Minimum redundancy maximum relevancy versus score-based methods for learning Markov boundaries

Silvia Acid, Luis M. de Campos, Moisés Fernández
Departamento de Ciencias de la Computación e Inteligencia Artificial, E.T.S.I. Informática y de Telecomunicación
CITIC-UGR, Universidad de Granada, 18071 – Granada, Spain
Email: {acid,lci,moises}@decsai.ugr.es

***Abstract*—Feature subset selection is increasingly becoming an important preprocessing step within the field of automatic classification. This is due to the fact that the domain problems currently considered contain a high number of variables, and some kind of dimensionality reduction becomes necessary, in order to make the classification task approachable. In this paper we make an experimental comparison between a state-of-the-art method for feature selection, namely minimum Redundancy Maximum Relevance, and a recently proposed method for learning Markov boundaries based on searching for Bayesian network structures in constrained spaces using standard scoring functions.**

***Keywords*-Feature subset selection, minimum Redundancy Maximum Relevance, Bayesian networks, Markov boundary**

I. Introduction

Feature subset selection (FSS) is the technique of selecting a subset of relevant features for building robust classification models. By removing most irrelevant and redundant features from the data, FSS can turn an automatic classification problem approachable (alleviating the curse of dimensionality in problems with a high number of variables), improves model interpretability, speeds up the learning process, reduces the cost of adquisition and storage requirements and may enhance generalization capability (thus bringing improvements in classification accuracy).

There are many algorithms in the literature that perform FSS [6], [13], [14], [16], where they are basically grouped into filter and wrapper approaches. Wrappers use a search algorithm to search through the space of possible features and evaluate each subset according to their predictive power, by running the model (the learning machine of interest) on the subset. Wrappers can be computationally expensive and have a risk of overfitting to the considered data. Filters select subsets of variables as a preprocessing step without utilizing the classifier that will be applied. They use some metrics to assess some intrinsic relation between the features and the class. We are interested in this paper in the filter methods.

The goal of an optimal algorithm for FSS would be to find a minimal set of features that renders the class variable C conditionally independent from the rest of the features in the domain, thus eliminating irrelevant as well as redundant features. In the literature about Bayesian networks, such a set is known as a Markov boundary (MB) for variable C [17], which coincides with the set of strongly relevant features as defined in [14].

A naive approach to obtain the Markov boundary of C could be to learn a complete Bayesian network from data for the domain and then to extract the MB for C from the topology of the graph (the MB of any variable is composed by the set of its parents, children and parents of the children in the graph). But this approach is not practical for high dimensional databases: learning a global Bayesian network would employ a lot of computing effort on discovering superfluous connectivity between variables which are irrelevant or redundant for C. A better option could be to learn a local Bayesian network focused on C and then to compose the MB for it.

The dominant approach in the literature for discovering Markov boundaries is based on the use constraint-based methods, i.e. learning algorithms which try to infer the structure of a Bayesian network through statistical tests of conditional independence on the training data (adapted to the search of MBs) [19], [21]. However, the use of methods based on the score+search paradigm, which seek a structure that best fits the given training data according to a selected scoring function and a search strategy [7], [11], has scarcely been considered.

In this paper we aim to study the capabilities of a recently proposed method for learning MB based on score and search, called RPDMB [3], and compare it with a state-of-the-art FSS method, namely the minimum Redundancy Maximum Relevance (mRMR) algorithm [8], [18]. This is a filter method which does not rank single variables independently in the set but it tries to find a balance between relevance (dependence between the features and the class) and redundancy (dependence among features).

The rest of the paper is organized in the following way: in Section II we describe both the mRMR and the RPDMB algorithms. Section III contains the experimental work comparing these algorithms. Finally, Section IV gives the concluding remarks.

