Available online at www.sciencedirect.com
ScienceDirect
Pattern Recognition Letters 28 (2007) 2375–2384
Pattern Recognition Letters
www.elsevier.com/locate/patrec

Learning Bayesian classifiers from positive and unlabeled examples

Borja Calvo *, Pedro Larrañaga, José A. Lozano

Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizabal, 1 20018 Donostia-San Sebastián, Spain

Received 1 March 2007; received in revised form 27 June 2007
Available online 15 August 2007

Communicated by W. Pedrycz

Abstract

The positive unlabeled learning term refers to the binary classification problem in the absence of negative examples. When only positive and unlabeled instances are available, semi-supervised classification algorithms cannot be directly applied, and thus new algorithms are required. One of these positive unlabeled learning algorithms is the positive naive Bayes (PNB), which is an adaptation of the naive Bayes induction algorithm that does not require negative instances. In this work we propose two ways of enhancing this algorithm. On one hand, we have taken the concept behind PNB one step further, proposing a procedure to build more complex Bayesian classifiers in the absence of negative instances. We present a new algorithm (named positive tree augmented naive Bayes, PTAN) to obtain tree augmented naive Bayes models in the positive unlabeled domain. On the other hand, we propose a new Bayesian approach to deal with the a priori probability of the positive class that models the uncertainty over this parameter by means of a Beta distribution. This approach is applied to both PNB and PTAN, resulting in two new algorithms. The four algorithms are empirically compared in positive unlabeled learning problems based on real and synthetic databases. The results obtained in these comparisons suggest that, when the predicting variables are not conditionally independent given the class, the extension of PNB to more complex networks increases the classification performance. They also show that our Bayesian approach to the a priori probability of the positive class can improve the results obtained by PNB and PTAN.

Keywords: Positive unlabeled learning; Bayesian classifiers; Naive Bayes; Tree augmented naive Bayes; Bayesian approach

1. Introduction

In this paper we are interested in a particular case of classification problem, known in the literature as positive unlabeled learning (Denis et al., 2002) or partially supervised classification (Liu et al., 2002). This is an interesting problem that arises in the binary classification context when examples from one of the classes are not available.

Classically, the classification problems (Bishop, 2006; Duda et al., 2001) are divided into two groups: supervised and unsupervised. In the supervised classification framework, all the examples we have are labeled, while in the unsupervised context all the examples are unlabeled. Semi-supervised classification can be considered as an intermediate situation where both labeled and unlabeled examples are available.

In semi-supervised classification, examples from all the classes are needed. Nevertheless, in many real-life problems getting examples from one or more classes is either difficult or impossible (Calvo et al., 2007), while unlabeled examples are readily available. The lack of examples from some of the classes makes it unfeasible to directly apply the methodologies developed for semi-supervised classification problems. In positive unlabeled learning we have this problem, as the class can take two values (normally named positive and negative), but no negative examples are available.

In the text classification domain, solving positive unlabeled learning problems is interesting as it allows, for

* Corresponding author. Fax: +34 943015590.
E-mail addresses: borja.calvo@ehu.es (B. Calvo), pedro.larranaga@ehu.es (P. Larrañaga), ja.lozano@ehu.es (J.A. Lozano).

0167-8655/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.patrec.2007.08.003

instance, to retrieve, from a set of unlabeled documents, those related to a set of interesting texts without the tedious task of hand-labeling uninteresting documents (Denis et al., 2003; Li and Liu, 2003). Text classification is not the only domain where positive unlabeled learning problems can be found. In Calvo et al. (2007) and Wang et al. (2006), two computational biology problems are modeled as positive unlabeled learning. In these two examples, getting negative instances is not just difficult, but impossible.

New algorithms have been developed to cope with positive unlabeled learning. In Liu et al. (2002) the authors propose a procedure named spy-EM, based on a modification of the EM algorithm (Dempster et al., 1977). An adaptation of the naive Bayes induction algorithm (named positive naive Bayes, PNB) to partially supervised classification is proposed in Denis et al. (2002). In Section 2 more details about this algorithm will be given. (Calvo et al., 2007) present the application of a new model averaging algorithm, named DCDiv, to the prediction of genes associated with human genetic diseases. In Wang et al. (2006) an algorithm known as Positive Sample only Learning (PSoL) is used to search for non-coding RNA functional genes. This algorithm tries to obtain a set of negative examples by selecting, from the set of unlabeled cases, those most distant from the set of positive examples. Several other approaches based on support vector machines can be consulted in Li and Liu (2003), Liu et al. (2003), Yu et al. (2003).

Another interesting concept that is close to the positive unlabeled learning is the one-class classification (Tax, 2001; Tax and Duin, 2002). The one-class algorithms try to obtain, from only positive (target) examples, a classifier that is able to distinguish between target and non-target (outlier) instances. The main difference with the positive unlabeled learning methods is that one-class classification algorithms do not use unlabeled examples.

The PNB algorithm (Denis et al., 2002) is able to estimate the parameters of a naive Bayes model from positive and unlabeled data. In this paper we propose an extension of this idea that can be used to build more complex Bayesian classifiers such as tree augmented naive Bayes models (Friedman et al., 1997), where the dependencies between variables are represented as a tree structure or k Dependence Bayesian Classifier (kDB, Sahami, 1996), where each predicting variable can depend on up to k predicting variables (besides the class).

PNB and any other extension to more complex networks require the a priori probability of the positive class. As this probability is generally unknown and cannot be estimated from the data, it must be set by the user. In this work, we propose a new Bayesian approach to handle this a priori probability that models the uncertainty about this parameter by means of a probability distribution.

The rest of the paper is organised as follows. In Section 2 an adaptation of PNB to the case of general discrete variables is presented. Section 3 is devoted to the extension of PNB to more complex networks, and presents the PTAN algorithm. In Section 4 the averaged version of these positive Bayesian classifiers (APNB and APTAN) is introduced. The empirical comparison between the three new classifiers proposed and PNB is shown in Section 5. Finally, in Section 6 some conclusions and ideas about future developments of this work are given.

2. Positive naive Bayes

Before starting the description of the algorithm, some basic notation used throughout this paper will be introduced. Variables are represented with capital letters and their values with lower-case letters; vectors are represented with bold fonts. In each classification problem we have a predicting vector $\boldsymbol{X} = \{X_1, \ldots, X_n\}$ of discrete random variables and a discrete random class variable C. Each predicting variable X_i can take values from *1* to r_i, and the class variable C can take two values, *1* for the positive instances and *0* for the negative instances. The dataset from where the classifiers are induced is denoted as $\mathscr{E}$, and consists of a set of positive instances ($\mathscr{E}_1$) and a set of unlabeled instances ($\mathscr{E}_u$), $\mathscr{E} = \mathscr{E}_1 \cup \mathscr{E}_u$.

