Learning structurally consistent undirected probabilistic graphical models

Sushmita Roy
Department of Computer Science
University of New Mexico

Terran Lane
Department of Computer Science
University of New Mexico

Margaret Werner-Washburne
Department of Biology
University of New Mexico

Abstract

In many real-world domains, undirected graphical models such as Markov random fields provide a more natural representation of the dependency structure than directed graphical models. For example, Bayesian networks cannot explicitly capture cyclic dependencies which occur commonly in real-world networks such as in biology. Unfortunately, structure learning of undirected graphs using likelihood-based scores remains difficult because of the intractability of computing the partition function. We describe a new Markov random field structure learning algorithm which is motivated by canonical parameterization of Abbeel *et al.* We improve on their parameterization by learning per-variable canonical factors, which makes our algorithm suitable for domains with hundreds of nodes. Our algorithm is similar to learning dependency networks, but the learned structure is guaranteed to be consistent, and, therefore represents a consistent joint probability distribution. We compare our algorithm against several algorithms for learning undirected and directed models on simulated and real datasets from the biology domain. Our algorithm frequently outperforms existing algorithms, producing higher-quality structures, suggesting that enforcing consistency during structure learning is beneficial for learning undirected graphs.

1 Introduction

Probabilistic graphical models (PGMs) representing real-world networks capture important structural and functional aspects of the network by describing a joint probability distribution of all node measurements. The structure encodes conditional independence assumptions allowing the joint probability distribution to be tractably computed. When the structure is unknown, likelihood-based structure learning algorithms are employed to infer the structure from observed data.

Likelihood-based structure learning of directed acyclic graphs (DAGs), such as Bayesian networks, is widely used because the likelihood score can be tractably computed for all candidate DAGs. Unfortunately, the acyclic constraint of the network structure makes it difficult to represent cyclic dependencies, which occur commonly in real-world domains such as biology. While undirected graphical models, such as Markov random fields (MRFs), provide a more natural representation of the dependency structure, likelihood-based structure learning of these models is much harder. This is because likelihood computation for MRFs requires estimation of the partition function which is known to be NP hard [1].

To overcome this issue, researchers have opted several alternatives: learn graphical Gaussian models where the likelihood can be computed tractably [11]; restrict to lower order, often pairwise

functions, [12, 10]; use pseudo-likelihood as structure score instead of likelihood [3]; learn dependency networks [6, 15]; or more recently, learn Markov blanket canonical parameters [1]. Pairwise models are scalable, but, approximate higher-order dependencies by pairwise functions, which is limiting for domains where higher-order dependencies occur commonly. While dependency networks are scalable, each variable neighborhood is estimated independently, resulting in inconsistent structures when the data sample size is small. This is problematic for real-world data which often lack sufficient samples to guarantee a consistent joint probability distributions for the learned structure. Finally, Markov blanket canonical parameterization requires exhaustive enumeration of variable subsets up to size k, which is not scalable for networks with hundreds of nodes.

We have developed a new algorithm for learning undirected graphical models, that produces consistent structures and is scalable to be applicable for real-world domains. Our algorithm, Markov blanket search (MBS) is inspired by Abbeel *et al.*'s Markov blanket canonical parameterization, which establishes an equivalence between global canonical parameters and local Markov blanket canonical parameters (MBCP) [1]. We extend Abbeel *et al.*'s result to establish further equivalence between MBCPs and *per-variable canonical parameters*. Because per-variable canonical parameters require learning Markov blankets *per-variable*, rather than all subsets up to size k, we save $O(n^{l-1})$ computations during structure learning, where n is the number of variables. The equivalence of per-variable canonical parameters and global canonical parameters has been observed before [8, 13]. However, we are the first to use per-variable canonical parameters in the context of MRF structure learning to learn consistent MRF structures. Enforcing structural consistency during search, guarantees the structure to be a MRF, and also the existence of a joint distribution for the individual conditional distributions. Thus we need not perform additional post-processing to make guarantee consistent structures [15].

We compare our algorithm against two existing algorithms for learning undirected models: Accurate reconstruction of cellular networks (ARACNE) [12], and a Lasso regression based dependency network algorithm (GGLAS) [11]. ARACNE learns only pairwise dependencies, whereas GGLAS learns both pairwise and higher-order dependencies. On simulated data generated from networks of known topology, MBS is able to capture the structure better than ARACNE. Although GGLAS and MBS are often tied in performance, GGLAS's assumption that variable ordering is irrelevant, is true only for the Gaussian distribution. MBS uses a more general framework of minimizing conditional entropy, which can be used with other probability distribution families.

