Algorithms for Large-Scale Local Causal Discovery and Feature Selection In the Presence Of Limited Sample Or Large Causal Neighbourhoods

C.F. Aliferis and I. Tsamardinos

Department of Biomedical Informatics,
Discovery Systems Laboratory,
Vanderbilt University

Technical Report DSL-02-08
October 8th 2002

Introduction

When the Causal Neighbourhood of a target node T is so large relative to the available sample then one cannot obtain statistically reliable tests of independence and association metrics when conditioning on all members of the causal neighbourhood. Thus both the growing and the shrinking phase of the algorithms in the IAMB family [Aliferis and Tsamardinos 2002a] will fail to give reliable results. Such situations arise often for example in text categorization, in studies with limited sampling (e.g., gene expression experiments), or when one wishes to learn the causal neighbourhood of a constellation of variables (e.g., in epidemiology when learning the causal neighbourhood of a "disease" that in effect summarizes the states of several variables some or all of which are also measured).

We give here algorithms that circumvent the problem by trading off less sample (equivalently large causal neighbourhood when the sample is fixed) with computational complexity and with the possibility that false positives may be introduced in the estimated causal neighbourhood. We collectively call these algorithms S/LCN (Small Sample/Large Causal Neighbourhood). The basic structure for all such algorithms is identical, so we will first outline the main concept and see how different members of this family can be created by changes in the basic structure. We will also discuss how the same techniques we introduced for the IAMB family for chunking and parallelization can be directly applied in the S/LCN family to provide distributed processing/storage versions.

The Main Algorithm

S/LCN algorithm // basic structure
Input: - dataset D,
- target variable T,
- a parameter *desired_output* // indicating whether the desired output is *direct-causes- effects* or *Markov-blanket* ,
- the maximum conditioning set cardinality *k* allowed by the sample //alternatively *k* can be computed by the algorithm itself on the basis of statistical or heuristic criteria

Output: causal neighbourhood CN of T
1. Start with an empty CN
2. repeat
3. Phase I: use a heuristic function *F* to admit a candidate variable Vi into CN such that *F* returns a non-zero value for every variable that is a direct cause or direct effect of T
4. Phase II (interleaved):
   repeat
   eliminate from CN every member Vj for which there is a subset CNp of CN with size *k* such that: Independent (Vj : T| CNp)
   until no more members of CN can be eliminated
5. until no more variables can be admitted in CN (i.e., all variables outside CN have *F* value 0)
6. If *desired-output=direct-causes-effects* then return CN

7. else if *desired-output=Markov-blanket* then return:
Union$_i$ (S/LCN(D, Vi, *direct-causes-effects*) //where Vi is a member of CN)

The output of the algorithm can be post-processed by running PC, FCI [Spirtes et al., 2000], or custom methods to further filter out false positives or identify the presence of hidden variables. Such methods are given in note #e.

Properties & Other Notes

a. We require the same assumptions as for the IAMB family [Alifeis and Tsamardinos 2002] (i.e., assumption 1: faithfulness; assumption 2: i.i.d.; assumption 3: causal sufficiency; assumption 4: reliable independence tests for given sample and *k*). Under these assumptions the algorithm will not miss a member of the true causal neighbourhood but may introduce false positive members.
Proof: Direct causes and effects: First we need to show that a function exists that has the property we required for *F*, namely *F* returns a non-zero value for every variable that is a direct cause or direct effect of T. Trivial such functions are: mutual information (T, V), or simply a constant value of 1. Now we only need to show that phase II never eliminates a direct cause or effect of T. Indeed, by conditioning on a subset of the true set of causes and effects, due to assumptions 2 and 3 we cannot eliminate a direct cause or direct effect because this independence would violate faithfulness (assumption 2). However it is possible that *k* is smaller than the number of connecting paths between a variable X and T; hence such an X that is also not a direct cause or direct effect will not be excluded from the causal neighbourhood.
Markov Blanket: the union of direct causes and direct effects of every member of the set of direct causes and direct effects of T contains the Markov blanket of T by definition of the Markov Blanket in faithful graphs. However, it is possible for a variable X to belong to this union without belonging to the Markov Blanket of T (e.g., when it is a parent of a parent of T).

b. We will introduce several S/LCN algorithms each using a different *F* function. Proving that all return non-zero values for direct causes and effects of T is trivial and will be omitted.