The positive naive Bayes (Denis et al., 2002) is an adaptation of the naive Bayes induction algorithm (Minsky, 1961) to the positive unlabeled learning context. This algorithm estimates the parameters of a naive Bayes model from positive and unlabeled examples. In the original paper, instances (text documents) were represented as a bag of words, and thus the equations in Denis et al. (2002) are adapted to this particular way of representing the examples. We have generalised the algorithm to handle general discrete variables. The rest of the section shows this generalisation in detail.

Given the Bayes rule, and under the assumption of conditional independence between all the predicting variables given the class, we have that, for a given instance x

$$P(C = c|\boldsymbol{X} = \boldsymbol{x}) \propto P(C = c) \prod_{i=1}^{n} P(X_i = x_i|c)$$

The parameters required to define a two-class naive Bayes model are $P(C = 1)$, $P(X_i = j|C = 1)$ and $P(X_i = j|C = 0)$ for all $i = 1, \ldots, n$ and $j = 1, \ldots, r_i - 1$ (for the sake of simplicity, from now on the previous probabilities will be denoted as p, $P(x_{ij}|1)$ and $P(x_{ij}|0)$). In the classical naive Bayes algorithm these parameters are estimated from the data by maximum likelihood estimators, but in the positive unlabeled learning context the absence of negative examples makes it unfeasible to estimate $P(x_{ij}|0)$ and p from the data. However, if we take into account that

$$P(x_{ij}|0) = \frac{P(x_{ij}) - P(x_{ij}|1)p}{1 - p} \tag{1}$$

where $P(x_{ij})$ stands for $P(X_i = j)$, we can estimate $P(x_{ij}|0)$ based on p as

$$\frac{N_{ij\mathscr{E}_u} - P(x_{ij}|1)pN_{\mathscr{E}_u}}{(1 - p)N_{\mathscr{E}_u}} \tag{2}$$

being $N_{ij\mathscr{E}_u}$ the amount of unlabeled instances where $X_i = j$ and $N_{\mathscr{E}_u}$ the cardinality of the set of unlabeled examples. The problem with this estimator is that it can be negative. We solve this problem by replacing the negative estimations by 0, and then normalising all the probabilities such that, for each predicting variable X_i, $\sum_{j=1}^{r_i} P(x_{ij}|0) = 1$. Taking the normalisation and the Laplace correction (a smoothing correction that shifts the estimated probabilities toward the a priori probability of the class) into account, we can estimate $P(x_{ij}|0)$ as

$$P(x_{ij}|0) = \frac{1 + \max(0; R_i(j))\frac{1}{Z_i}}{r_i + (1-p)N_{\mathscr{E}_u}} \tag{3}$$

where $R_i(j) = N_{ij\mathscr{E}_u} - P(x_{ij}|1)pN_{\mathscr{E}_u}$ and the normalisation factor $Z_i = \sum_{j=1}^{r_i} \max(0; P(x_{ij}|0))$ (estimated using Eq. (2)). p cannot be estimated from the data and, therefore, the user must introduce it as a parameter.

To sum up, our generalisation of the PNB estimates $P(x_{ij}|1)$ from the positive examples by means of a maximum likelihood estimator with Laplace correction, p is a parameter set by the user, and $P(x_{ij}|0)$ is estimated by means of Eq. (3).

3. Extending PNB to more complex networks

Within the Bayesian classifiers, naive Bayes has the simplest structure, as it assumes conditional independence between all the predicting variables given the class. More complex classification paradigms arise when dependencies between variables are allowed. In the tree augmented naive Bayes model (TAN, Friedman et al., 1997) a tree structure of conditional dependencies is constructed. This tree structure is built, in a step known as structural learning, by an algorithm based on the conditional mutual information between pairs of predicting variables. Once the structure is induced, the parameters of the model are obtained by maximum likelihood estimators in the parametric learning step.

We have adapted the conditional mutual information estimation to the positive unlabeled context, and we have also generalised the equations proposed in the previous section to the case when the predicting variables can have more parents than just the class. In the following subsections we will show how these modifications can be used to induce TAN models from positive and unlabeled examples. The same concepts shown here can easily be applied to more general paradigms such as kDB (Sahami, 1996).

3.1. Structural learning

Friedman's TAN structural learning algorithm is based on the information theory and it computes the mutual information conditioned to the class variable between every pair of predicting variables. The conditional mutual information between two variables X_i and X_k given a binary class C (assuming they are all discrete) is defined as

$$\sum_{j=1}^{r_i}\sum_{l=1}^{r_k} P(x_{ij}, x_{kl}, 1)\log\frac{P(x_{ij}, x_{kl}|1)}{P(x_{ij}|1)P(x_{kl}|1)} + \sum_{j=1}^{r_i}\sum_{l=1}^{r_k} P(x_{ij}, x_{kl}, 0)\log\frac{P(x_{ij}, x_{kl}|0)}{P(x_{ij}|0)P(x_{kl}|0)} \tag{4}$$

where $P(x_{ij}, x_{kl}, 1)$ and $P(x_{ij}, x_{kl}|1)$ stand for $P(X_i = j, X_k = l, C = 1)$ and $P(X_i = j, X_k = l|C = 1)$, respectively, and the same when $C = 0$. As happens in the PNB algorithm, all the conditional probabilities when $C = 1$ can easily be estimated from the known positive examples, but neither those related to the negative class nor $P(x_{ij}, x_{kl}, 1)$ can be obtained from the data. Regarding $P(x_{ij}, x_{kl}, 1)$, it can be substituted by $P(x_{ij}, x_{kl}|1)p$. If we take into account that

$$P(x_{ij}, x_{kl}, 0) = P(x_{ij}, x_{kl}) - P(x_{ij}, x_{kl}, 1)$$

where $P(x_{ij}, x_{kl})$ represents $P(X_i = j, X_k = l)$, the conditional mutual information can be computed as

$$\sum_{j=1}^{r_i}\sum_{l=1}^{r_k} pP(x_{ij}, x_{kl}|1)\log\frac{P(x_{ij}, x_{kl}|1)}{P(x_{ij}|1)P(x_{kl}|1)} + \sum_{j=1}^{r_i}\sum_{l=1}^{r_k}(P(x_{ij}, x_{kl}) - P(x_{ij}, x_{kl}|1)p)\log\frac{P(x_{ij}, x_{kl}|0)}{P(x_{ij}|0)P(x_{kl}|0)} \tag{5}$$

where $P(x_{ij}, x_{kl})$ can be obtained from $\mathscr{E}_u$ by means of a maximum likelihood estimator. In order to compute the previous equation, we need to estimate $P(x_{ij}, x_{kl}|0)$, $P(x_{ij}|0)$, $P(x_{kl}|0)$ and p. The latter, as in the PNB algorithm, cannot be estimated from the data and thus, it is a parameter the user must set. $P(x_{ij}|0)$ and $P(x_{kl}|0)$ can be estimated by means of Eq. (3) and, following the same reasoning as in Section 2, we can see that

$$P(x_{ij}, x_{kl}|0) = \frac{1 + \max(0; N_{ijkl\mathscr{E}_u} - P(x_{ij}, x_{kl}|1)pN_{\mathscr{E}_u})\frac{1}{Z_{ik}}}{r_i r_k + (1-p)N_{\mathscr{E}_u}}$$

where

$$Z_{ik} = \sum_{j=1}^{r_i}\sum_{l=1}^{r_k}\frac{\max(0; N_{ijkl\mathscr{E}_u} - P(x_{ij}, x_{kl}|1)pN_{\mathscr{E}_u})}{(1-p)N_{\mathscr{E}_u}}$$

Being $N_{ijkl\mathscr{E}_u}$ the number of unlabeled cases where $X_i = j$ and $X_k = l$. All the probabilities related to the positive class can be obtained by maximum likelihood estimators from $\mathscr{E}_1$.

