Learning structurally consistent undirected probabilistic graphical models

Sushmita Roy* SROY@CS.UNM.EDU
Terran Lane* TERRAN@CS.UNM.EDU
Margaret Werner-Washburne$^+$ MAGGIEWW@UNM.EDU
Dept. of Computer Science* and Biology$^+$, University of New Mexico, Albuquerque, NM 87131, USA

Abstract

In many real-world domains, undirected graphical models such as Markov random fields provide a more natural representation of the statistical dependency structure than directed graphical models. 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, motivated by canonical parameterization of Abbeel *et al.* We provide computational improvements on their parameterization by learning per-variable canonical factors, which makes our algorithm suitable for domains with hundreds of nodes. We compare our algorithm against several algorithms for learning undirected and directed models on simulated and real datasets from biology. 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. However, in many domains such as biology, causal implication of directed edges is difficult to ascertain without perturbations, leaving only a correlation implication. In such situations, undirected graphical models are a more natural representation of statistical dependencies. Unfortunately, likelihood-based structure learning of these models is much harder due to the intractability of the partition function (Abbeel et al., 2006).

To overcome this issue, researchers have opted several alternatives: learn graphical Gaussian models where the likelihood can be computed tractably (Li & Yang, 2005); restrict to lower order, often pairwise functions, (Margolin et al., 2005; Lee et al., 2007); use pseudo-likelihood as structure score instead of likelihood (Besag, 1977); learn dependency networks (Heckerman et al., 2000; Schmidt et al., 2007); or learn Markov blanket canonical factors (Abbeel et al., 2006). 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 distribution for the learned structure. Finally, Markov blanket canonical parameterization requires exhaustive enumeration of variable subsets up to a pre-specified size l, which is not scalable for networks with hundreds of nodes.

We have developed a new algorithm for learning undirected graphical models, that produces consis-

Appearing in *Proceedings of the 26th International Conference on Machine Learning*, Montreal, Canada, 2009. Copyright 2009 by the author(s)/owner(s).

tent 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 established an equivalence between global canonical potentials and local Markov blanket canonical factors (MBCFs) (Abbeel et al., 2006). We extend Abbeel *et al.*'s result to establish further equivalence between MBCFs and *per-variable canonical factors*. Because per-variable canonical factors require learning Markov blankets *per-variable*, rather than all subsets up to size l, we save $O(n^{l-1})$ computations during structure learning, where n is the number of variables. The equivalence of per-variable canonical factors and global canonical factors has been observed before (Paget, 1999). However, we are the first to use per-variable canonical factors 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 guarantee consistent structures (Schmidt et al., 2007).

We compare our algorithm against two existing algorithms for learning undirected models: Algorithm for the Reconstruction of Accurate Cellular Networks (ARACNE) (Margolin et al., 2005), and a Lasso regression based dependency network algorithm (GGLAS) (Li & Yang, 2005). ARACNE learns pairwise dependencies, whereas GGLAS learns both pairwise and higher-order dependencies. On simulated data from networks of known topology, MBS captures the structure better than ARACNE for the majority of the datasets. 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 per-variable canonical factors, which can be used with any conditional probability distribution family.

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 the L1 regularization-based Markov blanket and sparse candidate algorithms (Schmidt et al., 2007; Friedman et al., 1999). 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 algorithm to four real-world microarray data sets. Subgraphs generated from MBS-inferred networks represent more biologically meaningful dependencies than subgraphs from the other algorithms.

To summarize, MBS has the following advantages: (a) it captures both higher-order and pairwise dependencies, (b) it learns consistent structures ensuring the existence of a joint distribution, (c) it provides a tractable implementation of the theoretical framework of Abbeel *et al.*. This final property allows MBS 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 a summation over all possible joint assignments of $\mathbf{X}$.

Structure learning of MRFs using likelihood is difficult in general because of Z (Abbeel et al., 2006). 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 use pseudolikelihood (Besag, 1977; Heckerman et al., 2000), or, have used Markov blanket canonical parameterization (MBCP) (Abbeel et al., 2006). 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 learn local per-variable factors, requiring estimation of Markov blankets of only individual RVs. Avoiding Markov blanket estimation of all subsets, makes our approach scalable to domains with hundreds of nodes.