(i) *F* is mutual_information(V ; T | CN) when |CN| is no larger than *k* or: mutual information(V, T) otherwise. We will call algorithm S/LPC when the goal is to find the set of direct causes and direct effects, and S/LMB when the goal is to find the Markov Blanket of T.

(ii) *F* is: Max(Min(association(Target variable ; Vi | subsetj))). Where the maximum is taken over all variables not yet selected, while the minimum over all subsets of size *k* of CN. The choice of association metric varies with the particular distribution (for example in the multivariate normal linear case, Fisher's z-test can be employed; in the multinomial case G^2, and so on). We will call the algorithm MMPC when the goal is to find the set of direct causes and direct effects of T, and MMMB ("McubeB") when the goal is to find the Markov Blanket of T.

(iii) *F* is: constant set to 1.

(iv) *F* is: same as heuristic I of PC [Spirtes et al. 2000]
(v) *F* is: same as heuristic II of PC [Spirtes et al. 2000]
(vi) *F* is: same as heuristic III of PC [Spirtes et al. 2000]. We will call algorithms (iii) to (vi) LocPC1, LocPCh1, LocPCh2, LocPCh3 respectively when the goal is to find the set of direct causes and direct effects of T; we will call them LocMB1, LocMBh1, LocMBh2, LocMBh3 respectively when the goal is to find the Markov Blanket of T.

c. Paralellization of all these algorithms is straightforward and very similar to parallelization of the IAMB family [Aliferis and Tsamardinos 2002]: all *F* functions discussed can be distributed among several processing nodes and the results collected and applied toward phase II. Phase II can also be directly parallelized by indexing all independence tests to be carried out according to variable member of the causal neighbourhood that is tested for exclusion. Then by dividing the task by variable to all available processors. To avoid unnecessary computation, once a variable is eliminated all nodes get notified to update their conditioning sets accordingly Once a node eliminates all its candidates it can request surplus variables from other nodes. Similarly, chunking of computation either to address data that does not fit the main memory or for parallelization in straightforward as in the case of the IAMB family algorithms.

d. When the number of paths between any candidate variable and T intersected with the paths from the direct causes and effects to T is smaller or equal than *k,* then the algorithms will return exactly the direct causes and effects of T.

e. Instead of running PC or FCI to filter out some false positives from the returned Markov Blanket, the following method can be used: omit all variables V of the estimated Markov Blanket such that (Independent (V ;T | S1) and not_Independent(V ; T | S2)) where S1, S2 are non-overlapping non-empty subsets of the estimated Markov Blanket of size less or equal to *k* . This criterion will not be satisfied by direct causes of T (first part), or by direct effects of T (first part), or by direct cases of direct effects of T (second part). But will catch direct causes of descendants of T that also have connecting non-colliding paths to T. The remaining members of the Markov Blanket can further be post-processed using the criterion of *symmetry*: if *A* is in the Markov Blanket of *T* then *T* is in the Markov Blanket of *A* too. If we find that *A* is in the output of a S/L Markov Blanket algorithm but *T* is not in the output of the algorithm when inducing the Markov Blanket of *A*, then we can safely eliminate *A* from the Markov Blanket of *T*.

f. We postulate (without proof currently) that under the distributional restriction of *non-synthesis*, the algorithms return strictly the direct causes and direct effects of T.

g. A variation on the max-min heuristic is to instead of *min* to take the average of the left tail of the distribution of values of: association(V;T | subsetj) to address possible errors due to sampling variance. Bootstrapped methods can also be employed to this effect (for all metrics discussed) but they are computationally more expensive.

