Identifying Markov Blankets with Decision Tree Induction

Lewis Frey[1], Douglas Fisher

Dept. of Electrical Engineering and Computer Science, Box 1679 Station B, Nashville, TN 37235
Frey@vuse.vanderbilt.edu, DFisher@vuse.vanderbilt.edu

Ioannis Tsamardinos, Constantin F. Aliferis, Alexander Statnikov
[1]*Dept. of Biomedical Informatics, Vanderbilt University, 2209 Garland Avenue; Nashville, TN 37232*
Ioannis.Tsamardinos@, Constantin.Aliferis@, Alexander.Statnikov@vanderbilt.edu

Abstract

The Markov Blanket of a target variable is the minimum conditioning set of variables that makes the target independent of all other variables. Markov Blankets inform feature selection, aid in causal discovery and serve as a basis for scalable methods of constructing Bayesian networks. This paper applies decision tree induction to the task of Markov Blanket identification. Notably, we compare (a) C5.0, a widely used algorithm for decision rule induction, (b) C5C, which post-processes C5.0's rule set to retain the most frequently referenced variables and (c) PC, a standard method for Bayesian Network induction. C5C performs as well as or better than C5.0 and PC across a number of data sets. Our modest variation of an inexpensive, accurate, off-the-shelf induction engine mitigates the need for specialized procedures, and establishes baseline performance against which specialized algorithms can be compared.

1. Introduction

The Markov Blanket (MB) is the minimum conditioning set that makes all other features independent for a particular target. Tsamardinos & Aliferis [20] show that in faithful Bayesian Networks Markov Blankets consist of *strongly relevant* features as defined in relation to optimal classifiers [11]. The Markov Blanket is particularly useful for data sets with large numbers of features. For example, gene expression data has tens of thousands of features. Arnone [3] has estimated that on average, four to eight genes interact in higher organisms. Thus, an algorithm that finds the Markov Blanket of genes could reduce the data set by three orders of magnitude.

The Markov Blanket can be used for causal discovery under the condition of *faithfulness*, which is defined in Section 2. In essence, it can reduce the number of variables that need to be tested experimentally to discover direct causes of a target *T*.

Additionally, Markov Blanket discovery can be used for guiding Bayesian Network construction. The Markov Blanket for each variable is identified and used as a guide to construct the Bayesian Network for the domain. Margaritis and Thrun [15] use this approach for Bayesian Network structure learning.

Several algorithms have been developed or proposed for identifying Markov Blankets ([13], [15], [21], [22]). Because of the relationship between Markov Blankets and feature relevance, and because decision tree induction has been used to perform feature selection ([2], [7]), we evaluate the ability of C5.0 [19], an inexpensive, off-the-shelf induction engine, to identify Markov Blankets. Our evaluation motivates a modest post-processing step that demonstrably yields very accurate, low cost approximation of Markov Blankets. Our study thus forwards a scalable, accurate, and accessible algorithm for Markov Blanket identification. The observation that classification-motivated rule induction performs well at MB identification, also highlights the common principles of feature relevance that underlie induction of classifiers and Bayesian Networks [20].

The remainder of the paper reviews basic concepts of Bayesian Networks, describes C5.0, our variation called C5C, and the PC algorithm [18], a prototypical algorithm for Bayesian Network construction, which we use for comparison purposes. An experimental comparison of PC, C5C, and C5.0 evaluates how accurately each method finds the Markov Blankets for target variables in a variety of domains. The paper concludes with a discussion of the results.

2. Bayesian Networks

A *Bayesian Network, <V,G,J>,* consists of a set of variables *V*, a directed acyclic graph *G* and a joint probability distribution *J* over the variables *V* where the

Markov Conditions holds (i.e., a variable is independent from all variables other than its descendants when conditioned on its parents) [18].
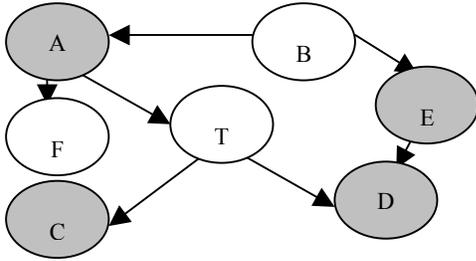

Figure 1. Example Bayesian Network Graph

The *Markov Blanket* (MB) of a target variable is the minimal set of variables X in V such that when a variable T in V is conditioned on X, any variable in $F \subseteq V$-X-T is independent of T (i.e., $P(T|XUF)=P(T|X)$).