2.1. Hammersly-Clifford theorem and canonical potentials

The Hammersly-Clifford theorem establishes a one-to-one relationship between MRFs and strictly positive distributions such as the Gibbs distributions. The *canonical potentials* (also called $\mathcal{N}$-potentials (Paget, 1999)) are used together with the Möbius inversion theorem to prove the Hammersly-Clifford theorem (Lauritzen, 1996). 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:

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

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

The joint probability distribution for a MRF using 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. This is true by an application of the Möbius inversion and setting $\psi^*_{\mathbf{D}} = 0$ for all $\mathbf{D} \notin \mathcal{C}$ (Lauritzen, 1996; Paget, 1999).

2.2. Markov blanket canonical parameterization

The computation of the canonical potentials is not feasible for real-world domains as they require the estimation of the full joint distribution (Abbeel et al., 2006). 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 factors* (MBCFs).

The MBCF, $\widetilde{\psi}$ 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 MBCF, $\widetilde{\psi}$ for $\mathbf{D}$ is also defined using the default joint instantiation, $\overline{\mathbf{x}} = \{\overline{x}_1, \cdots, \overline{x}_{|\mathbf{X}|}\}$ as:

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

For MRFs of unknown structure, MBCFs are identified by searching exhaustively among all subsets $\mathbf{F}_i \subset \mathbf{X}$, up to size l and finding MBs for each $\mathbf{F}_i$. Unfortunately, exhaustive enumeration of variable subsets becomes impractical for moderately sized networks (Abbeel et al., 2006). We show that the MBCFs can be further reduced to smaller per-variable canonical factors, which are computed using an RV and its Markov blanket.

2.3. Per-variable MB canonical factors

We now show that the MBCFs can be replaced by smaller, local functions: *per-variable MB canonical factors*, which does not require enumeration of all subsets up to size l. Specifically, for every MB canonical factor $\widetilde{\psi}$ there exists an equivalent per-variable canonical factor ψ^+. To illustrate how the per-variable factors are derived from MBCFs, we first consider a specific case of $\mathbf{D} = \{X_i, X_j\}$ in Eq 1 (Section 2.3.1), followed by a proof for the general case (Section 2.3.2).

2.3.1. Special case of two variables

Let $\mathbf{D} = \{X_i, X_j\}$ and $\mathbf{d} = \{x_i, x_j\}$. Note, because $\mathbf{D} \cap \mathbf{M}_{\mathbf{D}} = \emptyset$, $\sigma(\mathbf{U}, \mathbf{M}_{\mathbf{D}}, \mathbf{d}) = \overline{\mathbf{m}}_{\mathbf{d}}$, the default instantiation to $\mathbf{M}_{\mathbf{D}}$ from $\overline{\mathbf{x}}$. We first expand the sum inside the exponential of Eq 1 with $\mathbf{D} = \{X_i, X_j\}$:

$$\begin{aligned}
&\sum_{\mathbf{U}\subseteq\mathbf{D}} (-1)^{|\mathbf{D}\setminus\mathbf{U}|} \log P(\{X_i, X_j\} = \\
&\qquad \sigma(\mathbf{U}, \{X_i, X_j\}, \mathbf{d})|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&= (-1)^{|\{X_i,X_j\}|} \log P(X_i = \overline{x}_i, X_j = \overline{x}_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad +(-1)^{|\{X_j\}|} \log P(X_i = x_i, X_j = \overline{x}_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad (-1)^{|\{X_i\}|} \log P(X_i = \overline{x}_i, X_j = x_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad +(-1)^{|\emptyset|} \log P(X_i = x_i, X_j = x_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \qquad (2)
\end{aligned}$$

where the first term corresponds to $U = \emptyset$, the second term corresponds to $U = \{X_i\}$ and so on. Applying the chain rule to every term in the RHS:

$$\begin{aligned}
&= \log[P(X_i = \overline{x}_i|X_j = \overline{x}_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\qquad P(X_j = \overline{x}_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}})] \\
&\quad -\log[P(X_i = x_i|X_j = \overline{x}_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\qquad P(X_j = \overline{x}_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}})] \\
&\quad -\log[P(X_i = \overline{x}_i|X_j = x_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\qquad P(X_j = x_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}})] \\
&\quad +\log[P(X_i = x_i|X_j = x_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\qquad P(X_j = x_j|\mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}})]
\end{aligned}$$