978-1-4577-1675-1

II. THE ALGORITHMS

Let **U** denote a set of unidimensional discrete random variables $\{X_1, X_2, \ldots, X_n\}$ and let C be a distinguished class variable which takes its values in the set $\{c_1, c_2, \ldots, c_k\}$. $S \subseteq \mathbf{U}$ will represent any subset of **U**.

A. The mRMR algorithm

The first idea behind mRMR is that we should not use features which are highly correlated among themselves, i.e. the redundancy between features should be taken into account, thus keeping features which are maximally dissimilar to each other. A way of globally measuring redundancy among the variables in S is

$$W_I(S) = \frac{1}{|S|^2} \sum_{X_i, X_j \in S} MI(X_i, X_j) \tag{1}$$

where $MI(X_i, X_j)$ represents the measure of mutual information between the variables X_i and X_j.

The second idea of mRMR is that minimum redundancy should be supplemented by the use of a maximum relevance criterion of the features with respect to the class variable. A measure of global relevance of the variables in S with respect to C is

$$V_I(S) = \frac{1}{|S|} \sum_{X_i \in S} MI(C, X_i) \tag{2}$$

The simplest way of combining redundancy and relevance to obtain a good subset of features is

$$S^* = \arg\max_{S \subseteq \mathbf{U}} (V_I(S) - W_I(S)) \tag{3}$$

As an exhaustive search of the best subset is impractical, the selected subset is obtained in an incremental way, starting with the feature having maximum value of $MI(C; X_i)$ ($S_0 = \{X_{i_0}\}$) and progressively adding to the current subset S_{m-1} the feature which maximizes

$$\max_{X_j \in \mathbf{U} \setminus S_{m-1}} \left(MI(C, X_j) - \frac{1}{m-1} \sum_{X_i \in S_{m-1}} MI(X_j, X_i) \right)$$

B. The RPDMB algorithm

The RPDMB algorithm is based on the idea of searching for a local (surrounding the class variable) Bayesian network structure in a restricted space of graphs using standard scoring functions, and then extracting the Markov boundary of C. RPDMB is based on a modification of the space of C-RPDAGs, which have been used for building Bayesian classifiers [2]. C-RPDAGs combine the concepts of independent-equivalence and equivalence in classification to get a more reduced search space. This search space is modified in order to adapt it better to the problem of searching MBs (avoiding the tendency of C-RPDAGs to get overfitted models, i.e. MBs having a high number of variables).

Formally, a fringed C-RPDAG is defined in the following way [3]: A partially directed acyclic graph $G = (\mathbf{V}, E_G)$ (E_G is the set of (directed) arcs and (undirected) edges of G), where $\mathbf{V} = \mathbf{U} \cup \{C\}$, is a fringed C-RPDAG iff it satisfies the following conditions:

1 G does not contain any directed cycle.
2 If $Pa_G(C) \neq \emptyset$ then $|Pa_G(C)| \geq 2$ and $Sib_G(C) = \emptyset$.
3 $\forall X_i \in \mathbf{U}$, if $Pa_G(X_i) \neq \emptyset$ then either a) $C \in Pa_G(X_i)$ and either $|Pa_G(X_i)| \geq 2$ or $|Pa_G(C)| \geq 2$, or b) $C \notin Pa_G(X_i)$ and $\forall X_j \in Pa_G(X_i)$, $X_j \in Ch_G(C) \cup Sib_G(C)$,
4 $\forall X_i \in \mathbf{U}$, if $Sib_G(X_i) \neq \emptyset$ then $Sib_G(X_i) = \{C\}$ and $Pa_G(X_i) = \emptyset$.

Where $Pa_G(X_i) = \{X_j \in \mathbf{V} | X_j \rightarrow X_i \in E_G\}$ is the parent set of X_i in G, $Ch_G(X_i) = \{X_j \in \mathbf{V} | X_i \rightarrow X_j \in E_G\}$ is the set of children of X_i in G and $Sib_G(X_i) = \{X_j \in \mathbf{V} | X_i — X_j \in E_G\}$ is the set of siblings of X_i in G.

