Marginal Pseudo-Likelihood Inference for Markov Networks

Johan Pensar
Henrik Nyman

Abstract

Since its introduction in the 1970’s, pseudo-likelihood has become a well-established inference tool for random network models. More recently, there has been a revival of interest towards the pseudo-likelihood based approach, motivated by several ’large p, small n’ type applications. Under such circumstances some form of regularization is needed to obtain plausible inferences. The currently available methods typically necessitate the use of a tuning parameter to adapt the level of regularization for a particular dataset, which can be optimized for example by cross-validation. Here we introduce a Bayesian version of pseudo-likelihood inference for Markov networks, which enables an automatic regularization through marginalization over the nuisance parameters in the model. We prove consistency of the resulting estimator for network structure and introduce an efficient algorithm for learning structures via harmonization of candidate sets of Markov blankets. The marginal pseudo-likelihood method is shown to perform favorably against recent popular inference methods for Markov networks in terms of accuracy, while being at a comparable level in terms of computational complexity.

Keywords: Bayesian inference, Markov networks, model selection, pseudo-likelihood, regularization

1. Introduction

Markov networks and more generally random graphs represent a ubiquitous modeling framework for multivariate systems, with applications ranging from statistical physics to computational biology and sociology (see [Lauritzen] 1996 [Koller and Friedman] 2009). However, statistical inference for such models has been a challenge from the start, in particular due to the generally intractable form of the normalizing factor, often called partition function, of these distributions. In physics, random network models have traditionally been estimated using the mean-field approximation, which has only recently started to become superceded by more elaborate approaches, such as the pseudo-likelihood method ([Aurell and Ekeberg] 2012 [Ekeberg et al.] 2013). The pseudo-likelihood approach was originally motivated by the difficulties of maximizing the likelihood function for random networks ([Besag] 1972)
and it simplifies the inference by a factorization of the likelihood over local neighborhoods of the random variables involved in the modeled system.

For certain subclasses of network models, such as chordal Markov networks, also known as decomposable graphical models for multivariate data, the inference problem is simplified due to a genuine factorization of the likelihood function under the network structure. However, since the chordality assumption is too restrictive for several types of applications, considerable interest has been targeted towards making inference tractable also for non-chordal networks. A revival of interest has in particular arisen from the need to consider high-dimensional models in a ‘large p, small n’ setting (Lee et al., 2006; Höfling and Tibshirani, 2009; Ravikumar et al., 2010; Aurell and Ekeberg, 2012; Ekeberg et al., 2013).

High-dimensional Markov networks usually necessitate some form of regularization to make the pseudo-likelihood inference problem feasible to solve. Some of the currently available methods necessitate the use of a tuning parameter to adapt the level of regularization for a particular dataset. The value of the tuning parameter can then be optimized for example by cross-validation. Here we introduce a Bayesian version of pseudo-likelihood inference to learn the structure of a Markov network without assuming chordality. Our method enables an automatic regularization of the resulting model complexity through marginalization over the nuisance parameters in the model. The structure of the remaining article is as follows. In the next section the basic properties of Markov networks are reviewed and the inference problem is formulated in Section 3. In Section 4, we introduce the marginal pseudo-likelihood (MPL) approach and prove its consistency. An algorithm for optimizing the MPL score for a given dataset is derived in Section 5 and the penultimate section demonstrates the favorable performance of our method against other popular recent alternatives. The last section provides some additional remarks and conclusions.

2. Markov networks

We consider a set of \(d\) discrete random variables \(X = \{X_1, \ldots, X_d\}\) where each variable \(X_j\) takes values from a finite set of outcomes \(\mathcal{X}_j\). A Markov network over \(X\) is a undirected probabilistic graphical model that compactly represents a joint distribution over the variables. The dependence structure over the \(d\) variables is specified by an undirected graph \(G = (V, E)\) where the nodes \(V = \{1, \ldots, d\}\) correspond to the indices of the variables \(X\) and the edge set \(E \subseteq \{V \times V\}\) represents dependences among the variables. We will use the terms node and variable interchangeably throughout this article. The complete set of undirected graphs is denoted by \(\mathcal{G}\).

A node \(i\) is a neighbor of \(j\) (and vice versa) if \(\{i, j\} \in E\) and the set of all neighbors of \(j\) is called the node’s Markov blanket (MB), which is denoted by \(mb(j) = \{i \in V : \{i, j\} \in E\}\). A clique in a graph is a subset of nodes, \(C \subseteq V\), for which every pair of nodes are connected by an edge, that is \(\{i, j\} \in E\) if \(i, j \in C\). A clique is considered maximal if it cannot be extended by including an additional node without violating the clique criterion. The set of maximal cliques associated with a graph is denoted by \(\mathcal{C}(G)\).

The variables corresponding to a subset of nodes, \(S \subseteq V\), are simply denoted by \(X_S = \{X_j\}_{j \in S}\) and the corresponding joint outcome space is specified by the Cartesian product \(\mathcal{X}_S = \times_{j \in S} \mathcal{X}_j\). The cardinality of an outcome space is denoted by \(|\mathcal{X}_S|\). We use a lowercase letter \(x_S\) to denote that the variables have taken on a specific joint outcome in \(\mathcal{X}_S\), that
is \(X_S = x_S\). A dataset \(X = (x_1, \ldots, x_n)\) refers to a collection of \(n\) i.i.d. complete joint observations \(x_k = (x_{k,1}, \ldots, x_{k,d})\) over the \(d\) variables, that is \(x_{k,j} \in \mathcal{X}_j\) for each \(k\) and \(j\).

We use \(p(x_A \mid x_B)\) as an abbreviated notation for the conditional probability \(p(X_A = x_A \mid X_B = x_B)\) while \(p(X_A \mid x_B)\) denotes a specific conditional probability distribution and \(p(X_A \mid X_B) = \{p(X_A \mid x_B)\}_{x_B \in \mathcal{X}_B}\) represents the corresponding collection of conditional distributions. Finally, we use \(\theta_G\) to denote the parameterization of a specific distribution satisfying \(G\) and \(\Theta_G\) represents the parameter space induced by \(G\), that is, the set of all possible values of the parameter vector \(\theta_G\).

In addition to the graph, to fully specify a Markov network one must also define a probability distribution that satisfies the restrictions imposed by the graph. We restrict the models to positive distributions. The joint distribution of a Markov network can then be factorized over the maximal cliques in the graph according to

\[
p(x) = \frac{1}{Z} \prod_{C \in \mathcal{C}(G)} \phi(x_C) \tag{1}
\]

where \(\phi(x_C) : \mathcal{X}_C \to \mathbb{R}_+\) is a clique factor (or potential) and \(Z = \sum_{x \in \mathcal{X}} \prod_{C \in \mathcal{C}(G)} \phi(x_C)\) is a normalizing constant known as the partition function. Markov networks are often also parameterized in terms of a log-linear model in which each clique factor is replaced by an exponentiated weighted sum of features according to

\[
p(x) = \frac{1}{Z} \exp \left( \sum_{f_K \in \mathcal{F}} w_K f_K(x_K) \right) \tag{2}
\]

where \(\mathcal{F} = \{f_K\}\) is the set of feature functions and \(W = \{w_K\}\) is the corresponding set of weights. A feature function \(f_K : \mathcal{X}_K \to \mathbb{R}\) maps each value \(x_K \in \mathcal{X}_K\) for some \(K \subseteq V\) to a numerical value, typically it is in the form of a indicator function that takes on value 1 if the value matches a specific feature and 0 otherwise. Every Markov network can be encoded as a log-linear model by defining a feature as an indicator function for every assignment of \(X_C\) for each \(C \in \mathcal{C}(G)\). In this case the weights in (2) correspond to the natural logarithm of the clique factors in (1). Conversely, a log-linear model over a graph of a Markov network by imposing an edge \(\{i,j\}\) for every pair of variables appearing in the same domain of some feature function \(f_K(x_K)\), that is \(\{i,j\} \in E\) if \(\{i,j\} \subseteq K\).

