HELSINGIN YLIOPISTO — HELSINGFORS UNIVERSITET — UNIVERSITY OF HELSINKI

Tiedekunta/Osasto — Fakultet/Sektion — Faculty	Laitos — Institution — Department
Faculty of Science	Department of Mathematics and Statistics

Tekijä — Författare — Author
Mikko Heikkilä

Työn nimi — Arbetets titel — Title

Quasi-Pseudolikelihood in Markov network structure learning

Oppiaine — Läroämne — Subject
Statistics

Työn laji — Arbetets art — Level	Aika — Datum — Month and year	Sivumäärä — Sidoantal — Number of pages
Master's thesis	October 2016	39 p.

Tiivistelmä — Referat — Abstract

Probabilistic graphical models are a versatile tool for doing statistical inference with complex models. The main impediment for their use, especially with more elaborate models, is the heavy computational cost incurred. The development of approximations that enable the use of graphical models in various tasks while requiring less computational resources is therefore an important area of research. In this thesis, we test one such recently proposed family of approximations, called quasi-pseudolikelihood (QPL).

Graphical models come in two main variants: directed models and undirected models, of which the latter are also called Markov networks or Markov random fields. Here we focus solely on the undirected case with continuous valued variables. The specific inference task the QPL approximations target is model structure learning, i.e. learning the model dependence structure from data.

In the theoretical part of the thesis, we define the basic concepts that underpin the use of graphical models and derive the general QPL approximation. As a novel contribution, we show that one member of the QPL approximation family is not consistent in the general case: asymptotically, for this QPL version, there exists a case where the learned dependence structure does not converge to the true model structure.

In the empirical part of the thesis, we test two members of the QPL family on simulated datasets. We generate datasets from Ising models and Sherrington-Kirkpatrick models and try to learn them using QPL approximations. As a reference method, we use the well-established Graphical lasso (Glasso).

Based on our results, the tested QPL approximations work well with relatively sparse dependence structures, while the more densely connected models, especially with weaker interaction strengths, present challenges that call for further research.

Avainsanat — Nyckelord — Keywords
Graphical models, undirected models, structure learning, pseudo-likelihood, quasi-likelihood

Säilytyspaikka — Förvaringsställe — Where deposited
Kumpula Campus Library

Muita tietoja — Övriga uppgifter — Additional information

Quasi-Pseudolikelihood in Markov network structure learning

Mikko Heikkilä

October 11, 2016

Contents:

Chapter 1

Introduction

Probabilistic graphical models offer a versatile way of presenting and utilizing complex probabilistic models in various tasks. Graphical models come in two main variants: directed and undirected ones. Both versions have attracted increasing interest during the last decades, that have seen a number of important theoretical and algorithmic developments. As a result, graphical models are becoming easier to use and steadily more useful in a wide range of applications.

Nevertheless, even though graphical models can be powerful tools for inference, they are naturally not without problems. The main impediment for their use is usually the heavy computational burden involved, especially with larger systems. One way to alleviate these problems is to find good approximations, that require less computations or enable circumventing some parts of the problem altogether.

Generally, it is possible to classify statistical inference problems into two main classes: parameter inference and structure learning. In this thesis we concentrate on the structure learning problem in the special case of undirected graphical models. Our main interest lies in testing two versions of a new family of approximation methods, called quasi-pseudolikelihood (QPL), for scoring different model structures. Due to having an analytical expression, the evaluation of the QPL scorings is computationally efficient compared e.g. to a full Bayesian solution of the problem. In the theoretical part of this thesis, we present the basic theory needed for utilizing graphical models and then derive the QPL scoring. On the empirical side, we test two variants of the QPL scoring by applying them to structure learning problem in noisy Ising models and related noisy Sherrington-Kirkpatrick models. As a novel theoretical contribution, we show that one of the tested criterions is inconsistent in a specific case.

The most important general references used in this thesis are two articles, the first one presenting an approximation called marginal pseudo-likelihood [24] and the second one presenting the quasi-pseudolikelihood [9]. The marginal pseudo-likelihood article introduces many of the basic principles and ideas that the quasi-pseudolikelihood is then based on. As a general reference for graphical models, we use a book by Koller & Friedman [16], from which most of the basic definitions come from. For Ising and related models we refer to a book by MacKay [19].

The organization of the thesis is as follows. In chapter 2 we start by introducing undirected probabilistic graphical models as a tool for handling information in complex systems. In the subsections, we give formal definitions that underpin the use of undirected models for representation, inference, and structure learning.

In chapter 3, we present the general structure learning problem in graphical models. In the subsections we first present the standard Bayesian approach to structure learning and find that, in general, it cannot be used as such for structure learning in undirected graphical models because of the computational burden involved. We therefore move on to define an approximation called marginal pseudo-likelihood, that enables structure learning in a computationally more efficient manner for discrete random variables. This method is then extended in the last subsection to cover continuous random variables by defining the quasi-pseudolikelihood scoring.

In chapter 4, we move on to testing the quasi-pseudolikelihood on noisy Ising models, as well as on the related noisy Sherrington-Kirkpatrick models. We therefore first define the models and present some of their general properties. After this we introduce the Metropolis-Hastings algorithm used for generating the data. We continue on to present the general settings used in the testing as well as the test results.

Chapter 5 is reserved for short conclusions.

Chapter 2

Graphical models

Graphical models have enjoyed increasing interest in the last few decades. Although the general idea of presenting interactions between variables as a graph has been traced back at least to J. Willard Gibbs in the beginning of the 20th century [11], graphical models started to garner more wide-spread recognition in Statistics from 1980 onwards, thanks to an article by Darroch, Lauritzen and Speed [7], where the authors combined the already well-established theory of Markov random fields and log-linear models, as well as some earlier work done on graphical models, to present graphical models as a separate class of models with several advantageous properties (see [16], [29]). Koller and Friedman credit the more widespread breakthrough of the graphical models towards the end of 1980s and the start of 1990s mainly to an influential textbook by Judea Pearl [23], and an article by Lauritzen and Spiegelhalter [17], which presented some key-ideas enabling more efficient inference using graphical models [16].

Arguably the most important aspect in considering graphical models is the equivalence of the dependence structure represented by a graph, and the factorization properties of a joint distribution. When this equivalence holds, we can firstly use a graphical model for *representation*, i.e. we can present complex dependencies in a way that is, in general, a lot easier for humans to understand at a glance than a mathematical formula representation of the joint distribution.

Besides offering a convenient way to present dependencies in probabilistic systems for ease of understanding, this form of representation also enables us in many cases to define a joint distribution over the system with relative ease, utilizing the factorization properties of the joint distribution to avoid defining redundant parameters. With joint distributions having potentially dozens or hundreds of variables, this reduction in the number of parameters can be essential for the task to be manageable.

Secondly, in addition to the good representation properties, graphical models can also be used for carrying out statistical *inference* tasks, such as calculating posterior probabilities for events given information on some of the variables in the model. Since the inference algorithms developed for graphical models utilize the graph structure, and hence also the dependence structure, of the model in question, they tend to be faster than inference algorithms that work on the joint distribution as a whole.

Thirdly, graphical models framework can be used for *learning* model structures. This enables a more data-driven approach, where human input defines some bounds on the possible models, and the final model or a set of models is learned algorithmically from the data at hand.

2.1 Basic definitions and notations

In this thesis we take some basic probability theory, such as random variables or density functions, as granted without reviewing the actual definitions and results. All the necessary background information should be found in any basic textbook, such as [1].

Concerning notation, we usually write density functions and factors with a short-hand notation without explicitly noting the variables in question, e.g. for the joint density of some variables X we may write $p(X)$ for the probability density or probability mass function $p_X(X)$. This should not cause any misunderstandings, since the variables can always be identified from the context. With this clarificatory note, we can start defining concepts more central to the theme of graphical models.

Graphical models are based on mathematical graph theory, from where we have the following basic definition:

Definition 2.1.1 (Graph)**.** Let V be a set of nodes, and E be a set of edges, $E \subseteq V \times V$, connecting the nodes. A *graph* is a collection $G = (V, E)$. For an edge connecting variable V_j to another variable V_i, we write $(j, i) \in E$.

To use a graph to represent a joint distribution over a set of random variables $X = \{X_1, \ldots, X_d\}$, we first interpret the nodes $V = \{1, \ldots, d\}$ to correspond to the indices of the variables. Henceforth, using this convention we usually refer to variables and nodes interchangeably. Depending on the way we define the edges, we get the two main types of graphical models: directed ones, usually called Bayesian networks (BNs), and undirected ones, which are also called Markov random fields (MRFs) or Markov networks (MNs). Since in this thesis the focus is on learning MN dependence structures, we do not consider BNs any further. For a more proper introduction, see e.g. [16].

For a MN, the edges connecting the variables are undirected, i.e. if $(j, i) \in E \Rightarrow (i, j) \in E$ for all $j \neq i$. Variables connected by an edge are called *neighbours*, and the set of all neighbours is called a *Markov blanket*, or more formally:

Definition 2.1.2 (Markov blanket)**.** Let $G = (V, E)$ be a graph corresponding to a Markov network, and let $j \in V$. A Markov blanket for j is the set of nodes $mb(j) = \{i \in V | (j, i) \in E\}$.

Informally, in representing dependencies of a joint distribution with a graph, we would like to have some simple relationship between the dependencies and the edges. A preferable solution would be to have an edge in the graph between two variables whenever those variables are directly related, so that the edges would represent possible paths along which variables can influence one another. It turns out this intuitive idea is more or less achievable, depending on some assumptions concerning the distribution in

question. We return to this issue in defining the co-called Markov properties in chapter 2.3.

Besides Markov blankets, another useful concept when dealing with MN dependencies is a clique. A clique is simply a set of nodes, such that all nodes belonging to the same clique are connected by an edge in the graph. A maximal clique is a clique that cannot be enlarged any further by adding new nodes.

Definition 2.1.3 (Clique)**.** Let $G = (V, E)$ be a graph corresponding to a Markov network. A *clique* C is a set of nodes $\{i \in V | i \in C \Rightarrow (i, j) \in E \text{ for all } j \neq i, j \in C\}$. A clique C is called a *maximal clique*, if $\nexists i \in V$ such that $i \notin C$ and $(i, j) \in E$ for all $j \in C$. We denote the set of maximal cliques in a graph G by $\mathcal{C}(G)$.

We first assume that the variables $X_j, j = 1, \ldots, d$ are discrete, with an outcome space $\mathcal{X}_j$. We denote the cardinality of the outcome space as $|\mathcal{X}_j| = r_j$. This assumption is redefined later, when we move on to consider continuous variables defined on a closed interval [0,1]. For a subset of variables $U \subseteq V$, we write $X_U = \{X_j\}_{j \in U}$. The outcome space of the subset is naturally a Cartesian product over the individual outcome spaces, i.e. $\mathcal{X}_U = \prod_{j \in U} \mathcal{X}_j$, and the cardinality is the corresponding product over the individual cardinalities $|\mathcal{X}_U| = \prod_{j \in U} r_j$. We denote an assignment to variables with a lower case letter, so that x_U is some specific joint assignment to variables X_U. For n independent, identically distributed (iid) joint observations, we generally write $\mathbf{x} = (\mathbf{x}_1, \ldots, \mathbf{x}_n)$, where $\mathbf{x}_k = (x_{1,k}, \ldots, x_{d,k})$. Throughout the thesis, we assume that the datasets are fully observed, meaning that there are no missing observations.

2.2 Parameterizing Markov networks

One important question that needs addressing, before we continue with the issues relating to the MN dependence structure, is how MNs can be parameterized. This issue turns out to be a lot thornier that might be expected. We therefore present the basic problem here only shortly and refer to [16] for a longer exposition.

Probably the first thing that comes to mind is to utilize conditional probability distributions (CPDs) to parameterize MNs. However, since in MNs the influences between variables run both ways, there is no clear way to factorize the corresponding distribution into CPDs in such a way that the factorization respects the properties implied by the graph.

The standard answer to this conundrum is to use more general factors to represent the distribution. A factor is simply a mapping from the outcome space of the random variables to real numbers:

Definition 2.2.1 (Factor)**.** Let X be a set of random variables. A function $\phi : \mathcal{X} \to \mathbb{R}$ is called a *factor*. If the image of $\mathcal{X}$ under ϕ is restricted to non-negative or to positive reals, then the factor is called a non-negative or a positive factor, respectively.

If not stated otherwise, henceforth we assume the factors to be non-negative. To utilize factors to represent distributions, we also need to have a suitable product rule, such as the one given in the following definition:

Definition 2.2.2 (Factor product)**.** Let A, B, C be three disjoint sets of variables, and let $\phi_1(A, B)$, $\phi_2(B, C)$ be two factors. A factor product $\phi_1 \times \phi_2$ is defined to be a factor $\psi : \mathcal{X}_{A,B,C} \to \mathbb{R}$, $\psi(A, B, C) = \phi_1(A, B) \cdot \phi_2(B, C)$.

A noteworthy point in this definition is that the factor product matches the common part of the individual factors.

It is immediately clear that joint distributions and CPDs are both special cases of general factors. Using the general factor definition we can define a Gibbs distribution, which is an undirected parameterization for a distribution. As such it forms a suitable link to the MN representation.

Definition 2.2.3 (Gibbs distribution)**.** A distribution p_Φ is a *Gibbs distribution* parameterized by a set of factors $\Phi = \{\phi_1(D_1), \ldots, \phi_K(D_K)\}$, if it is defined as follows:

$$\begin{aligned} p_\Phi(X_1, \ldots, X_d) =& 1/Z \cdot \tilde{p}_\Phi(X_1, \ldots, X_d), \text{ where} \\ \tilde{p}_\Phi(X_1, \ldots, X_d) &= \phi_1(D_1) \times \phi_2(D_2) \times \cdots \times \phi_m(D_m) \\ &\text{is an unnormalized measure and} \\ Z &= \sum_{X_1, \ldots, X_d} \tilde{p}_\Phi(X_1, \ldots, X_d). \end{aligned}$$

The normalizing constant Z is often called a *partition function* following the terminology used in statistical physics, since it is a function of the model parameters.

The key idea in unifying the Gibbs distribution formulation and a MN representation is to make a connection between factors in a distribution and cliques in a graph: a factor in the Gibbs distribution corresponds to a clique in the graph. This factorization property eventually allows us to formulate the Markov properties that define the dependence structure implied by a MN.