Using Eq. (5) and the estimators proposed for the probabilities conditioned to the negative class, we can compute the conditional mutual information without negative examples and thus, we can apply Friedman's structural learning algorithm in the positive unlabeled learning context.

3.2. Parametric learning

Once we have the structure of a Bayesian network (Pearl, 1988), we need to learn the parameters of the model.

These parameters are $P(x_{ij}|\boldsymbol{pa_i})\ \forall i,j,\boldsymbol{pa_i}$, where $P(x_{ij}|\boldsymbol{pa_i})$ represents the probability that $X_i = j$ given that $\boldsymbol{Pa}(\boldsymbol{X}_i) = \boldsymbol{pa_i}$, being $\boldsymbol{Pa}(\boldsymbol{X}_i)$ the set of parents of X_i defined by the network structure.

In Bayesian networks built with classification purpose (such as those obtained by TAN or kDB algorithms) all the predicting variables have the class variable as parent. Thus, the parameters of the model can be divided into two different subsets: $P(x_{ij}|C = 1, \boldsymbol{Pa}^*(\boldsymbol{X}_i) = \boldsymbol{pa}_i^*)$ and $P(x_{ij}|C = 0, \boldsymbol{Pa}^*(\boldsymbol{X}_i) = \boldsymbol{pa}_i^*)$ (from now on $P(x_{ij}|1, \boldsymbol{pa}_i^*)$ and $P(x_{ij}|0, \boldsymbol{pa}_i^*)$, respectively) being $\boldsymbol{Pa}^*(\boldsymbol{X}_i) = \boldsymbol{Pa}(\boldsymbol{X}_i)\backslash C$. In the rest of the section, we will show how these parameters can be estimated from positive and unlabeled examples.

As in PNB, $P(x_{ij}|1, \boldsymbol{pa}_i^*)$ can be estimated from the positive cases, but those related to the negative class cannot be obtained from the data. Following the same reasoning as in PNB, we can obtain $P(x_{ij}|0, \boldsymbol{pa}_i^*)$ as

$$P(x_{ij}|0, \boldsymbol{pa}_i^*) = \frac{1 + \max(0; R_i^*(j))\frac{1}{Z_i^*}}{r_i + (1-p)N_{\boldsymbol{pa}_i^* \mathscr{E}_u}}$$

where

$$Z_i^* = \sum_{j=1}^{r_i} \frac{\max(0; R_i^*(j))}{(1-p)N_{\boldsymbol{pa}_j^* \mathscr{E}_u}}$$

$$R_i^*(j) = N_{ij\boldsymbol{pa}_i^* \mathscr{E}_u} - P(x_{ij}|1, \boldsymbol{pa}_i^*)pN_{\boldsymbol{pa}_i^* \mathscr{E}_u}$$

and $N_{ij\boldsymbol{pa}_i^* \mathscr{E}_u}$ is the number of unlabeled cases where $X_i = j$ and $\boldsymbol{Pa}^*(\boldsymbol{X}_i) = \boldsymbol{pa}_i^*$ and $N_{\boldsymbol{pa}_i^* \mathscr{E}_u}$ is the number of unlabeled cases where $\boldsymbol{Pa}^*(\boldsymbol{X}_i) = \boldsymbol{pa}_i^*$.

This generalisation can be used to learn the parameters of a TAN structure, or any other structure where the class variable is parent of all the predicting variables.

Taking into account all the estimators proposed in this section, we can use Friedman's algorithm (Friedman et al., 1997) to learn TAN models from positive and unlabeled examples. We have named this new algorithm Positive TAN (PTAN).

4. Averaging positive Bayesian classifiers

In the previous section, we have seen how we can generalise PNB to more complex Bayesian classifiers. In the learning process, all the estimations of the probabilities conditioned to the negative class require, as a parameter, the a priori probability of the positive class (p). This is the main drawback of PNB (and of any extension to other Bayesian classifiers), as this probability is generally unknown in real-life problems. Nevertheless, in certain situations some information about this parameter may be available. For instance, we may know that the number of negative cases hidden in $\mathscr{E}_u$ is greater than the number of positive cases, and thus we can suppose that with high probability $p < 0.5$. The only way to introduce the prior knowledge about the parameter in PNB or PTAN is by setting it at a given value. For instance, if we know that $P(p < 0.5)$ is close to one, we could set p at 0.25.

In this section, we propose a new Bayesian approach to integrate our knowledge in the final classifier. This approach models the uncertainty about p by means of a probability distribution. In the previous example, where we know that the probability of $p < 0.5$ is very high, we can suppose that p follows a probability distribution like the one plotted in Fig. 1. Modeling the uncertainty about p by means of a probability distribution allows us to consider in our estimations all the possible values that this parameter can take. We can apply this idea to the estimation of the probabilities associated with the negative class, integrating the estimators over all the possible values of p.

C is a random variable that follows a Bernouilli distribution of parameter p. Thus, if we sample this variable n times, the probability of this sample $c_1, \ldots, c_n$ given the parameter p follows a Binomial distribution. As the conjugate to the Binomial is the Beta, we have selected this distribution to model the a priori probability of this parameter ($P(p) \sim \text{Beta}(\alpha, \beta)$, see definition in Appendix A) (Bernardo and Smith, 1994).

Modeling the uncertainty about p as a Beta distribution makes it possible to solve the integrals and thus, to obtain a closed formula for the averaged estimators. The process of integrating the estimators over all the possible values of p is shown in the Appendix A, taking as an example the estimator in Eq. (1).