The absence of edges in the graph \(G = (V, E)\) of a Markov network encodes statements of conditional independence. The variables \(X_A\) are conditionally independent of the variables \(X_B\) given the variables \(X_S\) if \(p(X_A \mid X_B, X_S) = p(X_A \mid X_S)\) holds. We denote this by \(X_A \perp X_B \mid X_S\). The dependence structure of a Markov network can be characterized by the following properties:

1. Pairwise Markov property: \(X_i \perp X_j \mid X_{V \setminus \{i,j\}}\) for all \(\{i,j\} \not\in E\).
2. Local Markov property: \(X_i \perp X_{V \setminus \{mb(i)\}} \mid X_{mb(i)}\) for all \(i \in V\).
3. Global Markov property: \(X_A \perp X_B \mid X_S\) for all disjoint subsets \((A, B, S)\) of \(V\) such that \(S\) separates \(A\) from \(B\).
Although the strength of the Markov properties differ in general, they are proven to be equivalent under the current assumption of positivity of the joint distribution (Lauritzen, 1996). While the last property is sufficient in the sense that it captures the entire set of independences induced by a network, the first two properties are also useful since they allow one to focus on smaller sets of independences. This can facilitate inference processes such as structure learning. In particular, our MPL score for structure learning is based on the local Markov property.

3. Structure learning

By structure learning, we refer to the process of inferring the dependence structure from a set of data assumed to be generated under an unknown Markov network. Under this framework, the structure learning problem can be considered a model class learning problem in the sense that each specific structure alone represents a class of models. In many applications the structure is a goal in itself in the sense that one wants merely to gain a qualitative insight into the dependence structure of an underlying process. Still, given a known structure the more specific problem of model estimation is reduced to a simpler problem known as parameter estimation. Hence, if the distribution needs also to be explicitly estimated, this can be achieved by existing methods conditional on the fixed structure learned by our approach.

The structure of a Markov network can be formulated in terms of different degrees of granularity on the associated parameters. The most fine-grained structure learning methods aim at recovering distinct features in the log-linear parameterization [2], this approach is commonly referred to as feature selection (Pietra et al., 1997; Lowd and Davis, 2010; Lee et al., 2006). A very detailed structure allows the model to better emulate the properties of a distribution without imposing redundant parameters. One drawback of this formulation is the risk of overfitting. Since every pair of variables in a feature results in an edge, such parameterizations can obscure the connection to the graph structure in the sense that sparsity in the number of features does not correspond to sparsity in the number of edges in the graph. This is not desirable from a knowledge discovery perspective since it may in worst case hide the very pattern that one wants to identify. Furthermore, most inference algorithms for deriving probabilistic statements under a graphical model exploit the sparsity of the graph structure and dense graphs inevitably hamper the efficiency of such algorithms. In contrast to the very specific feature selection problem, the model space of our approach is formulated directly in terms of the graph structure alone and the complexity of a corresponding model is defined by the sizes of the cliques in the network.

Structure learning methods can roughly be divided into two categories; constraint-based and score-based methods. Constraint-based approaches aim at inferring the structure through a series of independence tests based on the Markov properties, (Spirtes et al. 2000; Tsamardinos et al. 2003; Bromberg et al. 2009; Anandkumar et al. 2012). This approach is appealing in the sense that the independently performed tests can be combined into a structure through a divide-and-conquer approach. Under the assumption that the network is a perfect map of its corresponding distribution and that the tests are correct, the true structure can be reconstructed. In practice, however, very large sample sizes may be required for the independence tests to yield correct answers.
The score-based approach formulates structure learning as an optimization problem. One defines an objective function that evaluates the plausibility of each candidate in the model space. The objective function is then optimized by some appropriate method.

The most common objective function is the likelihood of the data,

\[ l(\theta \mid X, G) = p(X \mid \theta, G) = \prod_{k=1}^{n} p(x_k \mid \theta, G), \]

or the corresponding log-likelihood,

\[ \ell(\theta \mid X, G) = \log l(\theta \mid X, G). \] (3)

By maximizing the (log-)likelihood, the model fit to the data is maximized. Although there is in general no analytical solution, the concavity of the likelihood function enables the maximization problem to be solved by numerical optimization. However, the maximum likelihood alone favors complex models and always obtains its maximum value under the complete graph due to noise in the data. One option is to constrain the expressiveness of the graphs in the model space, for example by only considering tree structures (Chow and Liu, 1968). However, such constraints can limit the model space to networks that are not plausible for modeling the data.

In the Bayesian formulation of the structure learning problem, one would ideally score a graph by its posterior probability given the data,

\[ p(G \mid X) = \frac{p(X \mid G) \cdot p(G)}{p(X)}. \] (4)

In practice, it suffices to consider the unnormalized posterior probability since the denominator in (4) is a normalizing constant that can be ignored when comparing graphs. The key term of (4) is the marginal likelihood, or evidence \( p(X \mid G) \). To evaluate the marginal likelihood one must integrate the likelihood function over all parameter values satisfying the restrictions imposed by the graph according to

\[ p(X \mid G) = \int_{\theta \in \Theta_G} l(\theta \mid X, G) \cdot f(\theta \mid G) d\theta \] (5)

where \( f(\theta \mid G) \) is a prior distribution that assigns a weight to each \( \theta \in \Theta_G \). Since (5) accounts for the parameter uncertainty through the prior, it implicitly regulates the fit to the data with respect to the complexity of the network. Unfortunately, the marginal likelihood is very difficult to evaluate for undirected non-chordal graphs. Instead, most methods regulate the fit explicitly by adding a sparsity promoting penalty function to (3). In particular, Schwarz (1978) introduced the Bayesian information criterion (BIC) as an asymptotic approximation of (5) that is consistent for exponential families.

Given a scoring function, it is necessary to specify an algorithm that identifies the network maximizing the score since the discrete search space is in general too large for an exhaustive evaluation. To avoid the discrete nature of the model space, Lee et al. (2006) used an \( L_1 \)-based penalty to reformulate the structure learning problem as a convex optimization problem over the continuous parameter space. This is especially well-suited
for binary pairwise Markov networks where each edge is associated with a single parameter in (2). In fact, in this case the problem formulation of feature selection and graph structure discovery are equivalent due to the one-to-one correspondence between edges and features. For more general networks, sparsity must be enforced to groups of parameters in order to achieve sparsity in the number of edges in the resulting graph (Schmidt and Murphy, 2010).

An advantage of score-based methods is that they are data-efficient in the sense that they tend to perform better than constraint-based methods when the amount training data is small relative to the network size. Unfortunately, this comes at the expense of an increased computational cost in the learning process. Due to the partition function, likelihood based methods, in particular, are rendered inefficient already for a moderate number of variables since each step of the search or optimization process must perform inference over the model. This is in general computationally infeasible and it thereby necessitates use of approximate inference. In this work we propose a hybrid method designed to combine the advantages of constraint- and score-based approaches.

4. Marginal pseudo-likelihood

To avoid problems associated with the evaluation of the true likelihood function, one can preferably use alternative objectives that possess favorable properties from a computational perspective. One of the most common choices is the pseudo-likelihood originally introduced by Besag (1972). The pseudo-likelihood function approximates the likelihood function by a factorization into conditional likelihood functions according to

$$pl(\theta \mid X) = \prod_{j=1}^{d} p(X_j \mid X_{V \setminus j}, \theta).$$

For a fixed graph structure the local Markov property implies that a variable is independent of the remaining variables given its Markov blanket, such that

$$p(X_j \mid X_{V \setminus j}, G) = p(X_j \mid X_{mb(j)})$$

must hold. Consequently, the pseudo-likelihood for a given graph $G$ is given by

$$pl(\theta \mid X, G) = \prod_{j=1}^{d} p(X_j \mid X_{mb(j)}, \theta) = \prod_{k=1}^{n} \prod_{j=1}^{d} p(x_{k,j} \mid x_{k,mb(j)}, \theta).$$

(6)

Since the global normalizing constant in the likelihood function now is replaced by $d$ local normalizing constants, the concave pseudo-likelihood approximation can still be applied when the number of variables grows large. For example, the pseudo-likelihood approximation of Höfling and Tibshirani (2009) allows the approach of Lee et al. (2006) to be extended to higher dimensions. Furthermore, pseudo-likelihood approaches have also been shown to be consistent under the assumption that the data is generated by a distribution in the model class. In particular, Ji and Seymour (1996) and Csiszár and Talata (2006) derived pseudo-counterparts to the BIC criterion which were proven to enjoy consistency.

