
<td>-0.0774</td>
<td>0.0277</td>
<td>0.0052</td>
</tr>
<tr>
<td>Time Remaining</td>
<td>-0.0022</td>
<td>0.0011</td>
<td>0.0352</td>
</tr>
</tbody>
</table>

\[ D = 25355.5, \quad \chi^2 = 20051.5, \quad \text{DOF} = 20068 \]
\[ C = 1192.3, \quad P = 0 \]

**Figure 12.** Stepwise Interactive Logistic Model Summary

The goodness of fit summary statistics are useful in determining overall performance of a model. The individual residuals used to calculate those statistics can give some additional insight [8]. In Figure 13 below, the Pearson residuals are plotted against the estimated probability of the models. The ‘x’s denote the missed shots while ‘o’s denote the made shots. For all three models, the residuals for the made shots were larger than the residuals for the missed shots. As the model complexity increased, the overall mean of estimated probability decreased.

Although the models poorly fit the data, the stepwise algorithms stated that the variables have some predictive power to approximate the underlying probability. In order to assess the models, the specificity and sensitivity of each model were calculated. Specificity of a model describes the percent of correctly classified negative occurrences, or missed shots, to all missed shots [9]. Sensitivity of a model describes the percent correctly classified made shots to all made shots [9]. These metrics can only be calculated when assuming a threshold probability for binary classification [9]. By varying the threshold from 0 to 1, a Receiver Operator Characteristic (ROC) curve is generated [9]. In Figure 14 below, the ROC curve for the three generated models. In order to compare the three models, the area under the ROC curve (AUC) is used as a quantifiable summary statistic [9]. The difference in AUC between the full model and the stepwise model is negligible (~0.1%). However, the introduction of interactive and quadratic terms increased the correct classification by 1.5 percent. The black dotted line shows the probability of flipping a coin for reference. All three models outperformed a coin flip by roughly 7 percent.

**Figure 13.** Pearson Residuals vs Estimated Probability

**Figure 14.** ROC Curve

\[ \text{AUC}_{\text{Full}} = 0.571, \quad \text{AUC}_{\text{Step}} = 0.572, \quad \text{AUC}_{\text{StepInter}} = 0.587 \]

V. CONCLUSION

Although the models obtained did not fit the data well, they gave some indication of rejecting the null hypothesis. The added complexity and nonlinear relationships improved the fit, but did not adequately describe the data. Due to the models barely outperforming a coin flip predictor, new approaches and techniques must be applied to solve this problem. The addition of features available for the stepwise algorithm could increase the GOF.

By analyzing some of the characteristics of the previous models, some new features are proposed. A feature which describes the area covered by the defense and offense could be beneficial. These features would utilize the area of a pentagon.
created by the player positions to describe the spacing of the floor. Offensive and defensive spacing are consistently mentioned as keys to success in basketball and could provide better results. Additionally, derivatives of the speed features could provide additional insight in fast paced vs slow offenses. The last proposed feature utilizes the z dimension of the basketball tracking data. A feature could be created to gauge the seconds between the jump shot and the last dribble. Unfortunately, these features were thought of much too late to contribute to the models in this project. Since some of the existing features contained statistical anomalies, cleaning up the existing data might increase the possibility of creating a sufficient model. By getting a more continuous shot distance and somehow removing the statistical jumps of integer Shot Clock seconds could prove beneficial to the model. Lastly, applying a nonlinear transformation to the x position data might make it more useful for regression.

Although R-squared metrics are used to gauge predictive power of regressions, there is no closed form for the logistic problem. Many were proposed, such as the McFadden and Tjur [9], and the field is still debating over the best method for evaluation.

The complexity of this problem was appreciated and the reasoning for NBA teams to pour money into their analytics programs was justified.

REFERENCES

Data Mining for the XXI Century

João Gama

Abstract—We are witnessing the big-bang of digital data. This talk discusses the impact of the exponential growth of digital streaming data on the design of learning algorithms. We discuss issues like learning and forgetting, non-stationarity and change detection, incremental learning and stability. We present illustrative algorithms for efficient learning in the XXI century.

Index Terms—Evolving Data, Change Detection, Data Streams.

I. INTRODUCTION

Nowadays, there are applications in which the data is modelled better as transient data streams rather than as persistent tables. In this article, we discuss the limitations of current machine learning and data mining algorithms. We discuss the fundamental issues in learning from dynamic environments like continuously maintain learning models that evolve over time, learning and forgetting, concept drift and change detection. Data streams produce a huge amount of data that introduces new constraints on the design of learning algorithms: limited computational resources in terms of memory, CPU power, and communication bandwidth. We present some illustrative algorithms, designed to take these constrains into account, for decision-tree learning, hierarchical clustering and frequent pattern mining. We identify the main issues and current challenges that emerge in learning from data streams that open research lines for further developments.

II. HIERARCHICAL CLUSTERING OF SENSORS

The Online Divisive-Agglomerative Clustering (ODAC) is an incremental approach for clustering streaming data sources using a hierarchical procedure [5,6]. It constructs a tree-like hierarchy of clusters, using a top-down strategy based on the correlation between data streams. The system uses two main operators: a splitting operator, that expands the structure by diving one cluster into two; and a merge operator that contracts the structure by merging a parent cluster with its two descendent clusters. The splitting operator triggers when there is enough information to define more detailed clusters. The merge operator triggers when the correlation structure in the most recent data is different from the correlation structure in the past data. The splitting and merge operators are based on the diameters of existing clusters and supported by a significance level given by the Hoeffding bound [7].

A. Incremental Dissimilarity Measure

We use Pearson’s correlation coefficient between time series as a similarity measure. Deriving from the correlation between two time-series and calculated in [1,5], the factors used to compute the correlation can be updated incrementally, achieving an exact incremental expression for the correlation:

\[ \text{corr}(a,b) = \frac{p - \frac{A.B}{n}}{\sqrt{A^2 - \frac{A^2}{n}} \sqrt{B^2 - \frac{B^2}{n}}} \]

The sufficient statistics needed to compute the correlation are easily updated at each time step: \( A = \sum a_i \), \( B = \sum b_i \), \( A^2 = \sum a_i^2 \), \( B^2 = \sum b_i^2 \), \( P = \sum a_i b_i \). In ODAC, the dissimilarity between variables and is given by an appropriate metric, the Rooted Normalized One-Minus-Correlation given by:

\[ \text{rnorm}(a,b) = \sqrt{1 - \text{corr}(a,b) \frac{2}{2}} \]

rnorm(a,b) is in [0,1], where 0 means high similarity, 1 means low similarity.

B. The splitting operator

The splitting operator works by selecting the two less correlated sensors for pivots of the descending clusters. Each pivot attracts the most correlated sensors.

The main problem in the splitting operator is: how many examples do we need to observe to make a decision? There are two approaches: 1) make a decision based on a predefined number of examples; 2) only make a decision when the information collected so far is enough to discriminate between the two best alternatives. In Hoeffding algorithms, we use the second approach. Assuming that \( d_1 = \text{rnorm}(a,b) \) is the \( \text{rnorm} \) distance between the two less correlated sensors, and \( d_2 = \text{rnorm}(c,d) \) is the second furthest \( \text{rnorm} \) distance. Any greedy algorithm will split using the diameter and as pivots. In ODAC, we split using a and b as pivots, only if there is statistical evidence in favor of this alternative. If \( d_1 - d_2 > \varepsilon \),
where $\varepsilon = \sqrt{\frac{R^2 \log (\frac{q}{n})}{2n}}$ is the Hoeffding bound [4]. If the difference between the two best alternatives does not satisfy the Hoeffding bound, we collect more information.

C. Change detection

The splitting rule used in ODAC, guarantees that when a cluster is divided, the diameter of descending clusters cannot be larger than the diameter of a parent cluster. This is true if, and only if, the process generating data is stationary. Assuming we have a cluster $C_i$ with two descendent clusters $C_i$ and $C_j$. If at any time we observe that $\text{diameter}(C_i) > \text{diameter}(C_k) + \varepsilon$ or $\text{diameter}(C_j) > \text{diameter}(C_i) + \varepsilon$ 1, this is an indication that the correlation structure in the most recent data (stored in the descendent clusters) is different from the data observed in the past. We merge the 3 clusters, and reset the sufficient statistics.

D. Analysis

ODAC has been previously tested on electricity demand sensor data streams, which proved its strengths regarding memory and processing load. From the raw data received at each sub-station, taking into account only the current intensity sensors, observations were aggregated on a hourly basis over more than two and a half years. This data set represents 2700 sensors along 22364 observations [5] (see Figure 1).

![Figure 1 Cluster structure after processing 6 months of electrical data](image)

The main characteristics of the system are constant in memory and time with respect to the number of examples. In ODAC, system space complexity is constant on the number of examples, even considering the infinite amount of examples usually present in data streams. An important feature of this algorithm is that every time a split is performed on a leaf with $n$ variables, the global number of dissimilarities needed to be computed at the next iteration diminishes at least $n-1$ (worst-case scenario) and at most $n/2$ (best-case scenario). The time complexity of each iteration of the system is constant given the number of examples, and decreases with every split occurrence, being therefore capable of addressing data streams.

ODAC is one of the so called Hoeffding algorithms: incremental adaptive algorithms. Similar algorithms for learning decision trees and rules [4,8], regression and model trees and rules [8,9], ensemble models [2], etc. A distinctive idea is the sample size required to make a decision; it is the sample that allows us to discriminate between the two best alternatives.

III. CONCLUSION

Current machine learning systems, that assume iid examples and stationarity of the data need to be reconsidering from scratch. With our ability to collect data online, the need of real-time machine learning is urgent. Learning from dynamic environments, requires learning algorithms able to reason about the learning process itself. Intelligent agents that adapt over time in a dynamic and sometimes in adversary conditions, should be capable of self-diagnosis (see Figure 2) [2,3]. A significant and useful intelligence characteristic is diagnostics—not only after failure has occurred, but also predictive (before failure) and advisory (providing maintenance instructions). The development of such self-configuring, self-optimizing, and self-repairing systems is a major scientific and engineering challenge. All these aspects require monitoring the evolution of the learning process itself, and the ability of reasoning and learning about it.

REFERENCES


1 Where $\varepsilon$ is the Hoeffding bound.
Dynamic Learning of Cases from Data Streams

Fernando Orduña-Cabrera and Miquel Sánchez-Marrè

Abstract—This paper presents a dynamic adaptive framework for building a case library being able to cope with a data stream in the field of Case-Based Reasoning. The framework provides a three-layer architecture formed by a set of case libraries dynamically built. This Dynamic and Adaptive Case Library (DACL), can process in an incremental way a data stream, and can be used as a classification model or a regression model, depending on the predicted variable. In this paper, the work is focused on classification tasks. Each case library has a first layer formed by the dynamic clusters of cases, a second one formed by the meta-cases or prototypes of the cluster, and a third one formed by an incremental indexing structure. In our approach, some variant of $k$-d tree have been used, in addition to an exploration technique to get a more efficient retrieval time. This three-layer framework can be constructed in an incremental way. Several meta-case learning approaches are proposed, as well as some case learning strategies. The framework has been tested with several datasets. The experimental results show a very good performance in comparison with a batch learning scheme over the same data.

Index Terms—Case-Based Reasoning, Data Streams, Dynamic Learning of Cases, Incremental Learning.

I. INTRODUCTION

In recent years, the problem of mining data streams has grown the attention of many researchers. Many real-world applications generate data continuously. For example, in network monitoring, telephone record calls, multimedia data, customer transactions, customer click streams, and so on. Advances in technology have facilitated new ways of continuously collecting data. In many applications, the volume of such data is so large that it may be impossible to store the data on disk. Furthermore, even when the data can be stored, the volume of the incoming data may be so large that it may be impossible to process any particular record more than once. Therefore, many data mining and database operations such as classification, regression, clustering, frequent pattern mining and indexing become significantly more challenging in this context [1]. The monitoring of many events in real time produces much information. In recent years, data stream mining field has grown rapidly. In [2] outlined some desirable properties for learning tasks in data streams: incrementality, constant time to process each example, single scan over the training set, and taking drift into account. Learning from data streams require incremental learning algorithms that take into account the problem of concept drift. The underlying concept or distribution of the data can change over time, and the mined models should be aware of the changes, and adapt themselves to the changes.

On the other hand, Case-Based Reasoning (CBR) systems solve new problems by retrieving and adapting the solutions to previously solved problems that have been stored in a case library [3][4][5].

CBR is a very flexible reasoning paradigm, which can be used both as a classification technique and as a regression technique. The most common approach to predict a class label is the use of a simple CBR scheme (k-nearest neighbor classifier), but it can also be used to predict numerical variables, in a regression problem. This flexibility makes it a powerful tool for data mining. Furthermore, CBR integrates a learning step in its basic reasoning cycle. This learning activity makes CBR to be very suitable to be used for dynamic learning purposes. CBR systems become more competent over time, because they learn from experience. CBR approaches can process a data stream with a fine grained time window of length one. They can process full the examples/cases one by one and adapt their model at each example.

In this paper, a dynamic adaptive framework is proposed to improve the CBR system performance coping especially with reducing the retrieval time, increasing the CBR system competence, and maintaining and adapting the case library to be efficient in size, especially in continuous domains (data streams) [6]. The framework proposed works for reasoning and learning both in supervised domains and unsupervised domains. One of the main contributions of the work is the proposal of a Dynamic Adaptive Case Library (DACL) framework. A DACL is composed of a set of dynamically built case libraries to cope with the heterogeneity and complexity of real domains. It learns cases and organizes them into dynamic cluster structures. The DACL is able to adapt itself to a dynamic environment, where new clusters, meta-cases or prototype of cases, and associated indexing structures (discriminant trees, $k$-d trees, etc.) can be formed, updated, or even removed. DACL offers a possible solution to the management of the large amount of data generated in an unsupervised continuous domain (data stream).

A very important aspect related to unsupervised continuous domains is the incrementality problem. General CBR systems assume that the set of cases available for building the case library is fixed and available at the beginning (batch learning). Then they build the memory indexing structures, like for instance, a $k$-d tree, decision/discriminant tree, etc. However,

This work was partially supported by the Spanish Thematic Network DAMA the Spanish Ministry MINECO, under Grant TIN2015-70308-RIDT. Fernando Orduña-Cabrera is with the Instituto Tecnológico Superior de Cajeme (ITESCA), Cajeme, Sonora, México (e-mail: forduna@itesca.edu.mx).

