Feature subset selection from positive and unlabelled examples

Borja Calvo a,*, Pedro Larrañaga b, Jose A. Lozano a

a Intelligent Systems Group, Department of Computer Science and Artificial Intelligence, University of the Basque Country, Paseo Manuel de Lardizabal, 1, E-20018 Donostia-San Sebastián, Spain
b Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, E-28660 Boadilla del Monte, Spain

A R T I C L E   I N F O

Article history:
Received 25 January 2008
Received in revised form 23 January 2009
Available online 12 May 2009
Communicated by L. Heutte

Keywords:
Positive unlabelled learning
Partially supervised classification
Feature subset selection
Filter methods

A B S T R A C T

The feature subset selection problem has a growing importance in many machine learning applications where the amount of variables is very high. There is a great number of algorithms that can approach this problem in supervised databases but, when examples from one or more classes are not available, supervised feature subset selection algorithms cannot be directly applied. One of these algorithms is the correlation based filter selection (CFS). In this work we propose an adaptation of this algorithm that can be applied when only positive and unlabelled examples are available. As far as we know, this is the first time the feature subset selection problem is studied in the positive unlabelled learning context. We have tested this adaptation on synthetic datasets obtained by sampling Bayesian network models where we know which variables are (in)dependent of the class. We have also tested our adaptations on real-life databases where the absence of negative examples has been simulated. The results show that, having enough positive examples, it is possible to obtain good solutions to the feature subset selection problem when only positive and unlabelled instances are available.

1. Introduction

Synthesising the knowledge contained in databases into classification models is a very powerful tool that can be used in a wide range of applications, from genome analysis to spam filtering. In principle, one can be tempted to think that the more information we have, the better the model we can induce, but this is only partially true.

In supervised databases we have instances characterised by some features or predicting variables and a category or class variable (Bishop, 2006; Duda et al., 2001). Given a new instance, a classification model tries to predict its class based on the value of the features, but not all the features are equally useful for the classification purpose. Non informative (poorly correlated with the class) and redundant variables (highly correlated with other features) can be harmful for some model induction algorithms. Irrelevant and redundant features are not only harmful, but they also lead to models that are too complex and increase the computational time required to obtain the classifier. Therefore, producing a small set of predictive and non-redundant features is becoming a very important step in many machine learning applications.

Two main ways to reduce of the dimensionality of classification problems have been proposed in the literature: feature extraction (Liu and Motoda, 1998) and feature subset selection (FSS) (Liu and Motoda, 2008, Guyon and Elisseeff, 2003). The former consists of combining the features in the database to obtain new, better features. The main problem with this approach is that the meaning of the original variables is lost in the newly constructed features. The latter consists of selecting the best subset of features for the classification purpose. In this work we will focus on the FSS approach.

There are three main approaches to the FSS problem (Guyon and Elisseeff, 2003, Saey et al., 2007), namely embedded, wrapper and filter methods. Some classifier induction algorithms, such as the C4.5 algorithm (Quinlan, 1993), do not use all the available variables. This sort of FSS is known in the literature as embedded FSS. The wrapper approaches (Kohavi and John, 1997) try to identify the subset of variables that, given a classification paradigm and a dataset, provides with the best classification function. The process consists of a search in the feature subset space guided by a performance measure (typically the accuracy, though other measures can be used). Each subset is evaluated by testing the performance of the chosen paradigm in the dataset, using only the variables in the subset at evaluation. The estimation of the performance of the classifiers requires a validation scheme, such as cross validation (Efron, 1983) or bootstrap estimation (Efron and Tibshirani, 1993). As a result, the evaluation of each subset involves the training and testing of several classification functions, increasing the computational time required for the FSS process. Besides, the search for the best subset is an NP-hard problem (Amaldi and Kann, 1998) and, thus, an exhaustive search quickly becomes computationally unfeasible and search heuristics have to be used to obtain a good feature subset in a reasonable time (Inza et al., 2000). This is the main drawback of these methods. Another characteristic...
of the wrapper methods (that can be good or bad, depending on the point of view) is that the subset produced by the algorithm depends on the classification paradigm considered in the search. This means that the selection obtained with a classification paradigm cannot be applied to other classification paradigms, as the solution is tuned up for that particular paradigm.