From a Bayesian perspective, the structural form of (6) offers an interesting possibility. In fact, under certain assumptions it enables an analytical evaluation of the integral

$$\hat{p}(X \mid G) = \int_{\theta \in \Theta} pl(\theta \mid X, G) \cdot f(\theta \mid G) d\theta$$

(7)
which is here referred to as the marginal pseudo-likelihood (MPL). We denote the conditional probabilities associated with the pseudo-likelihood function by

$$
\theta_{ijl} = p(X_j = x_j^{(i)} \mid X_{mb(j)} = x_{mb(j)}^{(l)})
$$

where $i = 1, \ldots, r_j$ and $l = 1, \ldots, q_j$ represent the configurations of the variable and its Markov blanket, respectively. Similarly,

$$
n_{ijl} = \sum_{k=1}^{a} I[(x_{k,j}, x_{k,mb(j)}) = (x_j^{(i)}, x_{mb(j)}^{(l)})]
$$

and

$$
n_{jl} = \sum_{i=1}^{r_j} n_{ijl}
$$

denote the counts of the corresponding configurations in $X$. The pseudo-likelihood function can now be expressed by our new notation in terms of

$$
\text{pl}(\theta \mid X, G) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}.
$$

The conditional distributions in (8) are connected to each other in the sense that they must satisfy certain algebraic properties for them to be consistent with a Markov network. However, to achieve an efficient analytical score, we make a necessary assumption about global and local independence among the parameter sets associated with the conditional distributions, $\theta_{jl} = (\theta_{1jl}, \ldots, \theta_{rjl})$. This is similar to the parameter independence assumption discussed by Heckerman et al. (1995) for Bayesian networks, however, we note that the assumption is not as natural here. Still, it is a fundamental assumption that ultimately allows us to factorize the parameter prior distribution into local Dirichlet distributions. Consequently, (7) may be reordered into a product of integrals which can easily be calculated analytically:

$$
\hat{p}(X \mid G) = \int_{\theta \in \Theta_G} \text{pl}(\theta \mid X, G) \cdot f(\theta \mid G)d\theta
$$

$$
= \prod_{j=1}^{d} \prod_{l=1}^{q_j} \int_{\theta_{ijl} \in \Theta_j} \theta_{ijl}^{n_{ijl}} \cdot f(\theta_{ijl} \mid x_{mb(j)}^{(l)})d\theta
$$

$$
= \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl} + \alpha_{jl})} \prod_{i=1}^{r_j} \frac{\Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}
$$

The result follows from the standard calculations in Bayesian inference by setting

$$
\theta \mid x_{mb(j)}^{(l)} \sim \text{Dirichlet}(\alpha_{1jl}, \ldots, \alpha_{rjl})
$$

where $(\alpha_{1jl}, \ldots, \alpha_{rjl})$ are hyperparameters such that $\alpha_{jl} = \sum_{i=1}^{r_j} \alpha_{ijl}$. We use symmetric Dirichlet distributions since we assume there is no prior knowledge favoring one parameter in $(\theta_{1jl}, \ldots, \theta_{rjl})$ over any of the others. More specifically, we use a modified version of the non-informative prior of Buntine (1991), originally defined for Bayesian networks, for which the hyperparameters are determined according to

$$
\alpha_{ijl} = \frac{N}{|X_j| \cdot |X_{mb(j)}|} = \frac{N}{r_j \cdot q_j}
$$
where $N$ is the equivalent sample size adjusting the strength of the prior.

The MPL for an undirected graph of a Markov network is to a considerable degree similar to the marginal likelihood for a directed acyclic graph of a Bayesian network. However, the parent sets of a directed acyclic graph are replaced by the Markov blankets of an undirected graph and the acyclicity constraint is replaced by a constraint of consistency among the Markov blankets. As a consequence of the parameter independence assumption, the MPL of a Markov network can in fact be considered the actual marginal likelihood of a bi-directional dependency network (Heckerman et al., 2001). Even though a bi-directional dependency network is more complex as a model, they are easier to evaluate from a computational perspective in terms of structure learning. Furthermore, Markov networks and dependency networks are strongly connected in the sense that the set of positive distributions that can be encoded by a Markov network with the undirected graph $G$ is equal to the set of positive distributions that can be encoded by a consistent dependency network with the same structural adjacencies as in $G$ (Heckerman et al., 2001, Theorem 3). Furthermore, in order for a consistent dependency network to be minimal it must be bi-directional (Heckerman et al.] 2001 Theorem 4). These observations motivate the MPL from a more theoretical perspective, in addition to its favorable properties in terms of computational complexity.

The MPL possesses several advantageous properties. The parameter prior offers a natural regularization that prevents the score from overfitting the data. Methods that explicitly penalize the degree of regularization are sensitive to the choice of some tuning parameter which usually has to be determined empirically. In contrast, the MPL requires specification of the hyperparameters in the Dirichlet distribution. In our above formulation, this boils down to setting a value on $N$, which is usually a less critical choice than selecting a parameter that explicitly penalizes overfitting. The fact that there exists an analytical solution makes the MPL easy and fast to evaluate. Furthermore, its variable-wise factorization

$$
\hat{p}(\mathbf{X} \mid G) = \prod_{j=1}^{d} p(\mathbf{X}_j \mid \mathbf{X}_{mb(j)})
$$

makes it a convenient candidate for search algorithms based on local changes. To compare the plausibility of two graphs, $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, we can calculate the ratio of their MPLs,

$$
K(G_1, G_2) = \frac{\hat{p}(\mathbf{X} \mid G_1)}{\hat{p}(\mathbf{X} \mid G_2)},
$$

which is basically the pseudo-version of Bayes factor or Bayes pseudo-factor. Assume there is a single edge difference,

$$
\{E_1 \cup E_2\} \setminus \{E_1 \cap E_2\} = \{i, j\},
$$

between the graphs. This implies that $mb(i)$ and $mb(j)$ are the the only Markov blankets that differ between the graphs. Consequently, Bayes pseudo-factor is simply evaluated by

$$
K(G_1, G_2) = \frac{p(\mathbf{X}_i \mid \mathbf{X}_{mb_1(i)})}{p(\mathbf{X}_i \mid \mathbf{X}_{mb_2(i)})} \cdot \frac{p(\mathbf{X}_j \mid \mathbf{X}_{mb_1(j)})}{p(\mathbf{X}_j \mid \mathbf{X}_{mb_2(j)})}
$$
since the rest of the factors cancel each other out.

Another important property preferably satisfied by a scoring function is consistency, that is, the score will eventually favor the true graph when the sample size tends to infinity. The following theorem establishes that MPL is indeed a consistent scoring function for Markov networks.

**Theorem 1** Let $G^* \in \mathcal{G}$ be the true graph structure, of a Markov network over $(X_1, \ldots, X_d)$, with the corresponding Markov blankets $mb(G^*) = \{mb^*(1), \ldots, mb^*(d)\}$. Let $\theta_{G^*} \in \Theta_{G^*}$ define the corresponding joint distribution to which $G^*$ is faithful and from which a sample $X$ of size $n$ is obtained. The local MPL estimator

$$\hat{mb}(j) = \arg \max_{mb(j) \subseteq V \setminus j} p(X_j \mid X_{mb(j)})$$

is consistent in the sense that $\hat{mb}(j) = mb^*(j)$ eventually almost surely as $n \to \infty$ for $j = 1, \ldots, d$. Consequently, the global MPL estimator

$$\hat{G} = \arg \max_{G \in \mathcal{G}} \hat{p}(X \mid G)$$

is consistent in the sense that $\hat{G} = G^*$ eventually almost surely as $n \to \infty$.