Definition 2.2.4 (Factorization)**.** A distribution p_Φ, $\Phi = \{\phi_1(D_1), \ldots, \phi_K(D_K)\}$, is said to *factorize* over a Markov network, if each $D_k, k = 1, \ldots, K$ is a clique in the graph.

Factors corresponding to cliques that are used to parameterize a MN are often called *clique potentials*.

It is worth noting that this factorization property generally does not lead to a unique parameterization, quite the contrary: if we have a parameterization with clique potentials that do not correspond to maximal cliques, then we can always find a new parameterization in terms of maximal clique potentials simply by taking factor products over all the factors that are encompassed in the same maximal clique.

This non-uniqueness of parameterizations is not limited to the case of maximal clique potentials but is a more fundamental issue: there is no single optimal parameterization for all purposes. Instead, what constitutes a useful parameterization depends on the situation. Different parameterizations can be loosely characterized by their *granularity*,

in other words by the level of detail they purpose to express. A parameterization more fine-grained than the one used in this thesis (that operates on the level of Markov blankets) can be expressed, for example, using factor graphs or log-linear models.

The key issue in choosing between the different parameterizations is the purpose of the representation. However, since this issue is not essential to this thesis, we stop here and proceed to treat the more central theme of dependence structures.

2.3 Markov properties

In order to meaningfully represent a joint distribution of some variables X with a graph, we need to know if the graph can properly capture all the intricacies inherent in the joint distribution and, on the other hand, if the graph can have some properties not satisfied by the distribution. The most basic properties we are now interested in are the conditional independencies between sets of variables. In the following, we write conditional independence statements as

$$X_A \perp\!\!\!\perp X_B | X_C,$$

meaning that variables X_A are conditionally independent of X_B given X_C, or equivalently that $p(X_A|X_B, X_C) = p(X_A|X_C)$.

With a joint distribution, as is evident from the notation above, conditional independence means that the distribution function factorizes into separate products, i.e. if $X_A \perp\!\!\!\perp X_B|X_C$, then

$$p(X_A, X_B|X_C) = p(X_A|X_C)p(X_B|X_C).$$

With MNs, we want to express the conditional independence statements with a graph structure. In order to do this, we need to define the so-called Markov properties, that articulate the conditional independencies we can encode with MNs.

Before giving the definitions, we need some auxiliary concepts, namely paths and separations. Intuitively, a path is an unbroken succession of edges between some variables, that allows a variable to have an influence on the non-neighbouring variables along the path. Since observing a variable fixes its value, it also blocks any influence from flowing through the observed variable. This means that we have to distinguish the so-called active paths that connect variables from all possible paths. Variables are separated if there is no active path connecting them.

Definition 2.3.1 (Path)**.** Let $G = (V, E)$ be a graph corresponding to a MN. The nodes $V_p = (V_1, \ldots, V_j)$ form a *path* in G, if $(i, i+1) \in E$ for all $i, i+1 \in V_p$.

Definition 2.3.2 (Active path)**.** Let $G = (V, E)$ be a graph corresponding to a MN, $V_p = (V_1, \ldots, V_j)$ a path in G, and let $Z \subseteq V$ be a set of observed variables. The path V_p is called *active* given Z, if for all $i \in V_p$, $i \notin Z$.

Definition 2.3.3 (Separation)**.** Let V be the set of variables in a MN, $A, B, C \subset V$. Set C *separates* A and B, if there is no active path between nodes $i \in A, j \in B$ given C.

Using these concepts, we can properly characterise the dependencies implied by MNs. There are three different Markov properties that a given graph can satisfy: pairwise, local, and global. The pairwise Markov property states that any two variables that are not neighbours are conditionally independent of each other given all the other nodes in the graph. The local Markov property says that a variable is conditionally independent of any non-neighbouring node given its Markov blanket. Finally, the global Markov property affirms that given two distinct sets of variables, if we can find a separating set of variables between them, then any pairs of variables in the original sets are conditionally independent given the separating set.

The following three definitions formulate the different Markov properties for MNs:

Definition 2.3.4 (Pairwise Markov property)**.** $X_j \perp\!\!\!\perp X_i | \, X_{V\setminus\{j,i\}}$ for all $j, i \in V$, such that $(j, i) \notin E$.

Definition 2.3.5 (Local Markov property)**.** $X_j \perp\!\!\!\perp X_{V\setminus\{mb(j)\cup j\}} | \, X_{mb(j)}$, for all $j \in V$.

Definition 2.3.6 (Global Markov property)**.** $X_A \perp\!\!\!\perp X_B | \, X_C$, for all disjoint subsets (A, B, C) of V, such that C separates A from B.

The structure scoring functions derived in the following chapters are based on the local Markov property, i.e. they utilize the Markov blankets for learning the dependence structure of the model in question. Generally speaking, the different Markov properties are not equivalent. However, it can be shown that in the special case of positive joint distributions, all the Markov properties are in fact equivalent (see e.g. [16]). We assume from now on, that the joint distributions are positive, unless noted otherwise.

We can also ask, are the dependence structures of a given joint distribution and some MN equivalent, i.e. do they encode the same conditional independencies. It is evident that in general, we can have a joint distribution with conditional independencies that we cannot represent with a MN (e.g. Bayesian networks provide counter-examples galore, see e.g. [16] for specific examples) and vice versa. A graph is called a *perfect map* for the joint distribution if it satisfies all the conditional independence statements encoded by the distribution and satisfies no conditional independence statements besides these. Similarly, if a graph is a perfect map for a distribution, then the distribution is called *faithful* to the graph. For the rest of the thesis we assume that the graphs are perfect maps, unless otherwise noted.

Having defined the dependence structures associated with MNs and their relations with the corresponding properties of joint distributions, we continue next to treat the structure learning problem in MNs.

Chapter 3

Structure learning problem

Fully learning a statistical model usually involves two main components: structure learning and parameter estimation, where by structure learning we mean learning the dependencies between the variables in a given model. In this thesis we are only interested in learning the model structure. Naturally, in real world problems it is not uncommon to encounter the two jointly: when doing statistical inference, we generally cannot depend on having a nice problem with a dependence structure fully known in advance, and when doing structure learning, we would in many cases also be interested in learning the parameters of the model. However, in both of these cases the complete problem can be solved in pieces, by first learning the model structure and then inferring the parameters given the structure learned.

In general, structure learning methods can be roughly divided into constraint-based and score-based methods. Constraint-based methods aim to recover the model structure by utilizing tests of independencies between variables, whereas score-based methods use an objective function to evaluate the different structures. The latter approach then also requires solving an optimization problem to find the structure that maximizes (or minimizes) the objective function used. Since constraint-based methods typically are more sensitive to mistakes made with individual edges and tend to require more data to achieve correct results, the choice between different methods usually involves a trade-off between accuracy of the discovered structure and computational burden [16],[24].

3.1 Bayesian solution to structure learning

The heart of Bayesian inference is contained in the Bayes' formula, which can be written as

$$p(\phi|y) = \frac{p(y|\phi)p(\phi)}{\int_\phi p(y|\phi)p(\phi)d\phi} \propto p(y|\phi)p(\phi), \tag{3.1}$$

where ϕ is some random quantity we like to make inferences on, y refers to some quantity the inference is based on (typically observed data), $p(y|\phi)$ is the likelihood function, and $p(\phi)$ is the prior. In many cases, especially with model selection problems, we are

not interested in the normalizing constant $\int_\phi p(y|\phi)p(\phi)d\phi$ and will therefore use just the unnormalized form. A thorough formulation of Bayesian theory can be found, for example, in [3].

Ideally, a full Bayesian solution to structure learning involves defining prior and likelihood functions for different graph structures, and then using Bayes' formula (3.1) to arrive at the posterior distribution. For a graph G, this results in a marginal posterior probability

$$p(G|\mathbf{x}) \propto p(\mathbf{x}|G)p(G) = \int_{\theta_G} p(\mathbf{x}|\theta_G, G)p(\theta_G|G)p(G)d\theta_G \tag{3.2}$$

$$= \int_{\theta_G} \{\prod_{k=1}^{n} p(\mathbf{x}_k|\theta_G, G)\}p(\theta_G|G)p(G)d\theta_G, \tag{3.3}$$

where the key term for structure learning, that can be termed *evidence*, is

$$p(\mathbf{x}|G) = \int_{\theta_G} p(\mathbf{x}|\theta_G, G)p(\theta_G|G)d\theta_G. \tag{3.4}$$

The evidence effectively measures how likely the observed dataset is under the assumed model G. Alas, (3.3) cannot usually be solved analytically without making some strict assumptions on the graph G. This might not be a huge problem, given that there are advanced methods like different flavours of Markov Chain Monte Carlo (MCMC) for evaluating just such integrals, if not for the size of the problem. Since there are $2^{\binom{d}{2}}$ possible model structures, using current techniques with non-trivial sized problems, it may simply be infeasible to utilize the full marginal posterior distribution as such.

3.2 Marginal pseudo-likelihood

One viable alternative to the use of the true evidence of a model, defined in (3.4), is to use some approximation that leads to an analytically solvable expression. One such formulation, called marginal pseudo-likelihood was derived by Pensar & al. [24]. Assuming the model consists of discrete variables $X = \{X_1, \ldots, X_d\}$, the MPL score for a graph is defined as

$$\prod_{j=1}^{d}\prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl}+\alpha_{jl})} \prod_{i=1}^{r_j} \frac{\Gamma(n_{ijl}+\alpha_{ijl})}{\Gamma(\alpha_{ijl})}, \tag{3.5}$$

where d, q_j and r_j refer to the number of variables in the graph, the states of the MB for variable j, and the states of variable j, respectively. In addition, n_{ijl} is the number of times the configuration $\{X_j = x^{(i)}, X_{mb(j)} = \mathbf{x}_{mb(j)}^{(l)}\}$ appears in the data. Using a similar notation, $\alpha_{ijl} > 0 \ \forall \ i, j, l$ refers to the corresponding prior parameters. Finally, by definition we have $n_{jl} = \sum_i n_{ijl}$ and $\alpha_{jl} = \sum_i \alpha_{ijl}$. In this thesis, we use a prior given by

$$\alpha_{ijl} = \frac{N}{|\mathcal{X}| \cdot |\mathcal{X}_{mb(j)}|} = \frac{N}{r_j \cdot q_j}, \tag{3.6}$$

where the parameter N is the so-called equivalent sample size that adjusts the strength of the prior.

In the rest of this section we derive the MPL approximation and discuss the assumptions needed to arrive at the analytical solution given in (3.5).

Starting from the true evidence (3.4), as a first step we replace the likelihood function $p(\mathbf{x}|\theta_G, G) = p(\theta_G; \mathbf{x})$ with a pseudo-likelihood function. The pseudo-likelihood, originally introduced by Besag [4], uses a product of conditional likelihood functions to approximate the true likelihood. For a graph G the pseudo-likelihood can be written as

$$pl(\theta_G; \mathbf{x}) = \prod_{k=1}^{n} \prod_{j=1}^{d} p(x_{j,k}|\mathbf{x}_{V\setminus j,k}, \theta_G) = \prod_{k=1}^{n} \prod_{j=1}^{d} p(x_{j,k}|\mathbf{x}_{mb(j,k)}, \theta_G), \tag{3.7}$$

where the first equality is by definition and the second one follows from the local Markov property for MNs (definition 2.3.5). Here $mb(j,k)$ refers to the kth observation of the variables in $mb(j)$. Replacing the true likelihood with the pseudo-likelihood effectively breaks the two-way interactions encoded by the edges in MNs locally into one-way influences.

As an intermediary step in deriving the marginal pseudo-likelihood, we parameterize the conditional probabilities in the pseudo-likelihood (3.7) as

$$\theta_{ijl} := p(X_j = x_j^{(i)}|X_{mb(j)} = \mathbf{x}_{mb(j)}^{(l)});\ \theta_{ijl} > 0, \sum_{i=1}^{r_j} \theta_{ijl} = 1,$$

where $i = 1, \ldots, r_j$, $r_j = |\mathcal{X}_j|$, $l = 1, \ldots, q_j$, $q_j = |mb(\mathcal{X}_j)| = \prod_{i \in mb(j)} r_i$. The indices i, l reference the configurations of the variable and its Markov blanket, respectively. Denote the counts of different configurations in the data as

$$\begin{aligned} n_{ijl} &= \sum_{k=1}^{n} \mathbf{I}\{(X_{j,k}, X_{mb(j,k)}) = (x_{j,k}^{(i)}, \mathbf{x}_{mb(j,k)}^{(l)})\}, \text{ and} \\ n_{jl} &= \sum_{i=1}^{r_j} n_{ijl}, \end{aligned} \tag{3.8}$$

where $\mathbf{I}\{\star\}$ is an indicator function. Using the above notation, the pseudo-likelihood (3.7) can be written as

$$pl(\theta_G; \mathbf{x}) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}. \tag{3.9}$$

In order to reach an analytical solution, we still need to define a prior distribution that factorizes in a way that fits in with the pseudo-likelihood formulation (3.9).

Defining sets of parameters by

$$\theta_{jl} = \cup_{i=1}^{r_j}\{\theta_{ijl}\},\ \theta_j = \cup_{l=1}^{q_j}\{\theta_{jl}\},\ \theta_G = \cup_{j=1}^{d}\{\theta_j\}, \tag{3.10}$$

we need a prior that factorizes as

$$f(\theta_G) = \prod_{j=1}^{d} f(\theta_j) = \prod_{j=1}^{d}\prod_{l=1}^{q_j} f(\theta_{jl}), \tag{3.11}$$

which implies that $\theta_j \perp\!\!\!\perp \theta_{j'}$ if $j \neq j'$ (global parameter independence), and $\theta_{jl} \perp\!\!\!\perp \theta_{jl'}$ if $jl \neq jl'$ (local parameter independence). This factorization together with some distributional assumptions leads to an analytical solution, as shown later.

This kind of prior formulation was presented by Heckerman & al. [14] for Bayesian networks, and consequently adapted by Pensar & al. [24] for MPL scoring prior. However, assuming this prior produces a problem with MNs. With Bayesian networks, the assumptions of global and local parameter independencies can be reasonable ones, although not always [14]. In contrast, with a MN these assumptions directly violate the internal consistency of the MN, which has to be taken into account later on. Nevertheless, the factorization is a necessary property if we want to arrive at an analytical formulation.

