Unsupervised Feature Subset Selection

Nicolaj Søndberg-Madsen [†], Casper Thomsen [†], and Jose M. Peña

Department of Computer Science
Aalborg University, Denmark
{nicolaj, raist, jmp}@cs.auc.dk

Abstract. This paper studies filter and hybrid filter-wrapper feature subset selection for unsupervised learning (data clustering). We constrain the search for the best feature subset by scoring the dependence of every feature on the rest of the features, conjecturing that these scores discriminate some irrelevant features. We report experimental results on artificial and real data for unsupervised learning of naive Bayes models. Both the filter and hybrid approaches perform satisfactorily.

1 Introduction

One of the main problems that arises in a great variety of fields, including artificial intelligence, machine learning, and statistics, is the so-called *data clustering problem*. Given some data in the form of a set of instances with an underlying group-structure, data clustering may be roughly defined as the search for the best description of this group-structure, when the true group membership of every instance is unobserved. Each of the groups in the data at hand is called a *cluster*. The lack of knowledge of the cluster membership for every instance in the data makes data clustering be also referred to as *unsupervised learning*.

Among the different interpretations and expectations that the term unsupervised learning gives rise to, we are concerned in this paper with those unsupervised learning problems basically defined by the following assumptions:

- A database $\boldsymbol{d}$ containing N instances or cases, i.e. $\boldsymbol{d} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$, is available. The l-th case of $\boldsymbol{d}$ is represented by an $(n+1)$-dimensional discrete vector $\boldsymbol{x}_l = (x_{l1}, \ldots, x_{ln+1})$ partitioned as $\boldsymbol{x}_l = (c_l, \boldsymbol{y}_l)$: c_l is the unobserved cluster membership of $\boldsymbol{x}_l$, and $\boldsymbol{y}_l = (y_{l1}, \ldots, y_{ln})$ is the n-dimensional discrete vector of observations of $\boldsymbol{x}_l$.
- The number of clusters underlying $\boldsymbol{d}$, denoted by K, is known.
- Each of the K clusters in $\boldsymbol{d}$ represents a physical process defined by an unknown joint probability distribution. Then, every case in $\boldsymbol{d}$ may be seen as sampled from exactly one of these K unknown joint probability distributions. This corresponds to assuming the existence of an $(n+1)$-dimensional discrete random variable $\boldsymbol{X} = (X_1, \ldots, X_{n+1})$ partitioned as $\boldsymbol{X} = (C, \boldsymbol{Y})$: C is a unidimensional discrete hidden random variable that represents the

[†] These authors contributed equally to this work.

unobserved cluster membership, and $\boldsymbol{Y} = (Y_1, \ldots, Y_n)$ is an n-dimensional discrete random variable that represents the observations. Each Y_i is called a *feature*. It is usual to assume that the mechanism that generated $\boldsymbol{d}$ works in two stages: first, one of the physical processes associated with the K clusters that exist in $\boldsymbol{d}$ is somehow selected according to a probability distribution for C and, then, an instance is somehow generated according to the joint probability distribution for $\boldsymbol{Y}$ that defines the selected physical process.
- The parametric forms of the joint probability distributions that govern the mechanism that generated $\boldsymbol{d}$ are all known to be multinomial.

Under the assumptions above, unsupervised learning can be approached from a *model-based* perspective: the description of the K clusters in $\boldsymbol{d}$ is accomplished through the probabilistic modelling of the mechanism that generated $\boldsymbol{d}$. Then, unsupervised learning reduces to learning a joint probability distribution for $\boldsymbol{X}$ from $\boldsymbol{d}$. This is the approach to unsupervised learning this paper deals with.

Among the many challenges that unsupervised learning involves, one that has received little attention so far in the literature is unsupervised feature subset selection. Given a database $\boldsymbol{d}$ over $\boldsymbol{X} = (C, \boldsymbol{Y})$, it may happen that a subset of the original features $\boldsymbol{R} \subseteq \boldsymbol{Y}$ contains (almost) all the information about the group-structure underlying $\boldsymbol{d}$. When this is the case, the features in $\boldsymbol{R}$ are considered relevant for unsupervised learning from $\boldsymbol{d}$, while the features in $\boldsymbol{I} = \boldsymbol{Y} \setminus \boldsymbol{R}$ are deemed irrelevant (e.g. noise). Unsupervised feature subset selection aims at identifying the feature subsets $\boldsymbol{R}$ and $\boldsymbol{I}$ so that a model describing accurately the clusters in $\boldsymbol{d}$ can be obtained from the projection of $\boldsymbol{d}$ onto the random variable $(C, \boldsymbol{R})$ via unsupervised learning. This provides the user with better understanding of $\boldsymbol{d}$ as relevant and irrelevant features are identified. Moreover, the final model is more comprehensible as it only involves relevant features.