**Proof** See Appendix A.

5. Search algorithm for MPL optimization

The variable-wise factorization of the MPL makes it directly applicable for search algorithms such as greedy hill-climbing and MCMC-based methods. Still, such global search methods tend to be rendered inefficient when the number of variables grows large enough. Hence, to optimize the MPL score, we propose a divide-and-conquer type approach that can be efficiently applied also in a high-dimensional setting. Initially, we ignore the Markov blanket consistency constraint and optimize the local MPL estimators independently. The first phase is considered a pre-scan reducing the size of the model space. The global MPL estimator is then optimized with respect to the reduced model space.

The straightforward MPL based optimization problem can be formulated as

$$\arg \max_{G \in \mathcal{G}} \hat{p}(X \mid G) \cdot p(G)$$

(9)

where $p(G)$ is the prior distribution over the graphs accounting for prior belief regarding for example degree of sparsity. In the remainder of the section we assume a uniform prior and the term $p(G)$ is therefore omitted. However, the following derivation could be done for any prior that follows the same factorization as the MPL.

Since each graph $G$ is uniquely specified by its collection of Markov blankets $mb(G) = \{mb(1), \ldots, mb(d)\}$, we can reformulate (9) as

$$\arg \max_{mb(G) \in \times_{j \in V} P(V \setminus j)} \prod_{j=1}^d p(X_j \mid X_{mb(j)})$$

subject to $i \in mb(j) \Rightarrow j \in mb(i)$ for all $i, j \in V$
where \( \mathcal{P}(V \setminus j) \) is the power set of \( V \setminus j \) representing all possible Markov blankets of node \( j \). From (10) it is easy to see that our problem is basically made up of \( d \) dependent subproblems that are connected through the consistency constraint. By omitting the constraint we remove the dependence among the subproblems and obtain a much simpler relaxed version of the original problem as

\[
\arg\max_{mb(G) \in \mathcal{X} \setminus j \mathcal{P}(V \setminus j)} \prod_{j=1}^{d} p(X_j | X_{mb(j)}),
\]

(11)

Since the \( d \) subproblems now are independent of each other, we can finally reformulate (11) by breaking it down into a collection of stand-alone subproblems,

\[
\arg\max_{mb(j) \subseteq V \setminus j} p(X_j | X_{mb(j)}) \quad \text{for } j = 1, \ldots, d,
\]

(12)

which can be solved very efficiently since the independent subproblems can be solved completely in parallel. This approach ultimately enables our search method to be applied to genuinely high-dimensional problems. In a sense, the relaxation shifts the focus from the strictly score-based view in (9) towards a more constraint-based view. It is worth noticing that the consistency result established in Theorem 1 still holds under the relaxed problem formulation due to the consistency of the local MPL estimators.

To identify the score-optimal Markov blankets we use the search method described in Algorithm 1. The algorithm is based on the two basic operations add and delete. It starts with an empty Markov blanket \( mb(j) = \emptyset \) and the candidate set \( C = V \setminus j \). At each iteration it adds to the Markov blanket the node \( i \in C \) that induces the greatest improvement to the local score \( p(X_j | X_{mb(j)}) \) and updates \( mb(j) \) and \( C \) accordingly. When the size of the Markov blanket grows larger than two, the algorithm interleaves each successful addition-step with a deletion phase.

In the deletion phase, the algorithm removes the node that induces the largest improvement to score. This deletion-step is repeated until removal of a node no longer increases the score or the size of the Markov blanket is smaller than three. After each successful removal phase, \( mb(j) \) is updated accordingly. When the addition-step is no longer successful in the sense that no node \( i \in C \) induces an improvement to the score, the algorithm is terminated and it returns the current \( mb(j) \). The recurring deletion-phase of the algorithm attempts to keep the size of \( mb(j) \) as small as possible during the search in order to improve both sample and time efficiency. Our search procedure is similar to the interIAMB algorithm described by Tsamardinos et al. (2003), however, the assessment of \( mb(j) \) is done differently. In the next section we compare the two approaches.

By solving the relaxed problem in (11) instead of the original problem in (10), we usually obtain a solution inconsistent with a Markov network. In fact, we actually identify the structure of a general dependency network maximized with respect to the MPL. Since the structure of a general dependency network is usually inconsistent with a Markov network, some post-processing is therefore necessary in order to make the solution satisfy the restrictions of a Markov network (or bi-directional dependency network). This is similar in spirit to the feature selection method of Lowd and Davis (2010). However, Lowd and Davis (2010) learn a probabilistic decision tree for each variable and the trees are then
Algorithm 1 Procedure for discovering the Markov blanket of node $j$.

Procedure discover-MarkovBlanket(
    $j$, // Current node
    $X$, // Complete data set
) returns $mb(j)$

1: $mb(j) \leftarrow \emptyset$
2: $C \leftarrow V \setminus j$
3: $addMB \leftarrow \text{True}$
4: while $addMB$
5:     $mb^*(j) \leftarrow mb(j)$
6:     for $i \in C$
7:         if $p(X_j \mid X_{mb(j) \cup i}) > p(X_j \mid X_{mb^*(j)})$
8:             $mb^*(j) \leftarrow mb(j) \cup i$
9:     end
10: if $mb^*(j) \neq mb(j)$
11:     $mb(j) \leftarrow mb^*(j)$
12:     $C \leftarrow C \setminus mb(j)$
13:     $addMB \leftarrow \text{True}$
14:     $delMB \leftarrow \text{True}$
15: else
16:     $addMB \leftarrow \text{False}$
17:     $delMB \leftarrow \text{False}$
18: end
19: while $delMB$ and $\text{size}(mb(j)) > 2$
20:     $mb^*(j) \leftarrow mb(j)$
21:     for $i \in mb(j)$
22:         if $p(X_j \mid X_{mb(j) \setminus i}) > p(X_j \mid X_{mb^*(j)})$
23:             $mb^*(j) \leftarrow mb(j) \setminus i$
24:     end
25: end
26: if $mb^*(j) \neq mb(j)$
27:     $mb(j) \leftarrow mb^*(j)$
28:     $delMB \leftarrow \text{True}$
29: else
30:     $delMB \leftarrow \text{False}$
31: end
32: end
33: $return \ mb(j)$
transformed into features. The authors mention the risk of overfitting by generating long specialized features. Since long features correspond to dense graphs, the implication of such overfitting is emphasized when the aim is graph structure discovery rather than the more specific feature selection.

In the next section we investigate three different approaches of transforming a set of inconsistent Markov blankets into a set of edges specifying the graph structure of a Markov network. A straightforward way of doing this is to use either an AND-criterion,

\[ E_{AND} = \{ \{i, j\} \in \{V \times V\} : i \in mb(j) \land j \in mb(i)\} \]

or an OR-criterion,

\[ E_{OR} = \{ \{i, j\} \in \{V \times V\} : i \in mb(j) \lor j \in mb(i)\} \].

These two basic approaches are very much in line with standard constraint-based methods. Therefore we also propose a third more elaborate method that allows us to take further advantage of our underlying score-based approach. We consider the edges in \( E_{OR} \) to be a result of a pre-scan that identifies eligible edges. The original problem \( (9) \) is then solved with respect to the subset \( G_{OR} = \{ G \in \mathcal{G} : E \subseteq E_{OR}\} \), that is

\[ \arg \max_{G \in G_{OR}} \hat{p}(X \mid G) \].