h. If one wishes to derive not only the direct causes, and/or direct effects but also indirect ones up to depth *d* recursive application of the induction and filtering methods gives the straightforward solution. Similarly for the induction of a more extended Markov Blanket of depth *d.*

i. A formal complexity analysis of the algorithms will not be given here. We note that clearly the main algorithm in phase II is exponential in the worst case. What makes the algorithms applicable for very-large-scale discovery is that in practical settings *k* is always small (because it is constrained by the available sample). When the true neighbourhood is small and synthesis does not occur, the algorithms are expected to run very fast. In experiments with a structural biology dataset with 2,000 cases and 140,000 variables the algorithm runs on a single PC in a few hours using interpreted matlab code. In gene expression experiments with 12,600 variables and 203 samples the algorithms run in minutes or few hours on the same computing platforms. Several simulated data experiments verify these efficiency claims. By optimizing code and employing parallel versions, such algorithms could be run in a few minutes or even seconds for such very-large-scale problems.

The following experiments show the quality and efficiency of the new algorithms in real and simulated data:

i.1 Quality of local Algorithms in Massive Real Dataset for finding the Markov Blanket

We applied local algorithms to the Thrombin dataset from KDD Cup 2001 [Cheng 2002]. The problem involves identification of active chemical substances with respect to binding to Thrombin (a key receptor in blood clotting), on the basis of molecular properties of each substance (the semantics of each variable were not released). *This dataset has more than a hundred thousand (139,351) variables, a large number of observations (2,543) and an equally large feature-to-sample ratio (up to 107 in our experiments depending on the data split).* Since the causal structure of the domain is unknown, we estimate the quality of causal discovery by the accuracy of classifiers trained with only the variables of the suggested output *MB(T)*: if the methods outputs the real *MB(T)*, the accuracy should be close to optimal (depending on the power of the classifier). Moreover there is no smaller variable set that gives better classification performance.

The classifier families used in our experiments were: *Linear and Polynomial Support Vector Machines* (LSVM and PSVM) [Vapnik 1992, Burges 1998, Scholkopf et al. 1999], *Neural Networks* (NNs), *K-Nearest Neighbors* (KNN), and the *Simple Bayes Classifier* (SBC) [Mitchell 1997]. The baseline comparison variable selection method was the state-of-the-art Recursive Feature Elimination [Guyon et al. 2002] and the method of comparison is the area under the ROC curve for different thresholds of confidence (as output by the different classifiers) for classification. A separate (from the training set) test set was used for estimating accuracy.

As can be seen from Table 1, the local algorithms produce small sets with excellent classification performance relative to using all the variables and Recursive Feature

Elimination. This is independent of the classifier used. Furthermore, the local algorithms produce small and highly-specific variable sets compared to the SVM method.

	ALGORITHM				
CLASSIFIER	**IAMB**	**IAMB Chunked**	**MMMB**	**RFE**	**All Variables**
LSVM	88.3%	93.5%	94.8%	93.3%	93.4%
PSVM	86.30%	93.8%	93.4%	92.5%	93.6%
SBC	92.8%	93.2%	94.8%	85.2%	80.3%
KNN	94.0%	91.2%	93.8%	89.7%	88.2%
NN	94.1%	93.3%	93.6%	92.0%	N/A
Averages	**91.1%**	**93.0%**	**94.0%**	**90.5%**	**88.9%**
Number of variables In Neighborhood	**8**	**9**	**27**	**8709**	**139351**

Table 1: Neighborhoods and classification on the Thrombin data set

i.2 Efficiency of Local Algorithms in Massive Real Dataset for MB

The following are the times needed to run different local algorithms on a single CPU: MMPC: 3.1 hours; MMMB: 15.3 hours; IAMB: 2.9 hours; IAMB Chunked: 2.6 hours.