The remainder of this paper is structured as follows. Section 2 introduces the class of probabilistic graphical models for unsupervised learning that we consider in this paper, namely naive Bayes models. Section 3 describes our proposals for unsupervised feature subset selection. Section 4 evaluates these proposals on artificial and real data. Finally, Section 5 closes with some discussion.

2 Naive Bayes Models for Unsupervised Learning

As seen above, solving an unsupervised learning problem from a model-based approach reduces to learning a joint probability distribution. One of the paradigms specially well suited for such a purpose are Bayesian networks.

Let $\boldsymbol{X} = (C, \boldsymbol{Y})$ be a random variable as stated above. A *Bayesian network (BN)* for unsupervised learning for $\boldsymbol{X}$ is a pair $(\boldsymbol{s}, \boldsymbol{\theta})$, where $\boldsymbol{s}$ is the *model structure* and $\boldsymbol{\theta}$ are the *model parameters* [13]. The model structure $\boldsymbol{s}$ is an acyclic directed graph whose nodes correspond to the unidimensional random variables in $\boldsymbol{X}$. Throughout the text, the terms node and random variable are used interchangeably. The model parameters $\boldsymbol{\theta}$ specify a conditional probability distribution for each node X_i in $\boldsymbol{s}$ given its parents $\boldsymbol{Pa}_i$ in $\boldsymbol{s}$, $p(x_i \mid \boldsymbol{pa}_i)$. These conditional probability distributions are all typically multinomial.

A BN $(\boldsymbol{s}, \boldsymbol{\theta})$ for unsupervised learning for $\boldsymbol{X}$ represents a joint probability distribution for $\boldsymbol{X}$, $p(\boldsymbol{x})$, through the following graphical factorization:

$$p(\boldsymbol{x}) = \prod_{i=1}^{n+1} p(x_i \mid \boldsymbol{pa}_i) \; . \tag{1}$$

Therefore, $\boldsymbol{s}$ encodes a set of conditional (in)dependencies between the random variables in $\boldsymbol{X}$. Moreover, $\boldsymbol{s}$ is usually constrained so that every Y_i is a child of C. This restriction is imposed by the assumption about how the generative mechanism underlying the modelled domain works (recall Section 1).

A simple but competitive class of BNs for unsupervised learning that has received much attention in the literature are the so-called *naive Bayes (NB) models* (e.g. [3]). They are regular BNs for unsupervised learning with the only peculiarity that their model structures enforce features being conditionally independent given C. Then, the only arcs in the model structures are those due to the structural constraint previously stated, i.e. every Y_i is a child of C. Eq. (1) can be rewritten as follows for a NB model for unsupervised learning for $\boldsymbol{X}$:

$$p(\boldsymbol{x}) = p(c, \boldsymbol{y}) = p(c) p(\boldsymbol{y} \mid c) = p(c) \prod_{i=1}^{n} p(y_i \mid c) \; . \tag{2}$$

Given some data $\boldsymbol{d} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$ as defined above, unsupervised learning of a NB model for $\boldsymbol{X}$ from $\boldsymbol{d}$ reduces to unsupervised parameter estimation, thanks to the fact that the model structure is fixed beforehand. Maximum likelihood or maximum a posteriori estimates can be effectively obtained via approximation techniques such as the EM algorithm. In the terms introduced in Section 1, a NB model for unsupervised learning for $\boldsymbol{X}$ induced from $\boldsymbol{d}$ represents a description of the K clusters in $\boldsymbol{d}$ through (i) $p(c)$ modelling the selector, and (ii) $p(\boldsymbol{y} \mid c) = \prod_{i=1}^{n} p(y_i \mid c)$ modelling the physical processes to select among.

3 Feature Subset Selection for Unsupervised Learning

Given a database $\boldsymbol{d}$ over $\boldsymbol{X} = (C, \boldsymbol{Y})$, it is not unusual that only a subset of the original features $\boldsymbol{R} \subseteq \boldsymbol{Y}$ is informative about the clusters in $\boldsymbol{d}$, while the rest of the features $\boldsymbol{I} = \boldsymbol{Y} \setminus \boldsymbol{R}$ are marginally informative or totally uninformative (e.g. noise). When this is the case, $\boldsymbol{d}$ and the projection of $\boldsymbol{d}$ onto the random variable $(C, \boldsymbol{R})$, denoted by $\boldsymbol{d}^{(C,\boldsymbol{R})}$, have the same underlying group-structure because the only difference between them are the features in $\boldsymbol{I}$, which are uninformative about this structure. Therefore, it is possible to obtain a NB model for unsupervised learning from $\boldsymbol{d}^{(C,\boldsymbol{R})}$ such that it describes the clusters in $\boldsymbol{d}$ (almost) as accurately as the best NB model for unsupervised learning induced from the original database $\boldsymbol{d}$. The features in $\boldsymbol{R}$ are named *relevant* for unsupervised learning from $\boldsymbol{d}$, whereas the features in $\boldsymbol{I}$ are deemed *irrelevant*.