After integrating Eq. (1) for all the possible values of p, the averaged estimator for $P(x_{ij}|0)$ is

$$P(x_{ij}|0) = \frac{\alpha(P(x_{ij}) - P(x_{ij}|1)) + (\beta - 1)P(x_{ij})}{\beta - 1} \tag{6}$$

Taking a look at the previous equation, we can see that the estimation can be negative. When this occurs it can be corrected by replacing these negative estimations by a small value and then normalising the probabilities so as their sum

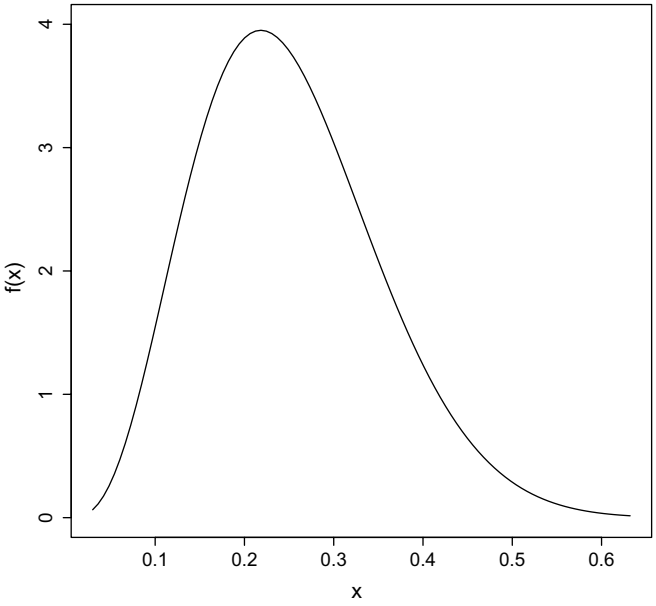

Fig. 1. Beta distribution with parameters $\alpha = 4.4$, $\beta = 13.17$. This parameters result in an expected value of 0.25 and a variance of 0.01.

is equal to one (we have replaced them by $\frac{1}{r_i}$). We cannot replace negative estimations by zero because then we would have null parameters in the model, and this can lead to problems in the estimation of the probability of each class for new instances. We can use the estimator in Eq. (6) in the parametric learning of a naive Bayes model. We have named this new algorithm Averaged PNB (APNB).

In order to extend this Bayesian approach to the previously described PTAN algorithm, we need to average all the estimators proposed in Section 3. Following the same procedure described in Appendix A, we can conclude that $P(x_{ij}, x_{kl}|0)$ can be estimated as

$$\frac{\alpha(P(x_{ij}, x_{kl}) - P(x_{ij}, x_{kl}|1)) + (\beta - 1)P(x_{ij}, x_{kl})}{\beta - 1}$$

and $P(x_{ij}|0, \boldsymbol{pa}_i^*)$ as

$$\frac{\alpha(P(x_{ij}) - P(x_{ij}|1, \boldsymbol{pa}_i^*)) + (\beta - 1)P(x_{ij})}{\beta - 1} \quad (7)$$

As happened with Eq. (6), these estimations can be negative. This is again solved by replacing negative estimations by a small number ($\frac{1}{r_i r_j}$ for $P(x_{ij}, x_{kl}|0)$ and $\frac{1}{r_i}$ for $P(x_{ij}|0, \boldsymbol{pa}_i^*)$).

In Section 3 we saw that $P(x_{ij}, x_{kl}, 1)$ can be expressed as $P(x_{ij}, x_{kl}|1)p$, whose averaged version is $P(x_{ij}, x_{kl}|1)\frac{\alpha}{\alpha+\beta}$.

We can replace all the estimations needed to compute the conditional mutual information by the new averaged versions, use them to obtain a tree structure and then, estimate the parameters by means of the averaged estimator shown in Eq. (7). We have named this new method for obtaining TAN models Averaged PTAN (APTAN).

5. Experimental evaluation

Comparing algorithms in real-life positive unlabeled learning problems is an unsolved issue, as the lack of negative examples makes it unfeasible to estimate performance measures such as accuracy or the F measure (see Fig. 2). An easy way to overcome this problem is by simulating the absence of negative examples. Starting from labeled data or a probability distribution, we can obtain a set of unlabeled instances just by selecting (or sampling, if our source of instances is a probability distribution) both positive and negative examples and then removing their labels. As we know which instances in this set are positive and which are negative, we can estimate any performance measure. Of course, the sets of positive instances are easily obtained just selecting (or sampling) only positive instances.

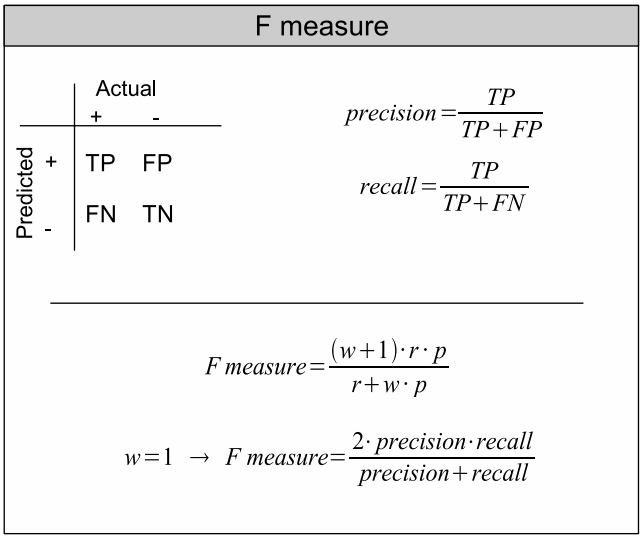

Fig. 2. Definition of the F measure. The general definition includes a weighting factor (w) that can be used to stress either the recall or the precision. In this work we have set w at 1, giving the same importance to both the precision and the recall.

Simulating the absence of negative examples not only allows us to evaluate the classification performance, but also allows us to set the ratio of positive examples hidden in the unlabeled instances at any desired value. This ratio is the value of p for the empirical distribution defined by the dataset.

Bearing this in mind, in order to obtain a dataset (containing both positive and unlabeled examples), we need: a source from where the instances are sampled (a probability distribution or a set of positive and negative examples); the cardinality of the positive cases ($N_{\mathscr{E}_1}$); the cardinality of the unlabeled instances ($N_{\mathscr{E}_u}$) and the ratio of positive examples in the set of unlabeled cases (p). Any positive unlabeled learning problem defined in this way can be instantiated, resulting in a particular dataset.

We have compared our three algorithms and PNB in two kinds of positive unlabeled learning problems: problems sampled from TAN models and problems sampled from real-life completely-labeled databases.

5.1. Synthetic data

PTAN and APTAN extend the PNB and APNB algorithms, respectively, as they are able to consider first order conditional dependencies between variables. Thus, TAN-based models should improve the performance of NB-based models in problems where the dependencies between the predicting variables fit in a tree structure. This has been experimentally tested in synthetic datasets obtained sampling TAN models.

We have built TAN models based on four different topologies (see Fig. 3). As we have mentioned previously, a positive unlabeled problem can be defined by $N_{\mathscr{E}_u}$, $N_{\mathscr{E}_1}$, p and the source from where the data are sampled (a TAN model in this case). We have defined 216 positive unlabeled learning problems, 54 based on each of the four topologies, combining three values for $N_{\mathscr{E}_u}$ (1000, 10,000 and 100,000), three values for $N_{\mathscr{E}_1}$ (100, 1000 and 10,000) and six values for p ($0.01, 0.1, 0.2, 0.3, 0.4$ and 0.5). In this study, we have not considered values greater than 0.5 for the p parameter because typically, in positive unlabeled learning problems, the amount of positive examples in the set of unlabeled cases is smaller than the amount of negative cases.