Miquel Sánchez-Marrè is with Universitat Politècnica de Catalunya-BarcelonaTech, Barcelona, Catalonia, Spain (e-mail: miquel@cs.upc.edu).
when a CBR system is facing an unsupervised continuous domain, the system should build and update the case library structure/s in an incremental way (incremental learning).

II. RELATED WORK

In machine learning literature, several works have addressed the problem of learning from data streams [7][8], and other works studied the time changing concept problem [9][10][11][12]. Most common techniques used are temporal windows, which determines the training set for the learning algorithm, and the weighting of examples, which attempts to decrease the relevance of the older examples, and increase the relevance of the new ones. Also there are some mixture techniques. Some authors [13][14][15] propose the use of adaptive time windows in order to minimize the generalization error of the classification models.

Continuous problem domains (i.e., domains where cases are generated from a continuous data stream) require different underlying representations and place additional constraints on the problem solving process [16]. Ram and Santamaria define three characteristics where the problem domain is continuous, and those are: First, they require continuous representations. Second, they require continuous performance. Third, these problem domains require continuous adaptation and learning. As the problems encountered become more varied and difficult, it becomes necessary to use fine-grained, detailed knowledge in an incremental manner to act, and to rely on continuous feedback from the environment to adapt actions and learn from experiences.

Reasoning about continuous domains is not an easy task. Moreover, this is a domain where CBR can rapidly extend its benefits because data is systematically collected for its analysis. A CBR system that continuously interacts with an environment must be able to create autonomously new situation cases (new concepts or clusters) based on its perception of the local environment in order to select the appropriate steps to achieve the current mission goal [17], but a general framework is still missing. Some systems that use case-based methods in continuous environment are described in [18][19][20].

There are two other central problems derived from the continuous nature of some domains. First of all, the size of the case library could grow very fast as the CBR system is learning new cases without an extensive improvement in the competence of the system, as pointed out in [21]. Two natural human cognitive tasks appear as the solution to these problems: forgetting ([22]) and sustained relevant learning [23]. On the other hand, learning many cases could provoke an overhead in the case library organization. As new cases are stored in the case library, it will be necessary to update the case library organization [24].

III. THE DYNAMIC ADAPTIVE CASE LIBRARY FRAMEWORK

When CBR systems are deployed in continuous domains, the case maintenance becomes critical. An uncontrolled growth of the case library can cause some serious problems of performance, where the retrieval efficiency and the quality of the retrieval is affected. When the case base has large amount of data, some inconsistent cases could be stored. This condition affects directly in the performance of the library.

The architecture proposed presents a Dynamic Adaptive Case Library (see figure 1), with the aim of giving a possible solution to the management of the large amount of data generated in a continuous domain. The library will be dynamically built and formed of several sub-libraries. Each sub-library is organized hierarchically at three levels: The Meta-case: The Meta-case is the prototype of a concrete cluster of cases. The clusters: The set of cases belonging to the same cluster, and that are being represented by the meta-case. The Indexing structures: They implement the way that all the cases are organized in the sub-library. In our proposal, the cases are organized in a hierarchical indexing structures (k-d trees, discriminant trees, etc.), but could be organized with other indexing approaches.

Each time a new cluster of cases is created, a new sub-library is grown up. Next, the Meta-case structure and the retrieval and learning steps in a DACL will be detailed.

A. The Meta-case

The idea of using a Meta-case as a representative case of several similar cases was introduced by Sánchez-Marré in [25]. The aim is to show a formal proposal of how to a Meta-case (Mc) can be built. For our goal, a Meta-case is the prototype of a set of related cases. The centroid value is generated taking into account the whole cases stored in the indexing structure. With the following formula: $Mc_j = \sum_{i=1}^{n_j} c_j^i / n_j$ where $j = 1, ..., m$. The average distance (centroid) of the set of cases in that cluster is computed. A case ($C_j^i$) and a Meta-case ($Mc_j$) are described by $m$ attribute values. That is the first proposal that was introduced in [26]. Our proposal of constructing Meta-cases [27] is where the stochastic methods is introduced and prove it. DACL have as representative case a Mc, this Mc works like a clustering filter where the decision to learn a new incoming case (Mc) is made, this decision concerns to a method to evaluate the Mc’s and find the most appropriate Mc where to learn the Nc. Here follows the formalization of this process:

$$C_j = (c_{j1}, c_{j2}, ..., c_{jn})$$  
$$Mc_j = (Mc_{j1},麦c_{j2}, ..., Mc_{jn})$$

Where $n_j$ = #Cases represented by the Meta – case($Mc_j$)

$$Mc_j = \frac{1}{m} \sum_{i=1}^{n_j} c_j^i$$  
If $j$ is a qualitative attribute, and

$$Mc_j = mode(c_j^k)$$

Where $n_j$ = #Cases represented by the Meta – case($Mc_j$)
If \( j \) is a quantitative attribute

The Meta-case structure improves the performance of the retrieval time according to the proposals of Orduña and Sánchez-Marré in \[27]\[26]. The Meta-case is related to the clustering and learning processes.

**B. Retrieval Process in a DACL**

Retrieving similar cases regarding to a new case is a process that needs to be done accurately. A new algorithm (DACL Retrieval algorithm) to retrieve the most similar case or cases in the DACL it is defined. First, the distance between the new case and all the Meta-cases must be computed. The most similar Meta-case will be selected, and its corresponding k-d tree, will be traversed for searching the most similar cases.

**C. Learning Process in a DACL**

Retain task aims to maintain (Basic retaining/learning algorithm) a competent Case Library with a high coverage. The Retain process decides whether the new case needs to be stored in the case library, by updating an existing sub-library, building a new sub-library or simply ignoring the case. The process to make a decision is guided by the learning algorithm. The basic retaining/learning algorithm (McSel-1) works as follows: it receives a solved new case (Ne) and then computes the distance to all the Meta-cases, with the aim to find the closest Meta-cases. Once the best Meta-case is found, it proceeds to compare whether the distance found falls within the \( \alpha \) threshold (previously defined by the experts or tuned by trial and error experimentation). Then, it proceeds to store the new case into the current library (see figure 2). Otherwise, if the distance is higher than the \( \alpha \) threshold value, a new sub-library must be created containing the new solved case. The last consideration in the algorithm is when the distance of the solved case falls within the ratio of two or more meta-cases. The solved case is randomly stored into one sub-library.

**D. Other Meta-case building strategies**

In addition to the MCSel-1 strategy, we have proposed other approaches. Using a relaxed maximum radius, and avoiding impasses with real Meta-cases. These strategies are described below.

1) **Using the Maximum Radius for building Meta-cases**

The criterion uses the same procedures for building the prototypes using the idea of a radius around the Meta-case (McSel-1) but being a dynamic one regarding the farthest case of the prototype. The case at maximum distance of the Meta-case (RMax) is used with an extra relaxation condition \( (\gamma) \), to decide whether a new case should be stored in the library or not. This means that the criteria must successfully compute the \( Mc^j \) where

\[
\gamma = \arg \min \{D(Nc,Mc^j) \mid D(Nc,Mc^j) \leq Mc^j \cdot RMax \cdot (1 + \gamma)\}, \quad \gamma \in [0,1]
\]

**Fig 3. Learning with RMax and relaxation factor (McSel-4)**

2) **Learning Meta-cases avoiding impasses with real MC (McSel-5)**

There is the possibility to work with real Meta-cases instead of the virtual Meta-cases. The difference relies in the fact that the prototypes of each cluster (virtual meta-cases) are replaced for real cases. This means that the prototype of a cluster of cases is a case existing within the set of cases of the cluster. Concretely, the real Meta-case will be the nearest real case to the virtual meta-case.

A variation of strategies McSel-1 and learning with real Meta-cases could be used as a good solution for impasse situations. Impasse situations happen when a new case which must be stored in the DACL is equally similar to more than one virtual meta-case (prototype). Even though a case could be at the same distance to several virtual Meta-cases, perhaps the distance to the corresponding real Meta-cases will not be the same, and the impasse situation could be solved. The strategy will be named as McSel-5. The strategy is depicted in figure 4.

**Fig 4. Impasse resolution using real Meta-cases (MCSel-5)**

**E. Incremental building of NIAR k-d Trees**

The proposed NIAR k-d tree technique [28] has two main steps based on the computation of the average value of the corresponding attribute among the sub-tree cases, and selecting for that attribute, the value of the Nearest Instance/case to the Average as the Root (partition value). Several experimental results with some databases have shown that the retrieval in NIAR k-d tree is faster than in the standard k-d tree approach, and that using a Partial Matching Exploration (PME) technique to traverse the k-d tree, the accuracy in classification tasks is just a little bit lower than some exhaustive exploration techniques (see [28][29]). In the current work, the NIAR k-d tree structure, in addition to the PME technique has been used as the indexing strategy in the DACL framework. A NIAR k-d tree can be created in an incremental way, processing each case at a time. Each new case arriving from the data stream, traverses the tree, and according to the comparison
with the corresponding partition values at each attribute node, follows the traversal until a non-full bucket is found (leave), or a full bucket is found (stored cases is equal to d). If the leaf is not full, then the case is stored in that leave, updating several statistics associated with each attribute node (mean value of the attribute, standard deviation of the attribute, current new mean value, current new standard deviation, number of cases, and number of disturbance values) in all cases in the subtree. If the leaf is full, then a split operations must take place. The leave node is converted into an attribute node, with the next attribute according to the specified criteria (random, etc.), and the corresponding statistics are initialized (mean value and deviation of the new attribute, etc.). Two new leaf nodes are created and the (d+1) cases are stored to the corresponding leaf. When the system is continuously learning new cases, it can happen that the splitting value of a node, which should be the nearest value to the average value of the attribute, is not anymore the nearest value. This will be caused by the fact that the average value of the attributes is changing continuously. This situation could provoke that the indexing k-d tree could start to be not well balanced in all its subtrees, worsening the disturbing.

For our proposal, we will consider that a value of an attribute, \( x_{n+1} \), is a dist. value \( \Rightarrow |Av(x_n) - x_{n+1}| > \beta \cdot \text{stdev}(x_n) \). \( \beta > 0 \)
\( x_{n+1} \) is a nor. value \( \Rightarrow |Av(x_n) - x_{n+1}| \leq \beta \cdot \text{stdev}(x_n) \). \( \beta > 0 \)
That means the values with high dispersion will be those that are far from the mean value of the attribute. \( Av() \) is the mean value and \( \text{stdev()} \) is the standard deviation of the distribution of values. \( \beta \) is a modifying factor for the \( \text{stdev()} \) value. Initially, \( \beta \) is proposed to be set to 1 (\( \beta = 1 \)), but other values can be used.

Fortunately, the DACL framework can easily and incrementally compute the new average values for all the attributes, according to the following formula:

\[
Av(x_{n+1}) = \frac{n \cdot Av(x_n) + x_{n+1}}{n + 1}
\]

Where \( Av(x_n) \) is the average mean value of the attribute \( x \) according to its first \( k \) values (\( x_1, ..., x_k \)).

Also the standard deviation can be computed in an incremental way through this formula due to Welford (Welford, 1962):

\[
\text{stdev}(x_{n+1}) = \text{stdev}(x_n) + (x_{n+1} - Av(x_n)) \cdot (x_{n+1} - Av(x_{n+1}))
\]

One first strategy is that the DACL will rebuild a concrete NIAR k-d subtree, when the following condition would be met:

\( x_{n+1} \) is a disturb. value \( \Rightarrow |Av(x_n) - x_{n+1}| > \beta \cdot \text{stdev}(x_n) \). \( \beta > 0 \)
This means that subtree rebuilding would be started each time the above condition is found, when a new case is going to be learnt into the DACL. This strategy is named as IncMTree-1.

Another strategy, which is a variation of the previous one, is that the DACL system will rebuild a concrete NIAR k-d subtree, at asynchronous time conditions, when the following condition would be satisfied, since the last time the task was fired:

\[
\#\text{Disturbance values} \geq \delta \cdot N
\]
where \( \delta \) is a specified percentage. We propose as initial trial that \( \delta = 0.2 \), and \( N \) is the size of the corresponding sub-library. This criterion means that when the number of disturbance values is higher than a specified percentage (for instance the 20%) of the number of cases of the sub-library, the task of rebuilding the indexing NIAR k-d tree corresponding to the sub-library will be started. A new NIAR k-d tree will be generated with the possible new splitting values at each node of the tree. This strategy is named as Incremental Maintenance of trees triggered by a percentage of disturbance values (IncMTree-2).

F. A strategy for learning relevant cases

One of the important problems in unsupervised continuous domains is the size of the case library. As the continuous data stream is being processed incrementally, it must be used a concrete case learning strategy.

The standard strategy propose learning all the cases that has been processed and solved, with the hope to increase the competence of the case library, by increasing the coverage of the cases within the case library. This strategy will be named as All Case Learning (CaseL-1). Of course, this strategy, even though could increase the competence of the CBR system, could enlarge too much the size of the case library.

In order not to increase the size of the Case library in an unnecessarily way, a possible new strategy could be thought to learn only the most relevant cases. A relevant case would be a case that increases the coverage of the case library (i.e., the competence of the CBR system). The coverage of the case library is increased when the new case learnt could solve different cases to the cases that could be solved previously without this new case. In our DACL approach, for each corresponding Meta-case, we have a NIAR k-d tree structure.

![Fig. 3. Relevant case learning (CaseL-2)]
IV. EXPERIMENTAL WORK

The experimental work comprised two major scenarios. First, testing several databases from the UCI repository (Abalone, Balance, Car evaluation, Ecoli, Glass, Ionosphere, Iris, Pima and Waveform) in a batch learning mode using the DACL framework, but in a supervised way. That means that the number of class labels was known, and the number of case libraries of the DACL was fixed a priori. Thus, DACL was used not in a dynamical mode but in a static multi-library style. In the second scenario, the same databases, used as they were unsupervised datasets, were tested in an incremental learning mode, with the whole Dynamic Adaptive Case Library framework, (with meta-case learning strategies and case learning approaches, the NIAR k-d trees, and the PME technique). Therefore, this way, the hierarchical structures (NIAR k-d trees) were incrementally constructed, and all the cases in the databases used were processed in a step by step mode.

The experimental setting was done under the following characteristics:

- The above nine databases were tested
- Different strategies were tested. These combinations are the result of the crossing of 3 Meta-case Selection strategies (Basic learning algorithm with radius α [MCSel-1], Avoiding impasses with real Mc [MCSel-5] and Learning using maximum radius RMax with relaxation [MCSel-4]), the incremental maintenance strategy for the NIAR k-d Tree (IncMaintTree [IncMtree-1] and two strategies for the Learning of cases (AllCaseLearning [CaseL-1] and RelCaseLearning [CaseL-2]).
- For each database and for each strategy, 10 execution runs were done sampling randomly the cases to get different ordering of the cases. In addition, one more run was done with the original ordering of each database.
- For each execution and for each database several statistics were computed: the average accuracy (in percentage); the average time retrieval (time in µs), the detection of new prototypes.