We also compare MBS to several algorithms for learning DAG structures. MBS not only outperforms the algorithms performing DAG searches, but provides a better pruning of the structure search space than L1 regularization-based Markov blanket and the sparse candidate algorithms [15, 5]. This suggests that learning consistent structures during structure search is better than post-processing learned structures to enforce consistency. We finally apply ARACNE, MBS and the sparse candidate-based Bayesian network structure search algorithm to four microarray data sets. Subgraphs generated from MBS-inferred networks represent more functionally coherent gene groups than subgraphs from the other algorithms.

To summarize, MBS has the following advantages: (a) it explicitly scores higher-order dependencies, capturing both higher-order and pairwise dependencies, (b) it learns undirected graphs allowing the representation of cyclic dependencies, (c) it learns consistent structures which ensures the existence of a consistent joint distribution for the learned structure, and (d) it does not require the estimation of likelihood, which allows it to scale to real-world domains with hundreds of nodes.

2 Markov random fields

A Markov random field (MRF) is an undirected, probabilistic graphical model that represents statistical dependencies among a set of random variables (RVs), $\mathbf{X} = \{X_1, \cdots, X_n\}$. A MRF consists of a graph $\mathcal{G}$ and a set of potential functions $\psi = \{\psi_1, \cdots, \psi_m\}$, one for each clique in $\mathcal{G}$. The graph structure describes the statistical dependencies, and the potentials describe the functional relationships between the RVs. The RVs encode the observed measurements for each node, $X_i \in \mathcal{R}$. The joint probability distribution of the MRF is defined to be: $P(\mathbf{X} = \mathbf{x}) = \frac{1}{Z}\prod_{i=1}^{m} \psi_i(\mathbf{F}_i = \mathbf{f}_i)$, where $\mathbf{x}$ is a joint assignment to $\mathbf{X}$, $\mathbf{F}_i \subseteq \mathbf{X}$ is the variable set in the i^{th} clique, associated with ψ_i; $\mathbf{f_i} \subseteq \mathbf{x}$ is a joint assignment to $\mathbf{F}_i$. Z is the partition function and is defined as an integral over all possible joint assignments of $\mathbf{X}$.

Structure learning of MRFs using likelihood is difficult in general because of the computation of Z [1]. This is because estimating Z requires a sum of exponentially many joint configurations of the RVs, making it intractable for real-world domains. To overcome this problem, researchers have proposed approaches that do not involve likelihood [3, 6], and, more recently, have used Markov blanket canonical parameterization (MBCP) [1]. We use an approach similar to MBCP, which requires the estimation of optimal Markov blankets for RV subsets, $\mathbf{Y} \subseteq \mathbf{X}$, $|\mathbf{Y}| \leq l$, where l is a pre-specified, maximum subset size. However, we estimate Markov blankets of only individual RVs, instead of all subsets.

2.1 Hammersly-Clifford theorem and canonical potentials

The Hammersly-Clifford theorem states that there is a one-to-one relationship between MRFs and Gibbs distributions. The canonical potentials (also called $\mathcal{N}$-potentials [13]) are used to prove the Hammersly-Clifford theorem in conjunction with the Mobius inversion theorem [9]. The canonical potential for a subset $\mathbf{D} \subseteq \mathbf{X}$ is defined using a default joint instantiation, $\overline{\mathbf{x}} = \{\overline{x}_1, \cdots, \overline{x}_{|\mathbf{X}|}\}$ to $\mathbf{X}$ as:

$$\psi^*_{\mathbf{D}}(\mathbf{D} = \mathbf{d}) = exp\left(\sum_{\mathbf{U} \subseteq \mathbf{D}} (-1)^{|\mathbf{D} \setminus \mathbf{U}|} \log P(\mathbf{X} = \sigma(\mathbf{U}, \mathbf{X}, \mathbf{d}))\right), \tag{1}$$

where $\sigma(\mathbf{A}, \mathbf{B}, \mathbf{a})$ is an assignment function to variables $X_k \in \mathbf{B}$ such that $\sigma(\mathbf{A}, \mathbf{B}, \mathbf{a})[k] = a_k$, if $X_k \in \mathbf{A}$ and $\sigma(\mathbf{A}, \mathbf{B}, \mathbf{a})[k] = \overline{x}_k$ if $X_k \notin \mathbf{A}$. σ returns an assignment for all variables in $\mathbf{B}$.