The filter approaches search for the best subset of variables, independently of the classification paradigm, considering the relationship between the predicting variables and the class and (sometimes) the relationship among the predicting variables. One of the most simple approaches consists of ranking the variables according to their usefulness and then selecting only those variables on the top of the ranking. The usefulness of a feature is measured univariately by means of different metrics. For instance, information theory related metrics (Cover and Thomas, 2006) evaluate the usefulness of the feature by measuring the reduction on the uncertainty of the class variable when the value of the feature at evaluation is known (Ben-Bassat, 1982). Once the features are ranked, a threshold must be set to obtain the final subset. The ranking methods are only concerned with the relevancy of the features considered and, thus, they do not filter out redundant variables.

The problem of searching for relevant and non-redundant features can be solved by a multivariate filter method known as correlation based filter selection (CFS) (Hall and Smith, 1997). This method searches for the best feature subset guided by a metric that measures both the correlation between each variable and the class and the correlation among the selected variables. The aim is to obtain a subset of relevant variables (i.e., features strongly correlated with the class) without redundancies (i.e., with a small correlation between them).

Filter methods are much faster than wrapper approaches and they are independent of the classification paradigm. Therefore, once we have a subset of features, this subset can be used in the training of any sort of model.

All the filter FSS methods mentioned above require examples from all the classes in order to measure the correlation between each feature and the class, but in some real situations, getting examples from one or more classes can be difficult or even impossible. For instance, suppose that we have a set of papers about a particular topic and we want to retrieve, from a database of (unlabelled) papers, those related to the ones in our set. We could try to obtain a set of uninteresting documents by hand labelling some papers, but this can be a very tedious task. In addition, the hand-labelled negative examples have to be representative of all the possible negative instances and this is an even harder task. Another example where getting negative instances is impossible is the identification of cancer genes (Furney et al., 2008). If we want to identify which genes are related with cancer, we have a list of positive examples (genes that have already been identified as cancer-related), but for the rest of the genes we have no information about their label (it is not possible to ensure that a given gene is not related to cancer in any possible way) and, thus, we have no negative instances. Therefore, it would be interesting to be able to build a classifier only with positive and unlabelled examples.

We can overcome the lack of negative instances by training a classifier using only positive and unlabelled examples. The problem of learning binary classifiers from only positive and unlabelled examples, known in the literature as partially supervised classification (Liu et al., 2002) or positive unlabelled learning (Denis et al., 2002), deals with this sort of situation. Many new methodologies have been developed to solve this problem (Calvo et al., 2007, Denis et al., 2003, Liu et al., 2003). In this paper we tackle the FSS problem when only positive and unlabelled examples are available. To the best of our knowledge, this is the first time the FSS problem is explicitly addressed in the positive unlabelled learning framework.

In this work we present an adaptation of the CFS algorithm that can be used without negative examples. The results obtained with this new algorithm have been compared with the ones obtained with the original CFS. For this comparison, we have used synthetic datasets obtained sampling Bayesian network models and real-life data based datasets where the absence of negative examples has been simulated.

The rest of the paper is organised as follows. In Section 2 the CFS algorithm is described and our adaptation to the positive unlabelled learning context is presented. In Section 3 our proposal is compared with the original CFS on synthetic and real-life data based problems. Finally, in Section 4 some conclusions and ideas about the future work are provided.

### 2. CFS based feature subset selection

Before presenting the CFS algorithm and its adaptation to the positive unlabelled learning context, some basic notation has to be introduced. Instances are characterised by a feature vector $X$ of $n$ components $(X_1, \ldots, X_n)$ and a class variable $C$ that can take only two values, 0 and 1 (also referred to as negative and positive); each feature $X_i$ can take $t_i$ values. For the sake of simplicity, the probabilities $P(X_i = j)$, $P(X_i = v | X_i = j)$ and $P(X_i = j | C = c)$ will be abbreviated as $P(x_i)$, $P(x_{iv} | x_i)$ and $P(x_i | c)$, $P(C = 1)$ will be denoted as $p$.