A. Experimental results

In this subsection, the results obtained regarding the following parameters are detailed: detection of meta-cases, evolution of the detection of Meta-cases, average accuracy, and average time retrieval obtained with the Iris and Balance datasets.

1) Detection of Meta-cases

Regarding the discovering of Meta-cases and prototypes, all the databases have been processed and the different meta-cases (prototypes) were found to correspond accurately with the real hidden class labels.

In some of the databases, the exact number of prototypes corresponding with the same number of classes in the original database was found. For instance, it is the case of Pima and Ionosphere (2 classes and 2 prototypes found), Iris andBalance (3 classes and 3 prototypes found), Car (4 classes and 4 prototypes) and Ecoli (8 classes and 8 prototypes). In the other databases, the exact number of classes was not discovered, but this that not means that the predictive performance of the system were worse. On the contrary, on same execution runs it will be observed that the predictive power of a different number of prototypes is higher than with executions having the same number of prototypes.

2) Performance evaluation

The average precision values found are reported in the following table for Iris and Balance datasets (table I). It is important to mention that the algorithm’s learning process updates its learning base each time a new case and/or meta-case appears, which has as a result that the initial precision value is lower and grows as more meta-cases are identified and more cases are learnt.

Table I. Mean precision values for Iris and Balance databases

<table>
<thead>
<tr>
<th>Database</th>
<th>#CL</th>
<th>#MCs</th>
<th>#NCL</th>
<th>Accuracy</th>
<th>RMax</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>50</td>
<td>3</td>
<td>87%</td>
<td>0.425</td>
<td></td>
</tr>
<tr>
<td>Balance</td>
<td>32</td>
<td>3</td>
<td>68%</td>
<td>5.9</td>
<td></td>
</tr>
</tbody>
</table>

Notwithstanding, if we compare the accuracy results of the non-incremental version of our DACL approach and the incremental version, we can observe that the incremental versions only have an accuracy decrease of 5-7%. See table II.

Table II. Comparison of accuracy in non-incremental DACL versus incremental DACL approaches

<table>
<thead>
<tr>
<th>Database</th>
<th>Non-Incremental DACL</th>
<th>Incremental DACL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>91.8</td>
<td>87</td>
</tr>
<tr>
<td>Balance</td>
<td>74.9</td>
<td>68</td>
</tr>
</tbody>
</table>

Some results obtained from the tests done with the incremental strategies are detailed for the first six strategies for the Iris dataset (see table III). Other results from other datasets are not described due to lack of space.

Table III. Iris database performance with the 6 strategies

<table>
<thead>
<tr>
<th>Strategy</th>
<th>IncMTree-1 (CaseL-1)</th>
<th>CaseL-2</th>
<th>Incremental k-d Tree</th>
<th>Strategy</th>
<th>IncMTree-1 (CaseL-1)</th>
<th>CaseL-2</th>
<th>Incremental k-d Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iris</td>
<td>101.5</td>
<td>90</td>
<td>20.15</td>
<td>3</td>
<td>106.45</td>
<td>80</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>129.4</td>
<td>90</td>
<td>20.15</td>
<td>2</td>
<td>105.6</td>
<td>100</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>113.95</td>
<td>90</td>
<td>20.15</td>
<td>3</td>
<td>128.05</td>
<td>85</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>119</td>
<td>90</td>
<td>20.15</td>
<td>2</td>
<td>103.4</td>
<td>85</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>99.95</td>
<td>80</td>
<td>20.15</td>
<td>2</td>
<td>104.55</td>
<td>95</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>104</td>
<td>80</td>
<td>20.15</td>
<td>3</td>
<td>81.95</td>
<td>80</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>30.85</td>
<td>90</td>
<td>20.15</td>
<td>3</td>
<td>13.15</td>
<td>75</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>15.15</td>
<td>90</td>
<td>20.15</td>
<td>2</td>
<td>27.6</td>
<td>90</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>15.8</td>
<td>90</td>
<td>20.15</td>
<td>2</td>
<td>22.15</td>
<td>95</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>27.5</td>
<td>85</td>
<td>20.15</td>
<td>2</td>
<td>23.85</td>
<td>85</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>29.5</td>
<td>85</td>
<td>20.15</td>
<td>2</td>
<td>23.85</td>
<td>85</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>28.1</td>
<td>90</td>
<td>20.15</td>
<td>3</td>
<td>29.2</td>
<td>100</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>19.65</td>
<td>45</td>
<td>20.15</td>
<td>8</td>
<td>5.05</td>
<td>65</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>43.65</td>
<td>50</td>
<td>20.15</td>
<td>8</td>
<td>7.7</td>
<td>75</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>58.55</td>
<td>65</td>
<td>20.15</td>
<td>8</td>
<td>7.75</td>
<td>70</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>105.25</td>
<td>90</td>
<td>20.15</td>
<td>6</td>
<td>7.95</td>
<td>70</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>36.45</td>
<td>75</td>
<td>20.15</td>
<td>13</td>
<td>16.55</td>
<td>75</td>
<td>10.6</td>
</tr>
<tr>
<td></td>
<td>24.15</td>
<td>80</td>
<td>20.15</td>
<td>4</td>
<td>21.3</td>
<td>70</td>
<td>10.6</td>
</tr>
</tbody>
</table>

The results in the nine databases tested showed a good time performance and accuracy average on several execution runs, even though the case library was incrementally populated. They achieved good accuracy values, just a little bit worse than...
These experiments provided the confirmation that the proposed DACL framework is especially suitable to cope with incremental processing and learning of cases, can be considered as an incremental clustering technique. Notwithstanding, in the scenarios when the discovered number of prototypes is different from the number of existing ones, the accuracy of the CBR system is very good, and even better than when discovering the same number of prototypes. Moreover, the average final precision is just a little bit worse than the average precision obtained with the same DACL approach, but just using the usual non-incremental way. In such non-incremental scenarios, the Case Library is seeded up with some training set of cases, and tested with a testing set, and of course, it should have better accuracy results, because at the beginning, this configuration has many more cases in the Case Library than the corresponding incremental configuration. These experiments provided the confirmation that the proposed DACL framework is especially suitable to cope with incremental data streams. As showed later the DACL approach has been able to detect and construct the same number of meta-cases (prototypes) than the existing ones in the databases.

V. CONCLUSIONS

It is not easy to cope with data streams, where in an incremental way, a lot of cases are being generated, and must be processed by the CBR system. This way, the system has not so much cases in its Case Library like in a non-incremental processing scenario. This means that it is pretty more difficult to achieve good accuracy percentages, because at the beginning of the processing, normally the precision will be lower than when working in a non-incremental scenario. The experimentation done has outlined that the DACL approach is able to satisfactorily cope with unsupervised and incremental scenarios. The entire Dynamic Adaptive case Library framework has been tested with simulated unsupervised domains where the cases were incrementally processed, and all the DACL structures were incrementally created and updated, with good results relating to the automatic discovery of prototypes (meta-cases), which correspond to the actual existing prototypes. In addition, good retrieval time measures and accuracy measures were obtained. Finally, and mainly, the proposed DACL framework is able to cope, with good performance, with the incremental processing of data streams, than most of the techniques used in CBR cannot.

REFERENCES

An Exploratory Evaluation of Bayesian Principled Approaches to Solve Imbalanced Problems


Abstract—Bayesian decision theory provides a principled framework to deal with imbalanced problems. Its application merely requires to use estimates of the likelihood ratios, that can be obtained from the “a posteriori” probability estimates that some Bregman-type training cost functions offer at the machine outputs. Following this way, it is possible to establish “associated” decision problems that are easy—without imbalance difficulties—to solve, and, from them, to build a solution for the imbalanced original problem. Some preliminary experiments demonstrate the potential of this approach.

Keywords—Decision costs, imbalance, likelihood ratios.

I. INTRODUCTION

MAKING a decision is selecting one of a number \( M \) of alternatives \( \{ D_j \}, 0 \leq j \leq J - 1 \), according to some information. Its statistical form assumes that a sample of a random variable \( x \) is available, as well as the probability density functions \( p(x|S_i) \), \( 0 \leq i \leq I - 1 \), being nature states whose (a priori) probabilities \( \{ P_i \} \) are also given. A matrix of costs \( \{ C_{ij} \} \) indicating how much to pay for selecting \( D_j \) when \( S_i \) is true completes the description of decision problems. These include hypothesis testing and classification, that correspond to \( \{ S_i \} \) representing possible hypotheses or classes: deciding \( D_j \) means to accept hypothesis \( S_i \) or to include instance \( x \) in class \( S_i \).

In most practical situations, the above statistical description is unknown, and a decision machine, i.e., a function \( F(x) \) which takes discrete values \( j \) has to be designed from the available information, which can be a number of labeled examples, \( \{ x^{(n)}, t^{(n)} \}, n = 1, ..., N \), where \( t^{(n)} \) is the scene indicator for \( x^{(n)} \). There are different ways to construct \( F(x) \): Discriminative machines are those that apply a parameterized family of functions, \( F_w(x) \), whose parameters \( w \) are obtained from the labeled examples and a training cost criterion \( C[F_w(x), t] \), which is somewhat minimized for those examples but keeping an appropriate generalization capability, i.e., good decision results for unseen samples.

We will restrict the discussion from now on to standard binary problems, i.e., \( S_0 \) and \( S_1 \) are the states and \( D_0 \) and \( D_1 \) the decisions; concepts are conceptually easy. We will also speak in terms of classification and use \( t^{(n)}_i = -1 \) for practical reasons.

The design of these machines becomes really difficult for problems that can be qualified as singular. Population imbalanced problems are a first singular sub-family. They appear when one class is much more probable than the other; for example, \( P_0 >> P_1 \). Any direct attempt of designing a machine tends to significantly reduce the appearance of \( t = 1 \) classifications. The same occurs when the cost ratio

\[
Q_C = \frac{C_{10} - C_{00}}{C_{01} - C_{11}} (> 0)
\]

is very high\(^1\), because to make an error saying that a class 0 sample is class 1 is much more expensive than the opposite. These are the cost imbalanced problems. Finally, if the basic costs depend on the samples, \( \{ C_{ij}(x) \} \) —these are the sample-dependent cost problems—, similar difficulties can appear. We will not address these last problems in this contribution.

Of course, many procedures have been proposed to overcome these difficulties. \([1][2]\) include reviews of the basic techniques that inspire them. Such techniques are mainly of two types. First, to modify the relative cost of making an error —i.e., \( Q_C \) in the design algorithms to compensate the difficulty of getting minority results. Second, to modify the training example populations with the same purpose: Undersampling the majority class or generating new samples for the minority class. This is the case of algorithms that justly deserve much attention, such as SMOTE (Synthetic Minority Oversampling Technique) \([3]\) and its variants MSMOTE (Modified SMOTE) \([4]\), SPIDER (Selective Preprocessing of Imbalanced Data) \([5]\), and ADASYN (ADaptive SYNthetizing) \([6]\), as well as undersampling and generation combinations.

In this paper, we present a principled perspective on population and cost imbalanced problems which allows to design classification machines or machine ensembles in a systematic and complete manner. We start from two points. First, Bregman divergences \([7]\) permit to build machines that estimate the “a posteriori” class probabilities. Second, class likelihoods \( p(x|S_i) \), and, consequently, its ratio

\[
q_L(x) = \frac{p(x|S_1)}{p(x|S_0)}
\]

do not depend on “a priori” probabilities and cost values.

Since there is a one-to-one correspondence between the “a posteriori” probabilities and the likelihood ratio, it is possible to define “artificial” associated problems that provide good estimates of the likelihood ratio that are valid for the problem under analysis. These estimates serve to obtain good solutions

\(^1\) We select again \( S_0 \) as the dominant class.
for the real problem, including estimates of the “a posteriori” class probabilities.

The rest of the paper is as follows. Section II reviews the Bayes decision theory and the estimation of “a posteriori” class probabilities, extracts consequences for solving classification problems, and, finally, discusses the way of using the previous results to design machines or ensembles for imbalanced problems. Some preliminary experimental results are presented in Section III, comparing them with those provided by SMOTE. The main conclusions of our work and suggestions for future research close the contribution.

II. BAYESIAN CLASSIFICATION AND ESTIMATION OF “A POSTERIORI” PROBABILITIES

A. Bayes decision theory

It is immediate to conclude that minimizing the average cost for a standard binary problem under the statistical situation which is described in the first paragraph of the Introduction leads to the rule

\[ q_L(x) = \frac{p(x|S_1) P_0}{p(x|S_0)} = \frac{1}{1 - \frac{Q_C P_0}{P_1}} = Q_C P = Q \]

(3)

where \( Q_P \) and \( Q \) have obvious expressions. Equivalently, since \( p(x|S_i) = p(S_i|x)p(x) \),

\[ \frac{Pr(t = 1|x)}{Pr(t = -1|x)} = \frac{Pr(t = 1|x)}{1 - Pr(t = 1|x)} \approx \frac{1}{1 + \frac{Q_P}{Q_L(x)}} \]

(4)

We remark that

\[ Pr(t = 1|x) = \frac{p(x|S_1) P_1}{p(x)} = \frac{p(x|S_1) P_1}{p(x|S_1) P_1 + p(x|S_0) P_0} \]

\[ = \frac{1}{1 + \frac{Q_P}{Q_L(x)}} \]

(5)

in other form,

\[ q_L(x) = \frac{Q_P Pr(t = 1|x)}{1 - Pr(t = 1|x)} \]

(6)

B. Estimation of the “a posteriori” probabilities

Bregman divergences [7] make possible to estimate “a posteriori” probabilities, a topic which is further discussed in [8][9]. In our case, it is enough to see that a cost \( C(o,t) \) to estimate \( t(x) \) by means of \( o(x) \) having a form such as

\[ \frac{\partial C}{\partial o} = -g(o)(t - o) \]

(7)

\( g(o) > 0 \) leads to

\[ o(x) = E\{t|x\} = Pr(t = 1|x) - Pr(t = -1|x) \]

\[ = 2Pr(t = 1|x) - 1 \]

(8)

C. Consequences for classification machines

When working with a machine, \( E\{t|x\} \) is replaced by an estimate which depends on the expressive capability of the machine (i.e., its capacity to establish arbitrary input-output correspondences) and the quality of the training set for representing the statistics of the problem. Therefore, we have

\[ \hat{Pr}(t = 1|x) = \frac{o(x) + 1}{2} \]

(9)

Needless to say, the above estimate will be good if we use a powerful enough machine and a representative enough training set.