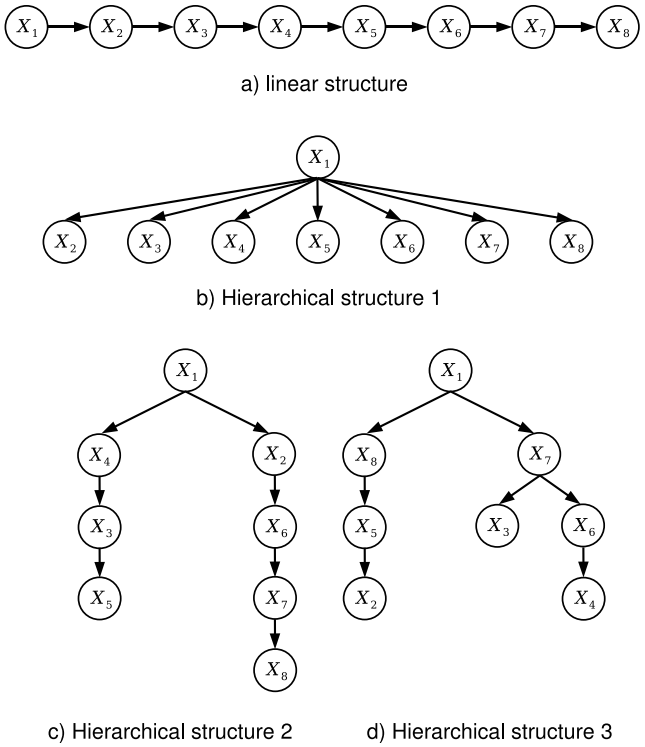

Fig. 3. Dependency structures used to build the TAN models from where datasets have been sampled. The class variable has been removed for clarity.

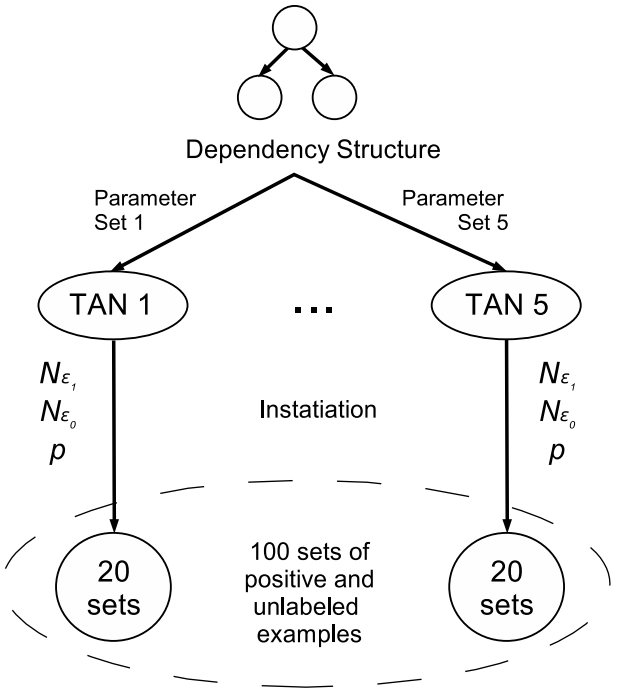

Fig. 4. Process followed to obtain the positive unlabeled learning problems based on TAN models.

In order to increase the variability between datasets, we have obtained, from each topology, five different models (same structure, but different parameters), obtaining 20 datasets from each model. As a result, 100 sets of positive and unlabeled cases have been obtained for each combination of $N_{\mathscr{E}_u}$, $N_{\mathscr{E}_1}$, p and topology. A scheme of the whole process can be seen in Fig. 4.

5.2. Real data

The four algorithms have also been compared in positive unlabeled learning problems built sampling real-life completely-labeled databases. In order to prepare these databases for the sampling process, we have to choose one of the classes as the positive one, and then label all the remaining instances in the database as negative. Then, $\mathscr{E}_1$ is built sampling the set of positive instances in the database, and $\mathscr{E}_u$ is obtained by randomly selecting positive and negative cases from the database (keeping the desired ratio of positive cases p) and then removing their labels.

Three different databases have been used in this process: `ACCDON`, a database of splice sites described in Castelo and Guigó (2004); `Letter Recognition`, a database of hand-written character identification (Blake and Merz, 1998) and `Nursery`, a database of applications for nursery schools (Blake and Merz, 1998).

`ACCDON` is a database of splice sites, which are small fragments of DNA sequence. In this database we can find both acceptor and donor sites. Each instance is a true or false donor or acceptor site, and its variables are the nucleotide at each position of the sequence (a nucleotide can take four values: *a*; *t*; *g* and *c*). Both acceptor and donor sites have a two-base-pair-long constant part that has been removed from the instances, as it does not give any information about the class (true or false site). For each type of splice site we have positive examples and two kinds of negative examples (obtained from coding and non-coding regions). We have built six different sets of positive and negative examples starting from this database: Acceptor Sites Coding, where the negative examples come only from the negatives obtained from the coding regions; Acceptor Sites Intron, where the negative examples are only from non-coding regions; Acceptor Sites Mixed, where the negative examples come from both coding and non-coding regions and the same with donor sites (Donor Sites Coding, Donor Sites Intron and Donor Sites Mixed).

`Letter Recognition` is a dataset of hand-written characters. The dataset contains examples of the 26 roman alphabet letters. We have built three sets of positive and negative examples from this database: Letter Recognition D (examples from the 'D' class are regarded as positive and the rest as negative), Letter Recognition P (the same, but with 'P' as the positive class) and Letter Recognition U (the same with 'U').

`Nursery` is a dataset designed to rank applications for nursery schools. This database has been used to build another pair of sets of positive and negative examples. The positive set contains all the examples from the 'spec_prior' class, and the set of negative cases contains the rest of the examples.

Starting from each of the ten pairs of positive and negative examples, we have built 21 different problems, combining three values of $N_{\mathscr{E}_l}$ (100, 500 and 1000 for sets coming from ACCDON database and 100, 200 and 300 for the rest) and six ratios of positive cases hidden in $\mathscr{E}_u$ (0.01; 0.1; 0.2; 0.3; 0.4 and 0.5). The number of unlabeled examples is constant for each dataset: 10,000 for ACCDON and 5000 for both Letter Recognition and Nursery. Thus, we have 180 positive unlabeled learning problems (defined by the database from where the data are sampled, the number of cases in $\mathscr{E}_l$ and $\mathscr{E}_u$ and the ratio of positive cases in $\mathscr{E}_u$). For each positive unlabeled learning problem, 100 different instances (pairs of $\mathscr{E}_l$ and $\mathscr{E}_u$ datasets) have been obtained.