Figure 1 displays an example of fringed C-RPDAG.

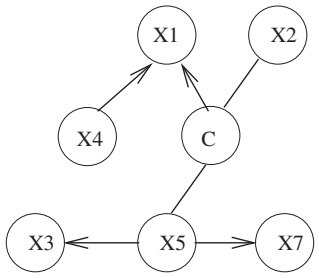

Figure 1. A fringed C-RPDAG.

The search process within this new space uses the twelve operators of addition and deletion considered for C-RPDAGs (see [2] for details) plus two new operators, in order to add and delete the fringes:

- *A_ChildOfChild*(X_i, X_j), addition of the arc $X_i \rightarrow X_j$ if X_i is a child of C (or a sibling of C).
- *D_ChildOfChild*(X_i, X_j), deletion of the arc $X_i \rightarrow X_j$ if X_i is a child of C (or a sibling of C).

Moreover, RPDMB uses a two phase exploration process: a growing phase that expands the current configuration while improving the score, where only the addition operators are considered, followed by a shrinking phase that reviews every connection in the current configuration (only the deletion operators are considered) in order to assess simpler configurations not explored, again while the score improves. These two phases can follow one another repeatedly until no improvement is achieved any more. This strategy is motivated by the observation that in the first stages the operators which are actually applied (improving the scoring function the most) are almost exclusively the operators for addition, which allow to grow the model reflecting the found dependencies. Thus, if we always try to apply the deletion

operators, we spend time evaluating configurations which will be, almost surely, useless.

RPDMB also uses a strategy for pruning bad candidates, that definitively discards candidate edges within the search space, thus pruning many branches in the exploration process. This is inspired in the behavior of many constraint-based algorithms for learning Bayesian networks, where once two variables have been found to be independent given some subset of variables, the corresponding edge connecting them is removed from the graph, and its inclusion is never reconsidered. At any step, every valid operator is tentatively applied to the current model and the new configurations are then assessed, giving to each one a scoring value. Among all these configurations, only the one having the highest score is recorded, the rest are disregarded. Some of these configurations may also improve the score of the current model (although to a lesser extent than the best configuration), whereas other even can make the score worse. As any operator can be evaluated by computing the difference of two local scores, for the addition operators, if this difference is negative (and therefore the global score gets worse), this may be interpreted as the existence of an independence relationship between the two connected nodes [9]. Therefore, those addition operators which turn the score worse obviously constitute a bad choice, and we could record this information in order to avoid to reconsider them definitively. Similarly, for the deletion operators, if the difference of local scores is positive, this means that the existing edge connecting the involved nodes should be removed, i.e. it again points out to an independence relationship and, therefore, we put the edge between these nodes in the list of forbidden links. In this way, if the operator is eventually applied, the removed edge will never be reconsidered for inclusion (see [3] for additional details).

III. Experiments

In this section we are going to experimentally compare the behavior of the two algorithms being considered, RPDMB and mRMR, on several domain problems and databases. RPDMB has been implemented on JAVA in the Weka platform [10], whereas we used the C++ implementation of mRMR provided by H. Peng[1].

A. Data sets and performance measures

We have used several synthetic databases sampled from six well-known Bayesian networks: Alarm [4], Barley [15], Boblo [20], Hailfinder [1], Insurance [5] and Pigs [12], all of them available from the Bayes net repository[2] except Boblo [3]; they come from different domains such as medical diagnosis, insurance risk, meteorology and agriculture. Table I gives some details on the networks from which the databases are sampled: the number of variables, the arcs that they contain and the minimum, maximum and average number of states per variable.

Table I
Characteristics of the Bayesian networks used in the experiments.