Obviously, this is not the case of population imbalanced problems. But let us assume that we consider an artificial problem in which the only difference is the imbalance ratio, \( IR = P_0/P_1 \), which is moderate enough to get a good estimate \( \hat{Pr}(t = 1|x) \) (from the trained machine output). It does not correspond to the real problem; but, if we take care of keeping the essential forms of the likelihoods,

\[ \hat{q}_L(x) = \hat{q}_L(x) = \frac{Q_P \hat{Pr}(t = 1|x)}{1 - \hat{Pr}(t = 1|x)} \]

(10)

\( \hat{Q}_P \) being the imbalance ratio of the associated artificial problem.

A good \( \hat{q}_L(x) \) can be applied to solve the real problem according to (3), and even to get estimates of the “a posteriori” probability of \( t = 1 \) for the real problem by means of (5).

It is clear that direct undersampling and generation without paying attention to the difficulty of examples or their proximity to the classification frontier, tend to maintain the form of the likelihoods, but introducing some changes. But this is the point at which the concept of diversity is helpful: for instance, if we undersample at random a number of times, an appropriate average of the results will serve to compensate these changes. The same can be said for generation. And there are several reasonable forms of averaging:

- A direct average of \( \{\hat{q}_{Lk}(x)\} \), \( k = 1,...,K \), obtained with \( K \) re-balancing associated machines;
- A direct average of \( \{\hat{Pr}_k(t = 1|x)\} \), \( k = 1,...,K \), obtained from \( \{\hat{q}_{Lk}(x)\} \) as explained above;
- Applying a majority vote to decisions of the from \( sgn(\hat{q}_{Lk}(x) - \hat{Q}) \);
- Idem to \( sgn(\hat{Pr}_k(t = 1|x) - 1/2) \)

D. Cost imbalance

In the above analysis, we have not addressed the role of \( Q_C \) for creating associated problems. There is not any difficulty of extending the discussion.

When training a machine for solving a problem with cost ratio \( Q_C \), the sample cost function

\[ C'[\{t(n),x(n)\}] = \sum_{n_1} C[1,o(x(n))] + Q_C \sum_{n_0} C[-1,o(x(n))] \]

(11)

where \( n_1(n_0) \) are class 1 (0) training instances, is minimized. Obviously, although \( C \) has the form (7), the resulting \( o(x) \) is
That means that for the training set, 286 majority class samples and 11 minority class samples. For training and 297 for test. The training set has 1147 majority class. Clearly, the best combination will be problem dependent.

We remark that minimizing (11) is (qualitatively) equivalent to solve a problem with the same likelihood ratio and unity cost ratio, but with an imbalance ratio \( P_0' / P_0' = Q_C(P_1 / P_0) \) (as (3) suggests). Therefore, it is evident that increasing the cost ratio can be also a way to re-balance a population imbalanced classification problem, and also that one can proceed in the opposite sense, using undersampling and/or generation to fight against cost sensivity difficulties.

In conclusion: Both types of imbalance problems (or combined forms) can be attacked by means of creating associated problems by cost ratio modifications, undersampling the majority class, but if we undersample at random many times, and then building easier associated problems would produce the risk of reducing the quality of the likelihood ratio estimates, as a consequence of changing too much the likelihoods. So, to oppose sense, using undersampling and/or generation to fight against cost sensivity difficulties.

### III. Exploratory Experiments

In this section we present three experiments whose aim is to illustrate the effects of applying the formulation exposed above, compared to SMOTE.

#### A. Data set

The first experiments involving this formulation were performed over the Yeast4 data set, from [10]. It derives from a bigger data set, the Saccharomyces Genome Database (SGD), containing microarray expression data and phylogenetic profiles from the budding yeast Saccharomyces cerevisiae. It is a multiclass problem with 24177 samples, 103 features, and many different classes usually grouped into 14. Yeast4 is a binarized version (one class against the rest) that contains 1484 samples with \( D = 8 \) features. From those samples, 1187 have been used for training and 297 for test. The training set has 1147 majority class samples and 40 minority class samples. The test set has 286 majority class samples and 11 minority class samples. That means that for the training set, \( IR = Q_P = 28.41 \).

#### B. Basic learning machine

A MultiLayer Perceptron (MLP) was the employed classifier for building the ensembles. In these preliminar experiments it was designed with one hidden layer with \( H = 55 \) neurons, a size which results appropriate for the problem under analysis.

The weights are initialized with Gaussian priors with zero mean and variance provided by

\[
\sigma_1^2 = \frac{1}{\sqrt{(D+1)H}} \quad \sigma_2^2 = \frac{1}{\sqrt{1+H}}
\]

for the first and second layer weights, respectively.

The MLPs were trained by BP. The number of epochs was set to 150 after studying the convergence of the classifier.

#### C. Metrics

Every time that an imbalanced problem is faced, the proper metrics to measure how the machine learns must assign importance and/or penalties to the hits/mistakes related to the relevant class. Keeping this in mind, a test accuracy, for instance, is not adequate, and we will prefer other ones related to the probability of detection. Concretely, we have chosen the F1 Score, which jointly penalizes the inability to detect and the excess amount of false alarms. It is obtained as a harmonic mean of Precision (Pre) and Recall (Re)

\[
P_{\text{re}} = \frac{TP}{TP + FP} \quad \text{Re} = \frac{TP}{TP + FN}
\]

\[
F_1 = 2 \cdot \frac{P_{\text{re}} \cdot \text{Re}}{P_{\text{re}} + \text{Re}}
\]

where \( TP \) means True Positives and \( FP \) (FN) means False Positives (Negatives).

#### D. Experiments on the data set

The first experiment consists of undersampling the majority class. This might lead to lose relevant information about this class, but if we undersample at random many times, and then we build ensembles, we may recover useful information for the original classification problem, through estimates of \( q_L(x) \) and \( P_r(t = 1|\mathbf{x}) \), applying (6) and (5), respectively. Inserting (9) into (6)

\[
q_L(\mathbf{x}) = \frac{Q_PPr(t = 1|\mathbf{x})}{1 - Pr(t = 1|\mathbf{x})} = \frac{Q_P(1 + o(\mathbf{x}))}{1 - o(\mathbf{x})}
\]

Then, an average of these magnitudes is considered just as in Section II-C. It should be kept in mind that we obtain an estimate of \( q_L(x) \) from an artificial problem, represented by \( Q_P \), but the decision is made under the conditions of the original problem, using \( Q_P \) as the threshold.

The second experiment adds a different cost policy \( \tilde{Q}_C \neq 1 \). Concretely, a good value for this problem was found to be
\( \hat{Q}_C = 1/3 \) during the training process. (6) and (5) can be rewritten, considering \( Q = Q_C Q_P \), as

\[
q_t(x) = \frac{Q(1+o(x))}{1-o(x)} \tag{16a}
\]

\[
Pr(t=1|x) = \frac{1}{1+Q \frac{Q}{\hat{Q}}(x)} \tag{16b}
\]

and using \( \hat{Q} \) and \( Q \) in the same conditions as in the previous experiment.

Results from the previous experiments will be compared to those of SMOTE, which is a generation algorithm which considers the \( K \) nearest neighbours of a minority class sample, and picking one of them at random, it generates a new sample allocated at a random point of the line that joins these two samples. This process can be repeated several times to reach a desired IR value. In two additional experiments were carried out with SMOTE \( K = 3 \); using \( Q_C = 1 \) in the first one, and \( Q_C = 1/3 \) in the second one.

In all the experiments different \( Q_P \) were tested by means of undersampling and oversampling, from the original \( Q_P \) to 1, with the intermediate values 20, 15, 12, 9, 6 and 3. Ensembles were built to take advantage of machines’ diversity, using up to 151 machines: (1), 21, 51, 101 and 151. Results are the average of 30 runs of each experiment.

E. Results

Now, subsampling, cost re-balance and generation are used to train ensembles, and then \( F_1 \) values obtained from a direct average of outputs, of likelihoods of the real problem, and of posteriori coming from that likelihood estimator, are shown.

The starting point, that is, without applying any of these techniques to the data, a direct solution for the imbalanced problem with the proposed MLP gives \( F_1 = 0.21 \).

1) Subsampling with \( \hat{Q}_C = 1 \): Table I shows the \( F_1 \) obtained in the first undersampling experiment, deciding using a direct average of \( o_t(x) \), the output of each machine for the “artificial” problem.

Table II and Table III are the \( F_1 \) values obtained considering the averaged estimates of \( q_t(x) \) and \( Pr(t=1|x) \), respectively, obtained for the original problem following the proposed formulas in (16).

<table>
<thead>
<tr>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.370 ±0.025</td>
<td>0.422 ±0.016</td>
<td>0.425 ±0.015</td>
<td>0.401 ±0.022</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.368 ±0.006</td>
<td>0.444 ±0.014</td>
<td>0.445 ±0.012</td>
<td>0.394 ±0.009</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.372 ±0.008</td>
<td>0.441 ±0.009</td>
<td>0.448 ±0.008</td>
<td>0.390 ±0.009</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.375 ±0.005</td>
<td>0.441 ±0.007</td>
<td>0.450 ±0.014</td>
<td>0.391 ±0.005</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.376 ±0.005</td>
<td>0.443 ±0.005</td>
<td>0.450 ±0.004</td>
<td>0.390 ±0.003</td>
</tr>
</tbody>
</table>

In this specific case, the likelihood ratio estimate was not properly learnt from the alternative problem. Lower results mean that Precision or Recall, one of them, is almost zero.

\( F_1 = 0.44 \) with 21 machines and IR=9 is the optimum using the outputs, and \( F_1 = 0.3 \) with 21 machines and IR=12 is the optimum using the likelihood estimate.

2) Subsampling with \( \hat{Q}_C = 1/3 \): The following tables show information as in the previous experiment, but now each machine is trained with \( Q_C = 1/3 \).

<table>
<thead>
<tr>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.291 ±0.008</td>
<td>0.297 ±0.013</td>
<td>0.294 ±0.027</td>
<td>0.284 ±0.023</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.293 ±0.002</td>
<td>0.299 ±0.004</td>
<td>0.297 ±0.006</td>
<td>0.281 ±0.002</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.291 ±0.007</td>
<td>0.295 ±0.010</td>
<td>0.297 ±0.013</td>
<td>0.278 ±0.009</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.294 ±0.011</td>
<td>0.299 ±0.018</td>
<td>0.298 ±0.011</td>
<td>0.274 ±0.003</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.294 ±0.016</td>
<td>0.299 ±0.022</td>
<td>0.298 ±0.016</td>
<td>0.274 ±0.003</td>
</tr>
</tbody>
</table>

TABLE I. \( F_1 \) WITH UNSUBSAMPLING, AVERAGE OF \( o_t(x) \)

<table>
<thead>
<tr>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
<th>( Q_{\text{IR}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.402 ±0.008</td>
<td>0.422 ±0.009</td>
<td>0.462 ±0.027</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.396 ±0.007</td>
<td>0.425 ±0.004</td>
<td>0.481 ±0.014</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.394 ±0.005</td>
<td>0.423 ±0.005</td>
<td>0.477 ±0.011</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.390 ±0.005</td>
<td>0.423 ±0.005</td>
<td>0.477 ±0.011</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.390 ±0.005</td>
<td>0.423 ±0.005</td>
<td>0.477 ±0.011</td>
</tr>
</tbody>
</table>

TABLE II. \( F_1 \) WITH UNSUBSAMPLING, AVERAGE OF \( \hat{Q}_{\text{IR}}(x) \)

TABLE III. \( F_1 \) WITH UNSUBSAMPLING, AVERAGE OF \( \hat{Pr}_k(t=1|x) \)

TABLE IV. \( F_1 \) WITH UNSUBSAMPLING, AVERAGE OF \( o_t(x) \)

TABLE V. \( F_1 \) WITH UNSUBSAMPLING, AVERAGE OF \( \hat{q}_t(x) \)
When adding a cost re-balance for this problem the likelihood estimates improve. \( F_1 = 0.48 \) with 21 machines and IR=20 is the maximum using outputs, and \( F_1 = 0.47 \) with 151 machines and IR=20 with the likelihood estimate, applying the averaging mechanism.

3) SMOTE with \( Q_C = 1 \): Now, instead of discarding majority-class samples, we will generate samples using SMOTE until a given \( Q_P \) is reached, considering \( K = 3 \) as the number of neighbours when generating samples. The following tables also present \( F_1 \) values considering the direct output of machines \( o(x) \), the likelihood estimate \( q_L(x) \), and the posterior estimate \( Pr(t=1|x) \), in the same way as before.

**TABLE VI.** \( F_1 \) with subsampling, average of \( Pr_k(t=1|x) \)

<table>
<thead>
<tr>
<th>IR 12</th>
<th>IR 15</th>
<th>IR 20</th>
<th>IR 28.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.472 ± 0.002</td>
<td>0.468 ± 0.002</td>
<td>0.457 ± 0.001</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.452 ± 0.002</td>
<td>0.457 ± 0.002</td>
<td>0.467 ± 0.002</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.432 ± 0.021</td>
<td>0.437 ± 0.021</td>
<td>0.459 ± 0.003</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.454 ± 0.015</td>
<td>0.459 ± 0.015</td>
<td>0.467 ± 0.002</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.452 ± 0.016</td>
<td>0.467 ± 0.002</td>
<td>0.486 ± 0.000</td>
</tr>
</tbody>
</table>

\( F_1 = 0.38 \) is the maximum with 21 machines and IR=1 using outputs, and \( F_1 = 0.25 \) with 51 machines and IR=20 using the likelihood estimate.