The Mobius inversion states that for any real functions f and g over subsets $\mathbf{A}, \mathbf{B}$ and $\mathbf{C}$

$$f(\mathbf{A}) = \sum_{\mathbf{B} \subseteq \mathbf{A}} g(\mathbf{B}), \qquad \text{is true if and only if,} \qquad g(\mathbf{B}) = \sum_{\mathbf{C} \subseteq \mathbf{B}} (-1)^{|\mathbf{B} \setminus \mathbf{C}|} f(\mathbf{C})$$

The joint probability distribution associated with a MRF using the canonical potentials is defined to be: $P(\mathbf{X} = \mathbf{x}) = P(\overline{\mathbf{x}}) \prod_{\mathbf{D} \in \mathcal{C}} \psi^*_{\mathbf{D}}$, where $\mathcal{C}$ is the set of maximal cliques in the graph [13]. This is true by the application of the Mobius inversion where $f(\mathbf{X}) = \log P(\mathbf{X})$, and $g(\mathbf{X}) = \psi^*$, followed by $\psi^*_{\mathbf{D}} = 0$ for all $\mathbf{D} \notin \mathcal{C}$.

2.2 Markov blanket canonical parameterization (MBCP)

The computation of the canonical potentials is not feasible for real-world domains as they require the estimation of the full joint distribution [1]. Markov Blanket canonical parameterization, developed by Abbeel *et al.*, allows the computation of global canonical potentials over $\mathbf{X}$, using local conditional functions called *Markov blanket canonical parameters* (MBCPs). We show that the MBCPs can be further reduced to smaller per-variable canonical parameters, which are computed using an RV and its Markov blanket.

The MBCP, $\widetilde{\psi}$ for a set $\mathbf{D} \subseteq \mathbf{X}$ is estimated using $\mathbf{D}$ and its Markov blanket (MB). The MB, $\mathbf{M}_i$ of a variable X_i, is the set of immediate neighbors of X_i in $\mathcal{G}$ and renders X_i conditionally independent of other variables, i.e., $P(X_i|\mathbf{X} \setminus \{X_i\}) = P(X_i|\mathbf{M}_i)$. The MB, $\mathbf{M}_{\mathbf{D}}$ of a set $\mathbf{D}$, is $\left(\bigcup_j \mathbf{M}_j\right) \setminus \mathbf{D}$ for all $X_j \in \mathbf{D}$. The MBCP, $\widetilde{\psi}$ for $\mathbf{D}$ is also defined using the default joint instantiation, $\overline{\mathbf{x}} = \{\overline{x}_1, \cdots, \overline{x}_{|\mathbf{X}|}\}$ as:

$$\widetilde{\psi}_{\mathbf{D}}(\mathbf{D} = \mathbf{d}) = exp\left(\sum_{\mathbf{U} \subseteq \mathbf{D}} (-1)^{|\mathbf{D} \setminus \mathbf{U}|} \log P(\mathbf{D} = \sigma(\mathbf{U}, \mathbf{D}, \mathbf{d}) | \mathbf{M}_{\mathbf{D}} = \sigma(\mathbf{U}, \mathbf{M}_{\mathbf{D}}, \mathbf{d}))\right), \tag{2}$$

For MRFs of unknown structure, MBCPs are identified by searching exhaustively among all subsets $\mathbf{F}_i \subset \mathbf{X}$, up to size l and finding MBs for each $\mathbf{F}_i$. MBs are chosen to minimize the conditional entropy, $H(\mathbf{F}_i|\mathbf{M}_{\mathbf{F}_i})$. Unfortunately, exhaustive enumeration of variable subsets becomes impractical for moderately sized networks [1].

2.3 Per-variable MB canonical factors

We now show that the MBCPs can be replaced by smaller, local functions: *per-variable MB canonical factors*, which does not require enumeration of all subsets up to size l. To illustrate how these

are derived from MBCPs, let $\mathbf{D}$ from Eq 2 be $\mathbf{D} = \{X_i, X_j\}$ and $\mathbf{d} = \{x_i, x_j\}$. Let $\mathbf{E} = \mathbf{M}_i \cup \mathbf{M}_j \cup \{X_j\}$. We first apply the chain rule to every term, $\log P(\mathbf{D} = \sigma(\mathbf{U}, \mathbf{D}, \mathbf{d}) | (\mathbf{M}_i \cup \mathbf{M}_j) = \sigma(\mathbf{U}, \mathbf{M}_i \cup \mathbf{M}_j, \mathbf{d}))$ in Eq 2, by first conditioning X_i on $\mathbf{E}$, followed by X_j on $\mathbf{E} \setminus \{X_j\}$. This allows $\widetilde{\psi}$ to be rewritten as:

$$\widetilde{\psi}_{\mathbf{D}}(\mathbf{D} = \mathbf{d}) = exp\left(\sum_{\mathbf{U} \subseteq \mathbf{D}} (-1)^{|\mathbf{D} \setminus \mathbf{U}|} \log P(X_i = \sigma(\mathbf{U}, \{X_i\}, \mathbf{d})) | \mathbf{E} = \sigma(\mathbf{U}, \mathbf{E}, \mathbf{d}))\right), \tag{3}$$

These canonical parameters are equal to those in Eq 2 because the $\log P(X_j = \sigma(\mathbf{U}, \{X_j\}, \mathbf{d}) | (\mathbf{E} \setminus \{X_j\}) = \sigma(\mathbf{U}, \mathbf{E} \setminus \{X_j\}, \mathbf{d}))$ terms cancel. We assert further independence in Eq 3 because X_i is independent of all variables other than $\mathbf{M}_i$. This allows us to write the original MBCF for $\{X_i, X_j\}$ as the per-variable canonical factor:

$$\psi^{+}_{\mathbf{D}}(\mathbf{D} = \mathbf{d}) = exp\left(\sum_{\mathbf{U} \subseteq \mathbf{D}} (-1)^{|\mathbf{D} \setminus \mathbf{U}|} \log P(X_i = \sigma(\mathbf{U}, \{X_i\}, \mathbf{d}) | \mathbf{M}_i = \sigma(\mathbf{U}, \mathbf{M}_i, \mathbf{d}))\right), \tag{4}$$

This implies that, instead of searching over all size l subsets of $\mathbf{X}$, we can estimate canonical factors by searching for MBs of individual RVs. Assuming that the variable MBs are estimated correctly, Eq 4 will produce the same canonical factors as Eq 2. Our structure learning algorithm therefore requires the estimation of MBs of each RV. We only need to ensure structural consistency (Section 2.4). Searching only for n MBs, as opposed to n^l MBs in MBCP, saves us $O(n^{l-1})$ computations.

The fact that ψ^{+} is a sum of conditional distributions, allows us to apply the Mobius inversion to estimate the conditional distribution, $P(X_i | \mathbf{M}_i)$, from ψ^{+}, by setting f to $\log P(X_i | \mathbf{M}_i)$ and g to ψ^{+}:

$$\log P(X_i | \mathbf{M}_i) = \sum_{\mathbf{Y} \subseteq \{X_i\} \cup \mathbf{M}_i} \psi^{+}.$$

In traditional MRFs described by the Gibbs distribution, estimating these conditional distributions would require inference, which is known to be hard for general graph structures. Estimating ψ^{+} using the conditional distributions not only guarantees a consistent joint distribution for the MRF structure, but also specifies the exact conditional distribution that can be inferred from the joint using the laws of probability. In [13], a similar statement is made about generating the conditional distributions from the original $\mathcal{N}$-potentials. However, this requires the computation of a local normalization factor which is not required for the per-variable canonical parameters. Although conditional probability distribution of a variable given its Markov blanket is estimated in dependency networks, this conditional distribution is not guaranteed to be consistent with a joint distribution.

The per-variable canonical parameters and the Markov blanket canonical parameters, do not deny the hardness of computing the likelihood in MRFs [1]. This is because computing $P(\mathbf{X} = \bar{\mathbf{x}})$ is equivalent to computing $\frac{1}{Z}$, where Z is the partition function.

2.4 Markov blanket search algorithm

We search for the best MB of individual RVs using conditional entropy, $H(X_i | \mathbf{M}_i)$ for each X_i, [4]. The best MB for all RVs can be identified by minimizing the following score:

$$S(\mathbf{G}) = \sum_{i=1}^{|\mathbf{X}|} H(X_i | \mathbf{M}_i) + \lambda \log(|\mathbf{M}_i|) \tag{5}$$

$\lambda \log(|\mathbf{M}_i|)$ penalizes large MBs and $0 \leq \lambda \leq 1$ is a regularization coefficient. The first term of Eq 5 is directly proportional to the data likelihood in Bayes nets and the pseudo-likelihood in MRFs.