5.3. Experimentation

The performance of PNB and our proposals (PTAN and the averaged versions, APNB and APTAN) have been compared in the positive unlabeled learning problems described in the previous section.

The algorithms have been compared in terms of both accuracy and the F measure (see Fig. 2). The F measure is a typical performance measure in information retrieval that combines the precision (the ratio of the instances labeled as positive that are actually positive) and the recall (the ratio of positive cases recovered). The F measure gives an idea of the amount of positive cases recovered and the quality of the recovery, and thus the higher the F measure the better the algorithm recovers the positive examples hidden in the set of unlabeled cases. In our comparisons, F measure is preferred because when the amount of positive cases is too low, the accuracy can be misleadingly good when all the cases are labeled as negative.

In both the PNB and the PTAN, p has been set at a value of 0.25. In the averaged versions of the algorithms, the parameters of the Beta distribution have been set at $\alpha = 4.4$ and $\beta = 13.17$, so the expected value of p is 0.25, and the variance is 0.01.

In order to check the significance of the differences between the performance of the algorithms, Wilcoxon signed rank tests have been run. The significance has been tested in both the accuracy and the F measure at a significance level of 1%. The complete set of results, including the mean and the standard deviation of the accuracy and F measure obtained by each algorithm as well as all the p-values of the Wilcoxon tests can be found in the supplementary data.[1]

5.3.1. PNB vs APNB

The parameter estimator used in the APNB algorithm is an averaged alternative to that used in the PNB (Eqs. (6) and (1), respectively). If we take a look at these two estimators, we can see that both are very similar. Indeed, if we rewrite Eq. (6) as

$$P(x_{ij}|2) = \frac{P(x_{ij}) - \frac{\alpha}{\alpha+\beta-1}P(x_{ij}|1)}{\frac{\beta-1}{\alpha+\beta-1}}$$

we can see that Eqs. (1) and (6) are equal when $p = \frac{\alpha}{\alpha+\beta-1}$. The expected value of a Beta distribution is $\frac{\alpha}{\alpha+\beta}$, so the averaged version of the PNB is equivalent to the original one, but setting p at a value slightly greater than its expected value.

We have modeled p as a Beta(4.4, 13.17) and thus, the expected value of p is 0.250. Bearing this in mind, in our experiments, the APNB would be equivalent to a PNB where p is set at 0.266. Thus, we would expect that the APNB algorithm will yield better results with datasets where the actual p is greater than 0.25, while PNB should outperform APNB for those datasets where $p \leqslant 0.25$. This is empirically confirmed in problems based on ACCDON (see Table 1) and Letter Recognition databases (see supplementary data), where PNB improves APNB for datasets where $p \leqslant 0.2$, while APNB outperforms PNB when $p \geqslant 0.3$ (except for some datasets where $p = 0.01$).

Taking a look at the F measure obtained in datasets based on Nursery database (see supplementary data), we can see something similar (when p is small PNB gives better results), but the threshold is not at 0.25 but at a lower level (APNB improves PNB in datasets where p is set at 0.2). On the other hand, in datasets obtained from TAN models, PNB outperforms APNB only in a few datasets where p is set at 0.01. This is reflected in Fig. 5a, where we can see that for ACCDON and Letter Recognition PNB beats APNB in approximately half the datasets, while APNB outperforms PNB in most of the datasets sampled from TAN models. From these results we can conclude that, compared to the original algorithm, our averaged version of the PNB makes a better recovery of the positive instances in most of the datasets.

Regarding the accuracy (see supplementary data), the results are very similar to those obtained for the F measure, except for the datasets sampled from TAN models, where the situation is inverted (PNB outperforms APNB in most of the datasets). This can be explained by the fact that PNB classifies more instances as negative and, given that most of the cases are negative, it improves APNB in accuracy, but not in the F measure (note that this measure gives an idea about the recovery of positive cases).

5.3.2. (A)PNB vs (A)PTAN

In the comparison between (A)PNB and (A)PTAN, the results we are interested in are those where the data have an underlying tree structure. The datasets based on synthetic data have been sampled from TAN models to ensure this point. From the results obtained in these datasets (see Fig. 5b–e), we can conclude that our extensions to TAN models clearly improve the results obtained by the naive

[1] Supplementary data can be found at http://www.sc.ehu.es/ccwbayes/members/borxa/PBC/.

Table 1
Comparison between PNB and APNB in terms of the F measure in datasets based on ACCDON (AccCod stands for acceptor sites coding, AccInt for acceptor sites intron, AccMix for acceptor sites mixed and DonCod, DonInt and DonMix the same, but for donor sites)

		ACCDON-F measure											
		AccCod		AccInt		AccMix		DonCod		DonInt		DonMix	
$N_{\mathscr{E}1}$	P	PNB	APNB	PNB	APNB	PNB	APNB	PNB	APNB	PNB	APNB	PNB	APNB
100	0.01	11.49	**11.90**	**10.42**	10.01	**11.57**	10.99	8.80	**9.38**	**10.15**	8.97	**10.17**	8.86
100	0.1	**67.21**	66.26	**58.85**	58.04	**62.70**	61.71	**66.79**	64.36	**64.16**	62.55	**65.79**	63.94
100	0.2	**83.86**	83.46	**75.33**	75.16	**79.05**	78.83	**84.32**	84.07	**80.98**	80.85	**82.61**	82.42
100	0.3	86.94	**87.18**	79.81	**80.09**	82.83	**83.08**	85.74	**86.24**	**83.31**	83.80	84.41	**84.93**
100	0.4	84.38	**84.97**	78.17	**78.77**	80.86	**81.45**	81.21	**82.11**	79.22	**80.03**	79.91	**80.76**
100	0.5	78.55	**79.35**	72.61	**73.46**	74.93	**75.80**	71.66	**72.85**	70.12	**71.22**	70.65	**71.88**
200	0.01	11.39	**12.26**	**10.67**	10.26	**11.70**	11.12	8.02	**9.45**	**10.16**	8.80	**10.19**	8.86
200	0.1	**68.59**	66.91	**59.88**	59.01	**63.69**	62.66	**67.77**	65.25	**64.68**	62.95	**66.39**	64.45
200	0.2	**85.22**	84.86	**77.10**	76.85	**80.65**	80.38	**85.37**	85.15	**82.40**	82.26	**83.80**	83.62
200	0.3	88.88	**89.02**	82.47	**82.64**	85.41	**85.58**	87.08	**87.60**	84.52	**84.99**	85.55	**86.07**
200	0.4	87.49	**87.95**	82.09	**82.56**	84.47	**84.97**	82.83	**83.66**	81.07	**81.78**	81.77	**82.59**
200	0.5	82.64	**83.36**	77.73	**78.46**	79.66	**80.42**	74.33	**75.49**	73.25	**74.36**	73.13	**74.38**
300	0.01	11.37	**12.32**	**10.73**	10.30	**11.78**	11.19	7.97	**9.42**	**10.15**	8.82	**10.17**	8.84
300	0.1	**68.77**	67.12	**59.93**	59.12	**63.73**	62.68	**67.80**	65.22	**64.75**	62.96	**66.69**	64.64
300	0.2	**85.34**	84.96	**77.53**	77.27	**80.85**	80.54	**85.59**	85.32	**82.57**	82.43	**84.09**	83.89
300	0.3	89.05	**89.21**	82.85	**83.02**	85.63	**85.81**	87.16	87.73	84.59	**85.14**	85.63	**86.22**
300	0.4	87.79	**88.26**	82.52	**82.98**	85.08	**85.55**	83.05	83.86	81.36	**82.10**	81.88	**82.64**
300	0.5	83.13	**83.85**	78.41	**79.16**	80.53	**81.26**	74.80	**75.98**	73.77	**74.81**	73.90	**75.09**

