Partially labelled data: classification and discovery of unknown labels

using subspaces of features

Luis Guerra

l.guerra@upm.es

CIG – Computational Intelligence Group

February, 2011

Outline

Data characteristics

- Let $\mathcal{D} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a data set of instances
- And $\mathbf{x}_i = \{x_{i1}, \ldots, x_{in}\}$, being n the number of observable features
- A class label can be observale or hidden for each feature, if observable $c_i \in (1, \ldots, c)$, the set of class labels
- The observable features could be continuous, discrete or a mix of them
- The set of c is the key to tackle the problem

$\mathcal{D}$	X_1	$\ldots$	X_n	C
$\mathbf{x}_1$	$x_{1,1}$	$\ldots$	$x_{1,n}$	c_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_N$	$x_{N,1}$	$\ldots$	$x_{N,n}$	c_N

Supervised classification

- $c_i, \forall i$ is known
- A **predictive** model is built based on a labelled data set
- This model will be able to predict the value of c_{i+1} given $\mathbf{x}_{i+1}$
- A honest validation is necessary in order to avoid overfitting
- There are many approaches to solve this kind of classification:
  - Bayesian approach
  - Decision trees
  - Logistic regression
  - SVM
  - ...

Unsupervised classification

- $c_i, \forall_i$ (and $|C|$) is (are) unknown
- This kind of problem is often called *Clustering*
- A **descriptive** model is built based on the set of observable features
- This model will partition the data into a certain number of groups, called clusters
- Validation is a very difficult task because of the absence of the ground truth
- There are different approaches to solve this kind of classification:
  - Partitional
  - Hierarchical
  - Density-based
  - Model-based
  - ...

Dimensionality reduction

- The number of observable features (n) can be too large (high-dimensionality problems)
- The *curse of dimensionality* appears
- Traditionally, the number of features is reduced globally
- There are two main approaches:
  - Feature extraction
  - Feature subset selection (FSS)

Constraints in unsupervised problems

- Some additional information may be available in unsupervised problems
- This information can be used not only for validating, but also for building the model
- There are different types of constraints:
  - Number of clusters
  - Size clusters restrictions
  - Pairwise contraints (must-link and cannot-link)
  - Partially labelled data

Partially labelled data set

- Two subsets of instances: $\mathcal{D} = \{\mathcal{X}_l, \mathcal{X}_u\}$, where:
  - $\mathcal{X}_l$ is the labelled subset (like a supervised problem)
  - $\mathcal{X}_u$ is the unlabelled subset (like an unsupervised problem)

$\mathcal{D}$	X_1	$\dots$	X_n	C
$\mathbf{x}_1$	$x_{1,1}$	$\dots$	$x_{1,n}$	c_1
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_L$	$x_{L,1}$	$\dots$	$x_{L,n}$	c_L
$\mathbf{x}_{L+1}$	$x_{L+1,1}$	$\dots$	$x_{L+1,n}$	?
$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
$\mathbf{x}_N$	$x_{N,1}$	$\dots$	$x_{N,n}$	?

Assumptions about partially labelled data set

- $\mathbf{x}_i = \{x_{i1}, \ldots, x_{in}\} \in \mathbb{R}^n$ (continuous features)
- $c \geq 2$ (multiclass data set is desirable)
- The final number of groups will be K, with $K \geq c$
- $C_s \in (1, \ldots, c, \ldots, K)$, if $\mathbf{x}_s \in X_u$
- Each group could be defined in a different subspace of features

Framework description

- The aim is to obtain a classification with the option of discovering new labels
  - This is a kind of semisupervised classification
- Each of the final labels could be described by a different subset of features
  - This is called subspace classification
- The final number of labels is suggested by the framework

Framework process

Phase 1. Translating knowledge

- To translate the instance-level knowledge (known labels) into feature-level knowledge (subspaces of features)
- To cluster all the instances in the identified subspaces

- If some instance is not grouped in the subspaces, then new groups must be found

Phase 2. Discovering new knowledge

- To find new subspaces that define new groups
- To cluster all the unclassified instances in the new groups