#### 2.1. The CFS metric

The CFS algorithm (Hall and Smith, 1997) is based on a metric that evaluates the merit of a given set of features. This metric is then used to guide a search for the best possible subset of variables. The merit function is based on the correlation between each feature and the class (relevancy) and on the correlation among the features in the subset (redundancy). This function can be expressed as:

$$G_S = \frac{k \tau_{ci}}{\sqrt{k + (k - 1) \tau_{r}}},$$

where $k$ is the number of variables in the subset $S$, $\tau_{ci}$ is the average correlation between the features in $S$ and the class, and $\tau_{r}$ is the average correlation among the features in $S$.

In (Hall and Smith, 1997) the authors measure the correlation between two variables $X_i$ and $X_j$ by means of the uncertainty coefficient $U(X_i, X_j)$, which is based on the mutual information $I(X_i; X_j)$ and the entropy $H(X_i)$ (Cover and Thomas, 2006). When the features are represented as random discrete variables (either because they are discrete or because they have been discretised) $U(X_i, X_j)$ is defined as:

$$U(X_i, X_j) = I(X_i, X_j) = \frac{H(X_i) - H(X_i | X_j)}{H(X_i)} = \frac{H(X_i) - H(X_i | X_j)}{H(X_i)}$$

$$I(X_i; X_j) = - \sum_{v=1}^{t_i} P(x_{iv}) \log P(x_{iv})$$

$$H(X_i) = - \sum_{j=1}^{t_i} P(X_j) \sum_{v=1}^{t_i} P(x_{iv} | x_i) \log P(x_{iv} | x_i)$$

Sets of irrelevant (poorly correlated with the class) and/or redundant variables (with a high correlation among them) will have a small $G_S$ value associated. Therefore, this metric can be used to guide the search for sets of relevant and non-redundant variables.

The CFS approach consists of a search in the feature subset space for a feature subset that maximises the $G_S$ score. As this search is an NP-hard problem, search heuristics are required to
obtain good solutions in a reasonable time. In the original work (Hall and Smith, 1997), greedy hill climbing forward selection and backward elimination algorithms are used.

2.2. CFS in absence of negative examples

In the supervised classification framework we have a dataset of labelled instances containing examples from all the classes. CFS uses these examples to obtain, for a candidate feature subset, the correlation between every pair of features and between each feature and the class variable.

In the positive unlabelled learning context only positive and unlabelled examples (Calvo et al., 2007), neither 

classifiers from positive and unlabelled examples (Calvo et al.,

tained so as, 

the lack of negative examples.

The problem with this estimator is that it can be negative. As 

the amount of unlabelled instances where 

he cannot be probabilities lower than 0, the negative estima-

The problem comes when we try to obtain 

the a priori probability of the positive class is available, its ex-

probability cannot be estimated from the data, the user must set 

in general its value is unknown. When some information about 

the probability cannot be estimated from the data, the user must set 

the a priori probability of the positive class is available, its ex-

pected value can be estimated and used in the algorithm.

In (Calvo et al., 2007) we presented a Bayesian approach to cope 
with the uncertainty about the p parameter. Following the same principle presented in that work, we can model the uncertainty about this parameter by means of a Beta distribution with parameters α and β. Having the knowledge of p modelled, we can average the conditional probability $P(x_0|0)$ for all the possible values of p (from 0 to 1):

$$P(x_0|0) = \frac{1}{\beta - 1} \int_0^1 P(x_0) - p(x_0|1)dp$$

This integral can be solved (more details can be found in (Calvo et al., 2007)), and the resulting estimator is:

$$P(x_0|0) = \frac{2}{\beta - 1} P(x_0) - p(x_0|1) + (\beta - 1)P(x_0)$$

Using this estimator instead of that shown in Eq. (1) we can propose a new algorithm that we have named averaged puCFS (apuCFS). In (Calvo et al., 2007) we shown that the averaged estimator (Eq. (2)) is equivalent to the non-averaged one (Eq. (1)) when the p is set at a value slightly greater than the expected value of the Beta distribution. As a consequence, both puCFS and apuCFS provide very similar results and, therefore, from now on we will focus only on the results obtained by the puCFS algorithm. The results obtained by the apuCFS can be consulted at the supplementary data web page.