Networks	N. vars	N. arcs	N. states (Avg.)
Alarm	37	46	2–4 (2.8)
Barley	48	84	2–67 (8.8)
Boblo	23	24	2–6 (2.7)
Hailfinder	56	66	2–11 (3.3)
Insurance	27	52	2–5 (3.3)
Pigs	441	592	3 (3.0)

Each network has been used to generate five databases of a given sample size. We have also considered five sample sizes, namely 100, 200, 500, 1000 and 10000 (using therefore a total of 150 databases).

An experimental unit consists in selecting a subset of features from the complete set using a variable X as the target variable, for a dataset of a given size in a domain problem, and then comparing the returned subset with the true MB of X extracted from the original network. The learning process is then repeated for each of the variables in the domain and each of the datasets. This represents a total of 15800 experimental units. Each one of these experimental units has been performed by the two algorithms RPDMB and mRMR. The results that we will show are the averages, across the number of variables and across the five datasets of the same size for each domain, of the selected performance measures.

To compare the learned and the true Markov boundaries we use the criteria of precision and recall. Precision is the number of true positives returned by the algorithm (i.e. the number of variables in the returned subset which are also in the true MB), divided by the number of variables in the output (the size of the returned set). Recall is the number of true positives in the output, divided by the size of the true MB. Precision is a measure of exactness, whereas recall is a measure of completeness, both ranging between 0.0 and 1.0. A high value for precision indicates that most of the features selected by the algorithm are correct, whereas a low value suggests that we have wrongly included many variables in MB. On the other hand, a high recall means that the algorithm has selected a large proportion of the attributes that are part of the true MB. None of these two measures makes sense in isolation from each other. For example, a non selective blind algorithm that systematically includes all the features in MB (hence the true positives are necessarily included) would obtain maximum recall but a very low precision; at the other extreme, a very selective algorithm that returns a MB composed of only a single true positive feature would get maximum precision, although the recall

[1]The code is available at http://penglab.janelia.org/proj/mRMR/, although we had to make some modifications of this code.

[2]http://www.cs.huji.ac.il/labs/compbio/Repository

[3]available at http://leo.ugr.es/elvira/

could be very low if the true MB contains many features.

So, it is necessary to consider both measures and combine them into a unique performance measure. The standard way of combining these two measures is the F measure, which is the harmonic mean of precision and recall:

$$F = \frac{2 * precision * recall}{precision + recall} \quad (4)$$

F ranges between 0.0 and 1.0, with $F = 1$ meaning that a totally precise and complete MB has been found.

B. Experimental results.

The results of RPDMB have been obtained by using the BDeu score [11] (with an equivalent sample size equal to 1), although other scoring functions existing in the literature could be considered.

A difficulty with mRMR is that it requires the number of variables to be selected as the input parameter. Therefore, in our experiments we have used, depending on the case, different values for this parameter. More precisely, this parameter has been fixed to the size of the true MB in each experimental unit. Notice that we are giving to mRMR a valuable information that will not be known by RPDMB, which has to estimate the size of the MB.

The results of the experiments with the different domains and sample sizes are displayed in Table II. From these results, first we are going to analyze the behavior of each algorithm depending on the size of the databases in every domain problem.

The filter mRMR is getting better results as the number of samples increases that is, the found subsets are more alike to the true MB. The measures of mutual information are best supported with more data. Exception is made for the Alarm and Hailfinder domains with 10000 samples, which decrease slightly respect to the size 1000.

We can reach the same conclusion observing the outcomes for RPDMB with respect to the different sizes, with the exception of Pigs for sizes 1000 and 10000. Thus, RPDMB improves the similarity of the learned MB to the true MB as the dependences among variables are more supported by the data.

Studying every algorithm through the different domains, we can observe that the domain marks strong differences in the performance measures reached by both algorithms. According to the performance measures of both algorithms, the lower values are got for the Barley domain, that is, the intrinsic relationships encoded in the data are harder to detect. This has been proven to be hard to be detected by other measures too [3].