i.3 Quality of Local Causal Discovery & Small Sample/Large Neighbourhood Algorithms in Gene Expression Dataset for Inducing the Markov Blanket of Disease

As a preliminary validation of the applicability of the local discovery methods we present here results showing the quality of Markov Blankets (measured by the ability of the Markov Blanket to classify the targets) when using as targets macro-states (i.e., states at the cellular or organism level rather than the gene level). We used as targets: (a) cancer (yes/normal), and (b) histological type (adenoCa/Squamus). The data set used is the one from [Bhattacharjee et al. 2001]. Again we used the area under the ROC curve as classification metric and a variety of classifiers.

As indicated by Table 2 we obtain perfect classification using a much smaller subset of the total 12,600 genes. The Markov Blanket is still large because we predict *composite states* (i.e., diseases that in effect represent the superimposition of a many genes that determine the disease state; the Markov Blanket of the disease state is then a superimposition of the Markov Blankets of the individual genes that cause the disease states).

	Cancer vs Non-Cancer		AdenoCa vs Squamus	
CLASSIFIER	**MMMB (Markov Blanket)**	**ALL GENES**	**MMMB (Markov Blanket)**	**ALL GENES**
LSVM	99.3%	99.6%	98.5%	99.0%
PSVM	99.2%	99.6%	98.7%	99.0%
KNN	93.2%	98.1%	93.0%	97.6%
NN	100.0%	N/A	99.3%	N/A
Variables In Neighborhood	**103**	**12,600**	**65**	**12,600**

Table 2: Neighborhoods of Disease States in Lung Cancer Data

i.4. Quality and Efficiency of Local Algorithms in Lung Cancer Gene Expression for Inducing the Markov Blanket of Target Genes

When switching the focus from composite (disease) states to individual gene targets we expect that the local neighbourhoods (i.e., Markov Blankets or Direct Causes and Effects) will be much smaller. Table 3 shows that in initial experiments we find that this is indeed the case. MMPC runs in 0.5 to 5 minutes (depending on the data split of the cross-validation) and MMMB runs in 15 to 70 minutes (depending on the data split).Additional analyses for other gene-selection methods (not shown here) suggest that the local neighbourhood methods give uniformly better prediction. These results are also consistent with independent estimates about the number of direct causes and effects of genes in higher organisms [Arnone et al. 1997].

	Tumor Protein 63kDA		
CLASSIFIER	**MMMB (Markov Blanket)**	**MMPC (Direct causes + Direct Effects)**	**ALL GENES**
LSVM	91.96	94.21	90.32%
PSVM	89.98	89.67	84.93%
KNN	88.68	89.48	84.98%
NN	93.78	93.38	N/A
Number of variables In Neighborhood	**63**	**9**	**12,600**

Table 3: Causal neighborhoods of individual genes in Lung Cancer data

i.5 Sample Efficiency of Small Sample/Large Neighbourhood Algorithms vs Large-Sample Algorithms in Simulated Data for Finding the Markov Blanket and Direct Causes/Direct Effects

In this experiment we generated a random Bayesian Network with 1000 variables such that the number of parents of each node was randomly and uniformly chosen between 0 and 10 and the free parameters in the conditional probability tables were drawn uniformly from [0, 1]. The *MB(T)* was set to contain three parents, two children, and one parent of one of the children. We then generated datasets from each network using *logic sampling* [Aliferis et al. 1994]. We applied the MMPC and MMMB small-sample algorithms to learn both the direct causes/direct effects and the Markov Blanket in the simulated dataset with 100, 200, and 500 samples using 1,000 variables. The small sample algorithms achieve the same quality (measured as area under the ROC curve) as IAMB by using 5 times less sample. The results are shown in Table 4.