3. Experimental evaluation

The experimentation has two parts. In the first part the algorithms are evaluated on datasets sampled from known probability distributions. As the probability distribution used to generate the data is known we know which features should be selected by the algorithms. In the second part of the experimental evaluation the algorithms are tested on real-life data based problems where the absence of negative examples has been simulated. In this second part the underlying probability distribution is unknown and, thus, we cannot know which features should be selected and which not. As the absence of negative examples is simulated we actually have both positive and negative examples. Therefore, supervised CFS can be applied to the datasets and

---

1 We are assuming that the set of unlabelled instances is a random sample drawn from the original probability distribution where the value of the class variable is missing.

2 http://www.sc.ehu.es/ccwivia/members/borja/positive.unlabelled.fss.
and the redundancy degree (100%–$\sigma$). For instance, given the variable $R2$, the redundant variable with $\sigma = 10\%$ is denoted as $R2$.

The most simple model (see Fig. 1a) consists of a naive Bayes (Minsky, 1961) and some independent variables. The second model (Fig. 1b) has a more complex structure, consisting of a tree augmented naive Bayes (TAN) (Friedman et al., 1997), five independent variables and a tree structure of five irrelevant features. The third model (Fig. 1c) includes a structure containing the relevant variables where the class variable is in the middle, five independent variables and a structure of eight irrelevant features. The fourth model (Fig. 1d) is the most complex, including a big structure containing the class and relevant and irrelevant features, a small structure of irrelevant variables and six independent variables. Redundant variables with different degrees of noise $\sigma$ (10%, 20% and 30%) were added to all the datasets sampled from the four models.

Starting from the structures shown in Fig. 1, Bayesian network models are constructed by setting the parameters of the model randomly. The probability distributions defined by these models were sampled to obtain sets of positive unlabelled instances. The set of positive cases was obtained by randomly sampling the models and then taking only the positive examples. The sets of unlabelled instances were obtained sampling the models to obtain positive and negative examples so that the ratio of positive cases is the desired one. From each model, 100 sets of positive cases of different sizes (10, 25, 50, 100, 250, 500, 750 and 1000) and 10 sets of unlabelled instances of different sizes (1000 and 10,000) and different ratios of positive cases (0.01, 0.15, 0.25, 0.35 and 0.75) were drawn. Each combination of a set of positive cases (of a given size) and a set of unlabelled instances (of a given size and with a given ratio of positive cases), both drawn from the same model, represents a positive unlabelled learning problem. Consequently, for a given model, a given number of positive and unlabelled examples and a given ratio of positive cases in the set of unlabelled instances, we have 1000 positive unlabelled learning problems. Globally, we are considering 320,000 positive unlabelled learning problems in the experimental evaluation of the algorithm based on synthetic data.

3.1. Synthetic datasets

The synthetic datasets were obtained sampling Bayesian network models of increasing complexity. These models include four kinds of variables (Fig. 1): the class variable (C), relevant variables (denoted by $R$ followed by a number), irrelevant non-independent variables (denoted by $I$ followed by a number) and irrelevant independent variables (denoted by $II$ followed by a number); redundant variables were added to the datasets after the model was sampled.

The relevant variables are those probabilistically related with the class (i.e., those on the Markov blanket of $C$). The irrelevant features are those not related with the class but related with other variables and the irrelevant independent variables are those not related neither with the class nor with any other variable.

Redundant variables are added to the datasets by copying all the relevant variables with a given amount of noise $\sigma$.\(^3\) Redundant variables are represented by the name of the original feature followed by $R$ and the redundancy degree (100%–$\sigma$). For instance, given the variable $R2$, the redundant variable with $\sigma = 10\%$ is denoted as $R2$.

The instances are actually labelled, but the labels are ignored when used with the positive unlabelled learning algorithms.

---

\(^3\) $\sigma$ represents the percentage of instances where the variable values are incorrectly copied.

---
cases in CFS, represented on the Y-axis). The ratios of selection for the puCFS algorithm shown in this figure are the average of the results obtained for all the ratios of positive cases in the set of unlabelled examples (from 0.01 to 0.75). Fig. 3 shows an example of the differences in the selection as a function of the ratio of positive cases in the set of unlabelled instances (the rest of these comparisons can be found at the supplementary data web page²).