We find that all $\log P(X_j|\mathbf{M}_{\mathbf{D}})$ terms cancel producing:

$$\begin{aligned}
&= \log P(X_i = \overline{x}_i, |X_j = \overline{x}_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad -\log P(X_i = x_i|X_j = \overline{x}_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad -\log P(X_i = \overline{x}_i|X_j = x_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \\
&\quad +\log P(X_i = x_i|X_j = x_j, \mathbf{M}_{\mathbf{D}} = \overline{\mathbf{m}}_{\mathbf{d}}) \qquad (3)
\end{aligned}$$

This allows $\widetilde{\psi}$ to be rewritten as:

$$\widetilde{\psi}_{\mathbf{D}}(\mathbf{D} = \mathbf{d}) = \exp\Bigg(\sum_{\mathbf{U}\subseteq\mathbf{D}} (-1)^{|\mathbf{D}\setminus\mathbf{U}|} \log P(X_i = \sigma(\mathbf{U}, \{X_i\}, \mathbf{d})|\{X_j\} \cup \mathbf{M}_{\mathbf{D}} = \sigma(\mathbf{U}, \{X_j\} \cup \mathbf{M}_{\mathbf{D}}, \mathbf{d}))\Bigg) \quad (4)$$

We assert further independence in Eq 4 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:

$$\psi^{+}_{\mathbf{D}}(\mathbf{D}=\mathbf{d}) = \exp\Bigg(\sum_{\mathbf{U}\subseteq\mathbf{D}} (-1)^{|\mathbf{D}\setminus\mathbf{U}|}\log P(X_i = \sigma(\mathbf{U},\{X_i\},\mathbf{d})|\mathbf{M}_i = \sigma(\mathbf{U},\mathbf{M}_i,\mathbf{d}))\Bigg) \qquad (5)$$

Thus we have equivalent the per-variable canonical factor, ψ^{+} from the original MBCF $\widetilde{\psi}$ in Eq 1.

2.3.2. General case

We now state the equivalence between per-variable and MB canonical factors more formally:

Theorem 2.1 *Every MBCP, $\widetilde{\psi}_{\mathbf{D}}$, of the form in Eq 1 possesses an equivalent per-variable factor, $\psi^{+}_{\mathbf{D}}(\mathbf{D}=\mathbf{d}) = exp\Big(\sum_{\mathbf{U}\subseteq\mathbf{D}}(-1)^{|\mathbf{D}\setminus\mathbf{U}|}logP(X_i = \sigma(\mathbf{U},\{X_i\},\mathbf{d})|\mathbf{M}_i = \sigma(\mathbf{U},\mathbf{M}_i,\mathbf{d}))\Big)$, where $X_i \in \mathbf{D}$.*

Proof The proof of this equivalence involves two steps: (a) deriving ψ^{+} from $\widetilde{\psi}$ for any general $\mathbf{D}$, and (b) identifying neighbors of an RV and making independence assertions described by the graph structure.