Directly minimizing this score by finding the best MB per variable independently may result in inconsistent MBs. In particular, we cannot guarantee that if $X_j \in \mathbf{M}_i$, then $X_i \in \mathbf{M}_j$. This problem also arises in MBCP estimation, since MBs of variable sets are identified independently[1]. This inconsistency can be handled as a post-processing of the learned MBs [15]. However, we found this post-processing approach to produce lower quality MBs (Section 3.2).

[1] MBCP requires an additional subset consistency check: if $\mathbf{X} \subset \mathbf{Y}$, then $\mathbf{M}_{\mathbf{X}} \subset (\mathbf{M}_{\mathbf{Y}} \cup (\mathbf{Y} \setminus \mathbf{X}))$

We propose a different approach such that we find consistent MBs during the search process. To find consistent MBs, we search MBs, $\mathbf{M}_i$, for X_i not only using the decrease in $H(X_i|\mathbf{M}_i)$, but also the net change in conditional entropy of all $X_j \in \mathbf{M}_i$, if their MBs were also constrained to include X_i. This is done by computing the net score gain per candidate MB for X_i.

Our approach is similar to [7] where consistent structures are learned using an edge-based score. However, their search strategy starts from a fully connected network and removes edges, whereas we add and replace edges starting with a completely disconnected network. For real-world domains, growing larger neighborhoods from smaller neighborhoods is more tractable than shrinking large neighborhoods, because we may not have enough data for reliably learning large neighborhoods.

We perform a greedy search, where we make one variable extensions to the current MB. Let $\mathbf{M}_i^k$ denote a candidate MB of X_i of size k, and $\widehat{\mathbf{M}}_i^{k-1}$ denote the best MB for X_i obtained so far. Then the score gain is:

$$S_i = H(X_i|\widehat{\mathbf{M}}_i^{k-1}) - H(X_i|\widehat{\mathbf{M}}_i^{k-1} \cup \{X_j\}) + H(X_j|\widehat{\mathbf{M}}_j^{k-1}) - H(X_j|\widehat{\mathbf{M}}_j^{k-1} \cup \{X_i\}). \quad (6)$$

The MBS structure learning algorithm uses the above score gains to identify the best MB for each variable. Each iteration of the search uses a combination of *add* and *swap* operations to learn the best structure. In the add stage of the k^{th} iteration, we make one variable extensions to the current Markov blanket $\widehat{\mathbf{M}}_i^{k-1}$ of each variable X_i restricting it to at most k RVs per MB.

In the swap stage, we revisit all variables Z in the Markov blanket $\widehat{\mathbf{M}}_i^{k}$ of each X_i, and consider other RVs $Y \notin (\{X\} \cup \widehat{\mathbf{M}}_i^{k})$, which if swapped in instead of Z, gives a score improvement. If so, we replace Z by Y in $\widehat{\mathbf{M}}_i$, and store Z in the *tabu list* of X_i. This prevents Z from being included into $\mathbf{M}_i$ in subsequent iterations. In the swap stage we allow a candidate neighbor to be present in more than k_{max} Markov blankets. However, no variable can be in more than $k_{hard} = 20$ Markov blankets. Thus the nodes in our inferred networks can have a degree of at most $d = 20$, which is reasonable for the domain of our interest and allows us to model hub nodes in the network.

In this paper, we assume all variables to have a Gaussian distribution. However, our general approach is applicable to both continuous and discrete variables, requiring only to be able to estimate conditional entropy. Using Gaussian distributions has the added advantage that the conditional and joint entropy can be computed in closed form.

3 Results

We compared the MBS algorithm against two other learning algorithms for undirected graphs and five DAG learning algorithms on both simulated data and real biological data.

3.1 Comparison on simulated datasets

We compared our Markov blanket search algorithm (MBS) to two undirected algorithms: An Algorithm for the Reconstruction of Gene Regulatory Networks in a Mammalian Cellular Context (ARACNE) [12], and a Lasso regression-based Graphical Gaussian model (GGLAS) [11]. We also compared MBS against several directed models provided in the DAGLearn software[2]: full DAG search (FULLDAG), LARs based order search (ORDLAS), DAG search using Sparse candidate for pruning (SPCAND), and DAG search using L1 regularization based Markov blanket estimation (L1MB) [15]. Because L1MB does not learn consistent Markov blankets, a post-processing step is required to make the structures consistent. The AND post-processing removes X_i from X_j's MB if X_j is not in X_i's MB. The OR post-processing includes X_i in X_j's MB if X_j is in X_i's MB. We refer to L1MB with AND and OR post-processing as MBAND and MBOR respectively. We also included an implementation of order Markov chain Monte Carlo (ORDMC) for Bayes net search.[3]