A third important assumption concerns the prior distributions of the parameter sets, which are assumed to follow a Dirichlet-distribution, i.e. we assume that

$$\theta_{jl} \sim \text{Dirichlet}(\alpha_{1jl}, \ldots, \alpha_{r_j jl}),\ \text{for all } j, l, \tag{3.12}$$

where $\alpha_{1jl}, \ldots, \alpha_{r_j jl}$ are hyperparameters, which means that the functional form of the prior (3.11) is

$$f(\theta_G) = \prod_{j=1}^{d}\prod_{l=1}^{q_j} f(\theta_{jl}) = \prod_{j=1}^{d}\prod_{l=1}^{q_j} \frac{\Gamma(\sum_{i=1}^{r_j}\alpha_{ijl})}{\prod_{i=1}^{r_j}\Gamma(\alpha_{ijl})}\prod_{i=1}^{r_j}\theta_{ijl}^{\alpha_{ijl}-1}. \tag{3.13}$$

This assumption is necessary, since the Dirichlet family of distributions is the conjugate family for a likelihood that follows a multinomial distribution. Following the earlier notation, we write $\alpha_{jl} = \sum_{i=1}^{r_j}\alpha_{ijl}$.

Combining all the above assumptions, we can finally derive an analytical solution for the marginalization over the parameter values in the evidence (3.4), using standard Bayesian calculations (see e.g. [3]) presented here for convenience:

$$\hat{p}(\mathbf{x}|G) = \int_{\theta_G} pl(\theta_G; \mathbf{x}) \cdot f(\theta_G) d\theta_G \tag{3.14}$$

$$= \int_{\theta_G} \{\prod_{j=1}^{d} \prod_{l=1}^{q_j} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}\} \cdot \{\prod_{j=1}^{d} \prod_{l=1}^{q_j} f(\theta_{jl})\} d\theta_G \tag{3.15}$$

$$= \int_{\theta_G} \prod_{j=1}^{d} \prod_{l=1}^{q_j} \{\prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}}\} \cdot \frac{\Gamma(\sum_{i=1}^{r_j} \alpha_{ijl})}{\prod_{i=1}^{r_j} \Gamma(\alpha_{ijl})} \prod_{i=1}^{r_j} \theta_{ijl}^{\alpha_{ijl}-1} d\theta_G \tag{3.16}$$

$$= \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\sum_{i=1}^{r_j} \alpha_{ijl})}{\prod_{i=1}^{r_j} \Gamma(\alpha_{ijl})} \int_{\theta_{jl}} \prod_{i=1}^{r_j} \theta_{ijl}^{n_{ijl}+\alpha_{ijl}-1} d\theta_{jl} \tag{3.17}$$

$$= \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\prod_{i=1}^{r_j} \Gamma(\alpha_{ijl})} \frac{\prod_{i=1}^{r_j} \Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\sum_{i=1}^{r_j} n_{ijl} + \alpha_{ijl})} \tag{3.18}$$

$$= \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl} + \alpha_{jl})} \prod_{i=1}^{r_j} \frac{\Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}, \tag{3.19}$$

where the integral in (3.17) can be readily calculated analytically by noting that it corresponds to a kernel of another Dirichlet distribution. Here we use the hat notation to differentiate the MPL score from the true marginal posterior, i.e. we write $\hat{p}(\mathbf{x}|G)$ instead of the true evidence (3.4).

To actually evaluate any concrete models in light of some given data, we also have to give values to the hyperparameters α_{ijl}. Pensar & al. [24] utilize a tweeked version of a prior originally introduced by Buntine [5] for Bayesian networks. This is just the prior given before in (3.6). The main motivation for using this type of prior is that, following [24], we are aiming at a more data-driven approach, where we do not favour any particular graph structure over any other. In other words we want to give equal weights to all parameters $\{\theta_{1jl}, \ldots, \theta_{r_jjl}\}$.

One important question concerns the sensitivity of the results to this choice of parameter value. Silander & al. [27] have demonstrated that the maximum a posteriori (MAP) results in structure learning using a so-called BDeu scoring can be sensitive to the equivalent sample size parameter value setting. Since MPL scoring is quite similar to the BDeu, it is expected to have similar properties, including the sensitivity to the prior equivalent sample size settings.

3.3 Quasi-pseudolikelihood

Hitherto we have assumed that the nodes V correspond to discrete random variables. Indeed, for Markov networks with continuous data there currently seems to exist precious few options for efficient structure learning. The well-established methods typically assume the data to follow a multivariate Gaussian distribution, which obviously can be

a problematic assumption. The reason for this scarcity of methods is also evident: the likelihood function, which is the most natural choice for a scoring function, generally leads to intractable expressions.

Quasi-PseudoLikelihood (QPL) scoring is based on the same ideas as the MPL scoring introduced in the previous section 3.2, but this time the data is assumed to be continuous. The actual formulation of the QPL was done by Dikmen [9] based on the ideas presented in [20].

In the next two sections we first define the QPL scoring and discuss its derivation, and then review its computational complexity and the optimization needed in order to find top-scoring graph structures.

3.3.1 Definition and derivation of the QPL scoring

As with the MPL scoring in the previous chapter, we start by defining the QPL scoring and then derive the approximation and discuss the assumptions mostly towards the end of this subsection.

Assume our true data-generating model consists of unobserved binary random variables $X = \{X_1, \ldots, X_d\}$. In addition, for each X_j we have an unobserved noise variable $\epsilon_j \in [0, 1]$ s.t. $\epsilon_j \perp\!\!\!\perp X_A, \epsilon_B$, where $A \subseteq \{1, \ldots, d\}, B \subseteq \{1, \ldots, d\} \setminus j$.

Our observed variables $Y = \{Y_1, \ldots, Y_d\}$ are then given by

$$Y_j = |X_j - \epsilon_j|, j = 1, \ldots, d. \tag{3.20}$$

In terms of graph structures, this means that for each binary variable in the original model, both the noise node and the binary node are connected to the new observed variable with a directed edge. In other words, we convert the original undirected binary model into a new noisy model, which includes both undirected and directed edges.

The QPL scoring is now defined as

$$\tilde{p}(\mathbf{y}|G) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(s_{jl} + \alpha_{jl})} \prod_{i=0}^{1} \frac{\Gamma(s_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}, \tag{3.21}$$

where d and q_j refer to the number of variables in the graph, and the states of the MB for variable j, in that order. We also have a prior similar to (3.5) given by $\alpha_{ijl} > 0, \alpha_{jl} = \sum_i \alpha_{ijl}$. The s_{ijl} and s_{jl} terms that correspond to the counts in the discrete version (3.5) can be interpreted as fuzzy counts. They are defined in terms of sample-specific *responsibilities* as

$$\begin{aligned} s_{ijl} &= \sum_{k=1}^{n} s_{ijlk}, \\ s_{jl} &= \sum_{k=1}^{n} s_{jlk}. \end{aligned} \tag{3.22}$$

Following [9], we define two different versions for the responsibilities leading to two distinct QPL scorings, termed QPL1 and QPL2. For QPL1, for variable j with a MB of size N_j the responsibility for the kth sample and configuration $(i, l_1, \ldots, l_{N_j})$ is given by

$$s_{ijl_1 \ldots l_{N_j} k} = (1 - |i - y_{j,k}|) \prod_{m=1}^{N_j} (1 - |l_m - y_{mb(j,m,k)}|) \tag{3.23}$$

$$= (2iy_{j,k} - y_{j,k} - i + 1) \prod_{m=1}^{N_j} (2l_m y_{mb(j,m,k)} - y_{mb(j,m,k)} - l_m + 1), \tag{3.24}$$

where, deviating from previous notation, we have written the variables in the MB explicitly. The notation $mb(j, m, k)$ refers to the kth observation for the mth variable in the MB $mb(j)$. The responsibility for s_{jlk} can be calculated from the same expression (3.24) by simply omitting the first factor that contains i from the product.

For QPL2, the responsibility for the kth sample and configuration $(i^*, l_1^*, \ldots, l_{N_j}^*)$,

$$(i^*, l_1^*, \ldots, l_{N_j}^*) = \arg \min_{i, l_1, \ldots, l_{N_j}} \{|i - y_{j,k}| + \sum_{m=1}^{N_j} |l_m - y_{mb(j,m,k)}|\}, \tag{3.25}$$

is given by

$$s_{i^* j l_1^* \ldots l_{N_j}^* k} = \frac{N_j + 1 - |i^* - y_{j,k}| - \sum_{m=1}^{N_j} |l_m^* - y_{mb(j,m,k)}|}{N_j + 1}. \tag{3.26}$$

For all other configurations, the QPL2 responsibility is defined to be zero. For the responsibility s_{jlk}, we make simple modifications to (3.25) and (3.26) to get the corresponding expressions

$$(l_1^*, \ldots, l_{N_j}^*) = \arg \min_{l_1, \ldots, l_{N_j}} \{\sum_{m=1}^{N_j} |l_m - y_{mb(j,m,k)}|\}, \tag{3.27}$$

$$s_{j l_1^* \ldots l_{N_j}^* k} = \frac{N_j - \sum_{m=1}^{N_j} |l_m^* - y_{mb(j,m,k)}|}{N_j}, \tag{3.28}$$

and the responsibilities for all the other configurations are zeros.

In the rest of this section we derive the QPL approximation and discuss the various assumptions and some implications of the definitions. Most of the discussion presented in section 3.2 in deriving the MPL formulation is also relevant here, since MPL and QPL are very closely related, but we do not repeat it here.

We start the derivation by assuming that $X_j \in \{0, 1\}, j = 1, \ldots, d$. Since this is a stricter assumption than the general discreteness assumed in the previous chapters, the

MPL score formulation given in (3.5) remains valid. The basic idea is then to extend MPL by utilizing an approximation to quasi-likelihood functions.

Quasi-likelihood functions were originally introduced by Wedderburn [28] to allow for parameter estimation with generalized linear models in situations, where the true distribution of the response variable is unknown.

More verbosely, the basic idea with the quasi-likelihood is to relax the assumptions we need to make in order to model a dataset. The usual way to model some datapoints is to specify a likelihood function, which is then used to make inferences. However, this requires us to define the complete distribution of the data. Instead, with quasi-likelihood we only specify the relation between the mean and the variance. With some assumptions, we can make inferences about various interesting quantities such as regression coefficients using the quasi-likelihood function. The reason this works is that the quasi-likelihood is defined so as to share some basic properties with the common likelihood function (see e.g. [21]).

Denote the response variable by Y, $\mathrm{E}(Y) = \mu$, a dispersion parameter by σ^2, and a variance function by $V(\star)$. Assuming the variance to have the form

$$\mathrm{Var}(Y) = \sigma^2 V(\mu),$$

the quasi-likelihood function is now defined as

$$Q(\mu|y) = \int_{\mu} \frac{y-t}{\sigma^2 V(t)} dt + g(y), \tag{3.29}$$

where $g(y)$ is some function that does not depend on the model parameters. As shown in [28], the quasi-likelihood function has properties analogous to a standard likelihood function with respect to μ, and in the special case of one-parameter exponential family models, the quasi-likelihood is actually identical to the standard likelihood.

Marttinen & al. [20] used a quasi-likelihood approach in a fuzzy clustering problem, deriving a scoring function with a closed-form solution for the marginals in question. Using the same idea, Dikmen [9] defined a quasi-likelihood-based scoring method for MN structure learning. We present here the basic idea behind the QPL approximation and refer to [20] for a more thorough, information theoretic justification for the original approximation.

For a single binary observation of the variable X_j, the marginal density is given by the Bernoulli distribution. Parameterizing the distribution by μ we have the standard log-likelihood given by

$$\log p(X_j = x_j) = x_j \log \mu + (1 - x_j) \log(1 - \mu), x_j \in \{0, 1\}. \tag{3.30}$$

We would like to have a similar equation for a continuous-valued variable $Y_j \in [0, 1]$. Assuming that the variance function $V(\star)$ in (3.29) has a similar form as the corresponding Bernoulli case, i.e. assuming that

$$\sigma^2 V(\mu) = \mu(1 - \mu), \mu \in (0, 1) \tag{3.31}$$

and calculating the integral in (3.29) without the constant terms we get

$$Q(\mu|y_j) = \int_\mu \frac{y_j - t}{t(1-t)} dt \tag{3.32}$$

$$= -y_j \log(\frac{1-\mu}{\mu}) + \log(1-\mu) \tag{3.33}$$

$$= y_j \log(\mu) + (1-y_j)\log(1-\mu), \tag{3.34}$$

which clearly is not a proper likelihood function, but is nevertheless very similar to the Bernoulli log-likelihood (3.30). This is the basic solution used in [20], which is then developed further by including the σ^2 term that is interpreted as measuring the amount of information in the observation y_j.

In our case however, we need to consider the conditional densities instead of the marginal ones. Starting therefore again with a single binary variable X_j, we can write the full conditional distribution as

$$p(X_j|X_{mb(j)}, \theta_G, G) = \frac{p(X_j, X_{mb(j)}|\theta_G, G)}{p(X_{mb(j)}|\theta_G, G)} = \frac{\prod_i \prod_l \theta_{ijl}^{n_{ijl}}}{\sum_{X_j} \prod_i \prod_l \theta_{ijl}^{n_{ijl}}}, \tag{3.35}$$

where n_{ijl} are the counts over indicator functions as defined in (3.8). Replacing the binary variables with continuous ones, we could write the full conditional in a similar fashion as

$$p(Y_j|Y_{mb(j)}, \theta_G, G) = \frac{\prod_i \prod_l \theta_{ijl}^{s_{ijl}}}{\int \prod_i \prod_l \theta_{ijl}^{s_{ijl}} dY_j}, \tag{3.36}$$

where s_{ijl} are the fuzzy counts, i.e. sums over some responsibilities. In other words, the individual responsibilities measure how similar the continuous variables are to the binary ones. This idea is somewhat similar to the information content tuned by σ^2 in [20]. The integral in (3.36) complicates things, since it cannot be evaluated analytically. To overcome this problem, we have to make an approximation, i.e. we simply drop the normalizing constant and use the resulting terms

$$\tilde{p}(Y_j|Y_{mb(j)}, \theta_G, G) = \prod_{l=1}^{q_j} \prod_{i=0}^{1} \theta_{ijl}^{s_{ijl}} \tag{3.37}$$

in the pseudo-likelihood defined in (3.7). This mean that QPL does not use a proper quasi-likelihood function but an approximation motivated by the quasi-likelihood approach. To evaluate how much information this approximation loses, we need to test the method on various models.