In Bayesian Networks (BN) the union of parents and children of T, and parents of children (spouses) of T is equivalent to the Markov Blanket [17]. In Figure 1, the Markov Blanket for T is $\{A, C, D, E\}$. This means that variables B and F are independent of T conditioned on $\{A, C, D, E\}$.

A Bayesian Network $N = <V, G, J>$ is *faithful* to joint probability distribution J over feature set V if and only if all dependencies entailed by G and the Markov Condition are also present in J. A data-generating process K is faithful to N, if K in the sample limit produces data with joint probability distribution P, and N is faithful to P. A data set is faithful to a BN N if in the sample limit the data was generated by a data generation process that is faithful to N.

Some algorithms work well in identifying the MB from faithful data. If the data set is faithful to the BN, then the MB of each variable is unique (proof in appendix) and the comparison of the algorithms' ability to find the unique MB is valid. If the data is unfaithful, then the MB is not unique. This can occur, for example, when deterministic relationships exist in the data [10]. Parity functions are examples of relationships that are not faithful. No current causal discovery methods can learn parity functions and most Bayesian methods assume faithfulness.

This paper focuses on the problem of inducing the Markov Blanket of the target concept in faithful distributions. For example, in the Bayesian Network depicted in Figure 1, the task of the algorithm is to predict A, C, D and E as the Markov Blanket and reject B and F as members of the Markov Blanket given a data set that is generated by the Bayesian Network. The algorithms for finding the Markov Blanket are discussed below.

3. Algorithms

This section describes the methods for Markov Blanket identification that we will compare. PC and C5.0 provide baseline performance to which C5C can be compared.

3.1 PC

The PC algorithm [18] starts with a fully connected, unoriented Bayesian Network graph and goes through three phases. In phase I, the algorithm eliminates edges by using the criterion that variable A has a direct edge to variable B if, and only if, for all subsets of features there is no subset S, such that A is independent of B conditioned on S. In phases II and III, the algorithm orients the edges by performing global constraint propagation. If the algorithm is not able to orient some edges, the output is actually a class of structurally equivalent BNs. The PC algorithm uses significance thresholds based on the G^2 statistic. In linear domains, Fisher's z-test is employed.

PC is intractable on densely connected data sets with a large number of variables. The algorithm has a complexity on the order of the number of variables raised to the maximal degree, d, for any node in the graph (i.e., $O(|V|^d)$). Consequently, it is exponentially bounded in the maximal degree. Similar complexity limitations hold for search and score Bayesian methods. The algorithm by Cheng et al. [9] improves the complexity (to $O(|V|^4)$) by introducing a distributional assumption ("monotone restriction") with properties currently being explored. In practice, these algorithms cannot run on more than a few hundred variables.

3.2 C5.0

A C5.0 decision tree is constructed using *GainRatio*. *GainRatio* is a measure incorporating entropy. Entropy ($E(S)$, Eq. 1) measures how unordered the data set is. It is denoted by the following equation when there are classes $C_1, ..., C_N$ in data set S where $P(S_c)$ is the probability of class C occurring in the data set S:

$$E(S) = - \sum_{c=1}^{N} P\left(S_c\right) * log_2 P\left(S_c\right) \qquad (1)$$

Information Gain is a measure of the improvement in the amount of order.

$$Gain(S, V) = E(S) - \sum_{v \in values(V)} \frac{|S_v|}{|S|} * E(S_v) \qquad (2)$$

Gain has a bias towards variables with many values that partition the data set into smaller ordered sets. In order to reduce this bias, the entropy of each variable over its *m* variable values is calculated as *SplitInfo* (Eq. 3).

$$SplitInfo(S, V) = \sum_{i=1}^{m} -\frac{|S_i|}{|S|} * log_2 \frac{|Si|}{|S|} \qquad (3)$$

GainRatio (Eq. 4) is calculated by dividing *Gain* by *SplitInfo* so that the bias towards variables with large value sets is dampened.

$$GainRatio(S, V) = \frac{Gain(S, V)}{SplitInfo(S, V)} \qquad (4)$$

C5.0 builds a decision tree greedily by splitting the data on the variable that maximizes gain ratio. CF-pruning is set to 25% (the default setting for C5.0) in this paper.