Partitional framework

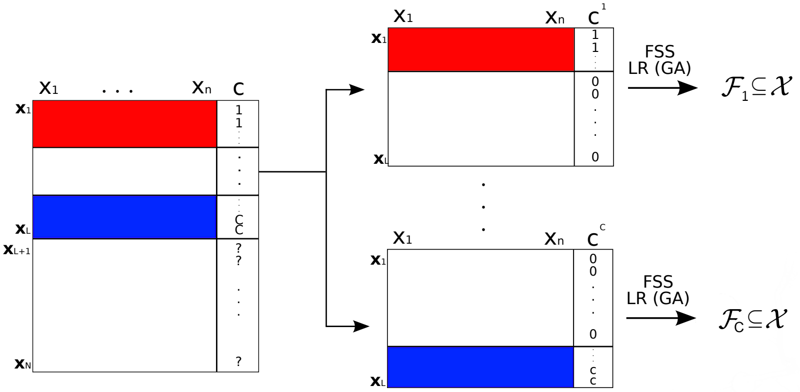
- The framework is developed using supervised techniques, whenever possible...
- ...and using unsupervised partitional-based approaches, when necessary
- The proposed solution was built using standard algorithms in order to check framework viability

Phase 1: Translating knowledge

Find subspaces that describe the known groups

- Only the subset $\mathcal{X}_l$ is used
- Separating the instances that belong to each known label from the instances that belong to the remaining known labels in each case
- c new smaller problems of binary supervised classification
- The subspaces of features are outputted using wrapper FSS (logistic regression + genetic algorithm)

Find subspaces that describe the known groups

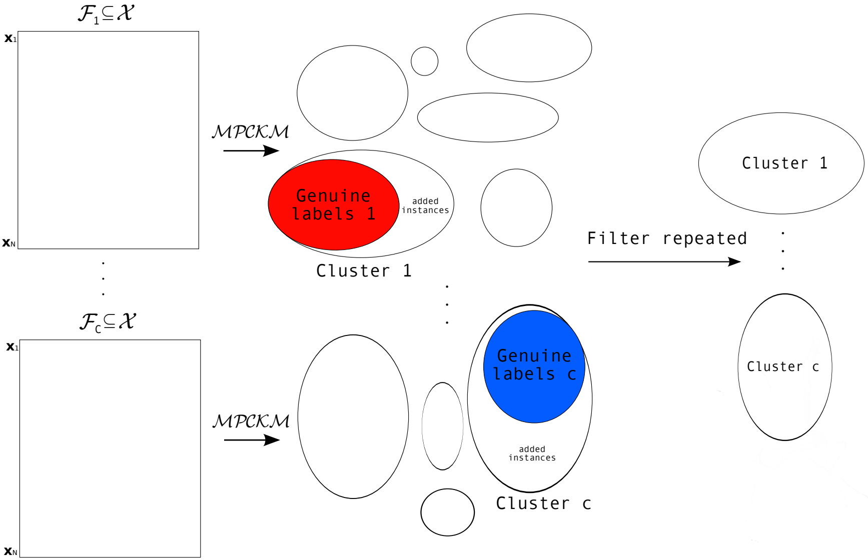


Phase 1: Translating knowledge

Clustering all the instances using the outputted subspaces

- All instances in $\mathcal{D}$ are used
- Clustering all the instances in each found subspace
- The aim is to find c *genuine clusters* (one cluster in each subspace)
- Any $\mathbf{x} \in \mathcal{X}_u$ can be grouped in the *genuine clusters*
- Metric pairwise constrained k-means (MPCKM) is used trying to satisfy the known constraints

Clustering all the instances using the outputted subspaces



Refining the genuine clusters

- An instance could belong to more than one genuine cluster
- It is not possible in a hard partitioning solution (each instance can belong to only one group)

Refining the genuine clusters

- An instance could belong to more than one genuine cluster
- It is not possible in a hard partitioning solution (each instance can belong to only one group)

Eliminating repeated instances

- If $\mathbf{x}_i$, with $i \in \{1, \ldots, L\}$ and $c_i = z$, belongs, among others, to the genuine cluster where the majority of instances with z label are, then the instance x_i will be deleted from the other clusters
- If $\mathbf{x}_i$, with $i \in \{L+1, \ldots, N\}$ belongs to more than one genuine cluster, then the instance will remain in the group in which its distance to the centroid was the minimum after normalization