If we have a look at the results obtained for the most simple model (Fig. 3a), the CFS algorithm is able to select the relevant features, filtering out both the irrelevant and the redundant variables. The selection of relevant variables with our proposal is not as strong as the supervised CFS selection, but it follows a similar pattern (variables such as $R_3$ or $R_4$ are selected almost always while $R_2$ is only selected in some of the datasets). It is important to point out that, due to the experimental design, the squares in the CFS chart are the average of 20 selections (there are 20 different sets of unlabelled examples sampled from a given model with a given ratio of positive cases in the set of unlabelled examples, 10 of size 1000 and 10 of size 10,000) while the squares in the puCFS charts are the average of 10,000 selections (100 sets of positive examples combined with 100 sets of unlabelled cases, 10 for each size and ratio of positive examples). The difference in the strength of the selection between CFS and puCFS can be explained by a less accurate estimation of the probabilities due to the uncertainty on the a priori probability of the positive class $p$ (obviously, in positive unlabelled learning problems we have less information than in supervised classification and, therefore, the estimations are less accurate).

Having a look at the rest of the models we can see that, regarding the relevant features, puCFS and CFS follow a similar pattern, though again the CFS selection is stronger than that of the puCFS. The model where there are more discrepancies between the supervised and the positive unlabelled learning selection is model 2, where only one common feature ($R_5$) is selected by both CFS and puCFS. As can be seen in the most complex models, some supposedly relevant variables are not selected. This is due to the fact that, although these variables are graphically related with the class, once the parameters are (randomly) set, these variables are not (or not very strongly) correlated with the class in the final probability distribution. This can be seen in Table 1 that shows, for the relevant features in the fourth model, the information gain ratio with the class, a metric based on the information theory that
measures the correlation between two variables and whose definition is as follows:

\[
\text{GainR}(C, R_i) = \frac{I(C; R_i)}{H(R_i)} = \frac{H(C) - H(C|R_i)}{H(R_i)}
\]

Taking into account that \(I(C, R_i) = H(C) - H(C|R_i) = H(R_i) - H(R_i|C)\) (Cover and Thomas, 2006), we can see that the information gain ratio and the uncertainty coefficient introduced in Section 2 are equivalent. Therefore, this is the same metric used by the CFS to measure the correlation of the variables with the class. In Table 1 we can see that variables \(R_4\) and \(R_6\) are very poorly correlated with the class while variables \(R_8\) and \(R_9\) are not correlated at all with the class. This explains why these four variables are almost not selected by the algorithms in the different datasets.

It is important to point out that the results shown in Table 1 have been obtained from the probability distribution

<table>
<thead>
<tr>
<th>Feature</th>
<th>Info gain ratio</th>
</tr>
</thead>
<tbody>
<tr>
<td>R1</td>
<td>(1.1 \times 10^{-7})</td>
</tr>
<tr>
<td>R2</td>
<td>(3.4 \times 10^{-2})</td>
</tr>
<tr>
<td>R3</td>
<td>(5.6 \times 10^{-3})</td>
</tr>
<tr>
<td>R4</td>
<td>(3.1 \times 10^{-7})</td>
</tr>
<tr>
<td>R5</td>
<td>(2.7 \times 10^{-2})</td>
</tr>
<tr>
<td>R6</td>
<td>(2.3 \times 10^{-6})</td>
</tr>
<tr>
<td>R7</td>
<td>(6.2 \times 10^{-2})</td>
</tr>
<tr>
<td>R8</td>
<td>0.0</td>
</tr>
<tr>
<td>R9</td>
<td>0.0</td>
</tr>
<tr>
<td>R10</td>
<td>(1.5 \times 10^{-3})</td>
</tr>
</tbody>
</table>

Fig. 3. Variables selected by puCFS \((p = 0.25)\) on datasets sampled from model 2 with different ratios of positive cases hidden in the set of unlabelled examples. Each square in the charts represents, for a particular number of positive cases \((-Y\text{-axis})\) and a particular variable \((-X\text{-axis})\), the ratio of datasets the variable has been selected using puCFS. Similar results have been obtained with the rest of the models.
defined by model 4, and not from a sample drawn from the distribution.\textsuperscript{5}

Regarding the irrelevant variables, we can see that in most of the datasets CFS is able to filter them out, while puCFS tends to include them (specially the independent ones) when the number of positive examples is low. As we increase the number of known positive examples, the selection of these features decreases. This selection is due to the score we are using to select the variables. In the CFS algorithm, the score increases either when we increase the mean correlation with the class (which is in the numerator) or when we decrease the mean correlation among the features (which is in the denominator). Independent features do not increase the correlation with the class, but they decrease the mean correlation between the selected features and, thus, they increase the score. In the supervised CFS algorithm this is compensated with a similar decrease in the correlation with the class, but in puCFS, and especially when very few positive examples are available, the estimation of the correlation of the features with the class is less accurate and probably it is not able to compensate the decrease in the correlation among features.