To prove (a) we select an arbitrary $X_i \in \mathbf{D}$. We replace each $\log P(\mathbf{D}|\mathbf{M}_{\mathbf{D}})$ in $\widetilde{\psi}$ by $\log(P(X_i|\mathbf{D}\setminus\{X_i\}\cup\mathbf{M}_{\mathbf{D}})P(\mathbf{D}\setminus\{X_i\}|\mathbf{M}_{\mathbf{D}}))$. We have $2^{|\mathbf{D}|}$ number of $\log P(\mathbf{D}\setminus\{X_i\}|\mathbf{M}_{\mathbf{D}})$ terms, one for each $\mathbf{U}\subseteq\mathbf{D}$. X_i does not occur in these terms, as we have conditioned on it. These terms can be grouped into two sets, $\mathcal{S}_{od}$ and $\mathcal{S}_{ev}$, where $\mathcal{S}_{od}$ and $\mathcal{S}_{ev}$ correspond to subsets of $\mathbf{D}$ with odd and even number of elements, respectively. Assuming $|\mathbf{D}|$ is even, all elements in $\mathcal{S}_{od}$ have a $-$ve sign and all elements in S_{ev} have a +ve sign. Further, for every $t \in \mathcal{S}_{ev}$ corresponding to $\mathbf{U}\subseteq\mathbf{D}$ there exists $t' \in \mathcal{S}_{od}$ corresponding to $\mathbf{U}'\subseteq\mathbf{D}$, such that $\mathbf{U}$ and $\mathbf{U}'$ differ only in X_i. Because X_i does not occur in either t or t', these two terms cancel. Applying this to all elements of $\mathcal{S}_{od}$ and $\mathcal{S}_{ev}$, the two subsets cancel each other, thus proving (a). If $|\mathbf{D}|$ is odd, elements of $\mathcal{S}_{od}$ and $\mathcal{S}_{ev}$ have +ve and $-$ve signs, respectively, and the rest of the argument follows.

The final step is to identify the neighbors of X_i and using the local Markov property, $P(X_i|\mathbf{D}\setminus\{X_i\}\cup\mathbf{M}_{\mathbf{D}}) = P(X_i|\mathbf{M}_i)$, for strictly positive distributions (Lauritzen, 1996) □.

The equivalence of the per-variable factors and MBCFs 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 MBs are estimated correctly, Eq 5 will produce the same canonical factors as Eq 1. Our structure learning algorithm therefore requires the estimation of MBs of each RV. We only need to ensure structural consistency (Section 2.4). Searching for n MBs, as opposed to n^l MBs in MBCF, saves us $O(n^{l-1})$ computations.

The per-variable canonical factors and MBCFs do not deny the hardness of computing likelihood in MRFs (Abbeel et al., 2006). This is because computing $P(\mathbf{X}=\overline{\mathbf{x}})$ is equivalent to computing $\frac{1}{Z}$.

Algorithm 1 Markov Blanket Search

```
Input:
  Random variable set, $\mathbf{X} = \{X_1, \cdots, X_{|\mathbf{X}|}\}$
  maximum neighborhood sizes, $k_{max}$, $k_{hard}$
Output:
  Inferred graph structure $\mathcal{G}$
for $k = 1; k \le k_{max}; k++$ do
  for $X_i \in \mathbf{X}$ do {Add stage}
    Find best new MB variable $X_j$ that maximizes $\Delta S_{ij}$
    s.t. $|\mathbf{M}_i| \le k$ (Eq 6)
  end for
  for $X_i \in \mathbf{X}$ do {Swap stage}
    for $X_j \in \widehat{\mathbf{M}_i}^{k}$ do
      for $X_q \in \mathbf{X} \setminus (\widehat{\mathbf{M}_i}^{k} \cup \{X_i\})$ and $|\widehat{\mathbf{M}_q}^{k}| \le k_{hard}$
      and $X_q \notin$ tabulist$(X_i)$ do
        Delete $\{X_i, X_j\}$, add $\{X_i, X_q\}$, add $X_j$ to
        tabulist$(X_i)$ if swapping $X_q$ for $X_j$ gives maximal score improvement.
      end for
    end for
  end for
end for
```