With the fuzzy counts defined by (3.22), we can repeat the arguments and assumptions given in the previous section to arrive at the analytical solution given in (3.21), that is very similar to the MPL scoring given in (3.5).

As has already been made evident, different definitions for the responsibilities lead to different QPL scorings, i.e. we should regard QPL as a family of related scoring methods.

A potential problem with the QPL1 responsibility defined in (3.24) is that it might result in small fuzzy counts for relevant configurations especially in higher dimensions, since nothing constrains the normalizing constant dropped in (3.37) from being far from 1. In contrast, the QPL2 score first identifies the orthant the vector $(y_{j,k}, y_{mb(j,1,k)}, \ldots, y_{mb(j,N_j,k)})$ belongs to by evaluating Manhattan distances as in (3.25). We then give positive responsibility only to this closest configuration and set all other configurations to zero. The idea is that in this case, the normalizing constant that we drop in (3.37) is closer to or equal to one, so that omitting it has less effect on the final score.

One drawback in the QPL2 scoring definition is that by giving non-zero weight only to the single responsibility, we lose the property $\sum_i s_{ijl} = s_{jl} \forall j, l$ that can be used to evaluate the asymptotic properties of the fuzzy counts, as is done in appendix A.

We noted earlier in section 3.2 that the prior value is expected to have a notable influence on the MPL scoring. Since we use the same prior in the QPL formulation, it is also expected to have an influence on the QPL results.

One important theoretical property for estimators in general is consistency, which in this case guarantees that given enough observations, we are bound to identify the correct graph structure. Unfortunately, one novel contribution of this thesis is to show that there exists a case where QPL1 method is inconsistent, i.e. it is not consistent in a general case. In many cases, consistency could still be hoped for by restricting the possible graph structures to be e.g. chordal (meaning that there are no chordless cycles of length 4 or greater in the graph) or bipartite (meaning that the nodes in the graph can be clustered into two sets s.t. there are no edges connecting nodes in the same cluster). In the case of QPL1 however, the example constructed uses three vertices in a string, which means that the example also shows that QPL1 is not generally consistent even if the possible graph structures are quite strictly limited, e.g. to chordal or bipartite graphs. The proof can be found from appendix A. For QPL2 the consistency is still an open question.

3.3.2 Computational complexity and optimization

The MPL and QPL scorings derived in 3.2 and 3.3.1 define objective functions that can be used to evaluate different graph structures. The computational complexities of the MPL (covered in [24]) and QPL scorings are similar. With QPL scoring (3.21), using logarithms to avoid computational problems, to evaluate one graph structure we need to calculate the sum

$$\sum_{j=1}^{d} \sum_{l=1}^{q_j} \{\log \Gamma(\alpha_{jl}) - \log \Gamma(s_{jl} + \alpha_{jl}) + \sum_{i=0}^{1} [\log \Gamma(s_{ijl} + \alpha_{ijl}) - \log \Gamma(\alpha_{ijl})]\}, \quad (3.38)$$

which contains $\sum_{j=1}^{d}(q_j(2+2\cdot 2)) = \sum_{j=1}^{d} 6q_j$ terms. The number of variables d is obviously constant for a given graph, but the number of different Markov blanket configurations q_j goes up exponentially with the size of the blanket.

Unlike with the MPL scoring, which can be implemented in a non-naive way to avoid at least part of the computations by only evaluating the blankets for which the configuration is also represented in the data, depending on the responsibility used the QPL scoring algorithm may actually need to evaluate all the possible Markov blanket configurations, since the fuzzy counts generally might not be zero for any configuration.

As with the MPL scoring, the variable-wise decomposition of QPL makes it suitable for quick evaluation of local changes in a graph. Given two graphs $G_1 = (V, E_1)$ and $G_2 = (V, E_2)$, their log-QPL-ratio can be calculated by

$$\log K(G_1, G_2) = \log \tilde{p}(\mathbf{y}|G_1) - \log \tilde{p}(\mathbf{y}|G_2), \tag{3.39}$$

which could be interpreted as a kind of QP-Bayes factor. Assuming that the two graphs differ from each other by a single edge between, say, nodes i and i', because of the decomposition properties of the QPL, equation (3.39) can be further modified by getting rid of some unnecessary terms. Since the single differing edge only affects the two Markov blankets $mb(i)$ and $mb(i')$, removing the terms that cancel each other leads to

$$\log K(G_1, G_2) = \log \tilde{p}_i(\mathbf{y}|G_1) + \log \tilde{p}_{i'}(\mathbf{y}|G_1) - \log \tilde{p}_i(\mathbf{y}|G_2) - \log \tilde{p}_{i'}(\mathbf{y}|G_2), \tag{3.40}$$

where e.g. $\tilde{p_{i'}}$ refers to the log-QPL score (3.38) with the index j set to i'.

Despite the fact that the QPL scoring is given by a relatively simple formula, this still leaves us with a non-trivial optimization problem. The reason for this is the possible number of different graph structures, which goes up exponentially with the number of nodes in the graph. Even for moderate sized problems there are still too many possible structures for the problem to be manageable by brute computational force.

Therefore, instead of simply enumerating all the possible scorings and choosing the winner, we utilize the optimization algorithm described in [24]. We give here a shorter formulation elucidating the basic problem and the ideas ideas used for tackling it, and refer to the aforementioned article and the references therein for a more specific description.

The basic optimization problem can be formulated as

$$\arg\max_{G\in\mathcal{G}} \log \tilde{p}(\mathbf{y}|G) + \log p(G), \tag{3.41}$$

where $\mathcal{G}$ is the set of all possible graph structures and $p(G)$ is the prior distribution (see equation 3.3). Assuming the prior is uniform over all the possible structures, we drop it in the following expressions. The same ideas would also work with a suitable non-uniform prior with some corrections to the formulas. Since the model dependence structure is uniquely specified by the collection of Markov blankets $mb(G) = \{mb(j)\}_{j=1}^{d}$

(see Section 2.3 on Markov properties), we can write equation (3.41) equivalently using Markov blankets as

$$\arg\max_{mb(G)\in\times_{j\in V}\mathcal{P}(V\setminus j)} \sum_{j=1}^{d} \log \tilde{p}(\mathbf{y}_j|\mathbf{y}_{mb(j)}) \tag{3.42}$$

with the constraint that $i \in mb(j) \Rightarrow j \in mb(i)$ for all $i, j \in V$, and where $\mathcal{P}(V \setminus j)$ denotes the power set of $V \setminus j$, i.e. it includes all the possible Markov blankets of node j.

As is readily visible from equation (3.42), the Markov blanket discovery problem consists of d interconnected sub-problems. The interconnectedness of the sub-problems results from the MN consistency requirements: without the constraint we could end up with a graph structure that includes one-sided influences between its nodes. However, from a computational perspective it can make sense to relax the constraint temporarily, since the resulting independent sub-problems can then be solved in parallel. This results in independent problems formulated as

$$\arg\max_{mb(j)\subseteq V\setminus j} \log \tilde{p}(\mathbf{y}_j|\mathbf{y}_{mb(j)}), j = 1, \ldots, d. \tag{3.43}$$

To solve the relaxed problem and find the local QPL maximum independently for each node, we use an approximate deterministic hill-climbing algorithm described in detail in [24].

After finding a solution to the relaxed problem given by equation (3.43), the MN consistency requirements could then be re-enforced simply by parsing the final Markov blankets from the relaxed problem solution, e.g. by including an edge between two nodes if it is included in the solution for either one of the individual nodes' Markov blankets, or alternatively including an edge if it is present in both of the individual solutions. Terming these strategies $\vee$-criterion (OR) and $\wedge$-criterion (AND), respectively, we can formalize the final edge sets for a graph $G = (V, E)$ as

$$E_\vee = \{(i, j) \in \{V \times V\} : i \in mb(j) \text{ or } j \in mb(i)\}, \tag{3.44}$$

$$E_\wedge = \{(i, j) \in \{V \times V\} : i \in mb(j) \text{ and } j \in mb(i)\}. \tag{3.45}$$

An alternative to these criterions, proposed by Pensar & al. [24], is to regard the constraint re-enforcement as a second optimization problem with the same QPL scoring function as before, but defined on a reduced edge set space given by the $\vee$-criterion:

$$\arg\max_{G\in\mathcal{G}_\vee} \log \tilde{p}(\mathbf{y}|G), \tag{3.46}$$

where $G_\vee = \{G \in \mathcal{G} : E \subseteq E_\vee\}$. The reduction in the edge space compared to the full problem might be considerable. Under some assumptions, this reduced problem can also

be solved exactly (see [24]). In general however, exact solutions may still be too much to hope for, since the computational cost grows exponentially with the Markov blanket size. To solve this second optimization problem, we again utilize an algorithm described in [24]. This is the solution used in all the tests in Chapter 4.3.

Chapter 4

Testing QPL on noisy models

We are now ready to test the QPL scorings on some actual problems to gauge their behaviour. We utilize two classes of models, one called Ising models, and the other called Sherrington-Kirkpatrick (S-K) models. Since the S-K model is a modification of the basic Ising model, we review the Ising model in some length and later just state shortly how the S-K model differs from the Ising model. Since we generate the data with known model structures, we can easily compare the results with the true structure to measure the performance of the QPL scores.

4.1 Ising model

The Ising model was originally proposed in statistical physics in the beginning of the 20th century by Lenz [18] as a model for ferromagnetism. The random variables in the Ising model, usually called spins, are discrete with an outcome space {-1,1}. In a one-dimensional Ising model, the variables form a chain so that each variable is connected to both of its neighbours, and to no other variable besides the neighbours. Higher dimensional Ising models are formulated similarly, so that each variable is connected to a specific number of neighbours. To test the QPL, we use a 2D Ising model, which means that the model is a 2D square lattice (see Figure 4.1).

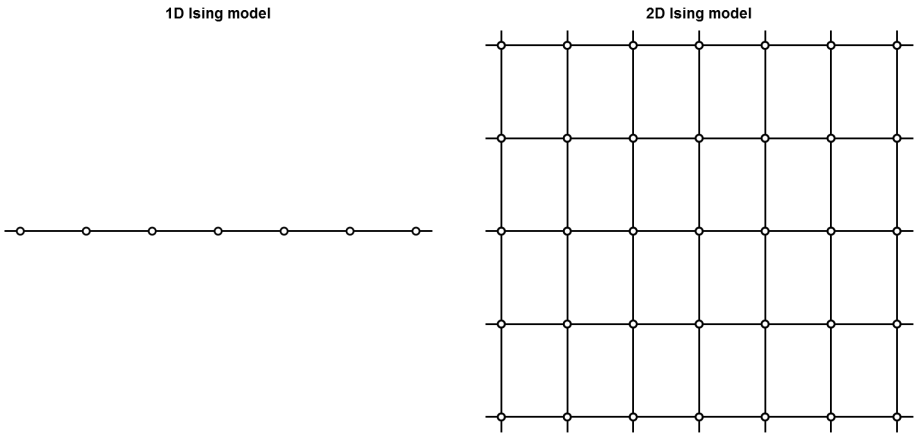

Figure 4.1 – Ising models.

We use periodic boundary condition so that all variables in the model have the same number of neighbours, i.e. the variables on the edges of the model are connected to the corresponding variables on the other edges.

The general Ising model has a probability distribution

$$P(\mathbf{x}) \propto P^*(\mathbf{x}) = \exp(-\beta \cdot \mathbb{E}(\mathbf{x})) = \exp(-\frac{\mathbb{E}(\mathbf{x})}{Tk_B}), x_j \in \{-1, 1\} \; \forall j, \tag{4.1}$$

where T is temperature, k_B is the Boltzmann constant, and $\mathbb{E}(\star)$ is an energy function. The energy function is defined as

$$\mathbb{E}(\mathbf{x}) = -\{\sum_{E} J_{j,i} x_j x_i + \sum_{j=1}^{d} H_j x_j\}, \tag{4.2}$$

where the summation over E refers to summing over the neighbours in the graph. In the energy function, $J_{j,i}$ is an interaction term that regulates the strength of the coupling between neighbours. In the following, we let $J_{j,i} = J$, if $(j, i) \in E$. If $J > 0$, the model is ferromagnetic (neighbouring spins tend to be in the same state), and if $J < 0$ it is antiferromagnetic (neighbours tend to be in opposite states). Obviously, with $J = 0$ there are no interactions between the variables. H_j represents an external magnetic field applied to the system, and similarly to the interactions, we let $H_j = H$ for all $j = 1, \ldots, d$. In the actual tests, we set $H = 0$.

In many cases, the most interesting question regarding the Ising models, and the original reason for their study, concerns phase transitions, i.e. transitions from ordered to unordered phase or vice versa. Analytically, a phase transition corresponds to a singularity in the second derivative of the partition function at some critical temperature. An infinite 2D Ising model is one of the simplest models known to exhibit a phase transition.

However, the phase transition does not fundamentally affect the problem of structure learning, and for this reason we do not consider it in any more detail. The same goes for other general properties of Ising models, such as magnetization behaviour. A bit more thorough introduction to Ising and related models can be found e.g. in [19].

The S-K model, introduced by Sherrington and Kirkpatrick in 1975 [26], is a modification of the basic Ising model, where the interactions between the nodes follow a Gaussian distribution. This means, among other things, that the number of neighbours for the nodes in an infinite model is infinite. For our purposes, the main difference compared to the standard Ising model is that the model has a random topology as well as random interaction strengths.

4.2 Metropolis-Hastings algorithm for Ising models

To test the QPL scorings, we would prefer to have iid observations from the Ising model with the selected parameter configurations. However, it is not straightforward to sample directly from a distribution such as given by equation (4.1). Therefore, to generate the observations, we utilize a Markov chain Monte Carlo (MCMC) technique called Metropolis-Hastings (M-H) sampling (see e.g. [12], for basic Markov chain theory, see e.g. [15]), first introduced in an article by Metropolis, Rosenbluth, Rosenbluth, Teller and Teller in 1953 [22], and subsequently generalized by Hastings in 1970 [13].