A final decision tree is changed to a set of rules by converting the paths into conjunctive rules and pruning them to improve classification accuracy. When using C5.0 to predict the Markov Blanket, each variable involved in any rule output by C5.0 is predicted to be in the Markov Blanket of the target/class variable.

3.3 C5C

The working hypothesis is that frequently occurring features in C5.0 production rules provide a good approximation of the *MB(T)*. This hypothesis is encapsulated as a simple augmentation of C5.0 to count the number of variable occurrences in C5.0 rules and use the most frequent to predict the Markov Blanket. This is called the C5C algorithm and it uses the C5.0 rules output to order the variables from most likely MB variables to least likely.. The modification consists of a simple script that counts the occurrence of the variables in the C5.0 rules output (i.e., if variable occurs in rule, then increment the variable's frequency count) A frequency threshold is chosen to distinguish between Markov and non-Markov Blanket variables.

Three methods will be used to identify the MB given the C5C ranking of variables. First is choosing the best frequency threshold given knowledge of the true MB (this is used as a comparison in sections 6.1 and 6.2 and is intended as a best-case analysis). The second strategy finds the frequency threshold that gives the best accuracy

of C5.0 on a test set. The third strategy employs the G^2-test to test for independence.

4. Experimental Methods

The experiments involve generating data from six Bayesian Networks and comparing the prediction of Markov Blankets by C5.0, C5C and PC. The first five data sets are publicly available Bayesian Networks: Alarm [5], Hailfinder [1], Insurance [6], Mildew and Barley [12]. These data sets are generated using a logic sampling data-generating approach, implemented in the Hugin package [4], run to produce a data set of 20,000 discrete valued instances. The last data set is an Artificial Bayesian Network that is constructed to examine the effect of sample size on learning.

Using these generated data sets, C5.0, C5C and PC's abilities to reconstruct the Markov Blanket for each variable in the data sets are compared. Since the original networks are available, the predicted Markov Blankets can be compared against the actual Markov Blankets in terms of sensitivity and specificity. *Sensitivity* is the ratio of correctly predicted MB variables over true MB variables. *Specificity* is the ratio of correctly predicted (i.e., excluded) non-MB variables over true non-MB variables. Ratios for predicted Markov Blankets of different sizes and characteristics are compared.

In comparing C5.0 and C5C, a measure of closeness to the true MB is determined by calculating the distance (*dist*) of sensitivity (*sen*) and specificity (*spec*) of a predicted MB to that of the true MB (Eq. 5). Thus, the smaller the distance, the closer the predicted MB is to the true MB.

$$dist = \sqrt{(1 - sen)^2 + (1 - spec)^2} \qquad (5)$$

The area under the Receiver Operating Characteristic (ROC) curve is calculated and serves as a measure of how well C5C and PC output the Markov Blanket with different thresholds. For C5C, we vary the frequency threshold, and for PC we vary the significance threshold. A one hundred percent area under the ROC curve equates to exactly identifying the Markov Blanket and fifty percent area is guessing.

5. Data Sets

5.1 Generated data from Bayesian Networks

A data set generated from the Alarm Network, a medical monitoring network, is used to explore how well the algorithms find the Markov Blankets for all of the

variables in the network. The alarm network has 37 variables, each of which has its own Markov Blanket.

The data set generated from the Hailfinder Network has 56 variables, which are used for severe weather forecasting [1].

The Insurance network has 27 variables and is used for estimating expected claim costs for insurance policies [6].

The data set generated from the Mildew Network has 35 variables, which determine amounts of fungicides to use against attacks of mildew on wheat.

The Barley data has 48 variables, which help predict yield and quality parameters for growing barley without pesticides [12].

5.2 Artificial Bayesian Network

A data set generated from an artificial Bayesian Network is used to explore the effect of the number of variables and the sample size upon the algorithms' ability to find the Markov Blanket. For this data set, the Markov Blanket is found for only one variable. The Markov Blanket for this variable has three parents, two children and one parent of a child for a total of six variables. The Markov Blanket for this variable is kept constant over the conditions of increasing noisy-variables and reduced sample size.