2.4. Markov blanket search (MBS) algorithm

The MBS algorithm learns the structure of a MRF by finding the best neighborhood or Markov blanket (MB) for each RV. To identify the best MB, we need to optimize a *score*, $S(X_i|\mathbf{M}_i)$ per RV X_i, which quantifies dependence between a RV and its MB. Examples of such scores include pseudolikelihood (dependency networks) or conditional entropy (MBCP) (Cover & Thomas, 1991). For example, the best MB identified via conditional entropy, $H(X_i|\mathbf{M}_i)$ is: $\mathbf{M}_i = \arg\min_{\widehat{\mathbf{M}}_i} H(X_i|\widehat{\mathbf{M}}_i)$. Best MB via pseudolikelihood, $PLL(X_i|\mathbf{M}_i)$, is: $\mathbf{M}_i = \arg\max_{\widehat{\mathbf{M}}_i} PLL(X_i|\widehat{\mathbf{M}}_i)$

Dependency networks and MBCP identify the best MB per RV by optimizing $S(X_i|\mathbf{M}_i)$ per RV[1]. However, optimizing $S(X_i|\mathbf{M}_i)$ per RV 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 inconsistency can be handled as a post-processing of

[1] In MBCP estimation, MBs of variable sets are identified independently. MBCP requires an additional subset consistency check: if $\mathbf{X}\subset\mathbf{Y}$, then $\mathbf{M}_{\mathbf{X}} \subset (\mathbf{M}_{\mathbf{Y}} \cup (\mathbf{Y}\setminus\mathbf{X}))$

the learned MBs (Schmidt et al., 2007). However, our experiments suggest that a post-processing approach produces lower quality MBs (Section 3.2).

We propose a different approach that finds consistent MBs during the search process. To find consistent MBs, we search MBs, not only using the improvement in $S(X_i|\mathbf{M}_i)$ on adding X_j, but also the score change in $S(X_j|\mathbf{M}_j)$ if X_i was added to $\mathbf{M}_j$. This is done by computing the net score gain per candidate MB for X_i. Let $\widehat{\mathbf{M}_i}^{k-1}$ denote the best MB for X_i obtained so far. Then the score gain is:

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

Our approach is similar to Hofmann & Tresp's edge-based score for guaranteeing consistency (Hofmann & Tresp, 1998). 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 feasible than shrinking large neighborhoods, because we may not have enough data for reliably learning large neighborhoods.

The MBS structure learning algorithm uses Eq 6 to greedily identify the best MB for each variable (Algo. 1). Each iteration 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 $\widehat{\mathbf{M}_i}^{k-1}$ of each X_i, restricting to at most $k \leq k_{max}$ RVs per MB.

In the swap stage, we revisit all variables $X_j \in \widehat{\mathbf{M}_i}^{k}$ for each X_i, and consider other RVs, $X_q \notin (\{X_i\} \cup \widehat{\mathbf{M}_i}^{k})$, which if swapped in instead of X_j, gives a score improvement. If so, we replace X_j by X_q with the maximal score improvement, in $\widehat{\mathbf{M}_i}^{k}$, and store X_j in the *tabu list* of X_i. This prevents X_j from being included in X_i's MB in subsequent iterations. In the swap stage, a variable can be present in $> k_{max}$ MBs. However, no variable can be in more than $k_{hard} = 20$ MBs. Thus, nodes in our inferred networks have degrees $\geq k_{hard}$, which reasonably models hub nodes in most domains.

The per-variable canonical factor equivalence exploited by MBS to identify the MRF structure does not make any specific assumptions of the parametric form of the conditional probability distributions. MBS only requires that the candidate MBs be scored using the conditional probability distributions. So MBS can potentially be instantiated with any probability distribution and choice of score. For empirical evaluation of our framework, we selected $P(X_i|\mathbf{M}_i)$ to be conditional Gaussians and $S(X_i|\mathbf{M}_i)$ to be the regularized conditional entropy for each X_i: $S(X_i|\mathbf{M}_i) = H(X_i|\mathbf{M}_i) + \lambda\log(|\mathbf{M}_i|)$. $\lambda\log(|\mathbf{M}_i|)$ penalizes large MBs and $0 \leq \lambda \leq 1$ is a regularization coefficient.