Finally, we are going to compare the two filters considered in the paper. The best results for every domain and size are displayed in the table in bold font. We can observe that RPDMB gets better results than mRMR most of the time, more precisely 25 times from the 30 trials. The

Table II
F VALUES OF THE ALGORITHMS, FOR DATABASES OF SIZE 100, 200, 500, 1000 AND 10000.

	Alarm	Barley	Boblo	Hailfinder	Insurance	Pigs
			size 100			
mRMR	0.5019	0.2226	0.3817	**0.3490**	**0.4671**	0.4227
RPDMB	**0.5423**	**0.2228**	**0.4124**	0.3152	0.4236	**0.7530**
			size 200			
mRMR	0.5789	0.2803	0.3926	0.3577	**0.5018**	0.4315
RPDMB	**0.6915**	**0.2831**	**0.4535**	**0.3867**	0.4696	**0.9273**
			size 500			
mRMR	0.6187	**0.3729**	0.4822	0.3892	0.5650	0.4347
RPDMB	**0.7931**	0.3368	**0.5527**	**0.5338**	**0.6088**	**0.9879**
			size 1000			
mRMR	0.6452	**0.4285**	0.4983	0.4035	0.5907	0.4362
RPDMB	**0.8570**	0.3923	**0.6443**	**0.6270**	**0.6810**	**0.9992**
			size 10000			
mRMR	0.6436	0.5053	0.5791	0.3912	0.6235	0.4408
RPDMB	**0.8942**	**0.5700**	**0.7959**	**0.7799**	**0.7773**	**0.9985**

RPDMB algorithm has a clearly better behavior in three of the six domains considered with significant differences, more precisely in Alarm, Boblo, and Pigs. It is noticeable the difference between the performance values got by the algorithms in Pigs. Looking more closely at the subsets returned by mRMR, the F values are lower because the algorithm miss the true positive features and includes some wrong ones (false positives) instead. On the other hand, it has to be noted that RPDMB reaches higher values for F (very close to 1.0) for the Pigs domain, that means that the returned MBs were almost perfect for all the nodes (neither false positives nor false negatives). RPDMB is in this case able to estimate correctly the number of nodes composing the true MB.

In the other domains where RPDMB loses in some occasion (Barley, Hailfinder and Insurance), this may be due to the advantage given to mRMR. In these cases the differences in performance between both filters are less significant. When giving the a priori parameter to mRMR, it starts with a valuable information (it may prevent low values for the precision) that is unknown by the RPDMB algorithm. RPDMB gets lower values for the F measure most of the time due to low values for precision, that is to say, the returned MBs contain more false positive in spite of higher values or at least comparable values for true positives with respect to mRMR, resulting in bigger MBs. In general, the differences between the algorithms are clear, resulting in a better behavior for RPDMB except for a few cases. We can also conclude that RPDMB outperforms clearly mRMR when using medium or high sample sizes.

IV. CONCLUDING REMARKS

In this paper we performed an experimental comparison between a well-known state-of-the-art method for feature selection, namely minimum Redundancy Maximum Relevance, and a recently proposed method, RPDMB, for learning Markov boundaries based on searching for Bayesian network structures in constrained spaces using standard

scoring functions (BDeu in the experiments). We have tested both algorithm through a variety of synthetic domains using databases with different sample sizes and we have carried out a comparative study between them.

In order to assess the quality of the resulting selected subsets obtained by each algorithm we compare them with the true MB (available in the originating networks) using the F measure (an aggregation measure of the precision and recall measures).

The experimental outcomes support the conclusion that, in general, the RPDMB algorithm has a better behavior than mRMR for all domains and all database sizes except for few cases. Moreover, we can conclude that RPDMB outperforms clearly mRMR for all the domains when using medium or high sample sizes. Additionaly, RPDMB has been shown to be a robust method for subset feature selection while decreasing the size of the databases.

Although the implementations for the algorithms are quite different, the mRMR is, in general, several times faster than RPDMB due to different computational costs for the algorithms. Nevertheless the additional effort benefits in the quality of the selected feature subsets wherever the domains and the size of the databases.