Feature subset selection for unsupervised learning or *unsupervised feature subset selection (UFSS)* aims at identifying the feature subsets $\boldsymbol{R}$ and $\boldsymbol{I}$ so that a NB model describing accurately the clusters in $\boldsymbol{d}$ can be obtained from $\boldsymbol{d}^{(C,\boldsymbol{R})}$ via unsupervised learning. This improves the interpretability of the induced model, as only relevant features are involved in it, without degrading its descriptive accuracy, as only irrelevant features are excluded from it. Additionally, the identification of relevant and irrelevant features for unsupervised learning provides valuable insight into the nature of the group-structure of $\boldsymbol{d}$. Finally, UFSS reduces the risk of overfitting and, thus, increases the reliability on the final model.

UFSS can be stated as an optimization problem in terms of search space, search strategy and objective function. The search space consists of all the subsets of the original features. An exhaustive search strategy is, therefore, unaffordable in most cases (2^n feature subsets exist for n original features). Instead, greedy hill-climbing is commonly used. This approach is referred to as sequential selection and it can be either forwards or backwards [10]. *Sequential forward selection (SFS)* starts with no feature being selected and, iteratively, the most rewarding feature among the unselected ones is selected until the stopping criterion is satisfied. Similarly, *sequential backward selection (SBS)* begins with the whole set of features being selected and, iteratively, the least rewarding feature among the selected ones is unselected until the stopping criterion is met. The objective function scores the relevance of every feature subset in the search space or, alternatively, the relevance of each feature alone. Objective functions for UFSS can be classified as being either *filters* or *wrappers* [9]. Filter approaches assess features or feature subsets from data exclusively, ignoring the subsequent unsupervised learning algorithm. On the other hand, wrapper approaches evaluate a feature subset according to the performance of the unsupervised learning algorithm on the original data projected onto the features in the subset.

3.1 Filter Unsupervised Feature Subset Selection

In this section we describe in detail our first proposal for UFSS: a filter approach combined with SFS. In particular, SFS proceeds as follows. Initially, no feature in $\boldsymbol{Y}$ is selected. Then, the most relevant feature among those unselected ones that are significantly relevant is iteratively selected. The search stops when no feature can be selected. We suggest below two filter relevance scores for features as well as a method for assessing whether or not a feature is significantly relevant.

The filter relevance scores that we describe below are based on the following observations. Each truly relevant feature in $\boldsymbol{R}$ must be dependent on the cluster random variable C, and in most cases strongly dependent. Therefore, the features in $\boldsymbol{R}$ must be all pairwise dependent. On the other hand, those features that are independent or weakly dependent on the rest of the features must be truly irrelevant for unsupervised learning from $\boldsymbol{d}$ and belong to $\boldsymbol{I}$. We conjecture that scoring the dependence of every feature in $\boldsymbol{Y}$ on the rest of the features helps in discriminating some irrelevant features. This reasoning assumes that there are at least two relevant features in $\boldsymbol{Y}$. Some other works that make use of similar observations are [6, 12, 17].

The first pairwise dependence score that we propose is mutual information. The dependence score for features Y_i and Y_j, $DS(Y_i, Y_j)$, is computed as the mutual information between Y_i and Y_j, $MI(Y_i, Y_j)$:

$$DS(Y_i, Y_j) = MI(Y_i, Y_j) = H(Y_i) - H(Y_i \mid Y_j) \tag{3}$$

where $H(Y_i)$ and $H(Y_i \mid Y_j)$ are the entropy of Y_i and the conditional entropy of Y_i given Y_j, respectively.

The second pairwise dependence score that we consider is based on mutual prediction. The dependence score for features Y_i and Y_j, $DS(Y_i, Y_j)$, can be seen as the mutual prediction between Y_i and Y_j, $MP(Y_i, Y_j)$:

$$DS(Y_i, Y_j) = MP(Y_i, Y_j) = 1 - \frac{1}{2}\left(\frac{PA(Y_i)}{PA(Y_i \mid Y_j)} + \frac{PA(Y_j)}{PA(Y_j \mid Y_i)}\right) \tag{4}$$

where $PA(Y_i)$ and $PA(Y_i \mid Y_j)$, similarly $PA(Y_j)$ and $PA(Y_j \mid Y_i)$, are:

$$PA(Y_i) = \max_{y_i} p(y_i) \; ; \qquad PA(Y_i \mid Y_j) = \sum_{y_j} p(y_j) \max_{y_i} p(y_i \mid y_j) \; . \tag{5}$$

$PA(Y_i)$, similarly $PA(Y_j)$, is the probability of predicting the state of Y_i correctly by predicting the most probable state of Y_i a priori. $PA(Y_i \mid Y_j)$, similarly $PA(Y_j \mid Y_i)$, is the expected probability of predicting the state of Y_i correctly by predicting the most probable state of Y_i given that the state of Y_j is known.

