Incremental Bayesian Network Learning for Scalable Feature Selection

Grégory Thibault[1], Alex Aussem[1], and Stéphane Bonnevay[2]

[1] University of Lyon, LIESP, F-69622 Villeurbanne Cedex, France
[2] University of Lyon, ERIC, F-69622 Villeurbanne Cedex, France

Abstract. Our aim is to solve the feature subset selection problem with thousands of variables using an incremental procedure. The procedure combines incrementally the outputs of non-scalable search-and-score Bayesian network structure learning methods that are run on much smaller sets of variables. We assess the scalability, the performance and the stability of the procedure through several experiments on synthetic and real databases scaling up to 139 351 variables. Our method is shown to be efficient in terms of both running time and accuracy.

1 Introduction

Feature subset selection (FSS for short) is an essential component of quantitative modeling, data-driven construction of decision support models or even computer-assisted discovery. No a priori information or selection of variables is required. Therefore, no previous knowledge premise will bias the final models. The FSS enables the classification model to achieve good or even better solutions with a restricted subset of features, and it helps the human expert to focus on a relevant subset of features. However, databases have increased many fold in recent years and most FSS algorithms do not scale to thousands of variables. Also, large-scale databases presents enormous opportunities and challenges for knowledge discovery and machine learning.

There have been a number of comparative studies for feature selection but few scale up to (say) 100 000 variables. Moreover, findings reported at low dimensions do not necessarily apply in high dimensions. While SVM are efficient and well suited for scalable feature selection [1] (e.g., SVM-RFE stand for SVM Recursive Feature Elimination), there is still much room for improvement. In microarray data analysis for instance, it is common to use statistical testing to control precision (often referred to as the false discovery rate) while maximizing recall, in order to obtain high quality gene (feature) sets. [1] show that none of the above SVM-based method provide such control. Moreover, not only model performance but also robustness of the feature selection process should be taken into account [2]. [3] show experimentally that SVM-RFE is highly sensitive to the "filter-out" factor and that the SVM-RFE is an unstable algorithm. [4, 5] showed recently through extensive comparisons with high-dimensional genomic data that none of the considered feature-selection methods performs best across all scenarios. Thus, there is still room for work to be conducted in this area.

In this paper, we report the use of a probabilistic FSS technique to identify "strongly" relevant features, among thousands of potentially irrelevant and redundant features. A principled solution to the FSS problem is to determine the *Markov boundary* (MB for short) of the class variable. A MB of a variable T is any minimal subset of $\mathbf{U}$ (the full set of variables) that renders the rest of $\mathbf{U}$ independent of T. If the probability distribution underlying the data can be faithfully represented by a Bayesian network, the MB of T is unique. In recent years, there have been a growing interest in inducing the MB automatically from data. Very powerful correct, scalable and data-efficient constraint-based (CB) algorithms have been proposed recently [6–9]. CB discovery methods search a database for conditional independence relations. In contrast to search-and-score methods, CB methods are able to construct the local MB structure without having to construct the whole BN first. Hence their ability to scale up to thousands of variables. This was, so far, a key advantage of CB methods over search-and-score methods.

Our specific aim is to solve the feature subset selection (FSS) problem with thousands of variables using an incremental procedure that combines the result of search-and-score methods run on small sets of variables. We assess the accuracy, the scalability and the robustness of the procedure through several experiments on synthetic and real-world databases scaling up to 139 351 variables.

2 Feature selection

Feature selection techniques can be divided into three categories, depending on how they interact with the classifier. Filter methods directly operate on the dataset, and provide a feature weighting, ranking or subset as output. These methods have the advantage of being fast and independent of the classification model, but at the cost of inferior results. Wrapper methods perform a search in the space of feature subsets, guided by the outcome of the model (e.g. classification performance on a cross-validation of the training set). They often report better results than filter methods, but at the price of an increased computational cost. Finally, embedded methods use internal information of the classification model to perform feature selection (e.g. use of the weight vector in support vector machines). They often provide a good trade-off between performance and computational cost.