The first two columns indicate the cardinality of the set of positive cases ($N_{\mathscr{E}1}$), and the ratio of unlabeled cases that are actually a positive case (p). The significance of the differences has been evaluated by a Wilcoxon signed rank test based on 100 results obtained for 100 different problems built following the definition shown in the first two columns. Best results where significant differences have been found (at a significance level of 1%) are highlighted with bold font.

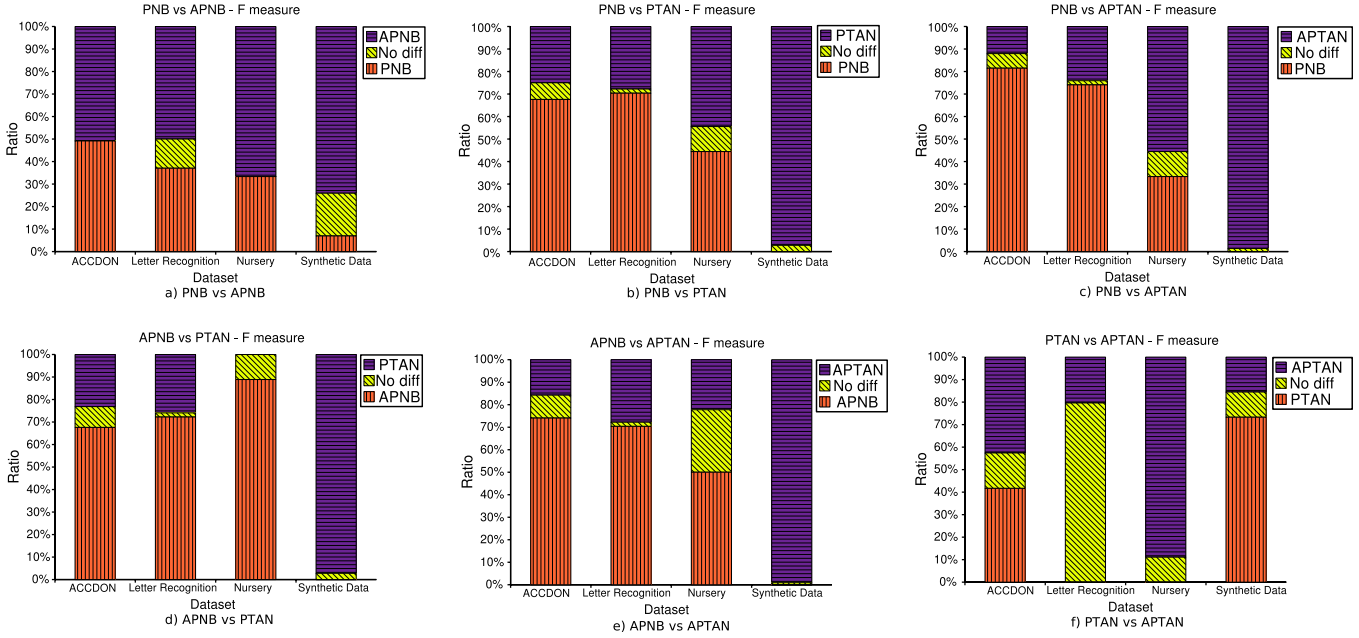

Fig. 5. Comparison between algorithms in terms of the F measure. The charts show, for the datasets obtained from each database, the ratio of the problems where each algorithm beats the other and the ratio of datasets where no significant differences were found at a significance level of 1%. It is important to point out that the amount of problems obtained from each of the databases is not the same (108 for ACCDON, 72 for Letter Recognition, 18 for Nursery and 216 for synthetic datasets).

Bayes-based algorithms when the conditional dependencies between the predicting variables fit in a tree structure.

The results obtained in the other datasets show that (A)PNB gives better results in some datasets and (A)PTAN

in others. The difference between the results obtained in synthetic datasets and the real-life problems can be explained by the fact that, in synthetic datasets, we know that there is an inner tree structure of dependencies between predicting variables and thus, TAN models give better results than naive Bayes models, but we know nothing about the structure in the real-life problems. It can be that the dependencies between variables in these problems are not as strong as in the synthetic datasets, allowing the naive Bayes models to improve the results obtained with TAN models.

In a real-world situation it is difficult to know which kind of model (naive Bayes or TAN) gives better results, but TAN models give us additional information regarding the relationship between variables. If we know that it is very likely that there are (strong) dependencies between predicting variables in our problem, then TAN induction algorithms will provide us with more realistic models than naive Bayes induction algorithms.

5.3.3. PTAN vs APTAN

In Section 5.3.1 we have seen that APNB is equivalent to PNB when p is set at a value slightly higher than the expected value of the Beta distribution and, as a consequence, the results obtained in certain datasets show a characteristic pattern.

The comparison between PTAN and APTAN is not that straightforward, as the averaging process includes the estimation of the mutual information, and thus the learning of the structure of the model. This is reflected in the results (see Fig. 5f), where no apparent pattern can be found. PTAN yields better results for certain datasets (those based on acceptor sites and synthetic data), APTAN improves the results obtained by PTAN in other datasets (those based on donor sites and `Nursery` database), and no significant differences can be found in the rest (those based on `Letter Recognition` database).

6. Conclusions and future work

In this paper we have seen that the concept behind PNB algorithm can be successfully extended to more sophisticated Bayesian classifiers. We have shown how to extend it to TAN models, but other induction algorithms could be adapted to the positive unlabeled learning context following a similar procedure. The resulting algorithm (PTAN) has been tested in datasets with an inner tree structure, showing a significant improvement with respect to PNB.

We have also proposed a new Bayesian approach to handle the p parameter that models its uncertainty by means of a Beta distribution. In the empirical comparison between APNB and PNB we have seen that our averaged version outperforms the original algorithm in most of the datasets. In the case of PTAN and APTAN, the comparison is not so clear and, depending on the dataset, PTAN or APTAN yields better results (or no significant differences are observed).