The simulated datasets were generated by a gene regulatory network simulator using differential equations for describing gene and protein expression dynamics [14]. The simulator models

[2] `http://www.cs.ubc.ca/~murphyk/Software/DAGlearn/`

[3] `http://www.bioss.ac.uk/staff/.adriano/comparison/comparison.html`

Table 1: Algorithm comparison on two datasets. Rows give different structure scores; columns are different structure learning algorithms; each entry is an E or V score. Bold* and italics indicate that MBS performs significantly better or worse than the algorithm compared. SPN: shortest path neighborhood, 1 and 2N: $r = 1$ and $r = 2$ neighborhood, 3C: cycles of size 3, 4C: cycles of size 4.

			MBS	ARACNE	ORDER	SPCAND	MBOR	GGLAS
G50	E	SPN	0.550	**0.418***	0.533	**0.516***	**0.417***	**0.463***
		1N	0.661	**0.440***	**0.587***	**0.560***	**0.432***	*0.836*
		2N	0.589	**0.444***	**0.532***	**0.510***	**0.438***	**0.562***
		3C	0.800	0.400	0.792	0.784	**0.250***	0.866
		4C	0.645	**0.440***	0.630	**0.580***	**0.367***	0.721
	V	SPN	0.308	*0.345*	**0.264***	**0.262***	*0.292*	*0.379*
		1N	0.352	*0.426*	**0.285***	**0.276***	*0.416*	0.231
		2N	0.335	*0.361*	**0.273***	**0.261***	*0.353*	0.340
		3C	0.328	**0.251***	**0.284***	**0.287***	**0.233***	0.271
		4C	0.324	**0.246***	**0.281***	**0.275***	**0.230***	0.260
ECOLI1	E	SPN	0.747	0.753	0.703	0.759	**0.729***	**0.729***
		1N	0.751	0.776	0.690	*0.778*	**0.705***	**0.700***
		2N	0.726	0.752	0.679	*0.749*	0.724	0.719
	V	SPN	0.514	*0.567*	**0.303***	**0.354***	0.520	0.522
		1N	0.608	*0.667*	**0.326***	**0.396***	*0.627*	*0.639*
		2N	0.591	*0.622*	**0.308***	**0.376***	0.585	0.594

Table 2: Number of times MBS loses/beats statistically significantly another algorithm. Rows are for different datasets.

DATA	ARACNE	ORDMC	FULLDAG	ORDLAS	SPCAND	MBAND	MBOR	GGLAS
G50	3/6	0/4	0/5	2/5	3/9	2/3	3/7	2/2
G75	0/3	0/6	0/5	0/5	0/5	0/5	0/10	3/4
ECOLI1	3/0	1/2	0/3	0/3	2/3	1/1	2/2	2/2
ECOL2	0/0	0/1	0/6	0/6	0/6	0/6	0/6	–

combinatorial control among regulator proteins to generate expression data resembling those from real-world networks. We used four simulated datasets: G50, G75, ECOLI1 and ECOLI2 with $n = 100, 150, 188$ and 188 nodes respectively. ECOLI1 and 2 were generated from the regulatory network of the bacteria, *E. coli*. G50 and G75 were generated *de novo* by the simulator. Each sample is a steady-state expression measurement after perturbing the transcription rate constants of the the genes. In G50, G75 and ECOLI1 all nodes are perturbed, whereas in ECOLI2 only the regulator proteins are perturbed.

As the true network topologies for these data are known, we compared the algorithms using the match between the inferred and true networks. Because we are most interested in analyzing higher-order dependencies, we compared subgraphs rather than edges. Briefly, we extracted subgraphs of different types (e.g. cycles, neighborhood) from the true network and used an F-score measure to match the vertex neighborhood and edge set per subgraph. We refer to the scores for vertex neighborhood as V-scores and for the edge set as E-scores. We use shortest path neighborhoods (SPN), r-radius neighborhoods ($r \in \{1, 2\}$, denoted by 1N and 2N) consisting of a vertex and its neighbors $\leq r$ steps away, and cycles of size r ($r \in \{3, 4\}$, denoted by 3C and 4C). ECOLI1 and 2 did not have any cycles. We moralize the inferred DAGs prior to comparison. We compare the algorithms using E and V-scores averaged over four different runs per algorithm.