Phase 2. Discovering new knowledge

Find a new subspace that describes the remaining instances

- The subset of instances grouped in the genuine clusters is $\mathcal{T}$ whereas the subset of instances that do not belong to any group is $\mathcal{R}$, with $\mathcal{T} \cup \mathcal{R} = \mathcal{D}$ and $\mathcal{T} \cap \mathcal{R} = \emptyset$
- A new subspace of features that distinguishes between $\mathcal{T}$ and $\mathcal{R}$ can be identified
- This subspace is found according to the same process explained in the previous phase

Phase 2. Discovering new knowledge

Clustering the remaining instances

- Using instances from R only and the last identified subspace of features
- Hierarchical clustering, for readily observing the distances between clusters
- The challenging task is to select the number of clusters in the hierarchy
- Internal clustering validation indices
  - Very dependent on data
  - A parallel research was done using outliers and noise dimensions

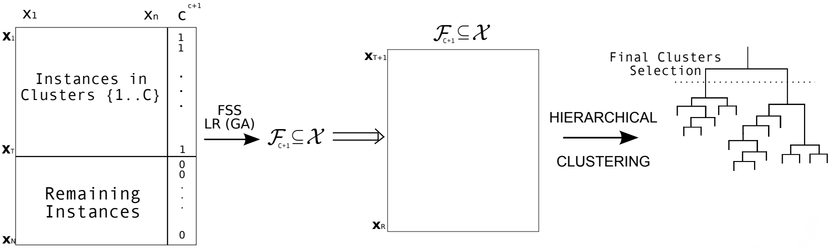
Phase 2. Discovering new knowledge

- Based on our experience and in the commented research

Voting scheme to select the number of clusters

- Gamma, Calinski, Silhouette and Davies-Bouldin indices
- Voting scheme to select the number of clusters
- The indices are ranked based on our study

Phase 2. Discovering new knowledge



Output of the partitional approach

- Each instance belongs to one, and only one, cluster (hard partition)
- K clusters, with $K = c + k$
- c is the number of a priori known classes
  - Each cluster $\in C$ is described by a different subspace of features
- k is the number of clusters found in the last phase
  - All k clusters in another subspace of features

Validation process

- Real data sets from UCI and synthetic data sets generated in subspaces
- All instances are labelled but only a percentage of labels is used in each case to build the models
- 1000 executions with 10 %, 20 %, 30 % and 40 % of randomly labelled instances. Some of the class labels are completely unknown
- It is assumed that original class labels match natural clusters
- Seven external validation measures used
- Results are compared using a Wilcoxon signed-rank test

Summary of results (I)

Accuracy of classification

- Similar results when compared with MPCKM in real data sets although the real data sets do not have labels separated in different subspaces
- In synthetic data sets, results are dependent on the number of features:
  - With 15 features, MPCKM obtained better results in the majority of indices
  - With 25 features, the framework outperformed MPCKM in all the indices
  - With 50 features, results also depend on the percentage of a priori knowledge
    - 10 % and 20 % of labelled instances, MPCKM obtained better results in the majority of indices
    - 30 % and 40 % of labelled instances, the framework outperformed MPCKM in all the indices

Summary of results (II)

Number of clusters

- The proposed framework selected the number of clusters more accurately than MPCKM in all data sets
- The higher the data dimensionality, the better the number of clusters is approximated

Possible improvements

- Different supervised algorithms and FSS techniques could improve the subspace selection
- The goodness during FSS should be taken into account
- The number of clusters associated to each label could be different to 1
- The last clusters could be also defined in distinct subspaces of features
- Different unsupervised algorithms could improve the last clustering
- There are many methods to select the number of clusters

Motivation

- Once the idea was validated using the partitional approach, it must be improved
- Probabilist approach allows to integrate all the steps using a mixture of distributions
- The distributions are assumed to be Gaussians
- There are not model-based clustering works using:
  - A priori knowledge
  - Clusters in different subspaces
  - Automatically selection of the number of clusters