In real-life problems, where the actual value of the a priori probability of the negative class is not known, it is not possible to decide whether the normal or the averaged version of the classifiers will build better models. The advantage of the averaged version is that, thanks to the Beta distribution, it provides a softer way to incorporate the a priori knowledge about the problem domain.

Regarding the lines for future work, this paper opens the door to the extension of these ideas to other Bayesian classifiers such as kDB (Sahami, 1996). The evaluation of classifiers in the positive unlabeled learning context is still an unsolved problem and thus it is also an interesting line to follow in the future. As we said in the introduction, there are biological problems, such as the prediction of disease genes or the identification of RNA functional genes, where the absence of negative cases is inherent to the problem. For the future, we would also like to look for other computational biology problems that can be modeled as a positive unlabeled learning problem.

Acknowledgements

This work was supported by the University of the Basque Country (GIU03/67), the Basque Government (ETORTEK and SAIOTEK projects) and the Spanish Ministry of Education and Science (TIN2005-03824). The authors would also like to thank the reviewers for their interesting comments and useful suggestions that have helped us to improve the presentation of this work.

Appendix A. Averaging the estimators

This appendix shows how to obtain the averaged versions of the estimators that are used in APNB and APTAN. As an example, we will develop the integration process for the estimator in Eq. (1).

The density function of the Beta distribution is as follows:

$$f(x;\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} x^{\alpha-1}(1-x)^{\beta-1}$$

$$\Gamma(x) = \int_0^\infty t^{x-1}\mathrm{e}^{-t}\,\mathrm{d}t$$

$$x \in [0,1]; \quad \alpha,\ \beta > 0$$

The estimator in Eq. (1) depends on p, so we can integrate it over all the possible values of p as

$$P(x_{ij}|0) = \int_0^1 P(x_{ij}|0)\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}\,\mathrm{d}p$$

$$= \int_0^1 \frac{P(x_{ij}) - P(x_{ij}|1)\cdot p}{1-p}\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}\,\mathrm{d}p$$

separating the previous integral we obtain

$$P(x_{ij}|0) = \int_0^1 \frac{P(x_{ij})\Gamma(\alpha+\beta)\Gamma(\beta-1)}{\Gamma(\alpha+\beta-1)\Gamma(\beta)}\mathrm{Beta}(\alpha,\beta-1)\,\mathrm{d}p - \int_0^1 \frac{P(x_{ij}|1)\Gamma(\beta-1)\Gamma(\alpha+1)}{\Gamma(\beta)\Gamma(\alpha)}\mathrm{Beta}(\alpha+1,\beta-1)\,\mathrm{d}p$$

If we take the factors that do not depend on p out of the integrals, we have the integral over all the possible values of a Beta distribution with parameters $(\alpha,\beta-1)$ in the first term and $(\alpha+1,\ \beta-1)$ in the second one, both of which are equal to one. Thus, we have that

$$P(x_{ij}|0) = \frac{P(x_{ij})\Gamma(\alpha+\beta)\frac{\Gamma(\beta)}{\beta-1}}{\frac{\Gamma(\alpha+\beta)}{\alpha+\beta-1}\Gamma(\beta)} - \frac{P(x_{ij}|1)\frac{\Gamma(\beta)}{\beta-1}\alpha\Gamma(\alpha)}{\Gamma(\beta)\Gamma(\alpha)} = \frac{P(x_{ij})(\alpha+\beta-1)}{\beta-1} - \frac{P(x_{ij}|1)\alpha}{(\beta-1)}$$

Then, we can estimate the probabilities related to the negative class as

$$P(x_{ij}|0) = \frac{\alpha(P(x_{ij}) - P(x_{ij}|1)) + (\beta-1)P(x_{ij})}{\beta-1}$$

References

Bernardo, J.M., Smith, A.F.M., 1994. Bayesian Theory. John Wiley & Sons.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer.

Blake, C., Merz, C., 1998. UCI repository of machine learning databases. <http://www.ics.uci.edu/~mlearn>.

Calvo, B., López-Bigas, N., Furney, S.J., Larrañaga, P., Lozano, J.A., 2007. A partially supervised classification approach to dominant and recessive human disease gene prediction. Comput. Meth. Prog. Biomed. 85 (3), 229–237.

Castelo, R., Guigó, R., 2004. Splice site identification by idlBNs. Bioinformatics 4 (20), I69–I72.

Dempster, A., Laird, N., Rubin, D., 1977. Maximum likelihood from incomplete data via the EM algorithm. J. Roy. Statist. Soc. Ser. B 39, 1–38.

Denis, F., Gilleron, R., Tommasi, M., 2002. Text classification from positive and unlabeled examples. In: The 9th Internat. Conf. Information Processing and Management of Uncertainty in Knowledge-Based Systems, IPMU 2002, pp. 1927–1934.

Denis, F., Laurent, A., Gilleron, R., Tommasi, M., 2003. Text classification and co-training from positive and unlabeled examples. In: Proc. ICML 2003 Workshop: The Continuum from Labeled to Unlabeled Data, pp. 80–87.

Duda, R., Hart, P., Stork, D., 2001. Pattern Classification. Wiley Interscience.

Friedman, N., Geiger, D., Goldszmit, M., 1997. Bayesian network classifiers. Machine Learn. 29, 131–163.

Li, X., Liu, B., 2003. Learning to classify texts using positive and unlabeled data. In: Proc. 18th Internat. Joint Conf. on Artificial Intelligence (IJCAI-03), pp. 587–594.

Liu, B., Lee, W.S., Yu, P.S., Li, X., 2002. Partially supervised classification of text documents. In: Proc. 19th Internat. Conf. Machine Learn. (ICML-2002), pp. 387–394.

Liu, B., Dai, Y., Li, X., Lee, W.S., Yu, P.S., 2003. Building text classifiers using positive and unlabeled examples. In: 3rd IEEE Internat. Conf. Data Mining (ICDM'03), p. 179.

Minsky, M., 1961. Steps toward artificial intelligence. Proc. Inst. Radio Eng. 49, 8–30.

Pearl, J., 1988. Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann Publishers.

Sahami, M., 1996. Learning limited dependence Bayesian classifiers. In: Proc. 2nd Internat. Conf. Knowledge Discovery and Data Mining, pp. 335–338.

Tax, D.M.J., 2001. One-class Classification. Ph.D. Thesis, Technische Universiteit Delft.

Tax, D., Duin, R., 2002. Uniform object generation for optimizing one-class classifiers. J. Machine Learn. Res. 2 (2), 155–173.

Wang, C., Ding, C., Meraz, R.F., Holbrook, S.R., 2006. PSoL: A positive sample only learning algorithm for finding non-coding RNA genes. Bioinformatics 22 (21), 2590–2596.

Yu, H., Zhai, C., Han, J., 2003. Text classification from positive and unlabeled documents. In: Proc. 12th Internat. Conf. Information Knowledge Management, ACM Press, pp. 232–239.