The general idea of the M-H sampler, that it shares with other MCMC methods based on stationary Markov chains (MC), is to construct a MC that has the target distribution (i.e. the distribution we would like to sample from) as its stationary distribution. After the chain has been run long enough to converge, we can then use values from this MC as (dependent) samples from the target distribution. A common approach for drawing independent samples with MCMC methods is to *thin* the samples, which means that after the chain has converged we choose every kth sample from the chain. With large enough k these samples are then treated as (nearly) independent samples form the target distribution.

In a little more detailed level, to run the M-H sampler the chain is started from some initial state with the condition that the probability of the initial state $x^{(0)}$ under the target distribution f is positive. At each step $t = 1, 2, \ldots$ we generate a proposal x^* for the next state of the chain $x^{(t)}$ from a suitable *proposal distribution* $g(\star|x^{(t-1)})$. We calculate the *M-H ratio*

$$R(x^{(t-1)}, x^*) = \frac{f(x^*)g(x^{(t-1)}|x^*)}{f(x^{(t-1)})g(x^*|x^{(t-1)})} \tag{4.3}$$

and set the next state depending on the value of the ratio as

$$x^{(t)} = x^* \text{ with probability } \min(R(x^{(t-1)}, x^*), 1) \text{ and } x^{(t)} = x^{(t-1)} \text{ otherwise.} \tag{4.4}$$

For a more detailed view of the M-H sampler, the requirements for the distributions used along with the necessary proofs that the chain does converge to the target distribution, we refer to [12].

Usually the real issue with the M-H sampler is the requirement that the chain has to be run until it converges to the stationary distribution, which is only guaranteed to happen asymptotically. In practice we have to stop the chain at some point and try to estimate if the chain is close enough to the target distribution. In general, there is no guaranteed test that could prove that the chain has converged. In the case of Ising models, the convergence of the M-H sampler has been treated e.g. by MacKay [19].

Our concrete algorithm works by picking a random spin on each iteration, and then checking how flipping the state of that spin would affect the energy of the system. If the flip decreases the energy, it is accepted automatically (since the M-H ratio in (4.4) ≥ 1). Otherwise the move is accepted with a probability that depends on the change itself. In order to keep the actual code easily readable, we introduce some clarifying notation, in addition to the standard Ising model equations (4.1 & 4.2). In the following, we write

$$b_j^{(t-1)} = \sum_{i:(j,i)\in E} J x_i^{(t-1)} + H, \tag{4.5}$$

$$\Delta\mathbb{E} = 2x_j^{(t-1)} b_j^{(t-1)}, \tag{4.6}$$

where $\Delta\mathbb{E}$ is the change in energy resulting from flipping the spin j. Using this notation, we can write the basic algorithm for the M-H sampler in pseudocode (see e.g. [6]) as follows

Algorithm 1: Metropolis-Hastings algorithm for simulating Ising models

```
initialize the system to a random state
for t = 1 to maxIter do
    choose a random spin j
    calculate ΔE
    if ΔE ≤ 0 then
        assign x_j^(t) = −x_j^(t−1)
    else
        generate a Unif(0,1)-distributed random number u
        if u < exp(−ΔE/Tk_B) then
            assign x_j^(t) = −x_j^(t−1)
        else
            assign x_j^(t) = x_j^(t−1)
        end if
    end if
end for
```

The algorithm is quite self-explanatory, but we still venture to make a few comments. The initial step is here denoted by time $t = 0$. The assignments are written by equality

signs, so that the left-hand side is the variable that the value is assigned to, and the right-hand side is the value that is assigned. The integer maxIter, that is used to control the number of iterations, should normally be quite large, since as noted before the algorithm is guaranteed to converge to the target distribution only asymptotically and the convergence can be slow. The same algorithm can be used for generating observations from the S-K model with minimal changes ($\Delta\mathbb{E}$ needs to be calculated using the randomly drawn MBs and interaction strengths).

4.3 Test setups and results

The actual tests have been mainly run on the Ukko cluster, maintained by the Department of Computer Science at the University of Helsinki, and the Triton cluster, maintained by the Aalto University Science-IT project.

The performance in the tests is measured in Hamming distances, averaged over five independent datasets. All the numerical results and plots for the tests can be found from appendix B. We use Graphical lasso (Glasso) [10] as a reference method, since it is a well-established and widely used method.

We use five different test settings in all. The first test uses a standard 2D Ising model with varying temperatures and noise levels. In these tests the QPL1 performance is better than Glassos and QPL2 clearly outperforms both other methods with almost all settings. The second test uses the Sherrington-Kirkpatrick model, again with varying temperatures and noise levels. This time Glasso performs clearly better than either of the QPL scorings, while QPL2 again outperforms QPL1. To clarify this weak performance, the third test uses an Ising model with randomized interaction strengths with a single temperature and varying noise levels. With this setup, both QPLs tend to perform better than Glasso on lower noise levels and worse on the highest noise level. Again, QPL2 outperforms QPL1. The fourth test uses the same datasets as the third test with a single temperature, but the learning is done with different prior strengths. The results show that the prior parameter has a significant effect on the results. Finally, the fifth test uses the same datasets as the second test on a single temperature, but the prior strength is varied. Again, the prior has a clear effect on the results, but the Glasso still mostly outperforms both QPLs.

In the rest of this chapter we introduce each test setup in a more detailed manner and discuss the results.

4.3.1 First test: Ising model

For the testing, we first use 2D lattice Ising models with 16*16 = 256 variables, constant interaction strengths $J = 1$ for the neighbours, and no external field, i.e. $H = 0$. The temperature is increased from 2 to 4 with a stepsize of 0.5. We would expect that the structure learning gets harder in both directions from some optimum temperature value; with low temperatures, the system is mostly frozen, resulting in a low variance, whereas with high temperatures the changes in the system tend to get more and more random.

To generate data from the correct model, we use the simple implementation of Metropolis-Hastings sampling given before (Algorithm 1). Before the first sample is drawn, the model is run for 10^7 iterations to establish the correct temperature, and after this we choose every 30000th iteration as an independent sample from the model. We repeat this process to produce 5 independent sets of observations for each temperature.

After the Ising model observations have been generated, we add noise to the datasets. This is achieved by selecting a noise level θ, and for each binary observation $x_{j,k} \in \{0, 1\}$ (simply re-labeled from the original $\{-1, 1\}$ observations) adding independent, uniformly distributed noise. This results in noisy observations $y_{j,k} = |x_{j,k} - u_{j,k}|$, where $U_{j,k} \sim \text{Uniform}(0, \theta)$ for all $k = 1, \ldots, n$ and $j = 1, \ldots, d$. We use noise levels 0.15, 0.35 and 0.55 with a maximum of 10000 observations.

QPLs are run with the prior defined in (3.6) with the equivalent sample size parameter $N = 1$. For Glasso, a tuning parameter needs to be set in some informed manner to achieve any reasonable performance. We use a scheme based on separate training set and test set, i.e. we choose a parameter value that gives a good performance on a training set separate from the actual test set used for measuring the performance. As a training set, we use a sixth independent dataset generated in the same way as the five test datasets.

A good parameter value for the Glasso is searched by starting from a small value (0.01) and then increasing the value with a stepsize of 0.01 until the sum of errors starts to increase. This value is not optimal, since it is the same for all sample sizes, but it generally gives a good performance on the training sets. The exception is the first test with temperature $T = 2$. As noted before, the low temperature makes the structure learning very difficult. In this case we could not find a single parameter value resulting in a reasonable performance. The results reported in the appendix correspond to the same parameter setting scheme utilized with other tests, but the best performance in terms of mean Hamming distances found by manual testing was achieved by setting the parameter to such a high value, that the result was simply a fully independent model with all sample sizes and noise levels.

The general effects of varying temperature are highlighted in figure 4.2 using QPL1 with noise level 0.15. The results for all methods and noise levels follow roughly the same pattern: the best results are achieved with T=2.5 or T=3, while for both lower and higher temperatures the performance degrades. Increasing the sample size enhances the results.

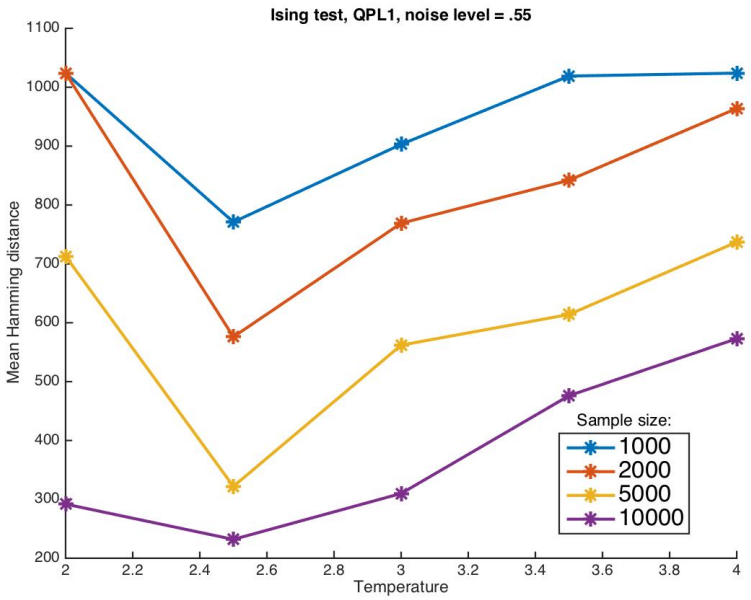

Figure 4.2 – QPL1: Effects of changing temperature with different sample sizes. The effects are roughly similar for all methods and configurations tested.

To assess the performance of each method compared to the other two, we calculate a ranking for each configuration of the test settings based on their average hamming distances. This leads to 60 rankings (5 temperatures * 3 noise levels * 4 sample sizes). In case of a tie, the tied methods are all awarded the best rank out of the tied ones (e.g. in a case where two methods have the same average hamming distance behind the best method, both of the tied methods get a rank of 2). The results are plotted in figure 4.3.

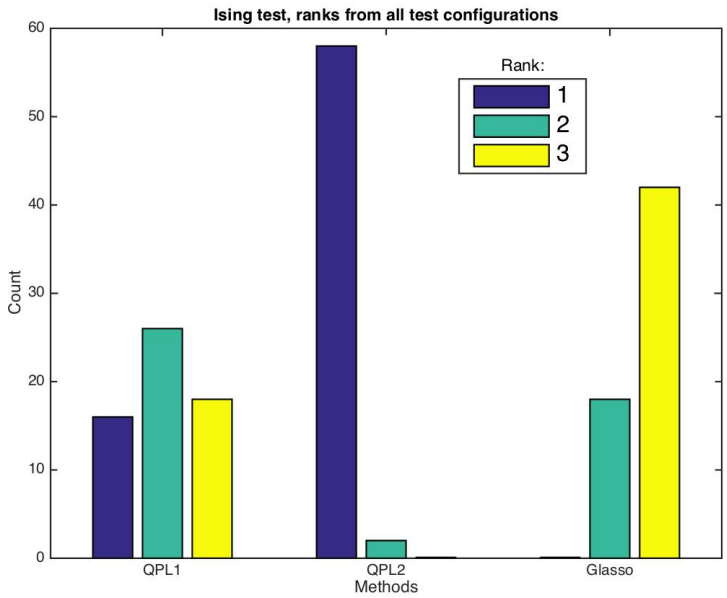

Figure 4.3 – QPLs & Glasso: Number of different rankings for each method in test 1. QPL2 clearly outperforms the other methods.

As can be seen from the results, in this test QPL1 tends to perform better than Glasso, and QPL2 is clearly superior to both other methods. The cases where QPL1 loses to Glasso correspond to the higher noise settings. As an example, figure 4.4 presents some of the results in the case of $T = 3$.

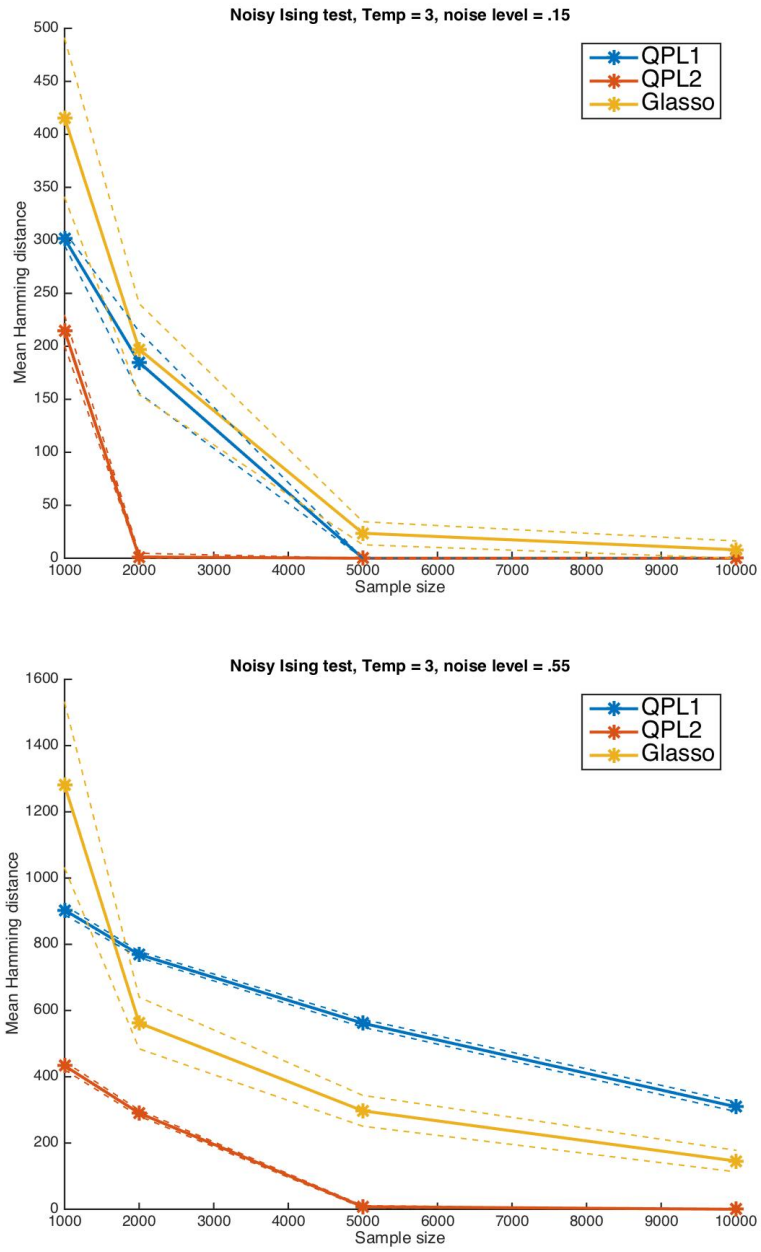

Figure 4.4 – QPLs & Glasso: Temp=3. QPL1 tends to perform better than Glasso with low noise level (top). Glasso outperforms QPL1 with high noise level (bottom).