4) SMOTE with \( Q_C = 1/3 \): Following the same order, results are:

**TABLE X.** \( F_1 \) with SMOTE, average of \( q_L(x) \)

<table>
<thead>
<tr>
<th>IR 9</th>
<th>IR 12</th>
<th>IR 15</th>
<th>IR 20</th>
<th>IR 28.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.484 ± 0.001</td>
<td>0.491 ± 0.002</td>
<td>0.480 ± 0.003</td>
<td>0.484 ± 0.003</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.495 ± 0.001</td>
<td>0.497 ± 0.002</td>
<td>0.499 ± 0.003</td>
<td>0.494 ± 0.003</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.495 ± 0.001</td>
<td>0.497 ± 0.002</td>
<td>0.501 ± 0.003</td>
<td>0.488 ± 0.003</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.496 ± 0.002</td>
<td>0.498 ± 0.003</td>
<td>0.500 ± 0.004</td>
<td>0.487 ± 0.004</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.495 ± 0.006</td>
<td>0.502 ± 0.005</td>
<td>0.491 ± 0.006</td>
<td>0.486 ± 0.006</td>
</tr>
</tbody>
</table>

**TABLE XI.** \( F_1 \) with SMOTE, average of \( \hat{q}_{Lk}(x) \)

<table>
<thead>
<tr>
<th>IR 12</th>
<th>IR 15</th>
<th>IR 20</th>
<th>IR 28.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.348 ± 0.001</td>
<td>0.377 ± 0.002</td>
<td>0.407 ± 0.003</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.352 ± 0.001</td>
<td>0.377 ± 0.002</td>
<td>0.403 ± 0.003</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.361 ± 0.001</td>
<td>0.378 ± 0.002</td>
<td>0.396 ± 0.003</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.361 ± 0.004</td>
<td>0.378 ± 0.005</td>
<td>0.397 ± 0.006</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.361 ± 0.004</td>
<td>0.379 ± 0.005</td>
<td>0.397 ± 0.006</td>
</tr>
</tbody>
</table>

**TABLE XII.** \( F_1 \) with SMOTE, average of \( \hat{Pr}_k(t=1|x) \)

<table>
<thead>
<tr>
<th>IR 12</th>
<th>IR 15</th>
<th>IR 20</th>
<th>IR 28.4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 mach.</td>
<td>0.260 ± 0.001</td>
<td>0.274 ± 0.001</td>
<td>0.298 ± 0.002</td>
</tr>
<tr>
<td>21 mach.</td>
<td>0.285 ± 0.001</td>
<td>0.264 ± 0.001</td>
<td>0.286 ± 0.002</td>
</tr>
<tr>
<td>51 mach.</td>
<td>0.262 ± 0.001</td>
<td>0.262 ± 0.002</td>
<td>0.290 ± 0.003</td>
</tr>
<tr>
<td>101 mach.</td>
<td>0.262 ± 0.002</td>
<td>0.259 ± 0.001</td>
<td>0.284 ± 0.001</td>
</tr>
<tr>
<td>151 mach.</td>
<td>0.262 ± 0.002</td>
<td>0.259 ± 0.001</td>
<td>0.285 ± 0.001</td>
</tr>
</tbody>
</table>

Likelihood and posterior estimates improve with respect to the previous experiment, but there is still more improvement to reach the same performance than the one from the output average.

Using outputs, the maximum is \( F_1 = 0.50 \) with 51 machines and IR=15, and using the likelihood estimate, it is \( F_1 = 0.44 \) with 1 machine and IR the original IR.

The above results must be carefully considered. It could seem that \( Q_C = 1/3, 51 \) machines, “classical” (i.e., averaging \( o_L(x) \) SMOTE solution is the best and that the proposed approach does not provide anything more that slightly worse results, probably because we are working with a small dataset. But if the original problem does not correspond to equal costs and/or the assumed IR, the only possibility for a “classical” solution is to empirically modify the aggregated output threshold to adapt to the new conditions, while the solutions based on \( \hat{q}_L \) do not need anything more that to introduce the new value of \( Q \) in (3).
Let us consider the following scenario, in which the test samples are known to have a different \( Q_P = Q_P / 2 \). This can be tested by randomly undersampling the majority class in the test set. We want to compare the performance of the SMOTE classifiers with \( Q_C = 1 / 3 \); those using the average of outputs with those that average likelihoods.

In our formulation, we only have to change \( Q \) properly: \( Q' = 2Q \). For the direct outputs average machine, we may think of adapt the output threshold to this change, in a way that may even not be working in every case. One way to modify this output threshold derives from (16):

\[
o(x) \approx \frac{1}{Q} \frac{Q - \hat{Q}}{Q + \hat{Q}}
\]

(17)

In the previous experiment, for the outputs averaged, we used \( Q = Q \), so 0 was the output threshold (corresponding to an artificial problem). But, if we set update thresholds for the likelihood estimate and for the outputs (corresponding to \(-1/3\) in the outputs’ domain), now the likelihood approach outperforms the SMOTE with direct average of the outputs: The likelihood estimates lead to \( F_1 = 0.58 \) with 101 machines and \( IR=15 \), and the outputs average gives a maximum \( F_1 = 0.55 \) with 101 machines and IR the original one.

IV. CONCLUSIONS AND FUTURE WORK

In this contribution, we have presented a Bayesian framework which, combined with the use of appropriate training costs --those providing estimates of the “a posteriori” probabilities-- permits to define “equivalent” problems for imbalanced situations, and to transform their results into results for the original problem. This allows to combine undersampling, generation, and cost re-balancing in controlled forms, as well as to directly extract the appropriate designs if there are changes in the original problem. Preliminary experiments show that elementary versions of this approach give promising results, although more work is needed to get their whole potential and benefits from their formal advantages.

APPENDIX A

The estimation requires to minimize the average cost

\[
\mathcal{C}(o) = \int_{\mathbf{x}} \int_{t} C(o, t)p(t|\mathbf{x})dt d\mathbf{x} = \int_{\mathbf{x}} \left[ \int_{t} C(o, t)p(t|\mathbf{x})dt \right] p(\mathbf{x})d\mathbf{x}
\]

(A.1)

and it is enough to minimize the inner integral

\[
\frac{\partial}{\partial o} \int p(t|\mathbf{x})dt = -g(o) \int (t - o)p(t|\mathbf{x})dt = 0\]

(A.2)

from which

\[
o = \int tp(t|\mathbf{x})dt = E\{t|\mathbf{x}\}
\]

(A.3)

g(o) > 0 means that (A.3) corresponds to a minimum.

REFERENCES


Machine learning decomposition models for partial ordering problems: An application to melanoma severity classification

Javier Sánchez-Monedero*, María Pérez-Ortiz†, Aurora Sáez‡, Pedro Antonio Gutiérrez‡ and César Hervás-Martínez‡
*Dept. of Quantitative Methods, Universidad Loyola Andalucía, 14004 Córdoba, Spain.
Email: {jsanchez,mariaperez}@uloyola.es
†Dept. of Signal Theory and Communications, University of Seville, 41092 Seville, Spain.
Email: aurorasaez@us.es
‡Dept. of Computer Science and Numerical Analysis, University of Córdoba, 14071 Córdoba, Spain.
Email: {pagutierrez,chervas}@uco.es

Abstract—Melanoma is a type of cancer that develops from the pigment-containing cells known as melanocytes. Usually occurring on the skin, early detection and diagnosis is strongly related to survival rates. In the present work, we propose a system combining image analysis and machine learning for detecting melanoma presence and severity. The severity is assessed in terms of melanoma thickness, which is measured by the Breslow index. We extract 100 features considering the shape, colour, pigment network and texture of the benign and malignant lesions, tackling this problem as a five-class classification problem, where the first class represents benign lesions and the remaining four classes represent different stages of the melanoma (as measured by the Breslow index). From a machine learning point of view, this problem is a partially ordered classification task. Because of this, we propose specific machine learning models to exploit the partial order information. In this sense, we experimentally demonstrate that specifically designed models achieve better performance than a set of nominal and ordinal classifiers, considering both the imbalanced nature of the problem and the magnitude of the ordinal error in the prediction.

Index Terms—melanoma, computer vision, machine learning, ordinal classification, imbalanced classification, partial ordering

I. INTRODUCTION

Melanoma is a type of cancer that arises from the pigment-containing cells known as melanocytes, and the most common type, the cutaneous melanoma, occurs on the skin. In Europe, approximately 100,000 cases are yearly diagnosed with a death ratio around 13% [1]. The patient prognosis directly depends on the tumour thickness, thus early detection and diagnosis can reduce mortality to a great extent [2].

The severity of the melanoma directly depends on its thickness. To improve survival rates, melanoma must be detected before the tumour has penetrated the epidermis (i.e. before the thickness is higher than 0.76mm). In the case of early detection, the five-year survival rate is about 99%, otherwise dropping to 15% for patients with advanced disease [3]. The current detection process consist in a visual inspection by trained professionals using a dermatoscope, and the prognosis is evaluated by measuring the depth of melanoma by a biopsy. However, recent works propose new tools to aid or to improve the process [3], mainly based on dermoscopic images analysis. Although there are different lines of undergoing research (e.g. those based on skin temperature variations in the lesion among others), image analysis methods present the advantage of being cheaper and relatively easy to be combined with existing detection procedures.

In the last years, computerised dermoscopy image analysis systems have been proposed to assist pigmented lesions diagnosis [4]. The majority of these works focus on the distinction of melanomas from benign lesions [5], [6]. However, a finer grain classification is required for appropriate prognosis. The scarcity of studies on this topic and its inherent difficulty makes it a promising line for research. The first work in that line is the characterisation of two types of melanoma based on their thickness [7]. It uses 49 features related to colour, geometry and texture, extracted from a private database of 141 images obtained with the company proprietary hardware system. Moreover, a recent study [8] focuses on the classification of three degrees of thickness for melanomas, but it excludes their distinction from benign lesions.

In this paper, we propose to simultaneously address the problem of melanoma detection and thickness estimation within a five-class classification problem. To do so, we combine image analysis and machine learning. The image analysis component considers a set of 100 input features, comprising characteristics focused on the distinction of the five classes. On the one hand, we have a class for benign lesions, and, on the other, we have four classes with a clear order relationship between them (the stages of the melanoma according to its depth, see Table I). Features for distinguishing melanoma from non-melanoma cases are based on the ABCD method, commonly used by dermatologists. This method analyses four clinical characteristics that describe a malignant melanoma: asymmetry (A), border irregularity (B), colour variegation (C) and differential structures (D). The rest of the features are extracted to estimate the melanoma thickness, and they are
based on clinical findings, relating certain visual characteristics present in dermoscopic images to tumour depth [9], [8].

For the different melanoma stages, it can be assumed that the labels are imbued with order information, which justifies the use of ordinal classifiers [10]. Ordinal methods exploit this ordered nature of the classes to build better classifiers and impose different misclassification errors penalizing the magnitude of the deviations (misclassifying a stage 0 melanoma with a stage I should not be considered the same as confusing it with a stage III melanoma). Generally, this order is also found in the input space and can be used to obtain more robust classifiers. However, for the benign lesion class of the problem considered, we can not assume an order relation in the input space with respect to the melanoma lesions. In this way, the problem can be considered to be a partially ordered classification task, and we propose several machine learning strategies to exploit this partial ordering. Fig. 1 illustrates the concept of partial ordering in a two dimensional dataset, where it can be seen that $C_1$ does not follow an order with respect to the rest of classes, while the rest are ordered in the input space ($C_2$ is closer to $C_3$ than to $C_4$ and so on).

Moreover, the dataset is characterised by being an imbalanced dataset with skewed class distribution (see Table I). In this sense, we present two approaches based on binary decompositions to deal with the partial order of the labels and the imbalance nature of the data. The purpose is to improve the classification performance of other state-of-the-art classifiers (both nominal and ordinal). Both proposed methods are based on a cascade binary utility ordinal model [11], which has been shown to obtain good results for problems with these characteristics (partial ordering and imbalance).

The rest of the paper is organised as follows. Section II-A presents the clinical problem and some characteristics of the dataset. Section II-B briefly presents the set of features selected to describe the images. Section III presents some previous notions and describes the two proposed binary decomposition methods. Section IV shows the experiments performed and analyses the results. Finally, Section V outlines some conclusions and future work.

II. DATA DESCRIPTION

A. Breslow Index

Tumour depth is directly correlated to survival rate. The reason is that thick tumours can access lymph capillaries, which is the most common way of cancer spreading. If the melanoma is confined to the epidermis it is referred to as 'in situ' melanoma, and it is removable by surgery. However, as the cancerous cells propagate to the deepest layer of the skin (the dermis), the melanoma is known as invasive, and the survival rate decreases with the depth of the invasion.

The Breslow index [12] is a method for prognosis of patient survival [9], that measures melanoma depth by means of pathological examination after an incisional or excisional biopsy of the suspected lesion [9]. It consists in a vertical measurement in millimetres from the top of the granular layer of the epidermis to its deepest part within the dermis. Different stages of melanoma based on the Breslow index are shown in Table I.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Location</th>
<th>Class</th>
<th>Number of cases</th>
</tr>
</thead>
<tbody>
<tr>
<td>Non-melanoma</td>
<td>-</td>
<td>1</td>
<td>313</td>
</tr>
<tr>
<td>Stage 0</td>
<td>In situ</td>
<td>2</td>
<td>64</td>
</tr>
<tr>
<td>Stage I</td>
<td>&lt;0.76 mm</td>
<td>3</td>
<td>102</td>
</tr>
<tr>
<td>Stage II</td>
<td>0.76 mm - 1.50 mm</td>
<td>4</td>
<td>54</td>
</tr>
<tr>
<td>Stage III</td>
<td>&gt;1.50 mm</td>
<td>5</td>
<td>29</td>
</tr>
</tbody>
</table>

In this paper, we use 562 images from the Interactive Atlas of Dermoscopy [13], a multimedia project for medical education with pigmented skin lesions images in which all lesions were biopsied and diagnosed histopathologically. As introduced, the images have been classified into five classes: non-melanoma (i.e. benign lesions) and the four stages of melanoma depth included in Table I. The imbalanced nature of the problem can be also be appreciated in this Table. All the images have a resolution of $768 \times 512$ pixels and have been segmented using the automatic segmentation algorithm proposed in [14], in which an edge based level-set technique is applied together with a perceptually adapted colour gradient [15]. Fig. 2 presents two examples of segmented melanomas.

B. Feature extraction

The feature extraction process proposed in this paper aims to mimic dermatologist assessment, using characteristics defined in the clinical ABCD rule (to distinguish between benign lesions and melanomas) and features inspired by the findings derived from clinical studies regarding the correlation between certain characteristics seen in dermoscopic images and melanoma thickness. A total of 100 descriptors $(x_1, x_{100})$.
based on shape, colour and texture have been extracted. Regarding the ABCD method, asymmetry and border irregularity are characterised by shape features, colour variegation by a feature set that contains the number of colours present in a lesion and differential structures by texture features, especially, by those based on a Markov random field model, that allows to identify different dermoscopic structures as proposed in [14]).