Finding the minimal set of features require an exhaustive search among all subsets of relevant variables, which is an NP-complete problem, and may not be unique. In this study, the FSS is achieved in the context of determining the Markov boundary of the class variable that we want to predict. Markov boundary (MB for short) learning techniques can be regarded as in between filter and embedded methods. They solve the feature subset selection (FSS) problem and, in the meantime, they build a local Bayesian network around the target variable that can be used afterwards as a probabilistic classifier.

3 Bayesian networks

For the paper to be accessible to those outside the domain, we recall first the principle of Bayesian network. We denote a variable with an upper-case, X, and value of that variable by the same lower-case, x. We denote a set of variables by upper-case bold-face, $\mathbf{Z}$, and we use the corresponding lower-case bold-face, $\mathbf{z}$, to denote an assignment of value to each variable in the set. We denote the conditional independence of the variable X and Y given $\mathbf{Z}$, in some distribution P with $X \perp_P Y|\mathbf{Z}$. In this paper, we only deal with discrete random variables.

Formally, a BN is a tuple $< \mathcal{G}, P >$, where $\mathcal{G} =< \mathcal{V}, \mathcal{E} >$ is a directed acyclic graph (DAG) with nodes representing the random variables $\mathcal{V}$ and P a joint probability distribution on $\mathcal{V}$. In addition, $\mathcal{G}$ and P must satisfy the Markov condition: every variable, $X \in \mathcal{V}$, is independent of any subset of its non-descendant variables conditioned on the set of its parents, denoted by $\mathbf{Pa}_i^{\mathcal{G}}$.

A Markov blanket $\mathbf{M}_T$ of the T is any set of variables such that T is conditionally independent of all the remaining variables given $\mathbf{M}_T$. A Markov boundary, $\mathbf{MB}_T$, of T is any Markov blanket such that none of its proper subsets is a Markov blanket of T. We say that $< \mathcal{G}, P >$ satisfies the faithfulness condition when $\mathcal{G}$ entails all and only conditional independencies in P.

Theorem 1 *Suppose $< \mathcal{G}, P >$ satisfies the faithfulness condition. Then X and Y are not adjacent in $\mathcal{G}$ iff $\exists \mathbf{Z} \in \mathbf{V} \setminus \{X \cup Y\}$ such that $X \perp_P Y|\mathbf{Z}$. Moreover, for all X, the set of parents, children of X, and parents of children of X is the unique Markov boundary of X.*

A proof can be found for instance in [10]. We denote by $\mathbf{PC}_T$, the set of parents and children of T in $\mathcal{G}$, and by $\mathbf{SP}_T$, the set of *spouses* of T in $\mathcal{G}$. The *spouses* of T are the parents of the children of T. These sets are unique for all $\mathcal{G}$, such that $< \mathcal{G}, P >$ is faithful and so we will drop the superscript $\mathcal{G}$.

Two graphs are said *equivalent* iff they encode the same set of conditional independencies via the d-separation criterion. The equivalence class of a DAG $\mathcal{G}$ is a set of DAGs that are equivalent to $\mathcal{G}$. The next result showed by [11], establishes that equivalent graphs have the same undirected graph but might disagree on the direction of some of the arcs.

Theorem 2 *Two DAGs are equivalent iff they have the same underlying undirected graph and the same set of v-structures (i.e. converging edges into the same node, such as $X \rightarrow Y \leftarrow Z$.*

Moreover, an equivalence class of network structures can be uniquely represented by a completed partially directed DAG (CPDAG), also called a DAG pattern. The DAG pattern is defined as the graph that has the same links as the DAGs in the equivalence class and has oriented all and only the edges common to all the DAGs in the equivalence class.

Algorithm 1 Generic Incremental FSS by MB Search

```
1: function IFSS(𝒟, target, selsize, Vars)
2:     MB ← ∅
3:     repeat
4:         Testvars ← {target} ∪ MB
5:         Testvars ← Testvars ∪ SELECTION(selsize)
6:         G ← BNLEARNING(𝒟, Testvars)
7:         MB ← EXTRACT_MB(G)                          ▷ MB extraction
8:     until stop_criterion
9:     return MB                                       ▷ features = variables in MB
10: end function
```

4 Incremental MB structure learning for scalable FSS

The key idea in this paper is that an incremental procedure could help in alleviating the complexity obstacle by aggregating the outputs of several feature selectors working on much fewer variables. More specifically, a collection of single FSS models is run on small subsets of variables in incremental fashion. The output of one feature selector serves as input to the next. The feature selector used in our method is based on a BN structure identification algorithm.