4.3.2 Second test: Sherrington-Kirkpatrick model

As a second testing setup, we use the S-K model. In this model, the interaction strengths are drawn randomly from a normal distribution, i.e. we set

$$J_{j,i} \sim N(0, 1/d), \tag{4.7}$$

where d is the number of variables. The result is that both the interaction strengths and the model topology are randomly generated.

This model presents some problems for the performance measurement, since interactions that are practically zero would still be counted as mistakes in learning. To avoid these kind of issues, we use a threshold of 0.15 for the interactions, i.e. all interactions with an absolute value smaller than the threshold value are set to zero.

This structure learning task is significantly harder than the one using regular Ising models. We increase the number of maximum observations to 10^5 and decrease the number of variables to 25, but otherwise the testing setup is kept unchanged.

The ranks from all test configurations calculated as in the first test are plotted in figure 4.5. In this task, both QPL methods perform poorly compared to the Glasso, while QPL2 again outperforms QPL1 very systematically.

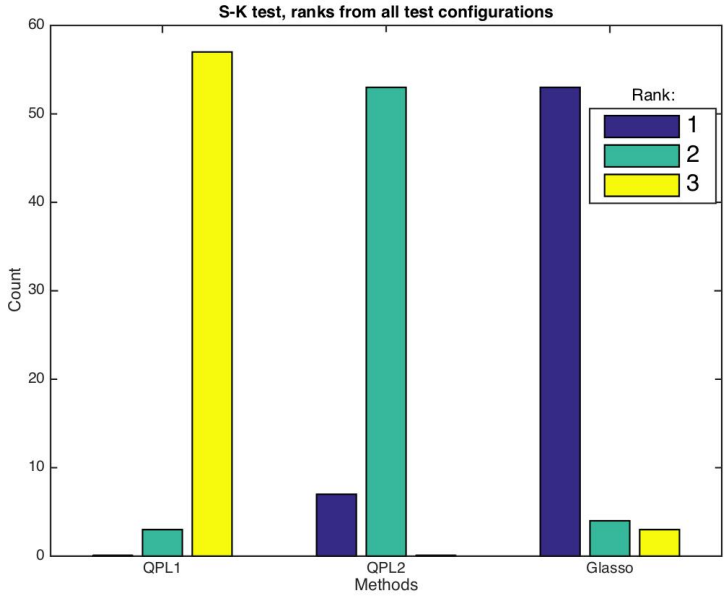

Figure 4.5 – QPLs & Glasso: Number of different rankings for each method in test 2. Glasso is clearly superior to the other methods.

Since the poor performance of QPLs persists with virtually all noise levels, it would seem that the problem is not mainly caused by the noise. To test this further, we also used MPL learning on the noiseless datasets with $T = 2.5$. As expected, the MPL results

are very similar to the QPL2 results with low noise settings (see figure 4.6). This lends some credence to the claim that the problem is related to the actual scoring used and not so much to the noise.

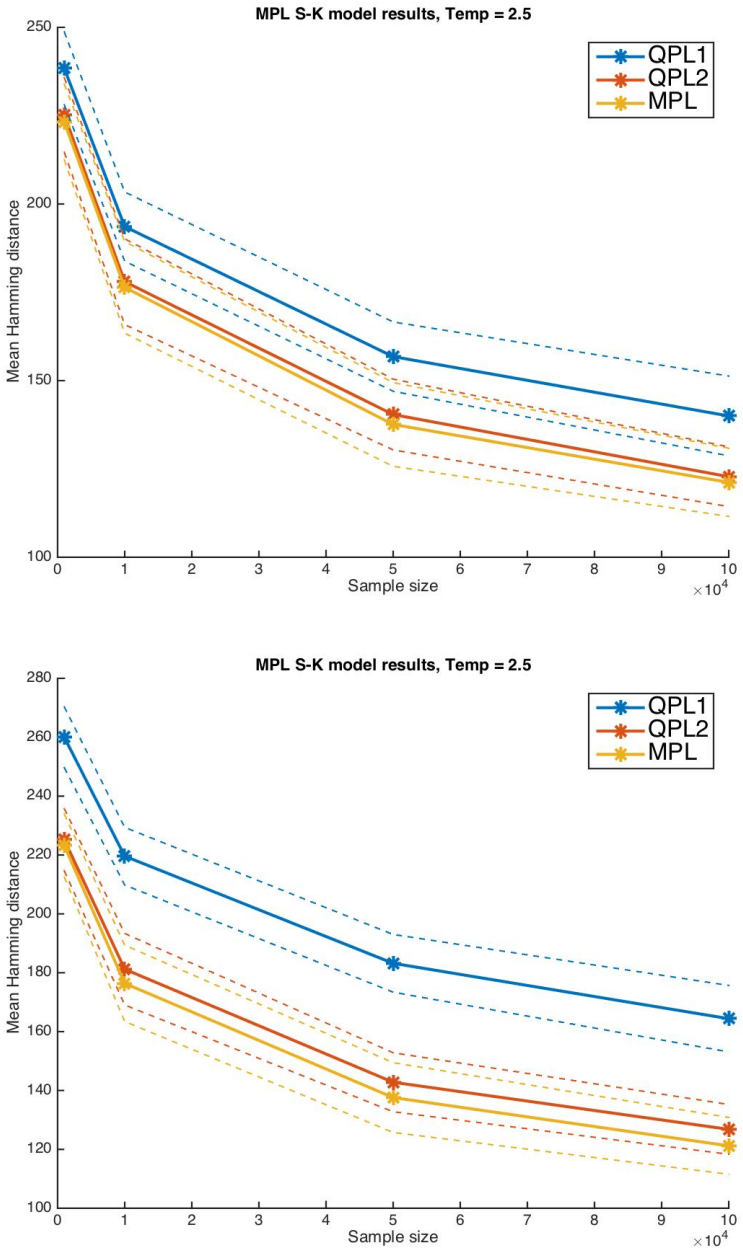

Figure 4.6 – QPLs & MPL with noise levels 0.15 (top) and 0.35 (bottom). QPL2 performance is very similar to MPL on lower noise levels.

4.3.3 Third test: Ising model with random interaction strengths

Based on the second experiment, the bad performance could be caused either by the interactions strengths, which are weak on average, or the more densely connected model topology; the average number of neighbours for a node is around 11. To further test these issues, as a third test setup we use the fixed 2D Ising model topology with $5*5 = 25$ nodes, but generate the interaction strengths randomly as in the S-K model. We also use a threshold $= 0.15$ in this setting, but since we do not want to randomize the topology, the interactions below the threshold are just randomly generated again until they are above the threshold level. We run the test with temperature $T = 3$.

With this setting, both QPL methods perform better again (see ranks as in the previous tests in figure 4.7). QPL2 again performs consistently better than QPL1, while both QPLs outperform Glasso on lower noise settings and get left behind on high noise level (see figure 4.8 for examples). In the case of lower noise levels, the results therefore seem to point to the high number of neighbours as the primary reason for the bad performance on the S-K tests, although the high-noise & weak interactions combination is also a troublesome case.

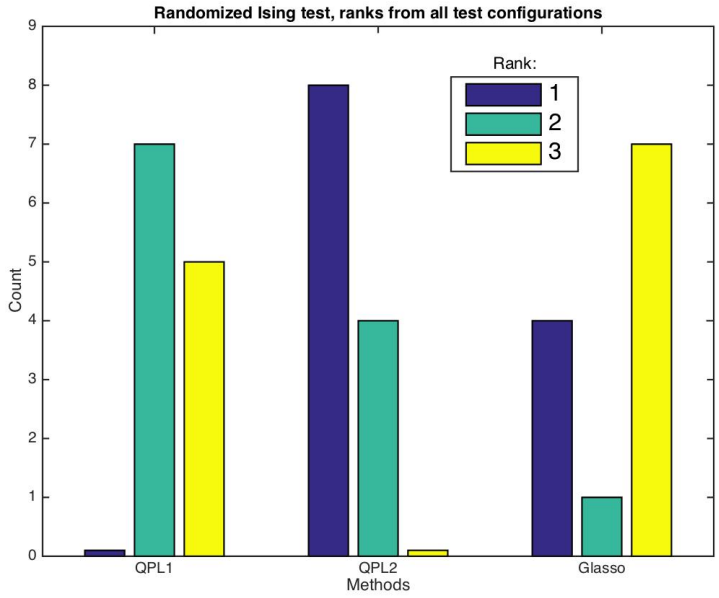

Figure 4.7 – QPLs & Glasso: Number of different rankings for each method in test 3. QPL2 performs better than the other two methods.

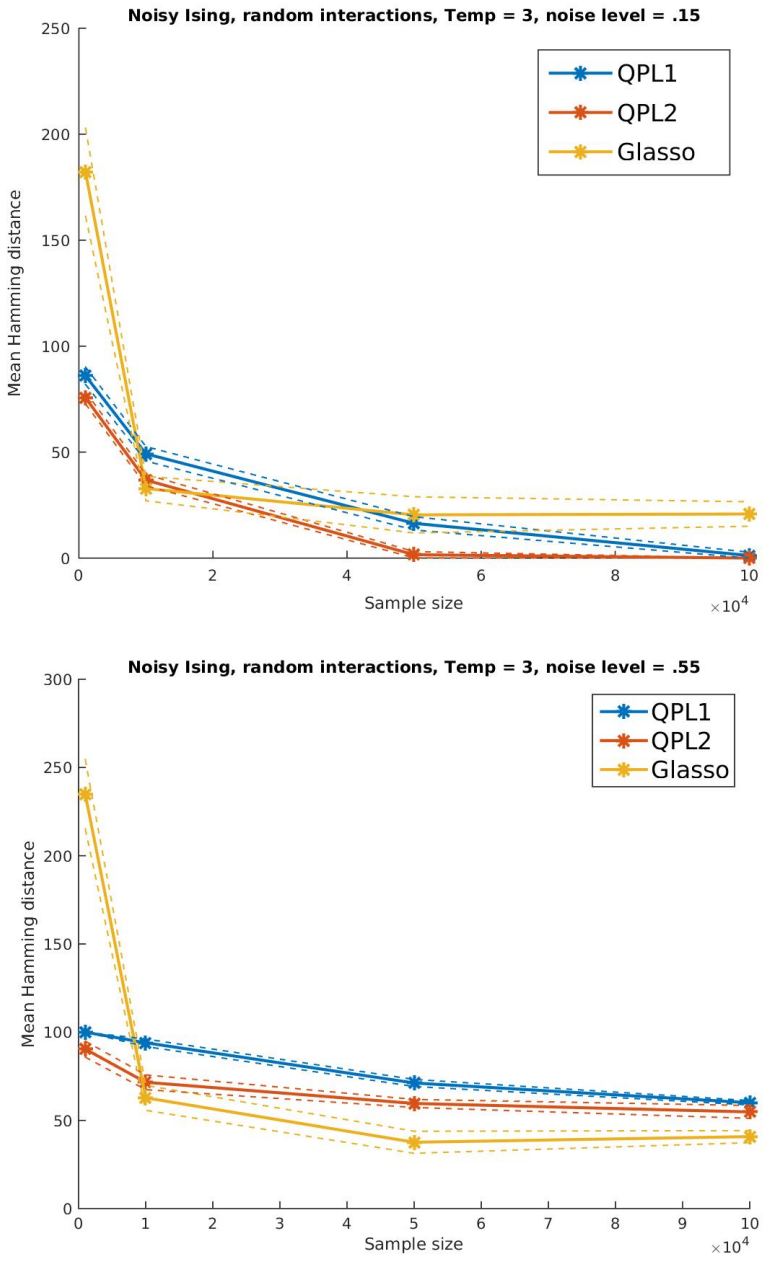

Figure 4.8 – QPLs & Glasso: QPLs perform better on lower noise levels (top), while Glasso outperforms others on the highest noise level (bottom).

One logical remedy for the problems with denser graphs and weak interactions is to tweak the prior used, since the problem basically boils down to over-regularization; based on the previous tests, the equivalent sample size value $N = 1$ used in the previous tests seems to correspond to quite a heavy penalty on denser connectivity between the

nodes. To set the prior in a more informed manner, we could resort e.g. to optimizing the prior strength on some separate training set similarly to the approach used with the Glasso parameter tuning.

4.3.4 Fourth test: Different priors on Ising model with randomized interaction strengths

To see the effect of the prior on the results, we use the same datasets generated in the previous test with fixed 2D Ising topology and random interactions, but run the learning with different values for the equal sample size prior N. Examples of the results with $N \in \{1, 10^1, 10^2, 10^3, 10^4, 10^5\}$ are plotted in figure 4.9. The results with other noise levels lead to the same conclusions. It seems that the prior can be used effectively to enhance performance in noisy environments with weak interactions. Based on this test, the prior strength should definitively be treated as a free parameter that needs to be set in an informed manner.

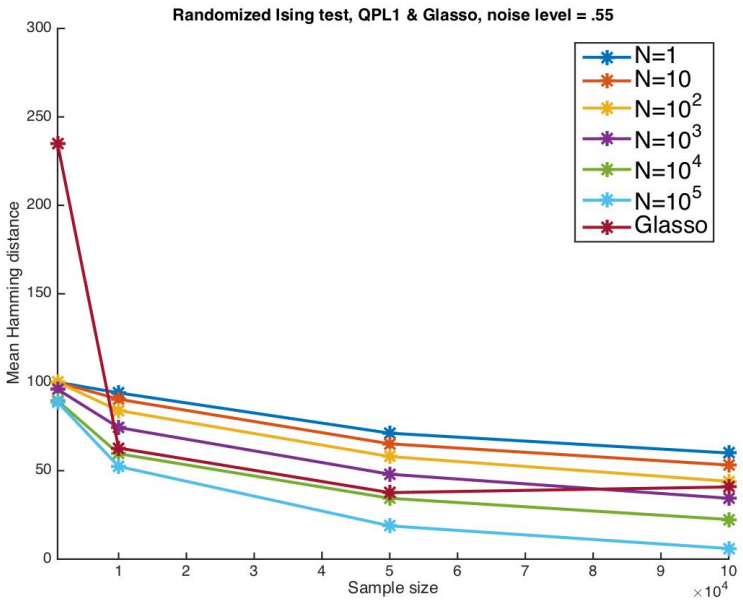

Figure 4.9 – QPL1 with varying prior strength and Glasso. Optimizing the prior value is clearly beneficial.