Some of the extracted features are based on two previous works [8], [16]. Moreover, in the current study, we include 14 additional shape and colour features. Shape features have been extracted to satisfy the A and B criteria. We use the circularity index (computed as 4 multiplied by the lesion area, divided by its squared perimeter), the perimeter normalised with respect to the equivalent diameter (a measure of elongation), length of major and minor axis of the lesion normalised with respect to the equivalent diameter (diameter of a circle with the same area as the lesion), and the variance of the distance of the border lesion points from the centroid location, eccentricity (a measure of elongation), length of major and minor axis of the lesion normalised with respect to the equivalent diameter (diameter of a circle with the same area as the lesion), and difference between these two values. The colour features have been extended to a total of 36 descriptors of colours white, blue-gray, dark brown, light brown, black and red (mean, standard deviation, kurtosis, skewness, entropy, and average of local standard deviation of each colour). For more details about the rest of features, we refer the reader to [8], [16].

III. METHODOLOGY

A. Previous notions: Support Vector Machines

Among kernel methods, the Support Vector Machine paradigm (SVM) [17] is one the most common learning methods for pattern recognition. The basic idea is the separation of two classes through a hyperplane specified by a normal vector $w$ and a bias $b$. The optimal separating hyperplane is the one that maximises the distance between the hyperplane and the nearest points in both classes (called margin). Beyond the application of kernel techniques to allow non-linear discriminators (the kernel trick), another generalisation was made to replace hard margins with soft margins [17], using the so-called slack-variables $\xi_i$ in order to avoid inseparability, relax the constraints and handle noisy data. Therefore, SVM seeks for a classifier $f : \mathbb{R}^K \rightarrow \mathbb{R}$ of the form $f(x) = w \cdot \Phi(x) + b$ ($\Phi$ being the mapping function induced by the kernel) that minimises the objective function:

$$\frac{1}{2}||w||^2 + C \sum_{i=1}^{N} \xi_i,$$  \hspace{1cm} (1)

for some cost parameter $C$ and subject to the constraints:

$$y_i((w \cdot \Phi(x_i)) + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i \in \{1, \ldots, N\}.$$  

Nonetheless, this 1-norm SVM solution for $w$ is the minimiser of the regularized empirical loss function, thus being also defined as:

$$\frac{1}{2}||w||^2 + C \sum_{i=1}^{N} (1 - y_i(w \cdot \Phi(x_i) + b))_+,$$

where $(x)_+ = \max(x, 0)$.

Originally, the SVM paradigm was proposed for binary classification problems, but it has been reformulated to deal with multiclass problems (one-against-one and one-against-all binary decompositions, among other proposals).

B. Ordinal cascade classification model

This paper considers a modification of the binary decomposition method known as the cascade linear utility model [18]. This procedure considers $Q - 1$ models (each model $D_i$ comprised of a projection $w_i$ and a threshold $b_i$), in such a way that model $q$ separates class $C_q$ from classes $C_{q+1} \cup \cdots \cup C_Q$, so that not all the classes are considered for the computation of each model. Fig. 3 graphically describes this decomposition. This methodology is naturally well-suited for the problem considered, because it balances the projections (due to the imbalanced character of the sample considered in this paper) and because it considers the partial order of the classes, and it does not assume any order constraint for $C_1$ (in this sense, $C_1$ is only considered by the first model, which discriminates $C_1$ and the rest of classes). In the literature, this approach is usually known as one-against-followers [10].

The training set for model or decision maker $D_q = \{w_q, b_q\}$ is specified by $\{X_{(i|i=q)}, X_{(j|j>q)}\}$. Therefore, a coding ma-

![Fig. 2: Examples of segmented melanomas](image)

(a) Melanoma $<0.76$ mm.  (b) Melanoma $\geq 0.76$ mm.
Fig. 3: Binary decompositions performed for a five-class problem, where \( w_i \) represents the \( i \)-th projection and \( b_i \) the bias associated to that projection. White-shadowed figures represent the negative class, black-shadowed ones the positive one and grey ones the classes not used in each model. The shape represents the original category. Note that this is an ordinal decomposition, as adjacent classes are grouped together (except for the first class, which by the problem definition presents a different relationship with the rest of classes).

Matrix \( M_{(Q-1 \times Q)} \) associated to the \( Q-1 \) binary decompositions of the cascade utility model can be defined as follows:

\[
M = \begin{pmatrix}
-1 & +1 & +1 & +1 & +1 \\
0 & -1 & +1 & +1 & +1 \\
0 & 0 & -1 & +1 & +1 \\
0 & 0 & 0 & -1 & +1
\end{pmatrix},
\]

where the label \(-1\) corresponds to the negative class patterns, the label \(+1\) to patterns belonging to the positive class, and finally, the patterns associated with label \(0\) are excluded from the training process. In this way, the approach considered is the same than in [11] but using a one-against-followers approach. A matrix of predictions can be obtained by means of a single multi-class model (e.g. using artificial neural networks) or by multiple models (training a binary classifier for each subproblem, as in this paper) [10]. Once the models have been trained, a set of \( Q-1 \) decision values \( f(x) \) are obtained for pattern \( x \). For the prediction phase, two different approaches can be considered (both tested in the experiments of this paper): a hierarchical approach or an approach based on the Error-Correcting Output Codes framework (ECOC).

The hierarchical approach is the most commonly used with the cascade binary utility model. In this case, \( w_1 \) is used in the first place, and all the patterns that are not predicted as positive (i.e. \( C_1 \)) but rather as negative (i.e. belonging to the set \( \{C_2, C_3, C_4, C_5\} \)) are used for \( w_2 \), and so on. In this sense, this approach emphasises more the first models, so that when these models fail, the final predictions are wrong without considering the rest of models. Concerning the ECOC framework [11], the principal idea is to associate each class \( C_q \in Y \) with a column of the binary coding matrix \( M \) (previously introduced). Prediction is then accomplished by choosing the column of \( M \) closest to the set of decision values \( f(x) = f_1(x), \ldots, f_{Q-1}(x) \). When the coding matrix contains a 0, this leads to an indifferent condition in the prediction phase. According to this, the final decision function is the following one:

\[
C(x) = C_q, \text{ where } q = \arg \min_{q=1 \ldots Q} d(M_q, f(x))
\]

where \( M_q \) is the \( q \)-th row of matrix \( M \) and \( d \) is the loss function considered. The main issue within this paradigm is the choice of a loss function which corresponds with the loss function used for deriving the binary classifier. In this case, due to the choice of the 1-norm SVM paradigm as the base methodology, the hinge-loss function is chosen. To see this, we formulate Eq. 1 in terms of the error:

\[
\frac{1}{2} \|w\|^2 + C \sum_{i=1}^{N} \text{loss}(y_i, f(x_i)),
\]

where the error function chosen is usually the hinge-loss:

\[
\text{loss}(y_i, f(x_i)) = (1 - y_i \cdot f(x_i))^+ = \xi_i = \max(0, 1 - y_i(w \cdot \Phi(x_i) + b)).
\]

In this way, the final decision function is \( C(x) = C_q \), where:

\[
q = \arg \min_{q=1 \ldots Q} \sum_{i=1}^{Q-1} \max(0, 1 - M_{q_i} \cdot f_i(x)))
\]

One of the main advantages of this methodology over a purely hierarchical approach is that all real values are used for prediction instead of binary predicted class values. Consequently, the model is provided with additional information which may be useful for improving its performance. Note that the decision values \( f(x) \) represent the distance to the threshold, which is a measure usually considered for estimating the class probabilities.

### IV. Experiments

Different classifiers (nominal and ordinal) are compared in this paper. The methods included are all single model methods or binary approaches to multi-class/ordinal classification. More specifically, the methods tested are the following:

- **Kernel Discriminant Learning for Ordinal Regression (KDLOR)** [10] extends the Kernel Discriminant Analysis (KDA) to ordinal classification using a rank constraint.
- **The Proportional Odds Model (POM)**, which adapts standard logistic regression to the ordinal case.
- **RED-SVM** [10], which applies the reduction from cost-sensitive ordinal ranking to weighted binary classification framework (RED) to SVM.
- **Logistic Regression where the classification model is composed of several binary models using the one-against-all scheme. We use the Regularized Multinomial Logistic Regression (RMLR) implemented in LIBLINEAR [19].**
- **Support Vector Classifier using the one-against-one approach (SVC1V1) [20].**
• The reformulation of SVMs for ordinal classification with implicit constraints (SVORIM) [10].
• Weighted Support Vector Machine with Ordered Partitions (WSVMOP), which considers a binary decomposition method using weight-based SVMs [21].
• The Ordinal Cascade binary utility model proposed in this paper, when using two different prediction approaches: the hierarchical approach (OC-H) and the ECOC framework (OC-ECOC) [11]. The model is described in Section III, and, in contrast to SVORIM and POM, it is able to consider the partial potential ordering of the classes. Given the nature of the decompositions, it also tackles the imbalanced nature of the data. We test these models using SVMs (OC-ECOC-SVM and OC-H-SVM) and RMLR (OC-ECOC-LR and OC-H-LR).

To take into account different aspects of the classification performance, we have selected different metrics that evaluate the global performance, the balance of performance for the different classes and the ordinal magnitude of the errors:

- **Accuracy (Acc)** is the percentage of correctly classified patterns:
  \[
  Acc = 100 \cdot \frac{1}{N} \sum_{i=1}^{N} [\hat{y}_i = y_i],
  \]
  where \([\cdot]\) is the indicator function (being 1 if the condition is true, and 0 otherwise), and \(\hat{y}_i\) is the predicted target for \(x_i\).
- The geometric mean of the sensitivities (GM) is typically used in imbalanced problems:
  \[
  GM = 100 \cdot \sqrt[5]{\prod_{q=1}^{5} S_q},
  \]
  where \(S_q\) is the sensitivity (accuracy ratio) of the classifier for class \(q\). If \(GM = 0\), the classifier is totally misclassifying at least one class.
- The **Mean Absolute Error (MAE)** is the average deviation in absolute value of the predicted class from the true class. It is the most commonly used ordinal classification metric. For imbalanced datasets, this measure is modified to consider the relative frequency of the classes, resulting in the Average MAE (AMAE) and Maximum MAE (MMAE) [22]:
  \[
  AMAE = \frac{1}{5} \sum_{q=1}^{5} MAE_q = \frac{1}{5} \sum_{q=1}^{5} \frac{1}{N_q} \sum_{i=1}^{N_q} e(x_i), \quad (3)
  \]
  \[
  MMAE = \frac{1}{5} \max_{q=1}^{5} MAE_q, \quad (4)
  \]
  where \(e(x_i) = |\mathcal{O}(y_i) - \mathcal{O}(\hat{y}_i)|\) is the distance between the true and the predicted ranks, and \(\mathcal{O}(C_q) = q\) is the position of the \(q\)-th label. AMAE values range from 0 to 4, and so do MMAE values.

The first class (\(C_1\)) is also considered for calculation of the errors, given that the partial ordering is assumed for the patterns in the input space, but the miss-classification costs of \(C_1\) with respect to \(C_2\) to \(C_5\) can be assumed to be the same than the ones applied in ordinal classification (i.e. 1, 2, 3 and 4, respectively). In this way, confusing a benign lesion with a Stage 0 melanoma will have a lower cost (i.e. will be less penalised) than confusing it with a Stage III melanoma, which at the same time will be more penalised than confusing it with a Stage II melanoma, and so on.

The experiments have been performed using a 10-fold partition design, and the metrics are calculated using the sum of all generalisation confusion matrices from the 10 folds. To adjust the kernel width and cost parameters for the SVM-based methods (RED SVM, SVC1V1, SVORIM, WSVMOP, OC-ECOC-SVM and OC-H-SVM), a nested cross-validation is applied to the training data, with a grid search with parameter values within the range \(10^{-3}, 10^{-2}, \ldots, 10^{3}\). The kernel width of KDLOH is optimised using the same range than SVM-based methods with regularization parameter values in the range \(u \in \{10^{-2}, 10^{-3}, \ldots, 10^{-5}\}\). The cost parameter of RMLR, OC-ECOC-LR and OC-H-LR is adjusted using the same values of the SVM-based methods. The criteria for selecting the parameters is AMAE, which had a positive impact on the performance of the metrics related to imbalance.

The results of the experiments performed can be seen in TABLE II, where the best performing method is highlighted in bold face and the second one in italics. **Acc** and **GM** are to be maximised, whereas **AMAE** and **MMAE** have to be minimised.

Several conclusions can be drawn from TABLE II. Firstly, it can be inferred that the problem can be addressed as standard ordinal regression, as the performance of state-of-the-art methods is quite satisfactory. Comparing the nominal SVC with other ordinal methods (e.g. SVORIM and RED SVM), it can be seen that the ordinal approaches obtain better performance in the ordinal metrics, thus validating the need of considering this problem as an ordinal one. **Acc** metric can be misleading, in such a way that the best accuracy method (SVC) obtains the second worst AMAE results and the worst MMAE values. In this way, high Acc values mask significant errors for some of the classes (specially, for minority classes) and do not take the order information into account. Secondly, from the

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc</th>
<th>GM</th>
<th>AMAE</th>
<th>MMAE</th>
</tr>
</thead>
<tbody>
<tr>
<td>RMLR</td>
<td>63.17</td>
<td>37.53</td>
<td>0.852</td>
<td>1.167</td>
</tr>
<tr>
<td>POM</td>
<td>61.57</td>
<td>32.04</td>
<td>0.850</td>
<td>1.240</td>
</tr>
<tr>
<td>SVC</td>
<td>66.55</td>
<td>40.17</td>
<td>0.910</td>
<td>1.450</td>
</tr>
<tr>
<td>SVORIM</td>
<td>63.35</td>
<td>40.11</td>
<td>0.820</td>
<td>1.210</td>
</tr>
<tr>
<td>REDSVM</td>
<td>63.52</td>
<td>35.94</td>
<td>0.830</td>
<td>1.140</td>
</tr>
<tr>
<td>KDLOR</td>
<td>54.63</td>
<td>35.66</td>
<td>0.880</td>
<td>1.340</td>
</tr>
<tr>
<td>WSVMOP</td>
<td>65.30</td>
<td>33.34</td>
<td>0.920</td>
<td>1.410</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Proposed methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>OC-H-SVM</td>
</tr>
<tr>
<td>OC-ECOC-SVM</td>
</tr>
<tr>
<td>OC-H-LR</td>
</tr>
<tr>
<td>OC-ECOC-LR</td>
</tr>
</tbody>
</table>
two variants proposed in this paper (hierarchical prediction and ECOC method), it seems that the hierarchical method works better for LR, while the ECOC framework is better suited for SVM. Furthermore, three of the variants of the two proposed strategies (OC-ECOC-SVM, OC-H-SVM and OC-H-LR) lead to the best performance considering the ordinal and imbalanced metrics ($GM$, $AMA E$ and $MMA E$). The differences favouring these methods with respect to SVRIM and REDSVM are clearer for ordinal metrics ($AMA E$ and $MMA E$). This is due to the fact that, given than partial ordering is not exploited by standard ordinal regression methods, they can misclassify class $C_1$ with a higher probability. As $C_1$ is an extreme class, the errors committed for this class tend to be of higher magnitude. Finally, the logistic regression in the OC-H framework significantly improves its performance considering all the performance metrics with respect to POM and RMLR. This allows the use of a linear model for melanoma detection and thickness classification that enables an interpretation of the corresponding equation.