Algorithm 1 displays our incremental feature selection process based on Markov Boundary search. Input parameters are:

- $\mathcal{D}$: data used for supervised learning,
- $target$: the target variable,
- $selsize$: number of new variables at each iteration,
- $Vars$: set of variables except the $target$ variable.

Standard $BNLearning$ methods do not scale to high-dimensional data sets of variables. The aim of the meta-procedure is to learn many small MBs (in regard to the whole set of variables) from many small subsets of variables. $BNLearning$ can be implemented by any BN structure algorithm. In this study, it is implemented with the GES scoring-based greedy search algorithm discussed by Chickering in [12].

At the beginning, the set of variables, $Testvars$, used to learn a Bayesian network, G, is chosen at random. A first Markov Boundary, MB, is extracted from G. At each iteration, variables in MB are kept into the set $Testvars$ and some other variables are added by a uniform random selection without replacement. The size of this selection, $selsize$, is adapted according to the size of the Markov Boundary, MB. Our variables selection process assumes that, in the first part of the algorithm, each variable of $Vars$ is selected once; then, when all variables have been selected once, the process restart with the whole set of variables, $Vars$. The algorithm stops when all variables have been selected twice. At the end, the selected features are returned. Under the faithfulness assumptions and assuming that the induction algorithm is correct, IFSS return

the correct Markov Boundary. This a sample limit property. In practice, our hope is to output the features that GES would have found on the complete database.

Indeed, after the first part of algorithm (when all variables have been selected once), MB contains all the parents and the children of the target, because by definition, the variable adjacent to the target cannot be d-separated from the target, given any other variable. During the second part of the algorithm (when all variables have been selected at least twice), the spouses of the *target* enter the candidate MB set.

5 Experiments

In this section, we assess the accuracy, the scalability and the robustness of IFSS through several empirical experiments on benchmark data sets. We use a state-of-the-art search-and-score BN structure learning algorithm called GES as our BN learner ($BNLearning$). First, we compare IFSS against GES in terms of accuracy on several synthetic data sets. Second, we assess the scalability of IFSS on a high-dimensional data sets that was provided at the KDD-Cup 2001. Third, we assess the IFSS's robustness.

5.1 Accuracy

Benchmark	# var	# edges	target	MB size	# samples
ASIA	8	8	OR	5	10 000
ASIA8	64	64	OR	5	10 000
ALARM	37	46	HR	8	30 000
INSULIN	35	52	IPA	18	50 000
INSURANCE	27	52	Accident	10	30 000
HAILFINDER	56	66	Scenario	17	50 000

Table 1. Description of the Bayesian networks used in these experiments to assess the comparative accuracy of IFSS and GES Markov boundary discovery on the target variable.

We report here the results of our experiments on six common benchmarks: ASIA, ASIA8 (ASIA tilled 8 times), ALARM, INSULIN, INSURANCE and HAILFINDER, (see [8] and references therein). For ASIA8, the tiling is performed in a way that maintains the structural and probabilistic properties of the original network, ASIA, in the tiled network. Description of the benchmarks is showed in Table 1. For each benchmark, 10 databases with independent and identically distributed samples were generated by logic sampling. The amount of data was chosen large enough to avoid the bias due to a lack of data. The task is to learn the MB of the variable that appears in the third column in Table 1. The size of the MB varies from 5 to 18 variables as may be observed. We compare IFSS against GES in terms of true positive rate (TPR, i.e., the number of

true positives variables in the output divided by the number of variables in the output), false positive rate (FPR, i.e., the number of false positives divided by the the number of variables in the output), the Kappa index (κ), the weighted accuracy (WAcc), computed as the average of the accuracy on true positives and the accuracy on true negatives and finally, the time in seconds. Kappa is a measure that assesses improvement over chance is appropriate. The following ranges of agreement for the Kappa statistic suggested in the literature are: poor $K < 0.4$, good $0.4 < K < 0.75$ and excellent $K > 0.75$. In all our experiments, GES is trained to maximize the Bayesian Dirichlet scoring criterion defined as :

$$BD(\mathcal{B} \mid \mathcal{D}) = p(\mathcal{B}) \cdot \prod_{i=1}^{n} \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij})}{\Gamma(N_{ij} + \alpha_{ij})} \prod_{k=1}^{r_i} \frac{\Gamma(N_{ijk} + \alpha_{ijk})}{\Gamma(\alpha_{ijk})}$$