The above data sets are examined because their true Markov Blankets are known. Because the predicted Markov Blankets of the methods are compared against the true Markov Blankets, this is necessary.

6. Results

6.1 Comparisons between C5C and C5.0

In Table 1, the average sensitivity, specificity and distance over target variables for C5.0 and the best thresholds for C5C are listed for the first five Bayesian Networks. Table 1 illustrates that C5C's average predicted MB is closest to the true MB. Examining the sensitivity and specificity values, it appears that C5C's distances are improved by the increase in specificity without the loss of sensitivity. Discarding low frequency variables from the MB usually causes non-MB variables to be discarded.

Table 1. Average over target variables of sensitivity (sen), specificity (spec) and distance (dist) for C5.0 and the best thresholds for C5C. Data sets have 20,000 instances. The asterisk (*) denotes the mean distance for C5C is significantly different from C5.0 by the paired Wilcoxon signed rank test of the equality of means (p<0.05).

DATA SET	C5.0			C5C		
	Sen	Spec	Dist	Sen	Spec	Dist
ALARM	0.83	0.84	0.32	0.82	0.99	0.18*
HAILFINDER	0.81	0.27	0.89	0.80	0.98	0.22*
INSURANCE	0.89	0.46	0.64	0.78	0.93	0.25*
MILDEW	0.94	0.11	0.95	0.80	0.90	0.26*
BARLEY	0.84	0.43	0.67	0.76	0.94	0.28*

Table 2 shows that for 156 variables (75% of the total variables) C5C's predicted MB is closer to the true MB than C5.0's predicted MB. C5C is equal to C5.0 for the remaining 47 variables.

Table 2. Number of target variables (var) out of the total for each data set that the distance of C5C's best predicted MB is closer to the true MB than C5.0.

DATA SET	FREQ THAT C5C IS CLOSER TO TRUE MB THAN C5.0	TOTAL VAR
ALARM	15	37
HAILFINDER	46	56
INSURANCE	22	27
MILDEW	34	35
BARLEY	39	48

Table 3 shows the improvement in average distance over the 156 variables where C5C is closer to the true MB than C5.0. It also shows the reduction in the average predicted MB size from C5.0 to the best C5C thresholds.

Table 3. Average Markov Blanket size and average distance to the true MB for C5.0 and best threshold for C5C. The asterisk (*) denotes significance by the paired Wilcoxon signed rank test of the equality of means (p<0.05) in comparing C5.0 and C5C distance.

DATA SET	AVG MB SIZE		DISTANCE	
	C5.0	C5C	C5.0	C5C
ALARM	16	3	0.40	0.05*
HAILFINDER	49	4	0.93	0.12*
INSURANCE	19	6	0.67	0.20*
MILDEW	31	6	0.98	0.27*
BARLEY	34	7	0.73	0.22*

Table 3 also demonstrates that C5C reduces the size of the predicted MB more than C5.0. The quality of the predicted MB is of interest. This also supports the hypothesis that the low frequency variables tend to be

non-Markov Blanket variables. This is why there can be such a reduction in the predicted MB size while improving the average distance.

6.2 C5C Identifying Markov Blanket variables

The results in Tables 1-3 show that C5C tends to rank MB variables higher than non-MB variables. Two methods that use C5C ranking of variables are tested in their ability to identify the MB: method one uses decision tree test set accuracy and method two uses the G^2 test.

For method one, C5.0 decision tree test set accuracy is assumed to improve when only MB variables are used in the training set. The filtered variable set with the best accuracy should correspond to the Markov Blanket because the Markov Blanket is the complete set of strongly relevant features. Variables are added to the training set for the decision tree starting with the variable(s) with the highest C5C ranking and proceeding to the lowest via a wrapper method [8]. The accuracy of the classifier is recorded for each of these subsets of variables. The subset with the highest test accuracy is proposed as the MB. The results in Table 4 are for running C5.0 with training set with sample size 16,000 for the training set and 4,000 for the test set.