V. CONCLUSIONS

This paper presents an approach for automatic melanoma characterisation via computational image analysis and machine learning methods. This paper proposes classifiers which simultaneously differentiates melanomas from benign lesions and, in the case of melanomas, predicts their depth. The depth is characterised by using a set of ordered labels, representing different stages according to the Breslow index. In this sense, the problem is an ordinal regression problem, where the order of the categories can be used to infer more robust classifier and minimise the magnitude of the errors. Additionally, we experimentally confirm that this problem is a partially ordered dataset in the sense that benign lesions can not be assumed to be ordered with respect to the different types of melanomas. We propose to use a cascade binary utility ordinal model based on binary decompositions, with two different strategies for fusing the predictions of the binary models. The results outline that this partial ordering should be taken into account to minimise the magnitude of the errors for minority classes, given the imbalanced nature of the dataset.

As future work, we intend to explore other paradigms and models to adapt the existing learners to this type of classification problems. Furthermore, we aim at interpreting the binary decomposition based on logistic regression to understand the most determining features for melanoma detection.

ACKNOWLEDGMENTS

This work was partly financed by the TIN2014-54583-C2-1-R project of the Spanish Ministry of Economy and Competitively (MINECO), by FEDER Funds and by the P11-TIC-7508 project of the Junta de Andalucía, Spain.

REFERENCES


Abstract—We review some of the latest proposals for accelerating the training of SVMs. In particular we consider two methods, one based on conjugate descent and another based on Nesterov’s Accelerated Gradient, and show that both lead to a substantial decrease in the number of iterations for SMO to converge to a given precision.

I. INTRODUCTION

Support Vector Machine (SVM) classification (SVC) and regression (SVR) are among the most powerful tools of modern Machine Learning, as they have been shown to give excellent results in a wide variety of problems. A formulation that works simultaneously for both SVC and SVR was introduced in [1]. Consider a set of triplets $S = \{(\mathbf{x}_i, y_i, p_i) : i = 1, \ldots, N\}$ with $y_i = \pm 1$ and $p_i$ some scalar values, and the following convex optimization problem:

$$
\min_{\mathbf{w}, b, \xi} \frac{1}{2}\|\mathbf{w}\|^2 + C \sum_i \xi_i \\
\text{s.t. } y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq p_i - \xi_i, \quad \xi_i \geq 0, \quad \forall i.
$$

(1)

If $p_i = 1$ and $y_i$ are the class labels, the previous formulation reduces to SVC, where $C$ is an hyper-parameter. Similarly, let $R = \{(\mathbf{x}_i, t_i) : i = 1, \ldots, N\}$ be the sample, where the $t_i$ are the targets associated to inputs $\mathbf{x}_i$. Define next an enlarged sample as $S = \{(\mathbf{x}_i, y_i, p_i) : i = 1, \ldots, 2N\}$ with $\mathbf{x}_i = \mathbf{x}_{i+N}, y_i = 1, p_i = t_i - \epsilon$ for $i = 1 \ldots N$ and $y_i = -1, p_i = -t_i - \epsilon$, for $i = N + 1 \ldots 2N$. Then (1) reduces to $\epsilon$-insensitive SV regression (SVR).

Assuming that there is at least one finite optimal solution $(\mathbf{w}^*, \mathbf{b}^*, \xi^*)$ of (1) (there may be instances of (1) for which this doesn’t hold), standard convex optimization theory can be applied. Solving (1) can be shown to be equivalent to solving its dual problem

$$
\min_{\alpha} f(\alpha) = \frac{1}{2} \sum_{i,j} \alpha_i \alpha_j Q_{ij} - \sum_i \alpha_i p_i \\
\text{s.t. } 0 \leq \alpha_i \leq C, \ 1 \leq i \leq N; \quad \sum_i \alpha_i y_i = 0.
$$

(2)

where $Q_{ij} = y_i y_j \mathbf{x}_i \mathbf{x}_j$. We shall refer to $\sum_i \alpha_i y_i = 0$ as the equality constraint. (2) is a constrained quadratic optimization problem and many techniques could be applied to solve it. However, some techniques like projected gradient are difficult to apply due to the the equality constraint. By assuming no bias term $b$ we can drop the constraint, but the high dimensionality (given by the sample size $N$) still makes applying projected gradient very costly. The alternative is to use other simpler descent directions that ensure the feasibility of the updates and have a cost per iteration as small as possible. Several algorithms have been proposed to optimize this problem but Sequential Minimum Optimization (SMO) [2], [3] is clearly the method of choice, particularly when implemented in high quality packages such as LIBSVM [4]. In fact, the second order algorithm implemented in LIBSVM, that we describe below, can be considered the state of the art for non–linear SVC and SVR.

Improving the performance SMO has received much attention in the literature and we will review here two recent proposals [5] and [6]. The first one tries to improve the descent direction given by SMO by adding a momentum term $m_k = \alpha^k - \alpha^{k+1}$. It is well known in unconstrained quadratic minimization that taking $d_k = -\nabla f(\alpha^k)$, properly scaled momentum is equivalent to conjugate gradient descent. As we will discuss in section III this momentum term implies some extra computation per iteration and, potentially, a projection step to ensure that the new coefficients satisfy the box constrains. In [5] they derive a conjugate version of SMO that 1) only needs $\frac{2}{N}$ extra float products per iteration and, 2) reduces the projection step to a simple clipping of the $\rho$ coefficient.

The second option is to apply Nesterov’s Accelerated Gradient (NAG) ideas to SMO. If one uses the Gaussian kernel and there is no pattern $x$ that appears in the sample with both $y = 1, -1$, the kernel matrix is positive definite [7], i.e., we have $\ell I \preceq \mathbf{K} \preceq L I, \ 0 \leq \ell \leq L$, and the SMO algorithm (see Section II), that performs a kind of approximated gradient descent, achieves a linear convergence rate, i.e., $f(\alpha^k) - f(\alpha^*) \leq C\lambda^k$ for some $\lambda < 1$ at iteration $k$.

For such a $Q$, $f(\alpha)$ is not only $L$ smooth but also $\ell$ strongly convex, although both $\ell, L$ are unknown. Then, gradient decent with a fixed rate $1/L$ achieves an $O(\gamma^k)$ linear convergence, where $\gamma = \frac{\ell - 1}{\ell}$ and $\kappa = L/\ell$ is the condition number of $f$ [8], Theorem 2.1.14. In addition, using a modified $\lambda_0$ sequence that includes $\kappa$ in its definition, NAG achieves a $O(\Gamma^k)$ convergence rate, with $\Gamma = \sqrt{\kappa + 1}$ (Theorem 2.2.2,
Since \( L \) and \( \ell \) are usually not known, this version cannot be used in practice. We will discuss here the proposal in [6], where they consider the use of NAG for solving the dual SVM problem, and we will check whether it reduces the number of iterations of standard SMO. The simplest approach would be the straight application of Nesterov’s procedure, but this would result in an algorithm much worse than SMO, as NAG is not monotone whereas SMO ensures \( f(\alpha^{k+1}) < f(\alpha^k) \) at each iteration. A solution for this is to use a monotone versions of NAG [9], where \( \alpha' \) becomes the new \( \alpha^{k+1} \), but only if \( f(\alpha') < f(\alpha^k) \).

If that is not the case, then the last \( \alpha \) retained is that is, \( \alpha^{k+1} = \alpha^k \), and we try with a new \( \alpha^{k+1} \) defined as \( \alpha^{k+1} = \rho_k x^{k+1} + (1-\rho_k)\alpha^k \), where \( \rho_k = \frac{\lambda_k}{\lambda_{k+1}} \). Note that since \( 0 \leq \rho_k \leq 1 \), \( \alpha^{k+1} \) can be seen as a convex combination of \( \alpha^k \) and \( \alpha^{k+1} \) that seeks to reduce the overshooting influence of \( \alpha^k \). However, this doesn’t guarantee that \( f(\alpha') > f(\alpha^{k+1}) = f(\alpha^k) \) and we may end up with a possibly long sequence of \( \alpha \) values with the same \( f \) value. In addition, there is the extra drawback of requiring the computation of the value \( f(x^k) \) of the SVM dual function, with a cost of \( N \) products if done using gradient information. In [6] they alleviate both drawbacks by taking advantage of the quadratic structure of \( f \) to directly compute the exact \( \rho_k \) and \( x^{k+1} \) that minimize \( f \) and they call this approach the monotone Nesterov acceleration (NA) of SMO, or monotone NA-SMO.

The rest of the paper is organized as follows. In Section II we will briefly review the SMO algorithm. In sections III and IV we give the details of the conjugate SMO and Nesterov Accelerated SMO proposals [5] and [6], respectively. We will numerically compare them in Section V together with standard SMO. A short discussion and conclusions section ends the paper.

II. Second Order SMO

We give a brief review focused on SMO’s computational costs; see [4] for more details. SMO iterations can be written as \( \alpha^{k+1} = \alpha^k + \rho_k d^k \), where \( d^k \) is a descent direction of the form \( d = y_L e_L - y_U e_U \) with \( e_j \) the canonical 0–1 basis vectors and where \( U^k, U^k \) correspond to a sample pair \( x_L, x_U \) which violates the most the SVM’s KKT conditions at iteration \( k \). That guarantees that \( d^k \cdot \nabla f(\alpha^k) < 0 \) and also that they are the most negative among all feasible choices of \( L \) and \( U \). The resulting unconstrained gain will be

\[
f^k - f^{k+1} = \frac{\langle d^k \cdot \nabla f(\alpha^k) \rangle^2}{2\|x_L^k - x_U^k\|^2}.
\]

This turns out to be maximal on the numerator but perhaps not so in the denominator. Thus it can be improved by using the same \( L^k \) as before but taking now \( U^k \) to maximize \( f \). This results in second order SMO [7]; we will refer to it simply as SMO from now on. The unconstrained \( \alpha \) updates would then be

\[
\alpha_L^{k+1} = \alpha_L^k + y_L \rho_k, \\
\alpha_U^{k+1} = \alpha_U^k - y_U \rho_k
\]

for an appropriate \( \rho_k^* \) which we may have to clip if necessary so that they verify the box constraints. Note that \( \sum_p y_p d_p^k = 0 \) and, hence, \( \sum_p y_p \alpha_p^k = 0 \), provided it does so for \( \alpha^k \). The SMO iterates continue until a stopping condition \( M(\alpha_t) - m(\alpha_t) < \epsilon_{KKKT} \) is met for a pre-selected KKT tolerance \( \epsilon_{KKKT} \), where

\[
m(\alpha) = \min_{p \in Z_p} y_p \nabla f(\alpha_p), \quad M(\alpha) = \max_{q \in Z_{out}} y_q \nabla f(\alpha_q),
\]

for appropriate index sets \( Z_p, Z_{out} \) (see [4], section 4.1.2). The floating point cost per iteration of SMO is determined by the second order choice of \( U^k \) and the update of the gradient \( g^k = \nabla f(\alpha^k) \). Selecting \( U^k \) requires \( 2N \) products and for the gradient update we have

\[
g^{k+1} = Q \alpha^{k+1} - p = Q \alpha^k - p + \rho_k Q d^k = g^k + \rho_k (y_L^k Q_L^k - y_U^k Q_U^k),
\]

where \( Q_j \) denotes \( Q \)’s \( j \)-th column. This requires \( N \) products and, hence, \( 3N \) floating point operations are needed in total for each SMO update.

III. Conjugate SMO

Momentum is often used to improve descent directions in several optimization algorithms. Writing the momentum term \( \rho \), we have \( g^k + \rho_k (y_L^k Q_L^k - y_U^k Q_U^k) \), where \( Q_k \) denotes \( Q \)’s \( j \)-th column. This requires \( N \) products and, hence, \( 3N \) floating point operations are needed in total for each SMO update.