Since \( G_{OR} \) is in general considerably smaller than \( \mathcal{G} \), the above problem can be solved efficiently by some appropriate method, such as the non-reversible MCMC approach of Corander et al. (2008). However, here we apply a simpler deterministic greedy hill-climb (HC) approach that at each iteration add or remove an edge according to the principle of maximum score improvement. The algorithm is terminated when no further improvement can be achieved and the current edge set \( E_{HC} \) is returned. This method is similar in spirit to the max-min hill-climbing algorithm for learning Bayesian networks by Tsamardinos et al. (2006). In our approach, however, both phases are derived from the notion of maximizing a single underlying score.

The performance of each method depends largely on the quality of the Markov blankets discovered in the first phase. In a setting where many false positives have been added in the first phase, it is beneficial to apply some form of pruning to stabilize the search. Consequently, the HC- or even the AND-method are to be preferred in this scenario. On the other hand, if the Markov blankets contain very few false positives (or even none), the OR-method is naturally the optimal choice since the other methods cannot reduce the number of false negatives but merely reduce the number of false positives.

6. Experimental results

The main focus of this section is to empirically investigate the performance of the MPL using our optimization algorithms. To illustrate the potential of our method, we also present a real high-dimensional problem on which our method was applied.

We used synthetic models to generate data sets of different sizes to systematically compare the performance of the MPL under our different optimization algorithms. Moreover,
we compared the MPL against different competing methods for structure learning of Markov networks. Since the generating underlying network structure is known in each individual experiment, we assessed the quality of the learned models in terms of rates of true positive (TP) and false positive (FP) edges as well as the Hamming distance (HD) from the true model. The HD reflects the structural difference between two models by accounting for the number of edge additions and deletions required to transform one of the models into the other. Since we are comparing with the true model, the HD is simply the sum of false positives and false negatives. Consequently, low values on the HD correspond to structural resemblance between the identified and true graph. In addition to the structural properties, we are monitoring the execution times for the different methods. The total runtimes of all algorithms are reported along with the average discovery time for a single Markov blanket. The average time can be considered a rough approximation of the total real Markov blanket discovery time required if the local problems were solved in parallel. In reality, this is indeed an underestimation since the real total time after a parallelization would equal the maximum time which depends on the size of the neighborhood.

For the synthetic part of the experiments, the prior $p(G)$ is kept uniform. For simplicity we restrict the simulations to binary variables, however, extending our method to non-binary variables is straightforward. In all experiments we set the equivalent sample size $N = 1$ which results in a weak non-informative prior.

The generating structures were formed by combining disconnected components in form of the four 16-node graphs illustrated in Figure 4 (see Appendix B). These graphs represent different structural characteristics present in realistic models and some of their properties are listed in Table 1 (see Appendix B). In particular, the hub network in Figure 4b illustrates a structural characteristic that is especially hard to capture for MPL type methods even though it is a rather simple tree-structure. This is emphasized by the AND-method since all edges must be captured from both directions. The OR-method can to some extent circumvent this problem since it suffices if each branch edge is captured from the corresponding leaf node. In this situation, the OR-method can be more data efficient in terms of HD, however, it requires a stable first phase with a low number of FPs. Finally, the MPL-HC is expected to yield a compromise by comparing the potential benefit from adding $j$ to $mb(i)$ against the potential loss of adding $i$ to $mb(j)$. This act of balancing yields a stability that will prove beneficial for large models when the sample size is limited.

Initially, one replica of each subgraph were combined to form a structure over 64 variables. This procedure was then repeated with 2, 4, and 8 replicas to form network structures over 128, 256, and 512 variables, respectively. Each final network structure thus contained all the structural characteristics present in the subgraphs. The disconnected nature of the generating networks facilitated the sampling procedure substantially since each distinct subnetwork was sampled directly from its corresponding joint distribution independently of the rest of the network. In practice, a distribution was created by randomly generating the factors in (1). Each factor value $\phi(x_C)$ was drawn, independently of the other values, from a uniform distribution over $(0,1)$. Consequently, the strength of the dependences entailed by the edges may have varied considerably. To increase the stability of our results, for each sample size and graph structure we generated 10 distributions from each of which

---

1. All experiments were carried out in Matlab on a standard PC architecture with 2.66GHz dual-core processors.
First we compare the different methods of transforming the set of (usually inconsistent) Markov blankets discovered by Algorithm 1 into an undirected graph. We refer to the methods as MPL-AND, -HC, and -OR. The quality of the models in terms of HD is illustrated in Figure 1. The HD-curves illustrate how the first phase of the search process affects the performance of the methods. The identification of the Markov blankets is naturally more unstable for smaller sample sizes. In such situations it is beneficial to perform some pruning to reduce the number of FPs. Consequently, MPL-AND and MPL-HC outperform MPL-OR when the number of data observations is small in relation to the network size. For small sample sizes, it is also clear that MPL-OR is the method that suffers most when the size of the model is increased. MPL-HC and MPL-AND are more stable in their performance. As the sample size is increased, so is the stability of the initial phase. At some sample size when

Figure 1: HD-curves for MPL-AND, -HC, and -OR.

10 datasets were generated. In total, under each setup we learned 100 model structures over which the final results were averaged. The experiments were performed for data sizes ranging from 250 to 32000. The results of the simulations are summarized in Table 2 and 3 (see Appendix B).
the number of FPs is brought to a sufficiently low level, it is obvious that the OR-method will be the optimal choice since both $E_{AND}$ and $E_{HC}$ are subsets of $E_{OR}$. The point where MPL-OR overtakes the other methods is shifted towards larger sample sizes when the size of the model is increased. Considering all the sample and network sizes, MPL-HC performs best throughout the experiments in terms of HD. At smaller sample sizes, its ability to prune FPs makes it much more stable than MPL-OR. At larger sample sizes, its compromising nature allows it to include more TPs than the overly conservative MPL-AND. Its advantageous properties are emphasized for medium-sized samples for which it outperforms both of the other methods. However, when choosing method it eventually comes down to what structural properties one prioritizes in terms of TPs contra FPs.

In the second part we compare MPL-HC against similar divide-and-conquer type structure learning methods that are applicable in high dimensions. The first method is based on the Markov blanket discovery approach of Tsamardinos et al. (2003) who use conditional mutual information $\text{CMI}$ to assess whether two variables are conditionally independent given some set of variables. For a fair comparison against MPL, we use the CMI measure combined with Algorithm 1. The second method is the $L_1$-regularized logistic regression $\text{L1LR}$ approach of Ravikumar et al. (2010). The L1LR approach is directly applicable on our models since we have restricted the experiments to binary variables. Ravikumar et al. (2010) briefly outline an extension of L1LR to general Markov networks, however, the MPL-and CMI-method may as such be readily extended to a non-binary setting. As for the first phase of MPL, these methods may yield inconsistent Markov blankets. Therefore the AND- and the OR-method were applied to form consistent Markov network structures. An issue with both the CMI- and the L1LR-method is that they require the user to specify a crucial tuning parameter in form of a threshold value and a regularization weight, respectively. Consequently, to circumvent this problem each method were executed for a range of values on the threshold value,

$$\lambda_{\text{CMI}} = \{0.1, 0.075, 0.05, 0.025, 0.01, 0.0075, 0.005, 0.0025, 0.001, 0.00075, 0.0005, 0.00025\},$$

and on the regularization parameter,

$$\lambda_{\text{L1LR}} = \{6, 8, 12, 16, 24, 32, 48, 64, 96, 128, 192, 256\}.$$

This resulted in a range of overly sparse graphs to overly dense. For each specific setup we then chose the value for which the methods performed optimally in terms of HD. The results of the simulations are once again summarized in Table 2 and 3. The quality of the identified graphs in terms of HD is illustrated in Figure 2. For CMI and L1LR, the HDs were computed with respect to either the AND- or the OR-graph depending on which one yielded the lowest distance. Overall, MPL-HC performed highly satisfactorily and was marginally inferior to the other methods only for the smallest sample size for $d = 128$, 256 and the two smallest sample sizes for $d = 512$.

Finally, to illustrate the MPL for a high-dimensional real application, we consider a dataset of 1,000 aligned whole-genome DNA sequences of *Mycobacterium tuberculosis*.