The second method uses the G^2-test [18], to test for independence between two variables given a conditioning set (p < 0.05). A weakness of the G^2-test is that its validity is sensitive to the amount of sample. The larger the Markov Blanket conditioning set the more sample needed (i.e., sample grows exponentially with conditioning set size). For this method all variables with zero C5C frequency counts are classified as non-Markov Blanket variables. Then the n top ranked variables of C5C are proposed as the conditioning set for the G^2-test. The top k ranked variables (where $k > n$) are then tested for independence from the target variable given the conditioning set. This is a heuristic approach to identify a larger MB than the sample size supports for the G^2-test. If the variables are not independent, they are included in the final MB. Each variable in the conditioning set is then tested for independence given the other features in the conditioning set. Those that are not independent are included in the final MB. In Table 4 $n=4$ and $k=10$ and the sample size is 20,000. The conditioning set is size 4 because with a sample of 20,000 the G^2-test tends to be valid for the data sets examined. This can be increased with larger sample size or reduced with smaller sample size.

One of the benefits of the ordered C5C output is that it provides a subset of the variables (i.e., k variables with high frequency counts) to be examined for Markov Blanket candidacy. For the examined data sets, the parameter k is arbitrarily set to 10. To minimize the exclusion of Markov Blanket variables, it is recommended that k is greater than the expected quantity of Markov Blanket variables. For example, in some biological data set, Arnone [3], there are on average less than eight interacting variables, so $k = 10$ is reasonable and $k = 15$ would be more conservative. Since the assumption of C5C is that variables in the C5.0 rules reflect the Markov Blanket, all variables with zero frequency counts (i.e., they do not occur in C5.0 rules) are not considered candidates for the Markov Blanket. For some data sets examined, many of the variables in the C5C output have zero frequency.

Table 4. Average over target variables of sensitivity (sen), specificity (spec) and distance (dist) for C5C with the C5.0 decision tree test set accuracy determining threshold and the G^2-test identifying the MB. For the test accuracy the training set is 16,000 instances and the test set is 4,000 instances. The G^2-test uses 20,000 instances. The asterisk (*) and plus (+) denote the mean distance for the method is significantly different from C5.0 (Table 1) by the paired Wilcoxon signed rank test of the equality of means at $p<.05$ and $p< 0.1$, respectively.

DATA SET	C5C – TEST ACC.			C5C – G^2		
	Sen	Spec	Dist	Sen	Spec	Dist
ALARM	0.79	0.97	0.23	0.83	0.97	0.20^+
HAILFINDER	0.76	0.95	0.27*	0.79	0.88	0.29*
INSURANCE	0.74	0.80	0.42*	0.76	0.88	0.31*
MILDEW	0.75	0.85	0.38*	0.78	0.80	0.35*
BARLEY	0.71	0.91	0.36*	0.76	0.87	0.31*

Table 4 shows that the two methods improve the identification of the MB over the C5.0 algorithm (see Table 1 C5.0 dist column). For some data sets the distances are close to the "best distance" (Table 1 C5C dist column), but both methods are significantly different from the "best distance". In comparing the two methods against each other they are not significantly different (p <0.05) except for Hailfinder and Insurance. This means the two methods are comparable in their ability to identify the MB.

6.3 Comparisons between C5C and PC

C5C and PC are run on the Bayesian Network data for each variable. The average area under the ROC is then computed for the algorithms by plotting the sensitivity and specificity values for the thresholds.

Six thresholds are used with C5C. The first threshold proposes all variables in the C5.0 rules as the Markov Blanket for the target variable (0% of variables in rules are excluded). The second threshold excludes from the Markov Blanket variables in the C5.0 rules whose counts are infrequent (20% of the most frequent feature count for

the given target variable). For example, if the most frequent variable in the rules occurred 100 times, then the second threshold would exclude any variable in the C5.0 rules that occurred fewer than 20 times. The thresholds occur in 20% increments (i.e., 20%, 40%, 60%, 80% and 100%). The area under the ROC curve drops if C5.0 rules don't contain Markov Blanket variables or if infrequent variables in the C5.0 rules are part of the Markov Blanket.

It is computationally cheap to get multiple thresholds from C5.0's output file, unlike PC that must be run for each threshold, so a finer granularity of thresholds (one hundred and one) are examined to see if better ROC area can be obtained. The thresholds are incremented by 1% of the maximum count variable in the C5.0 rules.