The probability distributions involved in the computation of the two dependence scores $DS(Y_i, Y_j)$ introduced above are estimated from $\boldsymbol{d}$ according to the maximum likelihood criterion. Note that both dependence scores are symmetric, i.e. $DS(Y_i, Y_j) = DS(Y_j, Y_i)$. In both cases the lower the value of $DS(Y_i, Y_j)$, the weaker the dependence between Y_i and Y_j. Finally, we suggest computing the relevance score for each feature Y_i, $RS(Y_i)$, as the average dependence score between Y_i and the rest of the features:

$$RS(Y_i) = \frac{1}{n-1} \sum_{\substack{j=1 \\ j \neq i}}^{n} DS(Y_i, Y_j) \; . \tag{6}$$

The lower the value of $RS(Y_i)$, the less relevant Y_i is for unsupervised learning from $\boldsymbol{d}$. Note that $RS(Y_i)$ is a filter measure, as it is based on $\boldsymbol{d}$ exclusively.

Recall that SFS keeps selecting features from unselected ones as long as they are significantly relevant. In order to decide upon the significance of the relevance of a particular feature, we propose carrying out a distribution-free hypothesis test. Given that Y_i is the feature being considered for selection, the main idea is to use $RS(Y_i)$ as the statistic value for a hypothesis test under the null hypothesis that Y_i is random noise and, thus, irrelevant. The probability distribution of the statistic under the null hypothesis is estimated empirically from the statistic values scored by m randomly chosen features that are known to satisfy the null hypothesis, being m sufficiently large. In practice, each of these m features can be obtained by randomly reshuffling the entries of Y_i in $\boldsymbol{d}$ or, alternatively, by filling each of these entries in with one of the states of Y_i drawn at random. Like in a regular hypothesis test, the empirically estimated probability distribution of the statistic under the null hypothesis is used to set a threshold for rejecting the null hypothesis at a given significance level α, with $0 \leq \alpha \leq 1$. In particular, if $RS(Y_i)$ is larger than $(1 - \alpha) \cdot 100$ % of the m statistic values scored by the randomly created irrelevant features, the null hypothesis that Y_i is irrelevant is rejected at significance level α. See [8] for a thorough introduction and extensive application of distribution-free hypothesis testing to clustering validation.

3.2 Hybrid Filter-Wrapper Unsupervised Feature Subset Selection

In this section we present our second proposal for UFSS: a wrapper approach combined with SBS built on top of the filter UFSS introduced in Section 3.1.

The filter UFSS described in the previous section is likely to behave conservatively: some truly irrelevant features may go undetected in the distribution-free hypothesis testing due to random effects. Preliminary experiments for unsupervised learning of NB models confirmed this and pointed out a simple way how to correct it, at least partially. The experiments showed that the accuracy of the description of the group-structure underlying the learning data never decreased by adding a feature selected by the filter UFSS. In other words, the performance of unsupervised learning of NB models was a non-decreasing function of the number of features selected by the filter UFSS. However, this function levelled off and became flat for the last few features selected (conservative behavior). These truly irrelevant features that go undetected by the filter UFSS can be spotted by running a wrapper UFSS with SBS afterwards. In the paragraphs below we give details about this hybrid filter-wrapper UFSS.

Our proposal proceeds in two steps as follows. First, we run the filter UFSS presented in Section 3.1. Second, we run a wrapper UFSS with SBS considering only those features identified as relevant by the filter UFSS, $\boldsymbol{R}_{filter}$. In particular, SBS starts by marking the features in $\boldsymbol{R}_{filter}$ as selected and those in $\boldsymbol{I}_{filter} = \boldsymbol{Y} \setminus \boldsymbol{R}_{filter}$ as unselected. Then, proceeding in reverse order as the features in $\boldsymbol{R}_{filter}$ were selected by the filter UFSS, i.e. from the least relevant to the most relevant, each feature is iteratively unselected until the stopping condition is met. The stopping criterion guarantees that the accuracy of the description of the clusters in $\boldsymbol{d}$ does not degrade dramatically. Let $\boldsymbol{R}_{current}$ denote the features that are still selected at the current iteration of the wrapper UFSS. The stopping criterion prescribes halting the wrapper UFSS if the performance of the NB model obtained from $\boldsymbol{d}^{(C,\boldsymbol{R}_{current})}$ via unsupervised learning falls below $(100 - \beta)$ % of the performance of the NB model obtained from $\boldsymbol{d}^{(C,\boldsymbol{R}_{filter})}$, where $0 \leq \beta \leq 100$ is a user-defined parameter. Intuitively, β represents the loss in performance that the user is willing to accept in the wrapper UFSS to identify new irrelevant features and, thus, improve the interpretability of the final model.