---

2. To calculate the conditional mutual information, the Matlab package of Peng (2007, Accessed 2013-10-14) was used.

3. The $L_1$-regularized logistic regression was performed using the Matlab package of Schmidt (2013, Accessed 2013-10-14).
A Markov network can be used to reveal direct associations between variation over genome positions that may be relatively distant from each other. This purpose is similar to the use of Markov networks for finding direct dependences among amino acid sequence positions that relate to the underlying crystal structure (Ekeberg et al., 2013). The original 5Mb multiple sequence alignment for *M. tuberculosis* had approximately 27,000 variable positions, out of which we chose 561 that displayed sufficient variability determined by the threshold that the most frequent DNA base in a variable position was not allowed to represent more than 90% of the total number of observations. Similar to the amino acid dependency modeling with Markov networks, associations between genome positions that are close neighbors in the sequences are trivial and uninteresting for the biological purposes. We applied the MPL-HC method on the 561 variables under a sparsity promoting prior of type

\[ p(G) \sim \prod_{j=1}^{d} 2^{-q_j(r_j-1)}. \]
Figure 3: Identified association pattern for 1kb genome intervals in *M. tuberculosis*. Black color indicates found linkage between positions that reside within the intervals located in the genome according to the values on the horizontal and vertical axes (in thousands of nucleotides).

The variable positions included in the analysis were separated maximally by almost 2.5 million bases since the bacterial genome is circular. To enable an efficient illustration of the results, the genome alignment was first split into non-overlapping intervals of 1000 (1kb) successive positions. Then, an adjacency matrix was created for the pairs of genome intervals such that a pair of intervals was determined adjacent if the MPL-HC learned Markov network contained an edge between any variable positions in the two intervals, respectively. The resulting adjacency structure is illustrated in Figure 3, which succinctly demonstrates the long-distance linkage of genome variation in this bacterium, as expected on the basis of its very low recombination rate.

7. Discussion

In this work we have introduced a novel approach for learning the structure of a Markov network without imposing restrictions on chordality. Our marginal pseudo-likelihood scoring method is proven to be consistent and it can also be considered as a small sample analytical version of the information theoretic PIC criterion (Csiszár and Talata [2006]). In comparison with the other methods for structural learning of Markov networks, our MPL method was overall best, and only marginally inferior to alternatives under fairly extreme 'large \( p \), small \( n \)' settings. Moreover, it should also be kept in mind that we chose the tuning and weight parameter values for the other methods by optimizing their performance against the known underlying structures in each analyzed generating model, to reduce the computational burden of the experiments. In a real data analysis scenario, it would be nec-
necessary to tune the other methods using for example cross-validation, which could plausibly lead to less good out-sample performance. In this sense the comparison was extremely fair for the alternative methods, since no parameter choice was made for the MPL method by the resulting performance.

By considering the execution times in Table 3, it can be deduced that MPL-HC is not as fast as the other methods, however, it is still at the comparable level of computational complexity. The additional computation times appears to be well balanced by the increase in the accuracy of inferences. The straightforward possibility of parallel use of the MPL makes it a viable candidate for high-dimensional knowledge discovery. Furthermore, the runtimes listed in Table 3 for CMI and L1LR are under a best case scenario. In a real application the value of the tuning parameter would have to be determined by some appropriate method. If this is done experimentally in a standard manner by cross-validation, the computation times would easily exceed those reported for MPL.

One drawback of the MPL compared to the true marginal likelihood is that the former in a way over-determines the dependence structure. Although this adds stability against overfitting, it also makes it less data efficient compared to the true marginal likelihood. On the other hand, calculation of true marginal likelihood remains still intractable for non-chordal Markov networks of realistic size and MPL enjoys consistency, which combined with its solid performance in our experiments suggests that this approach has considerable potential both for applications and further theoretical development.

Appendix A.

In this appendix we prove Theorem 1 from Section 4:

**Theorem** Let $G^* \in \mathcal{G}$ be the true graph structure, of a Markov network over $(X_1, \ldots, X_d)$, with the corresponding Markov blankets $mb(G^*) = \{mb^*(1), \ldots, mb^*(d)\}$. Let $\theta_{G^*} \in \Theta_{G^*}$ define the corresponding joint distribution to which $G^*$ is faithful and from which a sample $X$ of size $n$ is obtained. The local MPL estimator

$$\hat{mb}(j) = \arg\max_{mb(j) \subseteq V \setminus j} p(X_j | X_{mb(j)}) \quad (13)$$

is consistent in the sense that $\hat{mb}(j) = mb^*(j)$ eventually almost surely as $n \to \infty$ for $j = 1, \ldots, d$. Consequently, the global MPL estimator

$$\hat{G} = \arg\max_{G \in \mathcal{G}} \hat{p}(X | G) \quad (14)$$

is consistent in the sense that $\hat{G} = G^*$ eventually almost surely as $n \to \infty$.

**Proof** We investigate the asymptotic behavior of the global log-MPL for a complete graph,

$$\log \hat{p}(X | G) = \sum_{j=1}^{d} \log p(X_j | X_{mb(j)}),$$
by considering the local log-MPL for a single variable:

\[
\log p(X_j \mid X_{mb(j)}) = \sum_{l=1}^{q_j} \left[ \log \Gamma(\alpha_{jl}) - \log \Gamma(n_{jl} + \alpha_{jl}) \right]
\]

\[
+ \sum_{i=1}^{r_j} \left[ \log \Gamma(n_{ijl} + \alpha_{ijl}) - \log \Gamma(\alpha_{ijl}) \right]
\]

\[
= \sum_{l=1}^{q_j} \left[ -\log \Gamma(n_{jl} + \alpha_{jl}) + \sum_{i=1}^{r_j} \log \Gamma(n_{ijl} + \alpha_{ijl}) \right]
\]

\[
+ \sum_{l=1}^{q_j} \log \Gamma(\alpha_{jl}) - \sum_{i=1}^{r_j} \log \Gamma(\alpha_{ijl})
\]

Since

\[
\sum_{l=1}^{q_j} \log \Gamma(\alpha_{jl}) - \sum_{i=1}^{r_j} \log \Gamma(\alpha_{ijl})
\]

is constant, it is henceforth omitted, which introduces an \(O(1)\) error. We now let \(n \to \infty\) and apply Stirling’s asymptotic formula,

\[
\log \Gamma(n) = (n - \frac{1}{2}) \log n - n + \frac{1}{2} \log 2\pi
\]

\[
= (n - \frac{1}{2}) \log n - n + O(1),
\]

on the remaining terms:

\[
\log p(X_j \mid X_{mb(j)}) = \sum_{l=1}^{q_j} \left[ -(n_{jl} + \alpha_{jl} - \frac{1}{2}) \log(n_{jl} + \alpha_{jl}) + (n_{jl} + \alpha_{jl}) \right]
\]

\[
+ \sum_{i=1}^{r_j} \left[ (n_{ijl} + \alpha_{ijl} - \frac{1}{2}) \log(n_{ijl} + \alpha_{ijl}) - (n_{ijl} + \alpha_{ijl}) \right]
\]

\[
+ O(1)
\]

\[
= \sum_{l=1}^{q_j} \sum_{i=1}^{r_j} n_{ijl} \log \frac{n_{ijl} + \alpha_{ijl}}{n_{jl} + \alpha_{jl}}
\]

\[
+ \sum_{l=1}^{q_j} \sum_{i=1}^{r_j} \alpha_{ijl} \log \frac{n_{ijl} + \alpha_{ijl}}{n_{jl} + \alpha_{jl}}
\]

\[
+ \sum_{l=1}^{q_j} \frac{1}{2} \log(n_{jl} + \alpha_{jl}) - \sum_{i=1}^{r_j} \frac{1}{2} \log(n_{ijl} + \alpha_{ijl})
\]

\[
+ O(1)
\]