The six thresholds for the PC algorithm are 0.005, 0.01, 0.02, 0.05, 0.10 and 0.20. These thresholds are significance levels for the G^2 test used by the algorithm. The smaller the threshold, the higher the confidence level that the variables are independent. This means the smaller the threshold, the fewer the number of edges removed from the fully connected graph used by PC to determine Markov Blankets. PC must be run separately for each threshold confidence level.

Table 5 shows that both PC and C5C performed comparatively well in finding the Markov Blankets for Alarm, Hailfinder and Insurance data sets (i.e., not significantly different) across both sets of thresholds.

C5C does better than PC on both the Mildew and Barley data sets for both thresholds (Table 5). This is because these data sets have variables with large value sets (100 values). This condition makes it difficult for PC to eliminate edges from the fully connected graph.

Table 5. Average ROC over all variables in Network: PC & C5C with 6 thresholds (T'holds) and C5C with 101 thresholds. Data sets have 20,000 instances. The asterisk (*) denotes the mean ROC for C5C as significantly different from PC by the paired Wilcoxon signed rank test (p<0.05).

Data Set	Total Targets	PC	C5C	
# of T'holds		6	6	101
Alarm	37	96.3	91.1	91.4
Hailfinder	56	91.7	87.5	88.9
Insurance	27	82.0	86.3	88.0
Mildew	35	64.3	80.0*	88.0*
Barley	48	50.0	81.6*	85.9*

In summary, Table 5 shows that C5C is not significantly different from PC for three data sets and is significantly better for two data sets. C5C behaves similarly with 6 and 101 thresholds so the distinction will be dropped in further analysis. Importantly, C5C can be

run on data with many more variables than is reasonable with PC.

6.4 C5C over increasing noisy-variables and reduced sample size

Figure 2 shows that PC and C5C have similar performance across sample size and number of variables. C5C appears to do better at smaller sample size across the number of variables. C5C's greedy nature works in its favor while PC tends to get lower ROC for small sample size because it starts with a fully connected graph and needs enough evidence to remove edges. If there is not enough evidence to remove edges, then specificity is low.
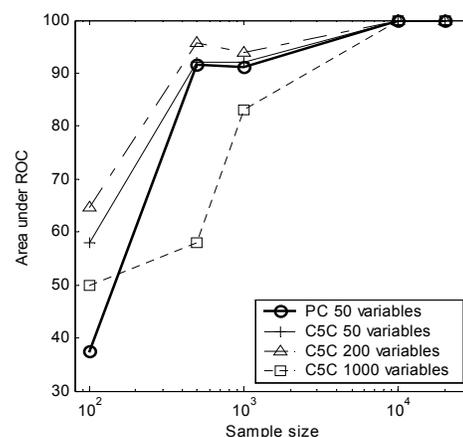

Figure 2. Area under the ROC for MB size 6 for PC with 50 variables and C5C with 50, 200 and 1,000 variables. The artificial Bayesian Network sample sizes are 100, 500, 1,000, 10,000 and 20,000 examples.

When there are a larger number of noisy variables (e.g., 1,000 variables), the C5C algorithm needs more data to distinguish the MB. Still, as Figure 2 demonstrates, C5C is able to find the Markov blanket for large sample and larger variable size.

C5C is also time-efficient. The largest computation (i.e., 1,000 variables for a sample size of 20,000) takes 30 seconds to perform on a 2.4GHz Xeon desktop machine for all thresholds.

7. Discussion

C5C is a simple modification to C5.0 that approximates the Markov Blanket as well as or better than C5.0 and PC in these experiments. Additionally, for the constructed Bayesian Network there is evidence that smaller sample sizes have good values of area under the ROC curve. This reduction in sample size could be due to the greedy nature of the algorithm. It chooses the best

variables (usually Markov Blanket variables) given the small sample. The C5C algorithm is quick, in that it greedily hill climbs through the space of possible decision trees.

Note that the concept of causal structure within a data set is, in general, different than the goal of improving classification accuracy. C5.0 is designed to improve classification accuracy. Thus, there is no explicit bias encoded into the algorithm for finding causal structure. C5.0 outputs a decision tree and a set of rules with the goal of learning to classify the data set with highest accuracy.