We show a subset of all the comparisons comprising two of the four datasets, G50 and ECOLI1 (Table 1). Our complete results are summarized in Table 2. For all datasets other than ECOL1, MBS significantly beats all algorithms at least as often as it is beaten (Student's t-test, p-value ≤ 0.05). On ECOLI1, ARACNE outperforms not only MBS, but all other higher order algorithms, suggesting that this dataset likely does not contain many higher-order dependencies. There is no significant performance difference between MBS and ARACNE on ECOLI2 (results not shown), which is generated from the same network as ECOLI1. GGLAS did not complete on ECOLI2.

We also find that the performance margin between MBS and the DAG learning models is greater than undirected learning algorithms, even though we convert the directed graph structures to moralized undirected graphs. Overall, we find that MBS performs at least as well as other algorithms.

3.2 Structural consistency for pruning DAGs space

To assess the value of enforcing consistency during learning, rather than as a post-processing step, we used the MBS-learned Markov blankets as inputs to DAG search algorithms as a constraint on variable families. We compared the DAG structures constrained using MBS Markov blankets against those constrained by Sparse candidate (SC) and L1 MB regularization (L1MB). The L1MB approach uses either an OR or AND of the Markov blankets to make inconsistent Markov blankets consistent.

We used the maximum size of L1MB AND and OR Markov blankets for selecting the neighborhood size, k, for MBS and SC. We first compared L1MB with OR post-processing (MBOR) using $k = 11$ for both G50 and G75 (Table 3). We found the structures constrained by MBS-learned Markov blankets to have higher E and V scores than the structures constrained by L1MB or SC, and to significantly outperform SC or L1MB-constrained DAGs more often than being outperformed. We had similar results for L1MB AND ($k = 4, 6$ for G50 and G75, respectively). This suggests that finding consistent structures during structure learning produces higher-quality Markov blankets than enforcing consistency as a post-processing step.

3.3 Comparison on real biological data

We compared MBS against ARACNE and SPCAND on real-world biological data. GGLAS did not complete within 48 hrs on this data, so is omitted. Each dataset measures the gene expression response of two different populations of yeast cells, *Quiescent* (Q) and *Non-quiescent* (NQ), to genetic perturbations [2]. Each dataset had a biological replicate, resulting in four datasets: Q1, Q2, NQ1 and NQ2. We pre-processed these data to include genes with $< 80\%$ missing data and with high variation resulting in $n = 1808$ genes.

As the true network for these data is not known, we used Gene ontology (GO) to identify subgraphs that were enriched in a biological process. For each inferred network we generated neighborhood subgraphs of radius $r = 1$. For each subgraph and GO term pair, we used the hyper-geometric distribution to compute a p-value enrichment. We used two criteria to compare the three algorithms. *Enrichment sensitivity* measured the ratio of the number subgraphs enriched in a GO term[4] to the total number of subgraphs. *Enrichment locality* measured the correlation between a GO term p-value and the number of subgraphs enriched in that term. A positive correlation suggests that terms with higher p-values (less enriched) are associated with many subgraphs, whereas terms with lower p-values (more enriched) are associated with fewer subgraphs. We used different significance thresholds to vary the stringency of enrichment (p-value $\in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$). Ideally, a good algorithm should identify good enrichment for the majority of the inferred subgraphs (high sensitivity), and also be able to associate highly enriched terms with a few subgraphs (high locality).

For each p-value threshold, we compared MBS against ARACNE and SPCAND using the enrichment sensitivity and locality of the four datasets. We found that at p-value $< 10^{-3}$ and $< 10^{-4}$, ARACNE and SPCAND had significantly higher sensitivity, but significantly lower locality than MBS (Wilcoxon rank sum test, $p < 0.05$). However, there was no statistically significant difference for higher stringency of enrichment.

These results suggest that there is a trade off between different algorithms for biological data. MBS is able to identify subgraphs that are local and ontologically coherent at the cost of having fewer subgraphs that are enriched in a term. On the other hand ARACNE and SPCAND identify more subgraphs with enrichment, but they may overly fragment coherent gene groups. Finally, there is no significant difference between algorithms at higher stringency, suggesting that the algorithms agree on the process terms in which we have greater confidence.