Note that no *a priori* information structure is used for tests on synthetic data (*i.e.*, $p(\mathcal{B})$ is uniform). Moreover, the prior on parameters is set so as to be non-informative, that is, an equivalent uniform Dirichlet prior with an equivalent sample size (ESS) equal to the greatest variable modality (see [10] for details). Table 2 summarizes the average performance indexes over 10 runs for each benchmark. As may be observed, IFSS performs as well as GES on all benchmarks, except on INSURANCE where IFSS outperform GES by a noticeable margin. This is quite a surprise as IFSS was not designed to outperform the underlying BN structure learning algorithm (here GES) but only to be scalable.

	GES					IFSS				
	κ	TPR	FPR	WAcc	Time	κ	TPR	FPR	WAcc	Time
ASIA	0.959	1.000	0.050	0.975	0.10	0.959	1.000	0.050	0.975	0.06
ASIA8	0.867	1.000	0.024	0.988	27.87	0.834	1.000	0.031	0.984	1.29
ALARM	0.916	0.875	0.000	0.938	6.79	0.916	0.875	0.000	0.938	2.56
INSULIN	0.840	0.933	0.094	0.920	15.16	0.870	0.933	0.063	0.935	7.83
INSURANCE	0.663	0.700	0.063	0.819	5.81	0.858	0.860	0.019	0.921	2.60
HAILFINDER	0.589	0.571	0.037	0.767	48.71	0.517	0.471	0.016	0.727	6.19

Table 2. Average performance of IFSS (with GES as underlying BN structure learning algorithm) and GES

In Table 3, the same indexes of accuracy are reported; the aim is to recover the MB given by GES (and not the true MB anymore). For instance, a True Positive is a variable given by GES and found by IFSS, *etc.*. IFSS is very close to GES in most cases. Some significant differences are observed on Hailfinder between IFSS and GES, output of IFSS is closer to the true MB than output of GES (see in Table Table 2). Moreover, the last column indicates the time saving when IFSS is used instead of GES.

	IFSS against GES				
	κ	TPR	FPR	WAcc	Time saving
ASIA	1.000 ± 0.000	1.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	0.04 ± 0.04
ASIA8	0.900 ± 0.107	0.940 ± 0.102	0.014 ± 0.014	0.963 ± 0.055	26.58 ± 1.72
ALARM	1.000 ± 0.000	1.000 ± 0.000	0.000 ± 0.000	1.000 ± 0.000	4.23 ± 2.21
INSULIN	0.959 ± 0.049	0.968 ± 0.037	0.007 ± 0.021	0.980 ± 0.023	7.34 ± 1.31
INSURANCE	0.730 ± 0.058	0.863 ± 0.031	0.111 ± 0.037	0.876 ± 0.025	3.21 ± 2.47
HAILFINDER	0.490 ± 0.093	0.530 ± 0.200	0.062 ± 0.058	0.734 ± 0.077	42.52 ± 33.73

Table 3. Average performance of IFSS where the task is to recover the variables output by GES

5.2 Scalability

In this section, experiments demonstrate the ability of IFSS to solve a real world FSS problem involving thousands of features. We consider the THROMBIN database which was provided by DuPont Pharmaceuticals for KDD Cup 2001. It is exemplary of a real drug design [13]. The training set contains 1909 instances characterized by 139351 binary features. The features describe the three-dimensional properties of the compounds. Each compound is labelled with one out of two classes, either it binds to the target site or not. The task of KDD Cup 2001 was to learn a classifier from 1909 given compounds (learning data) in order to predict binding affinity and, thus, the potential of a compound as anti-clotting agent. The classifiers submitted to KDD Cup 2001 were evaluated on the remaining 634 compounds (testing data) as the weighted average (WAcc) of the accuracy on true binding compounds and the accuracy on true non-binding compounds. The THROMBIN database is challenging for three reasons. First, it has a huge number of features. Second, the learning data are extremely imbalanced: Only 42 out of the 1909 compounds bind. Third, the testing data are not sampled from the same probability distribution as the learning data, because the compounds in the testing data were synthesized based on the assay results recorded in the learning data. Scoring higher than 60% accuracy is impressive as noted in [6].