	MMPC	**MMMB**	**IAMB using 500 samples**
Task	**DCE**	**MB**	**MB**
100 Samples	60%	57%	50%
200 Samples	70%	74%	50%
500 Samples	90%	91%	50%

Table 4: Relative Sample Efficiency of Small-Sample algorithms vs IAMB

Summary

Inducing the local causal neighbourhood (LCN) around one or more target variables of interest T provides a solution to the problem of efficient causal discovery when the total number of variables is large (i.e., in the hundreds or thousands) and the generating graph is dense. The LCN can be defined in a number of useful ways: as the set of direct causes of T; as the set of direct effects of T; as the set of direct causes and direct effects of T ; as the set of direct causes and/or direct effects, and/or direct causes-and-direct-effects up to depth *k*; and as the Markov Blanket of T.

Induction of the local causal neighbourhood answers fundamental causal questions about T; induction of the Markov Blanket in particular also provides the minimal set of predictors for T (i.e., provides a solution to the *Feature Selection Problem for Classification*).

Previously we had introduced sound and efficient algorithms for inducing the local neighbourhood, primarily by first inducing the Markov Blanket. In the present report we introduced several new algorithms that induce the Markov Blanket or directly the direct causes and direct effects. Most importantly, the new algorithms are practical when the sample size is small relative to the size of the local neighbourhood. The new algorithms are very efficient in practical experiments with more that a hundred thousand

variables. Initial evaluations suggest that they perform very well in terms of quality of output.

We outlined parallel and chunked versions of the S/L algorithms for cases where the number of variables exceeds the capacity of a single computer host. We also provided a post-processing method that may lead to tighter feature sets without compromising soundness.

The new algorithms are well suited to situations where either the size of the local neighbourhood is very large relative to any practical sample size (e.g., in text categorization tasks), or when samples are very small relative to even small neighbourhoods (e.g., in gene expression array data and other bioinformatics datasets).

References

Aliferis, C.F. and G.F. Cooper. *An Evaluation of an Algorithm for Inductive Learning of Bayesian Belief Networks Using Simulated Data Sets*. In *Tenth Conference on Uncertainty in Artificial Intelligence (UAI)*. 1994.

Aliferis, C.F. and I. Tsamardinos, *Methods for Principled Feature Selection, for Classification, Causal Discovery, and Causal Manipulation.* Technical Report DSL-02-01, 2002.

Aliferis, C.F., I. Tsamardinos, A. Statnikov [b]. *Large-Scale Feature Selection Using Markov Blanket Induction For The Prediction Of Protein-Drug Binding* (submitted). Available as Technical Report DSL-02-06, 2002.

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

Bhattacharjee, A., et al., *Classification of human lung carcinomas by mRNA expression profiling reveals distinct adenocarcinoma subclasses.* Proc Natl Acad Sci USA, 2001. 98(24): 13790-5.

Burges C.J.C. *A tutorial on support vector machines for pattern recognition. Data Minining and Knowledge Discovery.* 1998, 2(2): 1-47.

Cheng, J. et al. *KDD Cup 2001 Report.* SIGKDD Explorations. 2002, 3 (2): 1-18.

Guyon, I., J. Weston, S. Barnhill, et al., *Gene selection for cancer classification using support vector machines*. Machine Learning, 2002, 46: 389-422.

Hannon, G.J., *RNA interference.* Nature, 2002. 418(6894): 244-51.

Mitchell, T.M., *Machine Learning*. 1997, New York: McGraw-Hill Co., Inc.

Parker, S.L., et al., *Cancer statistics, 1996.* Cancer J Clin, 1996. 46(1): 5-27.

Scholkopf, B., C.J.C. Burges, and A.J. Smola, eds. *Advances in kernel methods: support vector learning.* The MIT Press; 1999.

Spirtes, P., C. Glymour, and R. Scheines, *Causation, Prediction, and Search*. Second ed. 2000, Cambridge, Massachusetts, London, England: The MIT Press.

Vapnik V.N., *Statistical learning theory.* John Wiley and Sons; 1992.