Note that each time the stopping criterion of the wrapper UFSS is evaluated, a NB model for unsupervised learning is induced. The number of evaluations of the stopping criterion is bounded by the number of features in $\boldsymbol{R}_{filter}$, $|\boldsymbol{R}_{filter}|$, because SBS performs a linear search. As mentioned above, preliminary experiments showed that the performance of unsupervised learning of NB models was a non-decreasing function of the number of features selected by the filter UFSS. Therefore, SBS can be easily modified to perform a binary search, instead of a linear search. This reduces considerably the number of evaluations of the stopping criterion of the wrapper UFSS so as to be bounded by $\log_2 |\boldsymbol{R}_{filter}|$. Finally, note that the assumptions made based on the preliminary experiments are harmless: if they are not satisfied in some domains, the hybrid UFSS does not perform worse than the filter UFSS. In fact, these assumptions are unnecessary if using sophisticated search strategies aimed at dealing with non-monotonic performance scores (e.g. floating search strategies [15]).

4 Experimental Evaluation

This section is dedicated to the empirical evaluation of the filter and hybrid UFSS described above for unsupervised learning of NB models. First, we introduce the artificial and real databases in the evaluation. Then, we discuss the experimental setting and performance assessment. Finally, we report the results obtained.

4.1 Databases

The first two artificial databases in the evaluation are created by sampling two BNs. These two BNs are constructed by adding 10 and 20 irrelevant features, respectively, to a BN for unsupervised learning reported in [14], which was induced from part of the Leukemia database (see below). Irrelevant features intend to simulate noise. They are added to the model as independent of the rest of the features, and their probability distributions are randomly generated. Therefore, these artificial BNs consist of 21 and 31 nodes, respectively, having three states all of them. From each of these two BNs we sample 10000 cases and, then, remove all the cluster labels. We refer to the samples as Syn10 and Syn20, respectively.

The third artificial database in the evaluation is obtained by processing the Waveform database [2]. This database contains 5000 instances with each instance being characterized by 40 continuous measurements. There are three classes. Measurements are discretized into three bins of equal width, and class labels are removed. We refer to the resulting database as Waveform.

The first real database in the evaluation consists of the training data of the CoIL Challenge 2000 [16]. This database contains 5822 instances with each instance being characterized by 85 features having between two and 41 states. There are two classes. Class labels are removed for our evaluation. The resulting database is referred to as CoIL in the forthcoming.

The other two real databases in the evaluation are obtained by processing the Leukemia database [7]. This database consists of 72 cases of leukemia patients with each case being characterized by the expression levels of 7129 genes. Gene expression levels are discretized into three states via an information theory based method [1]. The resulting database is split into two auxiliary databases: one containing the data of the 47 patients suffering from acute lymphoblastic leukemia (ALL), and the other containing the data of the 25 patients suffering from acute myeloid leukemia (AML). Finally, these two databases are transposed, so that the 7129 genes are the cases and the measurements for the corresponding patients are the features. These databases are simply denoted by ALL and AML, respectively, in the forthcoming discussion. Peña et al. (2003) report three sensible clusters for both the ALL and AML databases.

4.2 Experimental Setting and Performance Assessment

The experimental setting is as follows. The number of randomly created irrelevant features via random fill-in and the significance level for the distribution-free hypothesis testing are $m = 10000$ and $\alpha = 0.05$, respectively. The second step of the hybrid UFSS performs a binary search as SBS with $\beta = 3$ as stopping criterion. Unsupervised learning of NB models is carried out by running the EM algorithm with 10^{-6} as convergence criterion. Finally, the number of clusters is set to $K = 3$ for the Syn10, Syn20, Waveform, ALL and AML databases, and $K = 2$ for the CoIL database.

Let $\boldsymbol{R}$ be a feature subset considered by the filter or hybrid UFSS. We assess the performance of $\boldsymbol{R}$ by measuring both the amount of information about the clusters in $\boldsymbol{d}$ carried by $\boldsymbol{R}$ and the amount of information lost by ignoring the features in $\boldsymbol{I} = \boldsymbol{Y} \setminus \boldsymbol{R}$. We implement this performance score in three steps. First, we obtain a NB model from $\boldsymbol{d}^{(C,\boldsymbol{R})}$ via unsupervised learning. Second, we incorporate the features in $\boldsymbol{I}$ to the model. Third, we compute and report the log-likelihood of $\boldsymbol{d}$, $\log L(\boldsymbol{d})$, given the resulting model. Note that this way of assessing performance allows to compare feature subsets of different sizes.