We first define a momentum update writing

\[
\alpha' = \alpha^k + \rho_k' d^k + \sigma_{k-1} h^{k-1}
\]

for appropriate \( \rho_k' \) and \( \sigma_k \). Then \( m^{k+1} = \alpha^{k+1} - \alpha^k = \rho_k h^k \), and it follows that

\[
h^k = d^k + \beta_k m^k
\]

for appropriate \( \beta_k \) and \( \sigma_k \). As we mentioned before, in quadratic optimization a properly scaled momentum term is equivalent to conjugate descent. Once this conjugate direction \( h^k \) has been chosen, we find the unconstrained \( \rho_k' \) factor by minimizing \( f \) along the line of \( h^k \), that is, by solving

\[
0 = \nabla f(\alpha_k + \rho_k h^k) \cdot h^k = g^k \cdot h^k + \rho_k Q h^k,
\]

where \( g^k \) denotes the gradient \( \nabla f(\alpha^k) \). This yields

\[
\rho_k' = -\frac{g^k \cdot h^k}{h^k \cdot Q h^k} + \frac{d^k \cdot h^k}{h^k \cdot Q h^k}.
\]
If the previous line minimization along \( h^{k-1} \) has been unclipped, i.e., we have \( \alpha^k = \alpha^{k-1} + \rho'_k h^{k-1} \), the following orthogonality condition must hold
\[
g^k \cdot h^{k-1} = 0. \tag{9}
\]
We will call \( (9) \) the first orthogonality condition. As a consequence,
\[
g^k \cdot h = g^k \cdot d^k + \sigma_{k-1} g^k \cdot h^{k-1} = g^k \cdot d^k < 0,
\]
i.e., \( h^k \) is also a descent direction. Moreover we can then write \( \rho'_k \) as
\[
\rho'_k = -g^k \cdot d^k \quad \frac{1}{h^k \cdot Qh^k}, \tag{10}
\]
and it is easy to see that the unconstrained gain in \( f \) is now
\[
f^k - f^{k+1} = (\alpha^k) - f(\alpha')
= \frac{1}{2}(g^k \cdot d^k)^2,
\tag{11}
\]
where we have used our previous choice of the unconstrained \( \rho'_k \). Now we can maximize this gain by choosing \( \sigma_{k-1} \) to minimize the denominator \( h^k \cdot Qh^k \). Writing it as a function of \( \sigma \), we have
\[
\phi(\sigma) = (d^k + \sigma_{k-1} h^{k-1}) \cdot Q(d^k + \sigma_{k-1} h^{k-1}),
\]
and it follows that
\[
0 = \frac{1}{2} \phi'(\sigma_{k-1}) = d^k \cdot Qh^{k-1} + \sigma_{k-1} h^{k-1} \cdot Qh^{k-1}.
\]
Finally, this implies
\[
\sigma_{k-1} = -\frac{d^k \cdot Qh^{k-1}}{h^{k-1} \cdot Qh^{k-1}}. \tag{12}
\]
Now it is easy to see that this choice results in
\[
h^k \cdot Qh^{k-1} = 0, \tag{13}
\]
We will call \( (13) \) the second orthogonality condition. We can summarize now our conjugate SMO updates. If the iteration ending in \( \alpha^k \) along \( h^{k-1} \) has not been clipped, we can:
1) compute \( \sigma_{k-1} \) by \( (12) \) and \( h^k \) by \( (6) \),
2) compute \( \rho'_k \) by \( (3) \) and \( \alpha' = \alpha^k + \rho'_k h^k \) by \( (4) \) and, finally,
3) check whether \( (\alpha')^{k+1} \) satisfies the box constraints
\[
0 \leq \alpha'_p \leq C
\]
and, if not, clip \( \rho'_k \) accordingly (see \( (5) \)) to get the final \( \rho_k \) and \( \alpha^{k+1} = \alpha^k + \rho_k h^k \).
Note that if the preceding \( h^{k-1} \) verifies \( \sum_p y^p h^k_0 = 0 \), then
\[
\sum_p y^p h^k_p = \sum_p y^p (d^k + \sigma_{k-1} h^{k-1}) = 0
\]
and the new \( \alpha^k \) verifies the linear constraint, i.e.,
\[
\sum_p y^p \alpha^k + \rho_k \sum_p y^p h^k_p = 0.
\]
On the other hand, if we had had to apply clipping to arrive at \( \alpha^{k+1} \), i.e., we have hit the boundary of the box region, we will simply reset \( h^k \) to 0 after the update, as keeping the current conjugate direction may lead to further boundary hits. We will then have \( h^{k+1} = d^{k+1} \) at the new iteration, which becomes then a standard SMO update.

At first sight, the possible advantages of working with the conjugate directions may be offset by the higher cost that deriving them imposes on the standard SMO iterations, particularly that of computing the values \( h^k \cdot Qh^k \) required in the different denominators. However, working with the SMO descent directions \( d = y_{lce} - y_{ve} \) greatly simplifies these computations. For this, we will keep an auxiliary vector \( H = Qh \) and constant \( \delta = h \cdot Qh \) that we will update at each iteration and use to simplify the computation of the other elements as follows:
\[
\sigma_{k-1} = -\frac{d^k \cdot Qh^{k-1}}{h^{k-1} \cdot Qh^{k-1}} = \frac{\delta^{k-1}}{\delta^k} \tag{14}
\]
\[
H^k = Q(d^k + \sigma_{k-1} h^{k-1})
\]
\[
= y_{lce} Qh_k - y_{ve} Qh_k + \sigma_{k-1} H^{k-1}, \tag{15}
\]
\[
\delta_k = h^k \cdot Qh^k = d^k \cdot H^k
\]
\[
= y^k H^k + \sigma_{k-1} H^{k-1} - y_{lce} H^{k-1}, \tag{16}
\]
\[
\rho'_k = -g^k \cdot d^k
= g^k Qh^k - y^k Qh^k
\]
where \( Q_i \) denotes \( Q \)'s \( i \)-th column. This result in a general conjugate SMO algorithm where, starting in iteration \( k \) with \( \alpha^k \), its gradient \( g^k = Q \alpha^k - P, h^{k-1}, H^{k-1}, \delta^{k-1}, \delta^k \), we iteratively:
1) Select the \( L^k, U^k \) using the SMO procedure.
2) Compute the kernel matrix columns \( Q_{L^k} \) and \( Q_{U^k} \), if not previously cached.
3) Compute \( \alpha_{k-1} \) as in \( (14) \), update \( h^k = d^k + \sigma_{k-1} h^{k-1}, H^k \) as in \( (15) \) and \( \delta^k \) as in \( (16) \).
4) Compute an unconstrained step \( \rho'_k \) as in \( (17) \) and clip it to \( \rho_k \) if needed according to \( \rho'_k, \alpha^k, h^k \).
5) Update \( \alpha^k \) using \( (3) \) and \( g^k \) as \( g^{k+1} = g^k + \rho_k H^k \).
6) If clipping has taken place, reset \( h^k = H^k = 0, \delta^k = 1 \).
We finish this section discussing the computational cost of the conjugate SMO updates. If \( N_l \) and \( N_u \) denote the number of non-zero components of \( h \) and \( H \) respectively, the cost in products of each iteration is
1) \( 2N \) floating point operations in Item \( \text{I} \) when selecting \( L^k, U^k \).
2) \( N_l \) operations to update \( h \) in Item \( \text{I} \) to update \( \alpha \) in Item \( \text{I} \) and to compute a clipped \( \rho_k \) in Item \( \text{I} \).
3) \( N_H \) operations to update \( H \) in Item \( \text{I} \) and
4) \( N \) operations to update the gradient \( g \) in Item \( \text{I} \).
We expect \( N_H \approx N \) but \( N_l \) should coincide with the number of non-zero components in \( \alpha \); this number should be \( \ll N \) and, similarly, we should have \( N_U \ll N \). Thus a conjugate iteration should theoretically add a cost of
\[
3N + N_H + 2N_l \approx 3N + N_H \approx 4N,
\]
in contrast with $3N$ for a standard SMO iteration. Therefore, it should lead to a faster training if the number of SMO iterations is more than $4/3$ the number of CD-SMO ones. In any case, note that the cost of the iterations in which a non-cached kernel column matrix has to be computed will require a much larger $N \times d$ number of products when working with patterns in a $d$ dimensional space or even more in a kernel setting. Thus, in the starting iterations the cost of the SMO and conjugate SMO would be dominated by the much larger cost of computing the required $Q$ columns.

IV. MONOTONE NESTEROV ACCELERATED SMO

Standard NAG comprises two steps: to compute an intermediate point $x^{k+1} = \alpha^k + \mu_k(\alpha^{k-1} - \alpha^k)$ with a fixed $\mu_k$ and then to perform gradient descent to arrive at the new $\alpha^{k+1} = x^k - \epsilon_k \nabla f(x^k)$ where $\epsilon_k$ is a step parameter. When applied to SVM’s dual problem this last step can obviously be replaced by an SMO step from $x^k$ choosing $L, U$ according to the gradient $G^k = \nabla f(x^k)$ of $f$ at $x^k$. Note that we denote with $G$ the gradient at the intermediate point, while $g$ is still the final gradient. Moreover, we must also ensure the feasibility of $x^k$. This requires first that $\sum_i y_i^k x_i^k = 0$, which will clearly hold if both $\alpha^k$ and $\alpha^{k-1}$ are feasible, and second that $0 \leq x_i^k \leq C$, which can be easily achieved by clipping the coordinates.

In any case, while SMO guarantees a strictly monotone decrease of the dual function $f$, standard NAG is not monotone. Already mentioned, a monotone variant has been proposed in [9], where $f(\alpha^k)$ and $f(x^k)$ are compared at each iteration and if $f(\alpha^k) < f(x^k)$, we compute $x^{k+1} = \rho_k x^k + (1 - \rho_k) \alpha^k$, with $\rho_k = \frac{\gamma_k}{\gamma_{k+1}}$. Note that $x^{k+1}$ is automatically feasible, being a convex combination of $x^k$ and $\alpha^k$.

However, there is no guarantee that even after using $x^{k+1}$ to estimate the new $\alpha'$, we arrive at $f(\alpha') < f(\alpha^k)$, and a series of steps may follow in which $f$ remains constant. In [6], they compute an optimal $\rho$ that ensures $f(x^{k+1}) < f(\alpha^k)$. Let $x_\rho^k = \rho x^k + (1 - \rho) \alpha^k$, then $f(x_\rho^k)$ is a function $\phi(\rho)$ for which is easy to see that its minimum is reached at $\rho^* = \frac{(x^k - \alpha^k) \cdot 1 + \|\alpha^k\|_Q^2}{\|\alpha^k\|_Q^2 + \|x^k\|_Q^2}$.

To compute $\|\alpha^k\|_Q^2 = \alpha^k \cdot Q \alpha^k$ and $\|x^k\|_Q^2$, observe that $f(\alpha^k) = \frac{1}{2}\|\alpha^k\|_Q^2 - \alpha^k \cdot 1$, i.e., writing $s_\alpha = \alpha \cdot 1 = \sum \alpha_p$, we have $\|\alpha^k\|_Q^2 = 2f(\alpha^k) + s_\alpha^k$ and, thus,

$$
\rho^* = \frac{s_\alpha^k - s_\alpha + 2(f(\alpha^k) + s_\alpha)}{2f(\alpha^k) + s_\alpha + 2f(\alpha^k) + s_\alpha} = \frac{2f(\alpha^k) + s_\alpha^k + s_\alpha}{2f(\alpha^k) + f(x^k) + s_\alpha + s_\alpha^k}.
$$

To estimate the complexity of the previous steps, note that $2N$ products will be needed to compute $x^{k+1} = \alpha^k + \rho(x^k - \alpha^k)$ and $G^{k+1} = g^k + \rho G^k$ if we perform either a standard momentum or an exact monotone Nesterov step. To this we must add $N$ products to compute $f(x_k)$ as $f(x^k) = \frac{1}{2}(x^k \cdot G^k - s_\alpha^k)$ (we can retain $f(\alpha^k)$ from the previous iteration) plus another $N$ products to update $g_k$ from $G^k$. Summing things up, monotone NA–SMO adds $3N$ products to the $3N$ ones needed by SMO, with a total complexity of $6N$ products per iteration, i.e., twice that of standard SMO.

V. NUMERICAL EXPERIMENTS

In this section we will compare the behavior of second order SMO, conjugate descent SMO and monotone Nesterov Accelerated SMO in 8 binary classification datasets, namely, heart, diabetes, australiant, german (in its numeric version), adult4, adult8, web7 and web8, that are all available in the LIBSVM page. Table I shows for every dataset the number of samples for each class and dimension. We will work with a Gaussian kernel $k(x, x') = e^{-\delta (x - x')^2}$, where $\gamma = d$ (sample dimension); that is, we use LIBSVM’s default Gaussian kernel width. Features for the first 4 problems are scaled to $[-1, 1]$ while the adult and web problems already have binary (0-1) features.

We shall consider three different $C$ values, 1, 100, 1000 and two KKT tolerances 0.1, 0.001. Table II gives for each dataset, $C$ and $\epsilon_{\text{KKT}}$ values the number of iterations of plain SMO, conjugate descent SMO (CD5) and monotone Nesterov SMO (NAS) and their corresponding ratio. Ratios above 1 mean that the corresponding method performs less iterations than SMO while ratios below 1 mean more iterations. We have also checked that the three methods yield essentially the same final SVM model, in the sense that they arrive with high precision at the same value $f(\alpha)$ of the SVM objective function and have very similar number of support vectors (SVs).

As it can be seen in table II CD-SMO always performs less iterations than SMO, while NA-SMO only does so for $C = 10, 100$ in the first 4 problems, all $C$ values in the adult problem and none in web. Theoretically, these gains in iterations would correspond to faster execution times if the ratio is bigger than $4/3$ in CD-SMO and 2 in NA-SMO, which only happens for big $C$ values and small $\epsilon_{\text{KKT}}$ (more precise solutions). However, this does not take into account kernel computations which, in practice, dominate the entire iteration time if the cache does not kick in. Thus these methods would have a natural advantage in settings where the kernel matrix is very big and/or the size of the available cache is small, since the extra computation time per iteration is negligible when compared with the kernel operation, as shown in figure I.
VI. DISCUSSION AND CONCLUSIONS

In this work we have reviewed two variants of the SMO algorithm, that achieve a substantial reduction of the number of iterations needed to arrive at a given convergence precision while giving essentially the same final model in terms of its number of SVs and the final value of the SVC cost function. In particular we proposed two approaches, one based on conjugate descent and another one based on Nesterov’s accelerated gradient. For the first case our complexity analysis suggests that the reduction in the number of iterations can be achieved with an iteration cost about 4/3 larger than the second order SMO version implemented in LIBSVM, while for the second the iterations are twice as slow. In our experiments the ratio between the number of standard SMO and conjugate descent iterations was bigger than 4/3 for the ratio between the number of standard SMO and conjugate second order SMO version implemented in LIBSVM, while for the NAS worked better than conjugate descent, surpassing even the theoretical ratio of 2 for $C = 100$ and $\epsilon_{KKT} = 0.001$. Thus, from a practical point of view, and as we have made clear, the smaller number of iterations of NA-SMO are not likely to result in training times smaller than those of standard SMO. However, this extra cost won’t have an impact in the initial iterations, as their cost is dominated by that of building a cache for the kernel matrix. Thus, SMO acceleration may be worthwhile if it initially achieves a substantial reduction of the cost function while the kernel matrix cache is built, even if one reverts to standard SMO afterwards. This should be more pronounced in large sample problems and also in for high $C$ values.

In any case, and as mentioned before, the experiments reported here must be extended to timing results, for which a proper comparison should be done against LIBSVM, the state of the art solver for non-linear SVMs. This implies that our CD SMO and Nesterov’s implementations must be carried into the LIBSVM framework so that we can redo our comparisons here in terms of running times under a common LIBSVM setting. These comparisons must also include other variables like the size of the cache, which directly impacts the performance of the method.

Furthermore, the relevance of improvements on SMO depends on the difficulty of the problem at hand, that is broadly dependent on the $C$, $\gamma$ and (for SVR) $\epsilon$ choices. A discussion of this effects for SVC and Gaussian kernels is made in [10] and in principle, large $\gamma$ and $C$ should lead to more difficult SVM problems.

Another key factor for the effectiveness of an acceleration method is the condition number of SVM’s kernel matrix $Q$. For unconstrained quadratic problems over a positive definite $Q$, gradient descent has a linear convergence with an approxi-
Fig. 1. Time evolution of the objective function (top) and number of kernel operations (bottom) for different cache sizes.