4 Conclusion

We have described a new algorithm for inferring undirected graphical models that yields structurally consistent graphs, guaranteeing a consistent joint probability distribution for the random variables. We compared our algorithm against several algorithms for learning undirected and directed models.

[4] We actually considered the min(no. of subgraphs, no. of enriched terms) to prevent double counting.

Table 3: Comparison of MBS pruning against Sparse candidate and L1 MB regularization. Rows and columns are same as Table 1.

		G50			G75		
		SPCAND	MBOR	MBS	SPCAND	MBOR	MBS
E	SPN	0.48	$\mathbf{0.458}^*$	0.504	0.349	0.404	0.435
	1N	0.516	0.523	0.561	$\mathbf{0.467}^*$	$\mathbf{0.523}^*$	0.567
	2N	0.466	$\mathbf{0.485}^*$	0.538	0.424	$\mathbf{0.474}^*$	0.486
	3C	0.465	0.414	0.556	0.498	$\mathbf{0.481}^*$	0.612
	4C	0.508	$\mathbf{0.463}^*$	0.532	0.458	$\mathbf{0.447}^*$	0.595
V	SPN	$\mathbf{0.27}^*$	$\mathbf{0.27}^*$	0.348	0.269	*0.299*	0.257
	1N	$\mathbf{0.295}^*$	$\mathbf{0.328}^*$	0.413	*0.288*	*0.331*	0.256
	2N	$\mathbf{0.274}^*$	$\mathbf{0.296}^*$	0.367	0.27	0.287	0.247
	3C	0.274	0.316	0.318	0.247	0.241	0.241
	4C	0.256	0.276	0.327	0.274	$\mathbf{0.22}^*$	0.279

We show that learning structurally consistent graph structures during structure inference more accurately captures the graph structure. Our approach also produces higher-quality Markov blankets, that when used to prune DAG search space, yields better structures. On real data, MBS is able to identify functionally coherent local gene groups that indicate tightly co-regulated genes.

Acknowledgments

Withheld for peer review.

References

[1] P. Abbeel, D. Koller, and A. Y. Ng. Learning factor graphs in polynomial time and sample complexity. *JMLR*, 7:1743–1788, 2006.

[2] A. D. Aragon, A. L. Rodriguez, O. Meirelles, S. Roy, G. S. Davidson, P. H. Tapia, C. Allen, R. Joe, D. Benn, and M. Werner-Washburne. Characterization of differentiated quiescent and nonquiescent cells in yeast stationary-phase cultures. *Mol. Biol. Cell*, pages E07–07–0666+, January 2008.

[3] J. Besag. Efficiency of pseudolikelihood estimation for simple gaussian fields. *Biometrika*, 64(3):616–618, December 1977.

[4] T. M. Cover and J. A. Thomas. *Elements of information theory*. Wiley-Interscience, New York, NY, USA, 1991.

[5] N. Friedman, I. Nachman, and D. Pe'er. Learning bayesian network structure from massive datasets: The sparse candidate algorithm. In *Uncertainty in Artificial Intelligence*, 1999.

[6] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. M. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2000.

[7] R. Hofmann and V. Tresp. Nonlinear markov networks for continuous variables. In *NIPS '97: Proceedings of the 1997 conference on Advances in neural information processing systems 10*, pages 521–527, Cambridge, MA, USA, 1998. MIT Press.

[8] R. Kindermann and J. L. Snell. *Markov Random Fields and Their Applications*. American Mathematical Society, 1980.

[9] S. L. Lauritzen. *Graphical Models*. Oxford Statistical Science Series. Oxford University Press, New York, USA, July 1996.

[10] S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of markov networks using l_1-regularization. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 817–824. MIT Press, Cambridge, MA, 2007.

[11] F. Li and Y. Yang. Using modified lasso regression to learn large undirected graphs in a probabilistic framework. In *AAAI*, pages 801–806, 2005.

[12] A. Margolin, I. Nemenman, K. Basso, C. Wiggins, G. Stolovitzky, R. D. Favera, and A. Califano. Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, (Suppl 1): S7, 2005.

[13] R. D. Paget. *Nonparametric Markov random field models for natural texture images*. PhD thesis, University of Queensland, 1999.

[14] S. Roy, M. Werner-Washburne, and T. Lane. A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics (Oxford, England)*, April 2008.

[15] M. Schmidt, A. Niculescu-Mizil, and K. Murphy. Learning graphical model structure using l1-regularization paths. In *Twenty second international conference on AI*, 2007.