The second step is allowed since \(n_{jl} = \sum_{i=1}^{r_j} n_{ijl}\) and \(\alpha_{jl} = \sum_{i=1}^{r_j} \alpha_{ijl}\). As \(n \to \infty\) we have that

\[
\frac{n_{ijl} + \alpha_{ijl}}{n_{jl} + \alpha_{jl}} = \frac{n_{ijl}(1 + \frac{\alpha_{ijl}}{n_{ijl}})}{n_{jl}(1 + \frac{\alpha_{ijl}}{n_{ijl}})} = \frac{n_{ijl}}{n_{jl}}
\]
Pensar, Nyman and Corander

and, consequently,
\[
\sum_{l=1}^{q_j} \sum_{i=1}^{r_j} n_{ijl} \log \frac{n_{ijl} + \alpha_{ijl}}{n_{jl} + \alpha_{jl}} = \sum_{l=1}^{q_j} \sum_{i=1}^{r_j} n_{ijl} \log \frac{n_{ijl}}{n_{jl}}
\]
which is the logarithm of the maximum pseudo-likelihood (here denoted by \( \log PL_{\text{max}} \)) as defined by Csiszár and Talata (2006). Furthermore, since \( n_{ijl}/n_{jl} \) is the maximum likelihood of the corresponding conditional probability, omitting the second term introduces an \( O(1) \) error. Finally, the third term can be rewritten as
\[
\sum_{l=1}^{q_j} \left[ \frac{1}{2} \log(n_{jl} + \alpha_{jl}) - \sum_{i=1}^{r_j} \frac{1}{2} \log(n_{ijl} + \alpha_{ijl}) \right]
\]
\[
= \frac{1}{2} \sum_{l=1}^{q_j} \log \frac{n_{jl} + \alpha_{jl}}{n} - \log n - \sum_{i=1}^{r_j} \left( \log \frac{n_{ijl} + \alpha_{ijl}}{n} - \log n \right)
\]
\[
= \frac{1}{2} \sum_{l=1}^{q_j} \left[ - \log n + \sum_{i=1}^{r_j} \log n \right] + \frac{1}{2} \sum_{l=1}^{q_j} \left[ \log \frac{n_{jl} + \alpha_{jl}}{n} - \sum_{i=1}^{r_j} \log \frac{n_{ijl} + \alpha_{ijl}}{n} \right]
\]
\[
= \frac{1}{2} \sum_{l=1}^{q_j} \left[ - \log n + \sum_{i=1}^{r_j} \log n \right] + O(1)
\]
\[
= \frac{1}{2} (-q_j \log n + q_j r_j \log n) + O(1)
\]
\[
= \frac{q_j (r_j - 1)}{2} \log n + O(1)
\]
Every approximation made during the derivation has introduced an error of \( O(1) \) with respect to \( n \). In other words, when \( n \to \infty \) we have that
\[
\log p(X_j \mid X_{mb(j)}) = \log PL_{\text{max}} - \frac{(r_j - 1)q_j}{2} \log n + O(1).
\]
Since the \( O(1) \) term does not grow with \( n \), the local log-MPL is asymptotically equivalent to
\[
\log p(X_j \mid X_{mb(j)}) = \log PL_{\text{max}} - \frac{(r_j - 1)q_j}{2} \log n.
\]
Furthermore, as \( n \to \infty \), the original problem of (13) can be reformulated as
\[
\arg \min_{mb(j) \subseteq V \setminus j} - \log PL_{\text{max}} + \frac{(r_j - 1)}{2} \cdot q_j \log n
\]
which is equivalent to the PIC estimator of Csiszár and Talata (2006) except that the penalty term is multiplied with a variable specific constant \( c_j = (r_j - 1)/2 \). However, the PIC estimator is proven consistent by Csiszár and Talata (2006), who also note that their results remain valid when the PIC penalty term is multiplied by any constant \( c > 0 \). Consequently, the local MPL estimator (13) is also consistent.

Since the local MPL estimator is consistent, the true collection of Markov blankets is eventually identified when \( n \to \infty \). A set of Markov blankets uniquely specifies the structure of a Markov network. Since the true model structure satisfies the structural properties
of a Markov network, that is $i \in mb(j)$ if $j \in mb(i)$, the global MPL estimator (14) is consistent.

Appendix B.

This appendix contains supplementary material of Section 6:

(a) Grid network.

(b) Hub network.

(c) Loop network.

(d) Clique network.

Figure 4: Synthetic subgraphs.

| Network     | Grid | Hub | Loop | Clique |
|-------------|------|-----|------|--------|
| Number of nodes | 16   | 16  | 16   | 16     |
| Number of edges | 24   | 15  | 19   | 20     |
| MB sizes      | $\{2, 3, 4\}$ | $\{1, 8\}$ | $\{2, 4\}$ | $\{0, 1, 2, 3, 4\}$ |
| Average MB size | 3.25 | 1.88 | 2.38 | 2.5    |
| Decomposable  | No   | Yes | No   | Yes    |

Table 1: Properties of the synthetic subgraphs.
Table 2: TP and FP rates.