3. Results

We compared our Markov Blanket Search (MBS) algorithm against existing algorithms for undirected and directed graphs on both simulated and real data. Our test data is from biology. The simulated data are from networks of known structure, enabling a direct validation of the inferred structures. The real data is from microarray experiments. However, as the true network for the real data is not known, we use biological literature to validate the inferred structures. This test framework is common in bioinformatics (Margolin et al., 2005; Li & Yang, 2005).

3.1. Comparison on simulated datasets

We compared MBS to two undirected algorithms: ARACNE (Margolin et al., 2005), and a Lasso regression-based Graphical Gaussian model (GGLAS) (Li & Yang, 2005). We also compared MBS against several directed models provided in the DAGLearn software: 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) (Schmidt et al., 2007). 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 $\mathbf{M}_j$ if $X_j \notin \mathbf{M}_i$, where $\mathbf{M}_i$ and $\mathbf{M}_j$ are X_i's and X_j's MB, respectively. The OR post-processing includes X_i in $\mathbf{M}_j$ if $X_j \in \mathbf{M}_i$. We refer to L1MB with AND and OR post-processing as MBAND and MBOR, respectively. We also included an implementation of order MCMC (ORDMC) for Bayes net search (`http://www.bioss.ac.uk/staff/.adriano/comparison/comparison.html`).

The simulated datasets were generated by a gene regulatory network simulator using differential equations for describing gene and protein expression dynamics (Roy et al., 2008). We generated four datasets: G50, G75, ECOLI1 and ECOLI2 with $n = 100, 150, 188$ and 188 nodes, respectively. Each sample consists of steady-state expressions reached after perturbing the kinetic constants of the genes. Networks for G50 and G75 were generated *de novo* by the simulator. The network for both ECOLI1 and 2 belong to the bac-

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, 1N, 2N: $r = 1$ and $r = 2$ neighborhood, 3C, 4C: cycles of size 3 or 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. GGLAS did not complete on ECOLI2.

DATA	ARACNE	ORDMC	FULLDAG	ORDLAS	SPCAND	MBAND	MBOR	GGLAS
G50	3/6	0/4	0/5	2/5	0/9	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
ECOLI2	0/0	0/1	0/6	0/6	0/6	0/6	0/6	–

teria, *E. coli*. In ECOLI2 only a subset of the genes (regulators) are perturbed, whereas in ECOLI1, G50 and G75 all genes are perturbed.

As the true network topologies for these data are known, we compared the algorithms using the match between the inferred and true network structures. Because we are interested in higher-order dependencies, we matched 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 edge set as *E-scores*. We use shortest path neighborhoods (SPN), r-radius neighborhoods comprising a vertex and its neighbors $\leq r$ steps away ($r \in \{1, 2\}$, denoted by 1N and 2N), 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.

Our comparison used E and V-scores averaged over four runs per algorithm corresponding to different settings of an algorithm-specific parameter. This parameter is λ in MBS (Section 2.4), data processing inequality d in ARACNE, and hyper-prior parameter for the variance in GGLAS. For all DAG searches other than ORDMC, we used different random restart probabilities to generate different candidate graphs. In our experiments, $\lambda \in \{1e-5, 3e-5, 5e-5, 7e-5\}$ and $d \in \{0, 0.3, 0.5, 0.7\}$. All simulated experiments used $1 \leq k_{max} \leq 11$. For ORDMC, we varied the edge posterior probability. For each parameter setting, the graph with the highest average of E and V score is used. We compare the best graph per algorithm across different parameter settings.

We show results on two of the four datasets, G50 and ECOLI1 (Table 1). Our complete results are summarized in Table 2. For all datasets other than ECOLI1, 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 all algorithms, suggesting that ECOLI1 likely does not contain many higher-order dependencies. There is no significant difference between MBS and ARACNE on ECOLI2, which is generated from the same network as ECOLI1 using different perturbations.

We find that the performance margin between MBS and the DAG learning models is greater than undirected learning algorithms, suggesting undirected graphs may be better representations for this domain. Overall, MBS does a better job of learning the network structures compared to both directed and undirected algorithms for majority of the datasets.

Table 3. Comparison of MBS pruning against Sparse candidate and L1 MB regularization. Legend same as Table 1.