ACKNOWLEDGEMENT

This work has been supported by the Spanish research programme Consolider Ingenio 2010 and the Consejería de Innovación, Ciencia y Empresa de la Junta de Andalucía under projects MIPRCV:CSD2007-00018 and P09-TIC-4526, respectively.

REFERENCES

[1] Abramson B, Brown J, Murphy A, Winkler, RL (1996) Hailfinder: A Bayesian system for forecasting severe weather. International Journal of Forecasting 12:57–71

[2] Acid S, de Campos LM, Castellano JG (2005) Learning Bayesian network classifiers: searching in a space of partially directed acyclic graphs. Machine Learning 59(3):213–235

[3] Acid S, de Campos LM, Fernández M (2011) Score-based methods for learning Markov boundaries by searching in constrained spaces. Submitted to Data Mining and Knowledge Discovery.

[4] Beinlich IA, Suermondt HJ, Chavez RM, Cooper GF (1989) The alarm monitoring system: A case study with two probabilistic inference techniques for belief networks. In: Proceedings of the European Conference on Artificial Intelligence in Medicine, pp 247–256

[5] Binder J, Koller D, Russell S, Kanazawa K (1997) Adaptive probabilistic networks with hidden variables. Machine Learning 29:213–244

[6] Blum A, Langley P (1997) Selection of relevant features and examples in machine learning. Artificial Intelligence 97(1-2):245–271

[7] Cooper GF, Herskovits E (1992) A Bayesian method for the induction of probabilistic networks from data. Machine Learning 9:309–348

[8] Ding C, Peng H (2005) Minimum redundancy feature selection from microarray gene expression data. Journal of Bioinformatics and Computational Biology 3:185–205

[9] Gámez JA, Mateo JL, Puerta JM (2007) A Fast hill-climbing algorithm for Bayesian networks structure learning. Lecture Notes in Computer Science 4724:585–597

[10] Hall M, Frank E, Holmes G, Pfahringer B, Reutemann P, Witten IH (2009) The WEKA data mining software: An update. SIGKDD Explorations 11(1):10–18

[11] Heckerman D, Geiger D, Chickering DM (1995) Learning Bayesian networks: The combination of knowledge and statistical data. Machine Learning 20:197–243

[12] Jensen CS (1997) Blocking Gibbs sampling for inference in large and complex Bayesian networks with applications in genetics. Ph.D. Thesis, Aalborg University

[13] John GH, Kohavi R (1994) Irrelevant features and the subset selection problem. In: Proceedings of the Eleventh International Conference on Machine Learning, pp 121–129

[14] Kohavi R, John GH (1997) Wrappers for feature subset selection. Artificial Intelligence 97(1-2):273–324

[15] Kristensen K, Rasmussen IA (2002) The use of a Bayesian network in the design of a decision support system for growing malting barley without use of pesticides. Computers and Electronics in Agriculture 33:197–217

[16] Langley P, Sage S (1994) Induction of selective Bayesian classifiers. In: Proceedings of the Tenth Conference on Uncertainty in Artificial Intelligence, pp 399–406

[17] Pearl J (1988) Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufman

[18] Peng H, Long F, Ding C (2005) Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE transactions on Pattern analysis and Machine Intelligence 27(8):1226-1238

[19] Peña JM, Nilsson R, Björkegren J, Tegner J (2007) Towards scalable and data efficient learning of Markov boundaries. International Journal of Approximate Reasoning 45(2):211–232

[20] Rasmussen LK (1995) Bayesian Network for blood typing and parentage verification of cattle. Ph.D. Thesis, Research Centre Foulum, Denmark

[21] Tsamardinos I, Aliferis CF, Statnikov A (2003) Algorithms for large scale Markov blanket discovery. In: Proceedings of the Sixteenth International Florida Artificial Intelligence Research Society Conference, pp 376–380