The features in $\boldsymbol{I}$ are added to the model for $\boldsymbol{d}^{(C,\boldsymbol{R})}$ as dependent on the cluster random variable C but conditionally independent of the rest of the features given C. The probability distributions for these new nodes are estimated from $\boldsymbol{d}$ by re-running the last maximization step of EM algorithm. The fractional completion of the entries for C is the same as in the last expectation step of the EM algorithm, i.e. before the features in $\boldsymbol{I}$ were added to the model. Therefore, no additional evidence propagation is required. Note that we add the features in $\boldsymbol{I}$ to the model for $(C, \boldsymbol{R})$ for evaluation purposes exclusively. That is, they would have never been incorporated into the final model in a real application.

4.3 Results

The first experiment aims at evaluating $RS(Y_i)$ as a means to rank features according to their relevance for unsupervised learning of NB models. For this purpose, we run the filter UFSS on each database in the evaluation until all the features are selected (no matter whether or not they are significantly relevant). We report results in Fig. 1. For each graph in the figure, the vertical axis plots the relevance and performance of the feature subsets considered by the filter

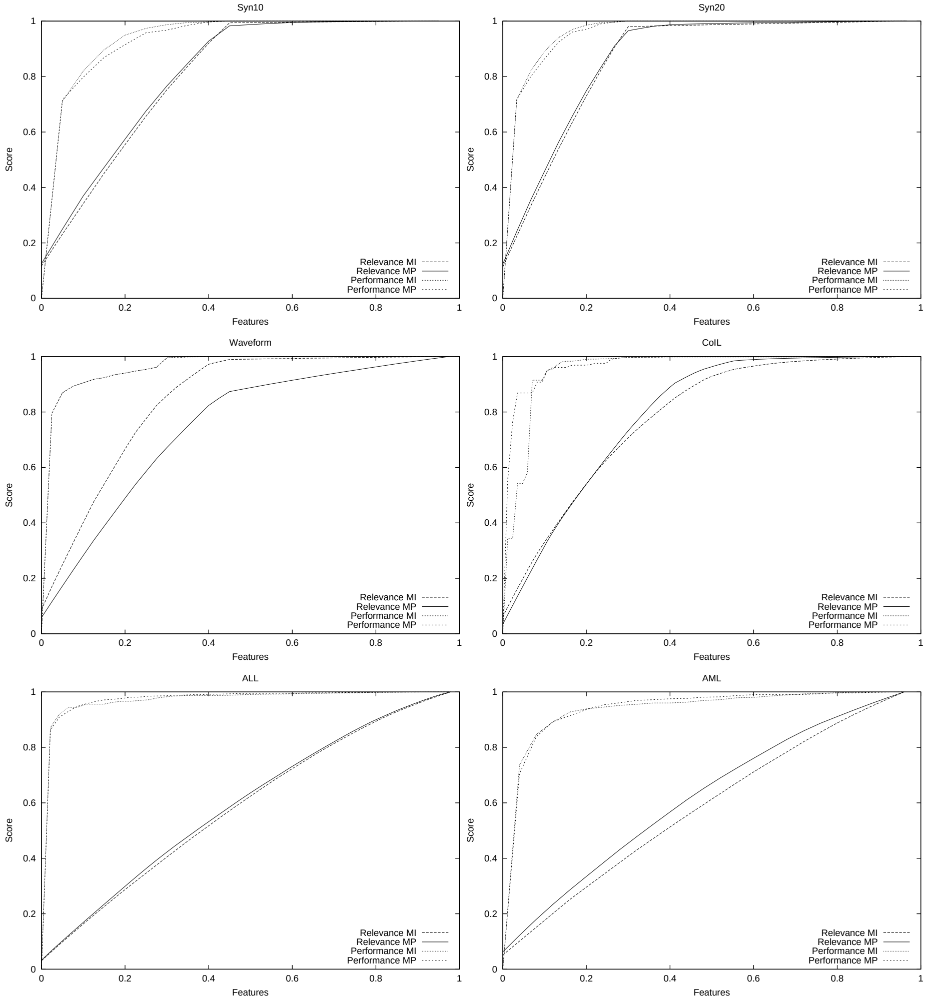

Fig. 1. Feature relevance rankings, and feature subset performance as a function thereof

UFSS, whereas the horizontal axis denotes the fraction of the original features in these subsets. The relevance of a feature subset is computed as the summation of $RS(Y_i)$ for the features in the subset. The performance of a feature subset is assessed as indicated in Section 4.2. In the graphs in Fig. 1, the legends "Relevance MI" and "Relevance MP" denote that $DS(Y_i, Y_j)$ is computed by using $MI(Y_i, Y_j)$ and $MP(Y_i, Y_j)$, respectively. Likewise, the legends "Performance MI" and "Performance MP" indicate that the performance scores correspond to the feature subsets examined by the filter UFSS with $DS(Y_i, Y_j)$ computed as $MI(Y_i, Y_j)$ and $MP(Y_i, Y_j)$, respectively. For the sake of visualization, relevance and performance scores are scaled between 0 and 1.