IFSS, with GES as the MB learner, was run 61 times in the time we have disposed for our experiments, with a prior over structures arbitrary fixed to $10^{-16\times f}$, where f is the number of free parameters in the DAG. The outputs were used as input of Naive Bayesian Classifier, and a classification on the test data was perfomed. As shown in Figure 3, IFSS scores between 36% (really bad) to 71% with an average 55% and only 46 runs of IFSS score more than 50% weighted accuracy, i.e. the random classifier. These results are comparable to MBOR [7] and IAMB [14] that achieve respectively 53% (over 10 runs) and 54% (both over 114 runs). This is however worse than PCMB [6] that achieves 63% (over 114 runs). Of course, we have no idea what GES scores on such data since GES do not scale to such high-dimensional database. Note that each launch of IFSS lasted approximately 3 hours, which is the same order of magnitude as the other algorithms mentioned above.

Nonetheless, the best MB over 61 runs consists of five variables 3392, 10695, 23406, 79651 and 85738. This MB is depicted in Figure 2. It scores 71,1% which is impressive according to [13, 6]. It worth mentioning that J. Cheng, the winner of the KDD cup 2001, only scores 71.1% accuracy and 68.4% weighted accuracy with four variables: 10695, 16794, 79651 and 91839. He used a Bayesian classifier to assess the accuracy of his feature set. It is shown in Figure 1. As may be seen, two variables are common with the winner's selection. IFFS outputs the THROMBIN MB in about 220 minutes on our laptop (2.6GHz Intel® Core™ 2 Duo with 1 GB of RAM). Of course, this time is highly dependent our MATLAB® implementation, and may significantly be reduced if written in C/C++ for instance.

The Figure 5 represents the ROC curves of the classifier given by IFSS with the best MB as input. The area under ROC curve is a well-known performance measurement. The ROC curve is the 2-D plot of sensitivity and 1-specificity acquired by applying a sequence of arbitrary cut-off threshold to the probabilities generated by the predictive model. A clear difference is observed between the ROC curve on the test set (in plain line) and the ROC curve on the training set (in dotted lineline, obtained by 10-fold cross validation). The reason is that the testing data was not sampled from the same probability distribution as the learning data, hence the difficulty of the task. The area under curve (AUC) is 0.6978 on the test set. This classifier scores 69% (and 71% when constructing a naive BN with the same variables) which seems highly competitive compared to PCMB [15] and IAMB [14] that achieves respectively 63% and 54% as shown in [6]. Table 4 reports the scores obtained with the best MB classifiers constructed from the sets of variables given by the respective algorithms.

	IFSS					Cheng				
	κ	TPR	FPR	Acc	WAcc	κ	TPR	FPR	Acc	WAcc
Output model	0.420	0.467	0.085	0.809	0.691	0.316	0.633	0.264	0.711	0.684
NaiveBN	0.437	0.547	0.120	0.801	0.713	0.297	0.600	0.258	0.708	0.671
SVM	0.464	0.500	0.076	0.823	0.712	0.312	0.313	0.056	0.795	0.629
RForest	0.439	0.513	0.099	0.809	0.707	0.312	0.313	0.056	0.795	0.629

Table 4. Results of classifiers with the output-model of the algorithm, the naive bayesian network model, the support vector machine classifier and the random forest classifier

5.3 Robustness

When using FSS on data sets with large number of features, but a relatively small number of samples, not only model performance but also robustness of the FSS process is important. For instance, in microarray analysis, domain experts clearly prefer a stable gene selection as in most cases these genes are subsequently analyzed further, requiring much time and effort [16]. With such