		G50			G75		
		SPCAND	MBOR	MBS	SPCAND	MBOR	MBS
E	SPN	0.48	**0.458***	0.504	0.349	0.404	0.435
	1N	0.516	0.523	0.561	**0.467***	**0.523***	0.567
	2N	0.466	**0.485***	0.538	0.424	**0.474***	0.486
	3C	0.465	0.414	0.556	0.498	**0.481***	0.612
	4C	0.508	**0.463***	0.532	0.458	**0.447***	0.595
V	SPN	**0.27***	**0.27***	0.348	0.269	*0.299*	0.257
	1N	**0.295***	**0.328***	0.413	*0.288*	*0.331*	0.256
	2N	**0.274***	**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	**0.22***	0.279

Table 4. Number of GO terms MBS has worse/better enrichment than other algorithms.

	Q1	Q2	NQ1	NQ2
ARACNE	556/641	471/734	409/685	518/487
SPCAND	434/570	278/784	325/809	567/762

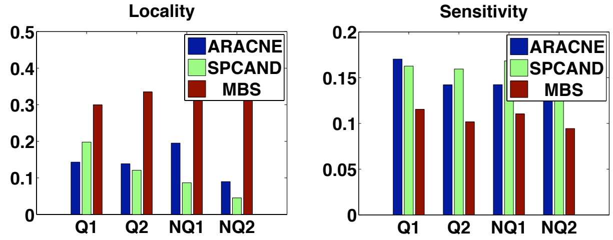

Figure 1. Enrichment sensitivity and locality for significance $p < 10^{-3}$. Higher values are better.

3.2. Structural consistency for pruning DAGs

To assess the value of enforcing consistency during learning, rather than as a post-processing step, we used the MBS-learned Markov blankets as family constraints in DAG search algorithms. We compared the DAG structures constrained using MBS Markov blankets against those constrained by Sparse candidate (SPCAND) and L1 MB regularization (L1MB). L1MB uses either an OR or AND of the Markov blankets to generate consistent Markov blankets.

We used the maximum size of L1MB AND and OR Markov blankets as the neighborhood size, k, for MBS and SPCAND. We first compared L1MB with OR post-processing (MBOR) using $k = 11$ for both G50 and G75 (Table 3). We found the DAGs constrained by MBS-learned Markov blankets to significantly outperform both SPCAND or L1MB-constrained DAGs more often than being outperformed. Using L1MB AND ($k = 4, 6$ for G50 and G75, respectively) MBS outperformed SPCAND or L1MB with a greater margin. This indicates that enforcing consistency, during structure learning produces higher-quality Markov blankets, than 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 (Aragon et al., 2008). Each dataset had a biological replicate, resulting in four datasets: Q1, Q2, NQ1 and NQ2. We pre-processed these data to include $n = 1808$ genes with $< 80\%$ missing data. We used MBS with $\lambda = 10^{-4}, k_{max} = 4$, ARACNE with dpi $= 0.3$ and SPCAND with 4 parents. A relatively large value of λ, compared to that in simulated networks was used because of the large number of nodes in this data.

As the true network is not known, we used statistical enrichment of subgraphs generated from the inferred networks, in biological processes from Gene ontology (GO), to assess the quality of the networks. For each inferred network, we generated neighborhood subgraphs of radius $r = 1$. For each subgraph g and term t, we computed a hyper-geometric p-value, assessing the statistical significance of observing g's genes to be annotated with t. The lower the p-value the better is the statistical enrichment. We first computed the number of GO terms MBS had better (lower p-value) or worse enrichment compared to other algorithms (Table 4). We found that MBS had more terms with better enrichment in all except one case (ARACNE, NQ2). This suggests that networks identified via MBS capture more significant biological dependencies.

We also compared the algorithms using two other measures (Fig 1). *Enrichment sensitivity* is the ratio of the min(no. of subgraphs, no. of enriched terms) to the total number of subgraphs. *Enrichment locality* is the correlation between average p-value of a term 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. Ideally, an algorithm should identify good enrichment for the majority of subgraphs (high sensitivity), and also associate highly enriched terms with a few subgraphs (high locality).