Table 1. Relevant feature subsets and their performance

		Original		Filter UFSS		Hybrid UFSS	
Database	$DS(Y_i, Y_j)$	$\|\boldsymbol{Y}\|$	$\log L(\boldsymbol{d})$	$\|\boldsymbol{R}\|$	$\log L(\boldsymbol{d})$	$\|\boldsymbol{R}\|$	$\log L(\boldsymbol{d})$
Syn10	$MI(Y_i, Y_j)$	20	−174924	10	−174924	6	−175248
	$MP(Y_i, Y_j)$	20	−174924	10	−174924	6	−175447
Syn20	$MI(Y_i, Y_j)$	30	−242393	10	−242394	6	−242769
	$MP(Y_i, Y_j)$	30	−242393	10	−242394	6	−242887
Waveform	$MI(Y_i, Y_j)$	40	−200439	19	−200439	13	−200528
	$MP(Y_i, Y_j)$	40	−200439	26	−200439	13	−200528
CoIL	$MI(Y_i, Y_j)$	85	−313942	72	−315817	16	−316960
	$MP(Y_i, Y_j)$	85	−313942	39	−315836	16	−316253
ALL	$MI(Y_i, Y_j)$	47	−305304	47	−305304	13	−306879
	$MP(Y_i, Y_j)$	47	−305304	47	−305304	8	−306902
AML	$MI(Y_i, Y_j)$	25	−171940	25	−171940	10	−172755
	$MP(Y_i, Y_j)$	25	−171940	25	−171940	9	−172565

The graphs in Fig. 1 confirm that both instances of $RS(Y_i)$, i.e. with either $MI(Y_i, Y_j)$ or $MP(Y_i, Y_j)$ for $DS(Y_i, Y_j)$ computation, are able to induce effective decreasing relevance rankings of the features of the databases in the evaluation. That is, the lower the value of $RS(Y_i)$, the less relevant the feature is for unsupervised learning of NB models. In other words, the lower the value of $RS(Y_i)$, the less the increase in performance when the feature is included in unsupervised learning of NB models. It should be mentioned that the order in which features are selected by the filter UFSS may vary depending on whether $MI(Y_i, Y_j)$ or $MP(Y_i, Y_j)$ is used in $DS(Y_i, Y_j)$. However, both instances of $RS(Y_i)$ agree on which features are added in the first steps and which ones at the end of the search. As a matter of fact, both instances rank the 10 and 20 truly irrelevant features in the Syn10 and Syn20 databases as the least relevant features. The same applies to the 19 truly irrelevant features in the Waveform database. Note in Fig. 1 that truly irrelevant features score relevance values very close to 0. So do some of the features in the CoIL database, indicating that they are irrelevant. On the other hand, it seems that all the features in the ALL and AML databases are almost equally relevant. This makes sense, because all the patients in each of these databases suffer from the same type of acute leukemia.

The second experiment evaluates the filter UFSS (with the stopping criterion introduced in Section 3.1). Table 1 summarizes the number of features declared relevant as well as their performance for unsupervised learning of NB models as indicated in Section 4.2. It can be appreciated from the table that the filter UFSS performs very well: a considerable number of features are deemed irrelevant for the Syn10, Syn20, Waveform and CoIL databases without degrading significantly the descriptive accuracy of the final models. This contributes to improve the comprehensibility of the final models and the databases themselves. Note that for the Waveform database two truly relevant features are marked as irrelevant when using $MI(Y_i, Y_j)$ (feature 0 and 20). It is known that these features are considerably less relevant than the rest of the truly relevant features [2, 12, 17].

Finally, the third experiment evaluates the hybrid UFSS. It is clear from Fig. 1 and Table 1 that the filter UFSS performs well but too conservatively, specially for the ALL and AML databases. Table 1 reports the results of the hybrid UFSS. The simplification of the final models obtained from the databases in the evaluation is huge, while their performance degradation is always kept under control. Note that the graphs in Fig. 1 support the observation made in Section 3.2 about the non-decreasing performance of unsupervised learning of NB models with respect the fraction of features selected by the filter UFSS.

5 Discussion

Feature subset selection is a central problem in data analysis, as evidenced by the large amount of literature dedicated to it. However, the vast majority of research is carried out within supervised learning, paying little attention to unsupervised learning (e.g. [4, 5, 11, 12, 17]). We contribute in this paper with novel research on feature subset selection for unsupervised learning, also called unsupervised feature subset selection (UFSS).