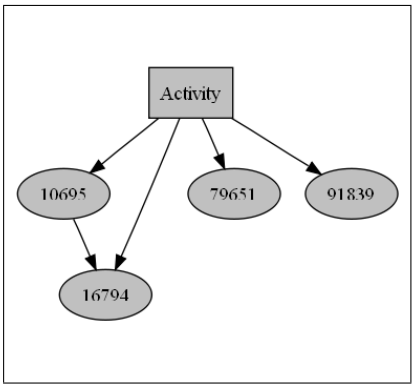

Fig. 1. BN of the KDD Cup winner

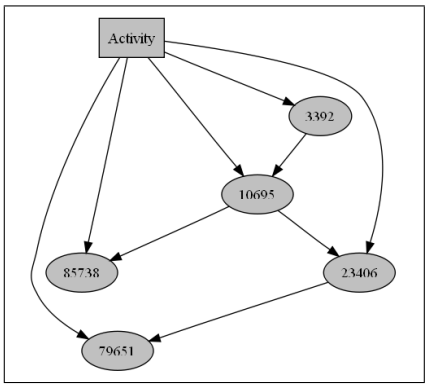

Fig. 2. Best MB output by IFSS

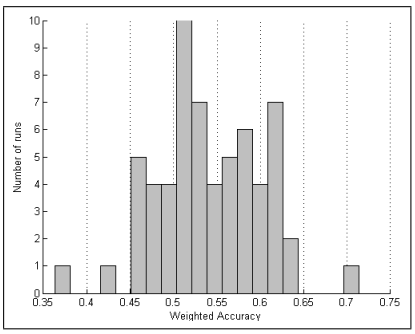

Fig. 3. Weighted accuracies of 61 runs of IFSS.

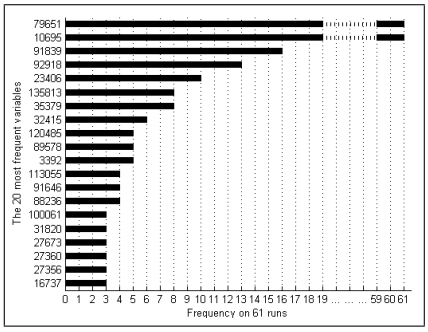

Fig. 4. Frequencies of twenty most frequent variables over 61 runs of IFSS.

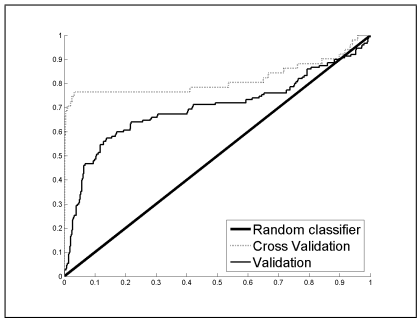

Fig. 5. ROC curves of the best MB output by IFSS on test set and on the training set using 10-fold cross-validation.

high-dimensional databases, all FSS algorithms are subject to some variability. Surprisingly, the robustness of FSS techniques has received relatively little at-

tention so far in the literature. As noted in [2], robustness can be regarded from different points of view: perturbation at the instance level (e.g. by removing or adding samples), at the feature level (e.g. by adding noise to features), or variation of the parameter of the FSS algorithm, or a combination of them. Here, we focus on the robustness of FSS selector as the variation of the output with respect to a random permutation of the variables. We consider again the 61 times runs of IFSS on THROMBIN data. A simple ensemble technique proposed in [2, 16] works by aggregating the feature rankings provided by the FSS selector into a final consensus ranking weighted by frequency. The variables returned by IFSS mostly differ by one or two variables. The top 20 ranked variables are shown in Figure 4 in decreasing order of frequency in the output of IFSS. As we can see, the variables 79651 and 10695 were always selected. These variables are also present among the four features of the winner of the KDD cup in 2001, and variable 79651 is always present in the top 10 MB output by KIAMB (see [6]). The third most frequent feature, namely 91839, is also one of the four features of the winner of the KDD cup.