Notation

From now on, $i = \{1, \ldots, L, L+1, \ldots, N\}$, $j = \{1, \ldots, F\}$ and $m = \{1, \ldots, c, \ldots, K\}$ are indices for instances, features and componentes, respectively

Introduction to Gaussian mixtures

- Each component of the mixture is assumed to be a cluster
- It is a soft clustering approach where each instance is assumed to be generated according to several probability distributions shaping a mixture model
- The mixture probability density is:

$$p(\mathbf{x} \mid \theta) = \sum_{m=1}^{K} \pi_m p(\mathbf{x} \mid \theta_m), \tag{1}$$

where θ_m is the set of parameters and π_m the mixing probability of the component m, with $\pi_m \geq 0$ and $\sum_m \pi_m = 1$.

Introduction to Gaussian mixtures

- The aim of this approach is to estimate the full parameter set
- An important parameter estimation method is maximum likelihood, which in a logarithm form is

$$lL(\theta) = \sum_{i=1}^{N} \ln p(\mathbf{x}_i \mid \theta) \tag{2}$$

- The EM algorithm iteratively approximates the ML estimation

Introduction to Gaussian mixtures

- If $\mathbf{x}_i = \{x_{i1}, \ldots, x_{in}\}$, x_{ij} represents each observable feature of an instance
- We assume the existence of unobserved data items, α_i
- In this case, $\alpha_i = \{\alpha_{i1}, \ldots, \alpha_{iK}\}$, (in crisp classification $\alpha_{im} = 1$ if instance i belongs only to the component m)
- Introducing this missing data into the data log-likelihood

$$lL(\theta) = \sum_{i=1}^{N} \ln \sum_{m=1}^{K} \alpha_{im} \left(\pi_m p(\mathbf{x}_i \mid \theta_m)\right) \tag{3}$$

Adapting Gaussian mixtures to our problem

- Separating our data set in $\mathcal{X}_l$ and $\mathcal{X}_u$ taking into account the existence of labels and the knowledge about the components for $\mathcal{X}_l$
- Equation 3 can be rewritten as

$$lL(\theta) = \sum_{i=1}^{L} ln \sum_{m=1}^{C} w_{im} \left(\pi_m p(\mathbf{x}_i \mid \theta_m)\right) + \sum_{i=L+1}^{N} ln \sum_{m=1}^{K} \alpha_{im} \left(\pi_m p(\mathbf{x}_i \mid \theta_m)\right), \tag{4}$$

where w_{im} is the knowledge about the labels, being $w_{im} = 1$ if instance i belongs to component m. Note the difference between the c known components for $\mathcal{X}_l$ and the K components (including the previous c) for $\mathcal{X}_u$

Instances and features independence

- Assuming that features and instances are conditionally independent given the component label, Equation 1 can be rewritten as

$$p(\mathbf{x} \mid \theta) = \prod_{i=1}^{N} \sum_{m=1}^{K} \pi_m \prod_{j=1}^{F} p(x_{ij} \mid \theta_m), \tag{5}$$

being F the total number of features

Subspace FSS in Gaussian mixtures

- A feature will be irrelevant for a component if the distribution of the feature is independent of the component
- A relevant feature for a component will use a specific component distribution
- An irrelevant feature for a component will use a shared distribution

$$p(\mathbf{x} \mid \theta) = \prod 1_{i=1}^{N} \sum_{m=1}^{K} \pi_m \prod_{j=1}^{F} [p(x_{ij} \mid \theta_m)^{v_{mj}} p(x_{ij} \mid \theta_s)^{(1-v_{mj})}], \quad (6)$$

where $v_{mj} = 1$ if feature j is relevant for component m and 0 in other case

Number of components selection

- We use a bottom-up components selection
- Starting from the number of known components, one new component (in a new subspace) is sought in each iteration
- Two models are built in iteration t:
  - $Model_t\ a$. It finds a new component in a subspace of features
  - $Model_t\ b$. It uses the known components and a hodgepodge in the complete space of features
- If $Model_t\ a$ is better than $Model_t\ b$, a new component is added to the set of known components
- This converges when
  - $Model_t\ b$ is better than $Model_t\ a$
  - $Model_t\ a$ is better than $Model_{t+1}\ a$ (penalized comparison)