We approach unsupervised learning from a model-based perspective. Our motivation for performing UFSS is two-fold: gain knowledge of which features are informative (relevant) and which ones are uninformative (irrelevant) for unsupervised learning, and improve the interpretability of the induced model without degrading its descriptive accuracy by discarding irrelevant features.

We propose and evaluate two novel UFSS techniques. The first proposal takes a filter approach based on the observation that features (almost) independent of the rest of the features can be deemed irrelevant. This observation has been made before [6, 12, 17]. In fact, our filter UFSS is inspired by [12], where the authors study a filter UFSS for continuous data. The main difference between both techniques is that our proposal is insensitive to the number of truly irrelevant features, while the method presented in [12] is very sensitive. As a matter of fact, it can become useless when a large number of irrelevant features exist. The second contribution to UFSS that we make is a hybrid filter-wrapper approach aimed at alleviating the conservative behavior of the filter UFSS proposed.

We evaluate the filter and hybrid UFSS for unsupervised learning of naive Bayes models on artificial and real data. However, it should be mentioned that both techniques can be readily applied to unsupervised learning of (unrestricted) Bayesian networks (BNs), i.e. they are not tailored to a particular class of probabilistic graphical models. The results obtained in the evaluation are very encouraging: the learnt models gain in interpretability, while keeping most descriptive accuracy. Unfortunately, a fair comparative study between our proposals and most existing UFSS techniques (e.g. [4, 5, 11, 12, 17]) is difficult: some of these works focus on prediction rather than on description, while some others are coupled with particular unsupervised learning algorithms and/or data types and, therefore, it is hard to fit them into the approach to unsupervised learning this paper deals with. Adapting some of these UFSS techniques to unsupervised learning of BNs may be a line of further research. Another topic for future investigation may be combining our hybrid UFSS with floating search strategies [15], in order to deal properly with non-monotonic performance scores.

References

1. Beibel, M.: Selection of Informative Genes in Gene Expression Based Diagnosis: A Nonparametric Approach. In: Proceedings of the First International Symposium in Medical Data Analysis (2000) 300–307
2. Breiman, L., Friedman, J.H., Olshen, R.A., Stone, C.J.: Classification and Regression Trees. Wadsworth International Group (1984)
3. Cheeseman, P., Stutz, J.: Bayesian Classification (AutoClass): Theory and Results. In: Advances in Knowledge Discovery and Data Mining (1996) 153–180
4. Devaney, M., Ram, A.: Efficient Feature Selection in Conceptual Clustering. In: Proceedings of the Fourteenth International Conference on Machine Learning (1997) 92–97
5. Dy, J.G., Brodley, C.E.: Feature Subset Selection and Order Identification for Unsupervised Learning. In: Proceedings of the Seventeenth International Conference on Machine Learning (2000) 247–254
6. Fisher, D.H.: Knowledge Acquisition Via Incremental Conceptual Clustering. Machine Learning **2** (1987) 139–172
7. Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., Lander, E.S.: Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring. Science **286** (1999) 531–537
8. Jain, A.K., Dubes, R.C.: Algorithms for Clustering Data. Prentice Hall (1988)
9. John, G.H., Kohavi, R., Pfleger, K.: Irrelevant Features and the Subset Selection Problem. In: Proceedings of the Eleventh International Conference on Machine Learning (1994) 121–129
10. Liu, H., Motoda, H.: Feature Selection for Knowledge Discovery and Data Mining. Kluwer Academic Publishers (1998)
11. Mitra, P., Murthy, C.A., Pal, S.K.: Unsupervised Feature Selection Using Feature Similarity. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) 301–312
12. Peña, J.M., Lozano, J.A., Larrañaga, P., Inza, I.: Dimensionality Reduction in Unsupervised Learning of Conditional Gaussian Networks. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 590–603
13. Peña, J.M., Lozano, J.A., Larrañaga, P.: Learning Recursive Bayesian Multinets for Data Clustering by Means of Constructive Induction. Machine Learning **47** (2002) 63–89
14. Peña, J.M., Lozano, J.A., Larrañaga, P.: Unsupervised Learning of Bayesian Networks Via Estimation of Distribution Algorithms: An Application to Gene Expression Data Clustering. Submitted
15. Pudil, P., Novovicová, J., Kittler, J.: Floating Search Methods in Feature Selection. Pattern Recognition Letters **15** (1994) 1119–1125
16. van der Putten, P., van Someren, M. (eds.): CoIL Challenge 2000: The Insurance Company Case. Sentient Machine Research (2000)
17. Talavera, L.: Dependency-Based Feature Selection for Clustering Symbolic Data. Intelligent Data Analysis **4** (2000) 19–28