4.3.5 Fifth test: Different priors on Sherrington-Kirkpatrick model

As a final test, to gauge the effect of the prior with dense graphs and weak interactions, we use the same S-K model data generated in the second test with $T = 3$ with prior values $N \in \{1, 10^1, 10^2, 10^3, 10^4, 10^5\}$. Similarly to the previous test, the prior strength can be tuned to improve the performance of both methods significantly. As already

noted before, the bad performance of both QPLs in learning the S-K model is caused by the dense graph structure together with the weak interactions. To see the effect of the different prior values in terms of average learned MB size, see figure 4.10.

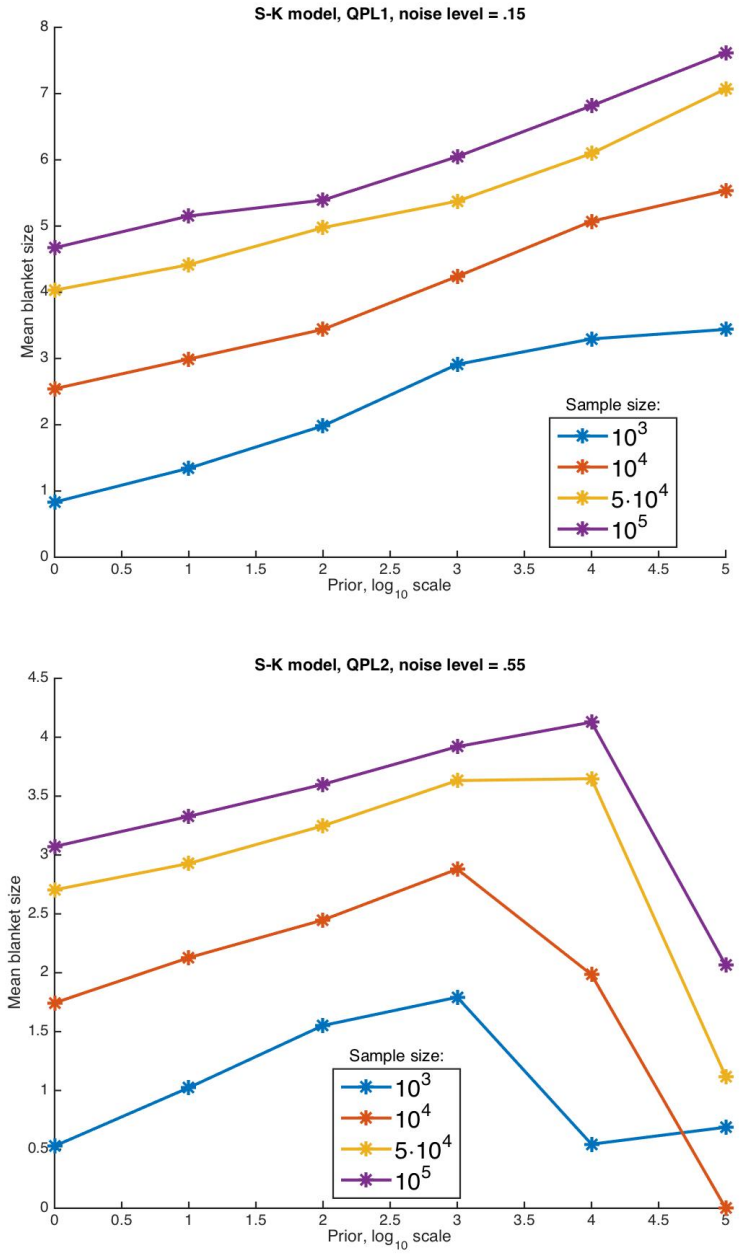

Figure 4.10 – QPLs: Increasing the prior strength increases the average size of the learned blanket (top). With too high prior values the learned blanket sizes start to shrink again (bottom).

One possible problem with the priors resulting in larger average MB sizes is that the larger blankets can include spurious edges more easily. As can be seen from figure 4.11, this problem starts to appear with the largest prior values together with the smallest sample sizes using QPL2 scoring.

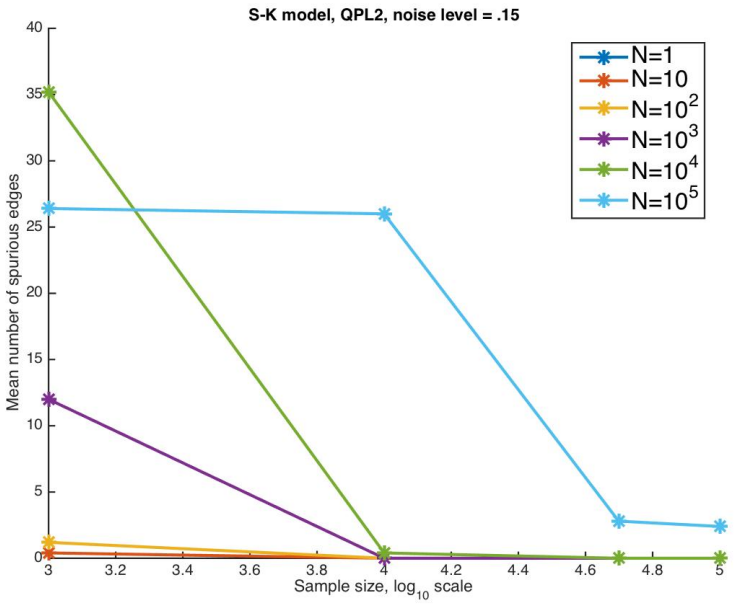

Figure 4.11 – QPL2: Increasing the prior strength increases the errors due to spurious edges especially with smaller sample sizes.

In conclusion, since the best results for both QPLs are still behind Glasso (see figure 4.12 where the best performance of both QPLs found when varying the prior is plotted with the corresponding results for Glasso), it seems that the denser graphs with weak interactions present clear problems for both methods. The performance can be affected up to a point by optimizing the prior, but it seems that further improvements would probably require choosing a different responsibility from the ones used in QPLs 1 & 2.

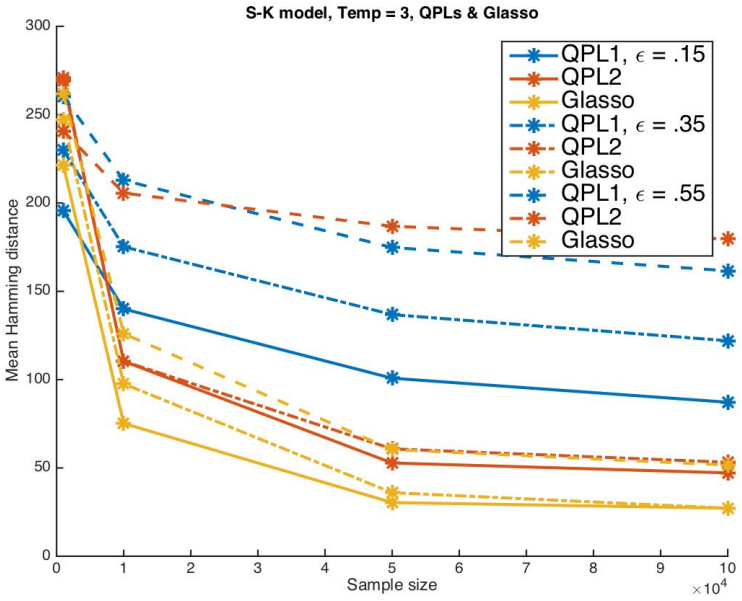

Figure 4.12 – QPLs & Glasso: All noise levels. QPLs correspond to best performance from tests with various prior values. In most cases Glasso performs better than QPLs with tuned priors.

Chapter 5

Conclusions

The QPL scorings derived and tested in this thesis represent one of few alternatives for learning continuous-valued MRFs, that is not based on assumed joint normality of the data. However, since the asymptotic consistency is quite a basic requirement for learning criterions, the example where the QPL1 criterion is not consistent found in this thesis is a setback for the method.

It is possible that the QPL2 scoring or some other alternative utilizing the same basic idea is found to be consistent. To this end, it would be important in future work to find a way to define the responsibilities in such a way as to preserve the dependence structure in the noiseless data. Assuming this kind of structure-preserving discretization is found, besides the QPL methods, we could then of course use any consistent structure learning method for discrete data with guaranteed consistency. Another possibility for searching consistency with the QPL-based methods would be to change the way the binary nodes are transformed into the noisy continuous nodes.

As for the performance in learning model structures compared to the Glasso method, the QPL criterions tested in this thesis perform very well on some problems, but both fall behind on others. Based on the tests used, the QPL methods seem to work well with relatively sparsely connected models, whereas more densely connected models with weaker interactions present problems for both scorings tested.

In conclusion, it seems possible that both of the main problems encountered, i.e. the lack of consistency and the poor performance with some models, could be remedied by changing the model specification, defining the responsibilities in some alternative manner, or possibly to some extent by using a different type of prior for the learning. Given these fixes, the QPL methods would be a welcome addition to the sparsely populated set of MRF learning methods for continuous, non-Gaussian data.

Appendix A

Appendix: Inconsistency example for QPL1

In this appendix we show that the QPL1 estimator is not consistent in the general case. This is done by finding a specific case where it can be shown to be inconsistent. As far as we know, this is a new result that has not been proven before.

The general idea of the proof is to consider the fuzzy counts constructed from the QPL1 responsibilities as noiseless counts coming from a new noiseless model, that is equivalent to the noisy model in terms of its dependence structure. This enables us to use one of the already proven results of consistency to find the asymptotic dependence structure implied by the QPL1 responsibility definition. With this dependence structure for the noisy model found, we can compare the conditional independence relations that hold in the noisy model with the corresponding relations that hold in the true noiseless model. To show that QPL1 is not generally consistent, we then construct a model where the conditional independencies differ between the noisy model and the true noiseless model.

The general idea of the proof could also be useful for establishing general consistency or finding cases of inconsistency for other related learning criterions defined using different responsibilities.

We start by some notational conventions and definitions used in the proof. After this, we prove a consistency criterion and five lemmas that we can use to test the consistency of QPL-type criterions. The final step is to put these together and provide a counter-example that shows that QPL1 with the assumptions made is not consistent in the general case.

A.1 Notations and definitions

The definitions and most of the notations have already been introduced in the main text, but the most important ones are repeated here for convenience. There are also a few notational conventions that are only used in this appendix.

Depending on the situation, a MB of size N_j is written either jointly as l or individually as $l_1, l_2, \ldots, l_{N_j}$. The noiseless true counts in a sample of n observations are denoted by

$$n_{ijl} = \sum_{k=1}^{n} n_{ijlk} = \sum_{k=1}^{n} n_{ijl_1 \ldots l_{N_j} k}, \tag{A.1}$$

which refers to the total number of occurrences of the configuration $(X_j = i, X_{mb(j,1)} = l_1, \ldots, X_{mb(j,N_j)} = l_{N_j})$ in the data. We write $mb(j,m)$ to reference the mth variable in the MB $mb(j)$. The kth observation for the variable $mb(j,m)$ is indexed as $mb(j,m,k)$. The true counts e.g. in the form n_{ijl}/n_{jl} are ML-estimates for the corresponding probabilities.

The fuzzy counts for QPL1 are defined as in the main text, i.e.

$$s_{ijlk} = (2iy_{j,k} - y_{j,k} - i + 1) \prod_{m=1}^{N_j} (2l_m y_{mb(j,m,k)} - y_{mb(j,m,k)} - l_m + 1) \tag{A.2}$$

$$= (1 - |i - y_{j,k}|) \prod_{m=1}^{N_j} (1 - |l_m - y_{mb(j,m,k)}|). \tag{A.3}$$

By definition we have

$$s_{ijl} = \sum_{k=1}^{n} s_{ijlk}. \tag{A.4}$$

The MPL criterion for binary observations $\mathbf{x}$ and graph structure G is defined as

$$\hat{p}(\mathbf{x}|G) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(n_{jl} + \alpha_{jl})} \prod_{i=0}^{1} \frac{\Gamma(n_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}, \tag{A.5}$$

where d is the number of variables, q_j is the size of the outcome space of the MB $mb(j)$, and $\alpha_{ijl}, \alpha_{jl}$ are pseudo-counts given by the prior. In turn, the QPL criterion for observations $\mathbf{y}$ in the [0,1] interval is defined as

$$\tilde{p}(\mathbf{y}|G) = \prod_{j=1}^{d} \prod_{l=1}^{q_j} \frac{\Gamma(\alpha_{jl})}{\Gamma(s_{jl} + \alpha_{jl})} \prod_{i=0}^{1} \frac{\Gamma(s_{ijl} + \alpha_{ijl})}{\Gamma(\alpha_{ijl})}. \tag{A.6}$$

We write

$$ijl|l_t = s \tag{A.7}$$

for a configuration $(i, l_1, \ldots, l_{t-1}, l_t = s, l_{t+1}, \ldots, l_{N_j})$ where the variable l_t is set to the value s.

For a given configuration, we use the logical negation symbol $\neg$ to denote the other binary value, e.g. if $i = 0$ then $\neg i = 1$ and vice versa.

Given a configuration $(i, l_1, \ldots, l_{N_j})$, we also use a special notation to denote all the possible combinations of the variables, when some number z of them are set to

their negations. Since this is only used in summations with the true counts, this is written as $\sum_{e(z)} n_{ijl}$. E.g. for a configuration (i, l_1, l_2), $\sum_{e(0)} n_{ijl} = n_{ijl_1l_2}$, $\sum_{e(1)} n_{ijl} = n_{ijl_1\neg l_2} + n_{ij\neg l_1l_2} + n_{\neg ijl_1l_2}$ and $\sum_{e(3)} n_{ijl} = n_{\neg ij\neg l_1\neg l_2}$.