| Size (MB) | MPL (TP rate / FP rate) | CMI (TP rate / FP rate) | L1LR (TP rate / FP rate) |
|-----------|---------------------------|--------------------------|--------------------------|
| 1000/0   | 0.36 / 0.0015            | 0.43 / 0.0032            | 0.51 / 0.0114             |
| 2000/0   | 0.29 / 0.0007            | 0.27 / 0.0004            | 0.31 / 0.0009             |
| 3000/0   | 0.43 / 0.0007            | 0.52 / 0.0014            | 0.61 / 0.0068             |
| 4000/0   | 0.54 / 0.0002            | 0.62 / 0.0006            | 0.70 / 0.0028             |
| 5000/0   | 0.64 / 0.0001            | 0.71 / 0.0002            | 0.78 / 0.0016             |
| 6000/0   | 0.74 / 0.0000            | 0.79 / 0.0001            | 0.87 / 0.0006             |
| 7000/0   | 0.80 / 0.0000            | 0.84 / 0.0000            | 0.91 / 0.0004             |
| 8000/0   | 0.81 / 0.0000            | 0.84 / 0.0000            | 0.89 / 0.0003             |
| 9000/0   | 0.84 / 0.0000            | 0.88 / 0.0000            | 0.92 / 0.0002             |
| 10000/0  | 0.80 / 0.0000            | 0.84 / 0.0000            | 0.85 / 0.0004             |
| 11000/0  | 0.80 / 0.0000            | 0.84 / 0.0000            | 0.85 / 0.0004             |
| 12000/0  | 0.80 / 0.0000            | 0.84 / 0.0000            | 0.85 / 0.0004             |
| $d$  | $n$  | **MPL (s)** | **CMI (s)** | **L1LR (s)** |
|------|------|-------------|-------------|-------------|
|      |      | MB discovery (Total / Node avg.) | MB discovery (Total / Node avg.) | MB discovery (Total / Node avg.) |
|      |      | MB discovery (AND)          | MB discovery (OR)          | MB discovery (AND)          | MB discovery (OR)          |
|      |      | HC                        |                          |                          |                          |
| 250  | 250  | 3.19 / 0.05               | 0.25 0.02                | 0.96 / 0.01              | 0.96 / 0.02              |
|      | 500  | 3.76 / 0.06               | 0.30 0.03                | 1.23 / 0.02              | 1.26 / 0.02              |
| 64   | 1000 | 5.21 / 0.08               | 0.41 0.12                | 1.65 / 0.03              | 1.64 / 0.03              |
|      | 2000 | 9.11 / 0.14               | 0.82 1.38                | 2.83 / 0.04              | 2.80 / 0.04              |
|      | 4000 | 19.01 / 0.29              | 1.55 4.38                | 6.32 / 0.10              | 6.18 / 0.10              |
|      | 8000 | 45.18 / 0.71              | 3.55 7.43                | 20.22 / 0.32             | 19.29 / 0.30             |
|      | 16000| 123.34 / 1.92             | 8.02 16.44               | 57.93 / 0.91             | 55.57 / 0.87             |
|      | 32000| 491.87 / 7.69             | 33.37 41.66              | 133.12 / 2.08            | 121.85 / 1.90            |
| 128  | 250  | 15.73 / 0.12              | 0.65 1.81                | 2.82 / 0.02              | 2.82 / 0.02              |
|      | 500  | 15.94 / 0.12              | 0.63 4.58                | 3.76 / 0.03              | 3.72 / 0.03              |
|      | 1000 | 21.94 / 0.17              | 0.88 16.15               | 6.36 / 0.05              | 6.36 / 0.05              |
|      | 2000 | 38.79 / 0.30              | 1.57 5.74                | 16.15 / 0.13             | 15.81 / 0.12             |
|      | 4000 | 79.72 / 0.62              | 3.08 16.84               | 36.17 / 0.28             | 34.65 / 0.27             |
|      | 8000 | 175.49 / 1.37             | 7.03 34.01               | 96.31 / 0.75             | 84.86 / 0.66             |
|      | 16000| 526.94 / 4.12             | 16.72 71.31              | 260.32 / 2.03            | 228.18 / 1.78            |
|      | 32000| 1892.42 / 14.78           | 64.61 36.17              | 676.22 / 5.28            | 523.68 / 4.09            |
| 256  | 250  | 81.31 / 0.32              | 1.86 6.75                | 8.51 / 0.03              | 8.57 / 0.03              |
|      | 500  | 76.09 / 0.30              | 1.61 9.15                | 14.23 / 0.6              | 14.56 / 0.06             |
|      | 1000 | 97.21 / 0.38              | 2.11 20.04               | 37.71 / 0.15             | 37.13 / 0.15             |
|      | 2000 | 152.94 / 0.60             | 3.25 22.95               | 95.94 / 0.37             | 84.11 / 0.33             |
|      | 4000 | 325.44 / 1.27             | 6.58 32.03               | 193.84 / 0.76            | 160.74 / 0.63            |
|      | 8000 | 724.05 / 2.83             | 13.81 120.79             | 538.15 / 2.10            | 386.50 / 1.51            |
|      | 16000| 2174.68 / 8.49            | 37.16 273.08             | 1433.34 / 5.60           | 1018.36 / 3.98           |
|      | 32000| 8242.36 / 32.20           | 127.37 704.34            | 3026.36 / 11.82          | 1888.86 / 7.38           |
| 512  | 250  | 458.97 / 0.90             | 6.18 28.14               | 38.48 / 0.08             | 43.69 / 0.09             |
|      | 500  | 347.58 / 0.68             | 4.21 193.27              | 90.39 / 0.18             | 94.10 / 0.18             |
|      | 1000 | 412.79 / 0.81             | 4.81 1005.10             | 189.10 / 0.37            | 177.72 / 0.35            |
|      | 2000 | 643.15 / 1.26             | 7.11 90.34               | 439.75 / 0.86            | 364.93 / 0.71            |
|      | 4000 | 1299.13 / 2.54            | 13.31 285.09             | 1088.14 / 2.13           | 804.38 / 1.57            |
|      | 8000 | 3043.55 / 5.95            | 30.25 469.76             | 1890.36 / 3.70           | 1428.42 / 2.79           |
|      | 16000| 8762.72 / 17.11           | 71.77 961.93             | 5402.52 / 10.55          | 2557.80 / 5.00           |
|      | 32000| 32440.35 / 63.36          | 250.39 2289.73           | 12602.53 / 24.61         | 8678.51 / 16.95          |

Table 3: Execution times.
References

A. Anandkumar, V. Y. F. Tan, F. Huang, and A. S. Willsky. High-dimensional structure estimation in Ising models: Local separation criterion. *The Annals of Statistics*, 40:1346–1375, 2012.

E. Aurell and M. Ekeberg. Inverse Ising inference using all the data. *Physical Review Letters*, 108:090201, 2012.

J. E. Besag. Nearest-neighbour systems and the auto-logistic model for binary data. *Journal of the Royal Statistical Society. Series B (Methodological)*, 34:75–83, 1972.

F. Bromberg, D. Margaritis, and V. Honavar. Efficient Markov network structure discovery using independence tests. *Journal of Artificial Intelligence Research*, 35:449–485, 2009.

W. Buntine. Theory refinement on Bayesian networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, pages 52–60. Morgan Kaufmann, 1991.

N. Casali, V. Nikolayevskyy, Y. Balabanova, S. R. Harris, O. Ignatyeva, I. Kontsevaya, J. Corander, J. Bryant, J. Parkhill, S. Nejentsev, R. D. Horstmann, T. Brown, and F. Drobniewski. Evolution and transmission of drug resistant tuberculosis in a population: Insights from a 1000 genome study. *Nature Genetics*, In press.

C. Chow and C. Liu. Approximating discrete probability distributions with dependence trees. *IEEE Transactions on Information Theory*, 14(3):462–467, 1968.

J. Corander, M. Ekdahl, and T. Koski. Parallel interacting MCMC for learning of topologies of graphical models. *Data Mining and Knowledge Discovery*, 17:431–456, 2008.

I. Csiszár and Z. Talata. Consistent estimation of the basic neighborhood of Markov random fields. *Annals of Statistics*, 34:123–145, 2006.

M. Ekeberg, C. Lökvist, Y. Lan, M. Weigt, and E. Aurell. Improved contact prediction in proteins: Using pseudolikelihoods to infer Potts models. *Physical Review E*, 87:012707, 2013.

D. Heckerman, D. Geiger, and D. M. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 20:197–243, 1995.

D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, 2001.

H. Höfling and R. Tibshirani. Estimation of sparse binary pairwise Markov networks using pseudo-likelihoods. *Journal of Machine Learning Research*, 10:883–906, 2009.

C. Ji and L. Seymour. A consistent model selection procedure for Markov random fields based on penalized pseudolikelihood. *Annals of Applied Probability*, 6:423–443, 1996.
D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT Press, 2009.

S. L. Lauritzen. *Graphical Models*. Oxford University Press, Oxford, 1996.

S.-I. Lee, V. Ganapathi, and D. Koller. Efficient structure learning of Markov networks using $\ell_1$-regularization. In *Advances in Neural Information Processing Systems*, 2006.

D. Lowd and J. Davis. Learning Markov network structure with decision trees. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 334–343. IEEE Computer Society, 2010.

H. Peng. Mutual information computation package, 2007, Accessed 2013-10-14. URL [http://www.mathworks.com/matlabcentral/fileexchange/14888-mutual-information-computation](http://www.mathworks.com/matlabcentral/fileexchange/14888-mutual-information-computation).

S. Della Pietra, V. Della Pietra, and J. Lafferty. Inducing features of random fields. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:380–393, 1997.

P. Ravikumar, M. J. Wainwright, and J. D. Lafferty. High-dimensional Ising model selection using $\ell_1$-regularized logistic regression. *Annals of Statistics*, 38:1287–1319, 2010.

M. Schmidt. Matlab code by Mark Schmidt (2005-2013), 2013, Accessed 2013-10-14. URL [http://www.di.ens.fr/~mschmidt/Software/code.html](http://www.di.ens.fr/~mschmidt/Software/code.html).

M. Schmidt and K. Murphy. Convex structure learning in log-linear models: Beyond pairwise potentials. In *Proceedings of International Workshop on Artificial Intelligence and Statistics*, 2010.

G. Schwarz. Estimating the dimension of a model. *Annals of Statistics*, 6:461–464, 1978.

P. Spirtes, C. Glymour, and R. Scheines. *Causation, Prediction, and Search*. MIT Press, 2nd edition, 2000.

I. Tsamardinos, C. Aliferis, A. Statnikov, and E. Statnikov. Algorithms for large scale Markov blanket discovery. In *The 16th International FLAIRS Conference, St*, pages 376–380. AAAI Press, 2003.

I. Tsamardinos, L. E. Brown, and C. F. Aliferis. The max-min hill-climbing Bayesian network structure learning algorithm. *Machine Learning*, 65:31–78, 2006.