We used different stringency levels of enrichment (p-value $\in \{10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}\}$) to assess the enrichment sensitivity and locality of the algorithms on all four datasets (Fig. 3.2 shows the sensitivity and locality for p-value$< 10^{-3}$). At p-value $< 10^{-3}$ and $< 10^{-4}$, ARACNE and SPCAND had significantly higher sensitivity, but significantly lower locality than MBS (Wilcoxon rank sum, $p < 0.05$). However, there was no statistical difference for higher stringency of enrichment. These results suggest that there is a trade off between different algorithms for biological data. MBS identifies subgraphs that are locally 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 may overly fragment coherent gene groups. Finally, there is no significant difference between algorithms at higher stringency, suggesting that the algorithms agree on the GO terms that are the most significant.

4. Conclusion

We have described a new algorithm for inferring undirected graphs that yields structurally consistent graphs, guaranteeing a joint probability distribution for the RVs. We compared our algorithm to several algorithms for learning undirected and directed models. On simulated data, we show that enforcing consistency during structure learning more accurately captures the graph structure. Our approach also produces higher-quality Markov blankets, that when used to prune DAG searches, yields better structures. On real data, MBS identifies more significant ontology terms associated with functionally coherent gene groups.

Acknowledgments

This work is supported by grants from NIMH (1R01MH076282-03) and NSF (IIS-0705681) to T.L., from NIH (GM-67593) and NSF (MCB0734918) to M.W.W. and from HHMI-NIH/NIBIB (56005678).

References

Abbeel, P., Koller, D., & Ng, A. Y. (2006). Learning factor graphs in polynomial time and sample complexity. *J. Mach. Learn. Res.*, *7*, 1743–1788.

Aragon, A. D., Rodriguez, A. L., Meirelles, O. *et al.* (2008). Characterization of differentiated quiescent and nonquiescent cells in yeast stationary-phase cultures. *Mol. Biol. Cell*, E07–07–0666+.

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

Cover, T. M., & Thomas, J. A. (1991). *Elements of information theory.* New York, NY, USA: Wiley-Interscience.

Friedman, N., Nachman, I., & Pe'er, D. (1999). Learning bayesian network structure from massive datasets: The sparse candidate algorithm. *Uncertainty in Artificial Intelligence* (pp. 206–215).

Heckerman, D., Chickering, D. M., Meek, C., Rounthwaite, R., & Kadie, C. M. (2000). Dependency networks for inference, collaborative filtering, and data visualization. *J. Mach. Learn. Res.*, *1*, 49–75.

Hofmann, R., & Tresp, V. (1998). Nonlinear markov networks for continuous variables. *Advances in Neural Information Processing Systems* (pp. 521–527). MIT Press.

Lauritzen, S. L. (1996). *Graphical models.* Oxford Statistical Science Series. New York, USA: Oxford University Press.

Lee, S.-I., Ganapathi, V., & Koller, D. (2007). Efficient structure learning of markov networks using l_1-regularization. *Advances in Neural Information Processing Systems 19* (pp. 817–824). MIT Press.

Li, F., & Yang, Y. (2005). Using modified lasso regression to learn large undirected graphs in a probabilistic framework. *American Association for Artificial Intelligence* (pp. 801–806).

Margolin, A., Nemenman, I., Basso, K. *et al.* (2005). Aracne: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context. *BMC Bioinformatics*, *(Suppl 1): S7.*

Paget, R. D. (1999). *Nonparametric markov random field models for natural texture images.* Doctoral dissertation, University of Queensland.

Roy, S., Werner-Washburne, M., & Lane, T. (2008). A system for generating transcription regulatory networks with combinatorial control of transcription. *Bioinformatics (Oxford, England)*, 1318–1320.

Schmidt, M., Niculescu-Mizil, A., & Murphy, K. (2007). Learning graphical model structure using l1-regularization paths. *22nd international conference on Artificial Intelligence* (pp. 1278–1283).