7.1 Limitations

Due to this bias, the C5C algorithm is not able to find the Markov Blanket for target variables that occur a disproportionate number of times in one class. In such a case, C5.0 implementation uses default classification and assigns the variable to the majority class. This happens once in Insurance, twice in Barley and three times in Alarm. When a default classification occurs, no rules are formed so no variables can be part of the Markov Blanket. This cannot be remedied by changing the pruning level.

The C5C algorithm is not able to predict the Markov Blanket when one variable predicts the target variable without error, even though there are other variables in the Markov Blanket. The single variable becomes the only variable in the rules so all other variables are excluded from the Markov Blanket. Changing the pruning level does not affect the rules when one variable perfectly predicts the target variable. This happens for eight variables in Hailfinder. However, this is an unfaithful distribution because there are deterministic relationships in the data set.

A related approach is that of Li et al. [14] who use Genetic Algorithms and frequently occurring features in their classifiers to determine feature relevance in gene expression data. Their motivation is establishing a set of predictors that are robust to noise. C5C, even in non-noisy data, gives a good approximation of the MB.

7.2 Future Work

Recall that the task of feature selection is to find a subset that maintains the optimal classifier. Markov blankets have been mapped to the concept of strongly relevant features in feature selection [21]. Hence, this simple and scalable method for finding the Markov Blanket is widely applicable in the area of feature selection in machine learning and data mining.

The method of using the C5C ranking of features to identify the minimal feature subset for improvement in classification accuracy had some success (Table 4). The thresholds of C5C produce different filtered feature sets. These feature sets can be used as input into the C5.0 decision tree algorithm with cross-validated accuracy. The filtered feature set with the best accuracy should correspond to the Markov Blanket because the Markov Blanket is the complete set of strongly relevant features. G^2-test method generally performed better than the decision tree accuracy test. The MB from this test could also be used as a filtered feature set to improve classification accuracy.

Additional areas of future exploration are methods for selecting pruning levels and weighting the importance of features as opposed to straight counting. The weighting could include factors such as size of rule, accuracy of rule and its coverage.

8. Conclusion

It has been shown empirically that a simple post-processing step using the C5C algorithm performs at least as well as the PC algorithm when finding the Markov Blanket on the data sets compared. The method of using the ranked C5C variables in conjunction with performing G^2-tests (i.e., the cheapest of the two methods that we examined) can be used to identify the Markov Blanket variables. In addition, C5.0 scales up very well. Since C5.0 can be used to obtain the Markov Blanket right out of the box, applying this counting algorithm, C5C, for useful applications is a simple matter.

9. Acknowledgements

10. Appendix

Denote d-separation of X and Y by Z in the graph of BN C, as $D_C(X ; Y | Z)$. Denote independent X and Y given Z in the probability distribution as $I(X;Y|Z)$. Denote the set of parents and children of T in the graph of BN C as $PC_C(T)$. Denote the set of the parents, children, and spouses of T in the graph of BN C as $MB_C(T)$.

Proposition 1: A faithful BN C is a perfect map of independencies and dependencies of the data. In other words, the d-separation criterion gives all the dependencies and independencies: $D_C(X;Y|Z) \Leftrightarrow I(X;Y|Z)$ [17].

Proposition 2: In a faithful BN C on variables V, X and T have a direct edge between them, if and only if $\forall S \subseteq V \backslash \{T,X\}, \neg I(X;T|S)$ [18].

Proposition 3: In any BN C on variables V, the set of parents, children, and spouses of T d-separates T from any other node in V [16].

Proposition 4: If BN N is faithful, then for every variable T, $MB(T)$ is unique and is the set of parents, children, and spouses of T [20].

<u>Theorem 1.</u> For any two BNs C and N, both faithful to the same distribution J on a variable set V, and for any variable T in that distribution, $PC_C(T) = PC_N(T)$ (i.e., the set $PC(T)$ is unique in all faithful BNs).

<u>Proof:</u> By Proposition 2 we get $X \in PC_C(T) \Leftrightarrow \forall S \subseteq V \setminus \{T, X\}, \neg I(X;T|S) \Leftrightarrow X \in PC_N(T)$.

<u>Theorem 2</u>: For any two BNs C and N, both faithful to the same distribution J on a variable set V, and for any variable T in that distribution, $MB_C(T) = MB_N(T)$ (i.e., the set $MB(T)$ is unique for all faithful BNs).