As can be seen in Fig. 2, both CFS and puCFS are able to filter out the redundant variables (though a shadow of selection can be seen in puCFS selections with few positive examples). Finally, Fig. 4 shows the comparison between the selection obtained with puCFS at different values of the parameter $p$. As can be seen in the figure, although there is a tendency to increase the number of variables selected when we increase $p$, the dependency with $p$ is not very strong.

\textsuperscript{5} The metric has been calculated with the probabilities obtained with the JavaBayes software (http://www.cs.cmu.edu/~javabayes/). This software can be used to load a Bayesian network and then make inference over the loaded model.

3.2. Real-life datasets

The algorithms have also been compared on positive unlabelled learning problems based on real-life databases where the absence of negative examples has been simulated.

Starting from real-life, completely labelled databases, we select one of the classes as the positive one and label all the remaining cases as negative. Sets of positive examples are obtained randomly selecting instances belonging to the class we have labelled as positive. The sets of unlabelled instances are obtained by mixing positive and negative examples (in the desired proportion) and then removing their labels. The real-life data based datasets used in this work have been obtained from the ACCDON, Letter Recognition and Nursery databases. The former is a database of splice site prediction described in (Castelo and Guigó, 2004) and the other two have been obtained from the UCI repository (Blake and Merz, 1998).

Six groups of positive unlabelled datasets have been obtained from the ACCDON dataset, corresponding to donor and acceptor splice sites where negative examples are obtained from coding regions, non-coding regions and a mixture of both types of negative examples. Three sizes of the set of positive examples have been used: 100, 500 and 1000. The number of unlabelled examples is 10,000 in all the datasets, and the ratios of positive cases in the set of unlabelled instances used in the experimentation are 0.01, 0.1, 0.2, 0.3, 0.4 and 0.5.

Regarding the UCI datasets, letters ‘D’, ‘P’ and ‘U’ have been used in the Letter Recognition database to construct the positive unlabelled problems; ‘spec_prior’ has been selected as the positive one in the Nursery database. Three sizes of sets of positive examples have been used: 100, 200 and 300 and the number of unlabelled examples has been set at 5000 in all the cases. The ratios

---

**Fig. 4.** Variables selected by puCFS using different values for the parameter $p$ on datasets sampled from model 4. Each square in the charts represents, for a particular number of positive cases (Y-axis) and a particular variable (X-axis), the ratio of datasets where the variable has been selected.
of positive cases in the set of unlabelled instances used are the same as in the ACCDON database. More details about how the positive unlabelled datasets have been constructed can be consulted in (Calvo et al., 2007), and all the datasets can be found at the supplementary data web page.2

3.2.1. Results on real-life datasets

Both puCFS and the original supervised CFS have been applied to the datasets obtained from real-life databases. In the synthetic datasets we know which features should be selected and which should be filtered out. In real-life data based dataset we do not have this information. Therefore, we have evaluated the puCFS algorithm comparing it with the original supervised CFS. Fig. 5 shows an extract of the results obtained in this comparison.

Fig. 5a and b shows some of the results obtained by CFS and puCFS on datasets based on the ACCDON dataset. In these figures we can see that both algorithms tend to select almost the same features. There are some exceptions, such as \( v_1 \) and \( v_3 \) in Acceptors.

![Fig. 5](image_url)

**Fig. 5.** Variables selected by CFS and puCFS in some of the real-life data based datasets. The X-axis represents the features and the Y-axis the number of positive examples (the ratio of instances that are positive in CFS and the size of the set of positive examples in puCFS). Each square in the charts represents, for a particular number/ratio of positive cases and a particular variable, the ratio of datasets where the feature has been selected using that algorithm.
Sites Intron datasets, that are clearly selected by CFS when the ratio of positive examples is high, but not so clearly selected by puCFS. In this same chart we can see that puCFS tends to select v19 and v21 when few positive examples are available, the same behaviour we saw in synthetic datasets for the completely independent features.