Complete data log-likelihood

- Completing the log-likelihood with the subspace FSS and introducing the unobserved items
- And separating between the log-likelihood associated with the classification (lL_1) and the associated with the discovery of new knowledge (lL_{2a} for $model\ a$ and lL_{2b} for $Model\ b$)
- The complete data log-likelihood is

$$logL(\theta) = lL_1 + lL_2, \tag{7}$$

Classification data log-likelihood

Classification

$$lL_1 = \sum_{i=1}^{L} \sum_{m=1}^{C} (w_{im}(\ln \pi_m + \sum_{j=1}^{F} [v_{mj} \ln p(x_{ij} \mid \theta_m) + \\ + (1 - v_{mj}) \ln p(x_{ij} \mid \theta_s)])), \quad (8)$$

Discovery new knowledge data log-likelihood

Classification a

$$lL_{2a} = \sum_{i=L+1}^{N} \sum_{m=1}^{C+1} (\alpha_{im} (\ln \pi_m +$$

$$+ \sum_{j=1}^{F} [v_{mj} \ln p(x_{ij} \mid \theta_m) + (1 - v_{mj}) \ln p(x_{ij} \mid \theta_s)])), \qquad (9)$$

Classification b

$$lL_{2b} = \sum_{i=L+1}^{N} (\sum_{m=1}^{C} (\alpha_{im} (\ln \pi_m + \sum_{j=1}^{F} (v_{mj} \ln p(x_{ij} \mid \theta_m) +$$

$$(1 - v_{mj}) \ln p(x_{ij} \mid \theta_s)))) + \alpha_{im} (\ln \pi_{c+1} + \sum_{j=1}^{F} \ln p(x_{ij} \mid \theta_{c+1})) \qquad (10)$$

EM algorithm

E-step

- The expected value of the hidden variable $E[\alpha_{ij} \mid \theta t] = \gamma(\alpha_{ij})$ given the current parameter estimate in iteration t is calculated. This was previously expressed as α_{im}

M-step

- Parameters are reestimated for maximizing the log-likelihood. The updated parameters are obtained by computing the partial derivatives of the complete log-likelihood described above with respect to the different parameters and equaling to zero

Probabilistic framework research

The study will consist of different comparisons between the explained models and:

- Selecting the number of components using traditional techniques top-down
- Iterating first the classification step and obtaining the subspaces of the known components before discovering new components
- Different initializations on known labels
- Letting w_{im} as a (almost) free parameter
- Soft knowledge
- Using different distributions of probability (not only Gaussians)
- Continuous and discrete features mixed

Neuronal data applications

- The aim of the framework is to solve the neurons classification problem
- Specifically, the framework emerges due to the interneurons data characteristics
- In any case, the proposal can be applied to any partially labelled data set
- Other possible neuroinformatics applications are:
  - Spines classification
  - Pyramidal neurons classification

Interneurons data

- The data set belongs to Columbia University (R. Yuste's laboratory)
- 220 interneurons characterized by 67 morphological continuous features
- 105 unlabelled instances + 115 labelled instances
- 5 known classes
- Possibility of new classes in the unlabelled interneurons
- Expert validation

Results in interneurons data

- Different executions of the partitional framework obtaining:
  - Labelled interneurons separated into different groups integrating some new cells
  - Identification of many labels in unlabelled instances
    - The original data set had 37 % of labelled instances. The current data set has 52 % (from 85 to 115 instances)
  - Detection of several outliers
    - An outlier was a bad reconstructed neuron
  - All validations are based in expert knowledge

Conclusions and contributions

- The framework covers an interesting gap in classification
- Partitional approach, although being a standard solution, outputted promising results
- All the characteristics of the framework will be better integrated in the probabilistic approach
- Multiple studies arise from both approaches
- Probabilistic approach results are expected to be better than partitional approach results
- Thus, interneurons data classification using probabilistic approach could be a great advance for the community

Partially labelled data: classification and discovery of unknown labels

using subspaces of features

Luis Guerra

l.guerra@upm.es

CIG – Computational Intelligence Group

February, 2011