Proof: Let us denote the set of parents, children, and spouses of T in C with $MB'_C(T)$. By Proposition 4 we have that $MB'_C(T) = MB(T)$. By the same proposition we also have that $MB'_N(T) = MB(T)$. So, $MB'_C(T) = MB'_N(T)$. □

11. References

[1] Abramson, B., Brown, J., Edwards, W., Murphy, A. and Winkler, R.L, Hailfinder: A Bayesian system for forecasting severe weather, *International Journal of Forecasting*, (1996), 12, 57-71.

[2] Aluallim, H., and Dietterich, T. G., Learning with many irrelevant features, *Proceedings, Ninth National Conference on Artificial Intelligence*, Anaheim, CA. AAAI Press/ The MIT Press, (1991), pp. 547-552.

[3] Arnone, M.I. and Davidson, E.H., The hardwiring of development: organization and function of genomic regulatory systems, *Development*, (1997), 12(4), pp. 1851-1864.

[4] Andersen, S.K., Olesen, K.G., Jensen, F. V. and Jensen, F., HUGIN - a shell for building bayesian belief universes for expert systems, in *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, (1989), pp. 1080-1085.

[5] Beinlich, I., Suermondt, G., Chavez R. and Cooper G., The ALARM monitoring system: A case study with two probabilistic inference techniques for belief networks, *Proceedings of 2'nd European Conference on AI and Medicine*, Springer-Verlag, Berlin, (1989).

[6] Binder, J., Koller, D., Russell, S. and Kanazawa, K., Adaptive Probabilistic Networks with Hidden Variables, *Machine Learning*, (1997), 29, 213-244.

[7] Cardie, C., Using decision trees to improve case-based learning, in *Proceedings of the Tenth International Conference on Machine Learning*, (1993), pp. 25-32.

[8] Caruana, R. and D. Freitag, Greedy Attribute Selection, *in International Conference on Machine Learning*, (1994).

[9] Cheng, J., Greiner, R., Kelly, J., Bell D. and Liu, W., Learning Bayesian networks from data: an information-theory based approach, *Artificial Intelligence*, (2002), *137,* pp. 43-90.

[10] Glymour, C. & Cooper, G.F., *Computation, Causation and Discovery*, AAAI/The MIT Press, (1999) .

[11] Kohavi, R. and G.H. John, Wrappers for Feature Subset Selection, *Artificial Intelligence*, (1997), 97(1-2), pp. 273-324.

[12] Kristensen, K. and Rasmussen, I.A., 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*, (2002), 33, pp. 197-217.

[13] Koller, D and M. Sahami, Toward Optimal Feature Selection, in *Thirteenth International Conference in Machine Learning*, (1996), pp. 284-292.

[14] Li, L., Pedersen, L.G., Darden, T.A. and Weinberg, C.R., Computational Analysis of Leukemia Micorarray Expression Data Using the GA/KNN Method, CAMDA'01, (2001).

[15] Margaritis, D. and Thrun, S., Bayesian Network Induction via Local Neighborhoods, Technical Report: CMU-CS-99-134, (1999).

[16] Neapolitan, R.E., *Probabilistic Reasoning in Expert Systems: Theory and Algorithms*, John Wiley and Sons (1990).

[17] Pearl, J., *Probabilistic Reasoning in Intelligent Systems*, San Mateo, CA: Morgan Kaufman, (1988).

[18] Spirtes, P., C. Glymour, and R. Scheines, *Causation, Prediction and Search*, Cambridge, MA: MIT Press, (2000).

[19] Quinlan, J.R., Induction of decision trees, *Machine Learning*, (1987), 1 pp. 81-106.

[20] Tsamardinos, I. and Aliferis, C.F., Towards Principled Feature Selection: Relevancy, Filters and Wrappers, in *Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics*, (2003).

[21] Tsamardinos, I., Aliferis, C.F. and Statnikov, A., Algorithms for Large Scale Markov Blanket Discovery to appear, in *Proceedings of the 16th International FLAIRS Conference*, (2003).

[22] Tsamardinos, I., Aliferis, C. F., and Statnikov A., Time and Sample Efficient Discovery of Markov Blankets and Direct Causal Relations**,** in *The Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, (2003).