In Donor Sites Coding datasets features v3, v4 and v5 are selected by both algorithms but, again, we can see that CFS selects v1 in datasets with ratios of positive examples higher than 0.2 while puCFS tends to filter this variable out. For v2 we have the opposite situation, CFS tends to filter it out (except for the lower ratios of positive examples) but puCFS includes this feature in more than half the datasets. puCFS also tends to include v6 and v7, but only in approximately 30% of the datasets. Similar results have been obtained in the remaining of the datasets based on the ACC-GS database.

Fig. 5c shows an extract of the results obtained in datasets based on the Letter Recognition database. The results obtained in these datasets are similar to those shown in Fig. 5a and b in the sense that almost the same features are filtered out by both puCFS and CFS. As it happened with v6 and v7 in Fig. 5b, v1 is selected by puCFS in about 30% of the datasets while CFS filters it out in almost all the datasets. Similar results have been obtained for the rest of the datasets.

Fig. 5d shows the results obtained in datasets based on Nursery. All the features in these datasets are selected almost always by both algorithms, though there are two features in CFS (comp and con1) and three in puCFS (comp, con1 and non) that have lower selection ratios. When the ratio of positive cases is lower than 0.1 CFS only selects clearly the last feature, rec. This phenomenon can also be observed in some features in other datasets (e.g., v1 in Fig. 5a and b and v3 in Fig. 5a), and is probably due to the big unbalance between the two classes in these problems.

4. Conclusions and future work

In this work we have extended the concept of FSS to the context of learning from positive and unlabelled examples. In particular, we have adapted the CFS algorithm to the positive unlabelled learning and we have tested this adaptation on both synthetic and real-life data based problems.

The two algorithms (the original CFS and our proposal) have been compared on synthetic datasets obtained from Bayesian network models where we know which features are related to the class and which are not. From the results of this comparison we can conclude that our adaptations tend to select more variables than the original version. The strong selection of completely independent features by puCFS when the number of positive examples is low is especially interesting. This is probably due to an increase in the $G_C$ score caused by a reduction in the averaged correlation between the selected features (as we are introducing completely independent variables) rather than by an increment in the average correlation with the class. This does not happen with the original supervised CFS because the decrease in the numerator (the mean correlation with the class) compensates the decrease in the denominator. This compensation does not occur in our adaptations, especially when the amount of known positive examples is low, probably due to a less accurate estimation of the correlation with the class (due to the lack of negative examples, we have less information than in the supervised framework).

Given the tendency of the CFS adaptation to select independent variables, it is advisable to perform a pre-filtering process based on the correlation with the class. Information theory measures, such as the uncertainty coefficient used by the CFS to compute the $G_C$ score, can be used at this step. With this pre-filtering the independent variables would be eliminated and, thus, they would not enter the puCFS selection process.

In the future, we would like to apply these algorithms to real-life problems. It would be particularly interesting to apply these FSS algorithms to biological problems, as the information about which variables are important for the classification purpose and which are not can provide very useful knowledge about the particular biological domain. Regarding the FSS techniques, in future work we would like to adapt, following the same principles shown in this paper, other univariate metrics such as those based on the information theory and use these adaptations to explore the ranking filter methods in the positive unlabelled learning context. Actually, we have already adapted one of these metrics in Section 2.2, as the uncertainty coefficient used in the $G_C$ score is equivalent to the information gain ratio, which can be used as a metric for ranking filtering methods. Another interesting problem for the future is the classifier validation in absence of negative examples, for it not only will allow us to compare different classifiers, but also will open the door to the wrapper FSS techniques. Finally, as a future research line, we would like to explore the possibility of developing an automated procedure to properly set the $p$ parameter.

Acknowledgements

This work has been partially supported by the Saitoek and Research Groups 2007–2012 (IT-242-07) Programs (Basque Government, TIN2008-06815-002-01, TIN2008-06815-002-02, and Consolidor Ingenio 2010 – CSD2007-00018 Projects (Spanish Ministry of Education and Science) and COMBIOMED network in computational biomedicine (Carlos III Health Institute).

References