Following [17], we take a similarity based approach where feature stability is measured by comparing the 61 outputs of IFSS. We use the Jaccard index as the similarity measure between two subsets S_1 and S_2. The more similar the outputs, the higher the stability measure. The overall stability can be defined as the average over all pairwise similarity comparisons between the $n = 61$ MBs:

$$I_{tot} = \frac{\sum_{i=1}^{n} \sum_{j=i+1}^{n} I(S_i, S_j)}{n(n-1)} \quad \text{with} \quad I(S_i, S_j) = \frac{|S_i \bigcap S_j|}{|S_i \bigcup S_j|}$$

An average of 0.336 (with a standard deviation of 0.116) was obtained.

6 Conclusion

We discussed a new scalable feature subset selection procedure. This procedure combines incrementally the outputs of non-scalable search-and-score Bayesian network structure learning methods that are run on much smaller sets of variables. The method was shown to be highly efficient in terms of both running time and accuracy. Future substantiation through more experiments with other BN learning algorithms are currently being undertaken and comparisons with other FSS techniques will be reported in due course.

References

1. Nilsson, R., Peña, J., Björkegren, J., Tegnér, J.: Evaluating feature selection for svms in high dimensions. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD. (2006)
2. Saeys, Y., Abeel, T., de Peer, Y.V.: Robust feature selection using ensemble feature selection techniques. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases, ECML PKDD, Antwerp, Belgium (2008) 313–325

3. Tang, Y., Zhang, Y., Huang, Z.: Development of two-stage svm-rfe gene selection strategy for microarray expression data analysis. IEEE-ACM Transactions on Computational Biology and Bioinformatics **4** (2007) 365–381
4. Ma, S., Huang, J.: Penalized feature selection and classification in bioinformatics. Briefings in Bioinformatics **5** (2008) 392–403
5. Hua, J., Tembe, W., Dougherty, E.: Performance of feature-selection methods in the classification of high-dimension data. Pattern Recognition **42** (2009) 409–424
6. Peña, J., Nilsson, R., Björkegren, J., Tegnér, J.: Towards scalable and data efficient learning of markov boundaries. International Journal of Approximate Reasoning **45**(2) (2007) 211–232
7. Rodrigues de Morais, S., Aussem, A.: A novel scalable and data efficient feature subset selection algorithm. In: European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases ECML-PKDD'08, Antwerp, Belgium (2008) 298–312
8. Tsamardinos, I., Brown, L.E., Aliferis, C.F.: The max-min hill-climbing bayesian network structure learning algorithm. Machine Learning **65**(1) (2006) 31–78
9. Yaramakala, S., Margaritis, D.: Speculative markov blanket discovery for optimal feature selection. In: IEEE International Conference on Data Mining. (2005) 809–812
10. Neapolitan, R.E.: Learning Bayesian Networks. Prentice Hall (2004)
11. Pearl, J.: Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference. Morgan Kaufmann (1988)
12. Chickering, D.M.: Optimal structure identification with greedy search. Journal of Machine Learning Research **3** (November 2002) 507–554
13. Cheng, J., Hatzis, C., Hayashi, H., Krogel, M., Morishita, S., Page, D., Sese, J.: KDD Cup 2001 Report. In: ACM SIGKDD Explorations. (2002) 1–18
14. Tsamardinos, I., Aliferis, C.F., Statnikov, A.R.: Algorithms for large scale markov blanket discovery. In: FLAIRS Conference. (2003) 376–381
15. Peña, J., Björkegren, J., Tegnér, J.: Scalable, efficient and correct learning of markov boundaries under the faithfulness assumption. In: 8th European Conference on Symbolic and Quantitative Approaches to Reasoning under Uncertainty (ECSQARU 2005). Volume 21., Lecture Notes in Artificial Intelligence 3571 (2005) 136–147
16. Aussem, A., Rodrigues de Morais, S., Perraud, F., Rome, S.: Robust gene selection from microarray data with a novel Markov boundary learning method: Application to diabetes analysis. In: European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty ECSQARU'09. To appear. (2009)
17. Kalousis, A., Prados, J., Hilario, M.: stability of feature selection algorithms: a study on high-dimensional spaces. Knowl. Inf. Syst. **12** (2007) Knowl. Inf. Syst.