This notation is also used for the configurations written e.g. as $ijl|l_t = s$. In this case, the variable t behaves as any other variable. For example, with the configuration $i, l_1|l_1 = s$ we would have $\sum_{e(1)} n_{ijl_1|l_1=s} = n_{ijl_1=\neg s} + n_{\neg ijl_1=s}$.

A.2 Inconsistency example

Before constructing the actual example of inconsistency for the QPL1 criterion, we first state some general assumptions and establish a consistency criterion that is a necessary condition for consistency. To utilize the consistency criterion, we then have to find a useful form for the fuzzy counts s_{ijl}, s_{jl} constructed from the QPL1 responsibilities, and show that with some assumptions the QPL1 scoring learns the structure inherent in the fuzzy data. We formulate these results as five lemmas in section A.2.3. The final step is then to show that QPL1 fails the consistency criterion in a specific case.

A.2.1 General assumptions

Let $X = (X_1, X_2, \ldots, X_d)$ be a set of binary random variables, and let $\mathcal{G}$ be the set of all MNs over the variables X. For each $G \in \mathcal{G}$ we have a family of joint distributions, with each member parameterized by $\theta_G \in \Theta_G$. In the following we assume that the joint distribution is positive and that the MN is a perfect map for the distribution.

Assume that for each binary variable X_j we have a noise variable $\epsilon_j \in [0, 1]$ s.t. for all j, $\epsilon_j \perp\!\!\!\perp X_A, \epsilon_B$, where $A \subseteq \{1, 2, \ldots, d\}$, $B \subseteq \{1, 2, \ldots, d\} \setminus j$. In other words, each noise variable is assumed to be mutually independent of every set of variables not including itself.

Assume we have a sample of size n of noisy observations

$$y_{j,k} = |x_{j,k} - \epsilon_{j,k}|, k = 1, \ldots, n, j = 1, \ldots, d. \tag{A.8}$$

Assume further that $\xi = \mathrm{E}(\epsilon_j)\ \forall j$ is the expected value of the noise variables, $\xi \in (0, 1/2)$. It is worth noting that this last assumption is somewhat modified in some of the arguments given below.

It is also worth defining properly, what we mean by consistency. Let $G^* \in \mathcal{G}$ be the true graph structure of a MN, with the corresponding Markov blankets $mb(G^*) = \{mb^*(1), \ldots, mb^*(d)\}$. Let $\theta_{G^*} \in \Theta_{G^*}$ define the corresponding joint distribution that is faithful to G^*, and assume that a sample $\mathbf{x}$ of size n from this distribution is transformed by noise variables ϵ_j, $\mathrm{E}(\epsilon_j) = \xi, j = 1, \ldots, d$, as defined above to produce an observed sample $\mathbf{y}$ of similar size. For an arbitrary variable $j \in \{1, \ldots, d\}$ with a true MB of size N_j^*, a local estimator

$$\hat{mb}(j) = \underset{mb(j) \subseteq V \setminus j}{\arg\max}\, \hat{p}(y_j | y_{mb(j)}) \tag{A.9}$$

is consistent, if $\hat{mb}(j) = mb^*(j)$ eventually almost surely as $n \to \infty$. Here we write $\hat{p}(y_j|y_{mb(j)})$ for the scoring function used.

If the local estimator is consistent, since the set of MBs uniquely define the dependence structure of the model, then it follows that the global estimator

$$\hat{G} = \arg\max_{G \in \mathcal{G}} \hat{p}(\mathbf{y}|G) \tag{A.10}$$

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

A.2.2 Consistency criterion

QPL-based graph structure scorings rely on constructing a set of fuzzy counts from the original noiseless counts that are then used for learning. A necessary condition for this kind of learning method to be consistent is that the fuzzy counts have asymptotically the same dependence structure as the true counts. The idea is that a consistent learning method eventually learns the dependence structure in the data, so the fuzzy counts used as data need to have the same structure as the true counts.

Since QPL-based methods are local, i.e. the scoring is a product of node-specific scores, we state the learning criterion in a local form for a MB of size N_j.

Theorem A.2.1. *(Consistency criterion)*
Let $(i, l_1, \ldots, l_{N_j})$ be an arbitrary configuration for node j and its MB, and $t \in \{1, \ldots, N_j\}$, and assume we have access to a consistent local learning method. With the assumptions made in A.2.1, the true noiseless counts n_{ijl}, n_{jl} constructed from the binary data, and the fuzzy counts s_{ijl}, s_{jl} constructed from some responsibilities that satisfy $s_{ijl} > 0, \sum_i \frac{s_{ijl}}{s_{jl}} = 1 \ \forall i, j, l$, if the learning method is locally consistent, then

$$\lim_{n\to\infty} \frac{n_{ijl|l_t=0}}{n_{jl|l_t=0}} = \lim_{n\to\infty} \frac{n_{ijl|l_t=1}}{n_{jl|l_t=1}} \Leftrightarrow \lim_{n\to\infty} \frac{s_{ijl|l_t=0}}{s_{jl|l_t=0}} = \lim_{n\to\infty} \frac{s_{ijl|l_t=1}}{s_{jl|l_t=1}}. \tag{A.11}$$

Proof. Assume we have a consistent learning method at our disposal. Since the fuzzy counts behave as probabilities, we can use the fact that the counts and the fuzzy counts are ML-estimates for some models (see lemma A.2.6 for an example). Consider first the implication from left to right. If t is part of the true MB, the l.h.s. is not true and the implication holds always. If t is not part of the true MB so that the l.h.s. is true, since our learning method is consistent, t cannot be part of the learned blanket and therefore the implication has to hold.

Consider next the implication from right to left. Similarly as before, if t is part of the learned blanket, the r.h.s. is not true and the implication holds always. If t is not part of the learned MB and the r.h.s. is therefore true, since our method is consistent, t cannot be part of the true MB either so the implication holds. □

Next, we prove five lemmas that help us utilize the consistency criterion.

A.2.3 Lemmas

The first lemma states that with QPL1, the fuzzy counts have properties that are similar to true probabilities.

Lemma A.2.2. *For a Markov blanket of size N_j, given the assumptions in A.2.1, the fuzzy counts given by the QPL1 responsibilities behave as true probabilities.*

Proof. By definition, concentrating on the summation over the index i, we have

$$\sum_i s_{ijl} = \sum_{i=0}^{1}\sum_{k=1}^{n} s_{ijlk} \tag{A.12}$$

$$= \sum_{k=1}^{n}\sum_{i=0}^{1}(2iy_{j,k} - y_{j,k} - i + 1)\prod_{m=1}^{N_j}(2l_m y_{mb(j,m,k)} - y_{mb(j,m,k)} - l_m + 1) \tag{A.13}$$

$$= \sum_{k=1}^{n}(1 - y_{j,k} + y_{j,k})\prod_{m=1}^{N_j}(2l_m y_{mb(j,m,k)} - y_{mb(j,m,k)} - l_m + 1) \tag{A.14}$$

$$= \sum_{k=1}^{n}\prod_{m=1}^{N_j}(2l_m y_{mb(j,m,k)} - y_{mb(j,m,k)} - l_m + 1) \tag{A.15}$$

$$= s_{jl}. \tag{A.16}$$

Consequently, with a sufficient sample size, we have

$$\frac{s_{ijl}}{s_{jl}} \in (0,1)\ \forall i,j,l, \text{ and} \tag{A.17}$$

$$\sum_i \frac{s_{ijl}}{s_{jl}} = 1\ \forall j,l. \tag{A.18}$$

□

In the second lemma we express the limit of the fuzzy counts using the true noiseless counts.

Lemma A.2.3. *For a Markov blanket of size N_j, with the assumptions made in subsection A.2.1 with the exception that here we assume the common expectation of the noise $E(\epsilon) = \xi \in [0, 1/2]$,*

$$\begin{aligned}
\lim_{n\to\infty} s_{ijl}/n &= \lim_{n\to\infty}[\sum_{z=0}^{N_j+1}(1-\xi)^{N_j+1-z}\cdot\xi^z(\sum_{e(z)} n_{ijl})]/n, \\
\lim_{n\to\infty}\frac{s_{ijl}}{s_{jl}} &= \lim_{n\to\infty}\frac{\sum_{z=0}^{N_j+1}(1-\xi)^{N_j+1-z}\cdot\xi^z(\sum_{e(z)} n_{ijl})}{\sum_{z=0}^{N_j}(1-\xi)^{N_j-z}\cdot\xi^z(\sum_{e(z)} n_{jl})}
\end{aligned} \tag{A.19}$$

Proof.

For a sample size $n \in \mathbb{N}$, by definition and the assumptions on the noise made in A.2.1 we have

$$s_{ijl} = \sum_{k=1}^{n}(1 - |i - y_{j,k}|) \prod_{m=1}^{N_j}(1 - |l_m - y_{mb(j,m,k)}|) \tag{A.20}$$

$$= \sum_{k=1}^{n}(1 - |i - |x_{j,k} - \epsilon_{j,k}||) \prod_{m=1}^{N_j}(1 - |l_m - |x_{mb(j,m,k)} - \epsilon_{mb(j,m,k)}||). \tag{A.21}$$

For each of the factors above, take for example $(1 - |i - |x_{j,k} - \epsilon_{j,k}||)$, we can rewrite it using indicator functions $\mathbf{I}()$ as

$$\begin{aligned} & \mathbf{I}(i = 0) \cdot [1 - |x_{j,k} - \epsilon_{j,k}|] + \mathbf{I}(i = 1) \cdot [|x_{j,k} - \epsilon_{j,k}|] \\ = \quad & \mathbf{I}(i = 0) \cdot [\mathbf{I}(x_{j,k} = 0) \cdot (1 - \epsilon_{j,k}) + \mathbf{I}(x_{j,k} = 1) \cdot \epsilon_{j,k}] \dots \\ & + \mathbf{I}(i = 1) \cdot [\mathbf{I}(x_{j,k} = 0) \cdot \epsilon_{j,k} + \mathbf{I}(x_{j,k} = 1) \cdot (1 - \epsilon_{j,k})] \\ = \quad & \mathbf{I}(x_{j,k} = i) \cdot (1 - \epsilon_{j,k}) + \mathbf{I}(x_{j,k} = \neg i) \cdot \epsilon_{j,k}. \end{aligned}$$

Writing all factors in A.21 using the indicators we get

$$s_{ijl} = \sum_{k=1}^{n}[\mathbf{I}(x_{j,k} = i) \cdot (1 - \epsilon_{j,k}) + \mathbf{I}(x_{j,k} = \neg i) \cdot \epsilon_{j,k}] \times \dots \tag{A.22}$$

$$\prod_{m=1}^{N_j}[\mathbf{I}(x_{mb(j,m,k)} = l_m) \cdot (1 - \epsilon_{mb(j,m,k)}) + \mathbf{I}(x_{mb(j,m,k)} = \neg l_m) \cdot \epsilon_{mb(j,m,k)}] \tag{A.23}$$

$$= \sum_{k=1}^{n}[\mathbf{I}(x_{j,k} = i, x_{mb(j,1,k)} = l_1, \dots, x_{mb(j,N_j,k)} = l_{N_j}) \cdot (1 - \epsilon_{j,k}) \prod_{m=1}^{N_j}(1 - \epsilon_{mb(j,m,k)}) \dots \tag{A.24}$$

$$+\mathbf{I}(x_{j,k} = \neg i, x_{mb(j,1,k)} = l_1, \dots, x_{mb(j,N_j,k)} = l_{N_j}) \cdot \epsilon_{j,k} \prod_{m=1}^{N_j}(1 - \epsilon_{mb(j,m,k)}) \dots \tag{A.25}$$

$$+ \cdots + \dots \tag{A.26}$$

$$+\mathbf{I}(x_{j,k} = \neg i, x_{mb(j,1,k)} = \neg l_1, \dots, x_{mb(j,N_j,k)} = \neg l_{N_j}) \cdot \epsilon_{j,k} \prod_{m=1}^{N_j} \epsilon_{mb(j,m,k)}], \tag{A.27}$$

where the suppressed terms in (A.26) enumerate all the possible combinations for the configurations $(i, l_1, \dots, l_{N_j})$ with the corresponding noise terms.

Using the strong law of large numbers (see e.g. [8]) we can assert that s_{ijl}/n converges almost surely to the corresponding expectation of the formula starting at (A.24). Since each of the noise terms is assumed to be independent of everything else and the expectation of an indicator is the probability of the event in question, we have

$$
\begin{aligned}
s_{ijl}/n \overset{a.s.}{\to} \; & P(X_j = i, X_{mb(j,1)} = l_1, \ldots, X_{mb(j,N_j)} = l_{N_j}) \ldots \\
& \times E[(1-\epsilon_j) \prod_{m=1}^{N_j} (1-\epsilon_{mb(j,m)})] \ldots \\
& + \cdots + \ldots \\
& P(X_j = \neg i, X_{mb(j,1)} = \neg l_1, \ldots, X_{mb(j,N_j)} = \neg l_{N_j}) \ldots \\
& \times E[\epsilon_j \prod_{m=1}^{N_j} \epsilon_{mb(j,m)}] \\
= \; & P(X_j = i, X_{mb(j,1)} = l_1, \ldots, X_{mb(j,N_j)} = l_{N_j}) \cdot (1-\xi)^{N_j+1} \ldots \\
& + \cdots + \ldots \\
& P(X_j = \neg i, X_{mb(j,1)} = \neg l_1, \ldots, X_{mb(j,N_j)} = \neg l_{N_j}) \cdot \xi^{N_j+1},
\end{aligned}
$$

which shows that we can replace the individual noise terms by their common expectation without affecting the limiting behaviour.

We can take a further step by replacing the probabilities for all the events above by their ML-estimates, i.e. with the corresponding true counts. Furthermore, combining all the common noise terms we get

$$
\begin{aligned}
s_{ijl}/n \overset{a.s.}{\to} \; & \lim_{n\to\infty} [n_{ijl}/n \cdot (1-\xi)^{N_j+1} \ldots && \text{(A.28)} \\
& + (\sum_{e(1)} n_{ijl})/n \cdot (1-\xi)^{N_j} \cdot \xi \ldots && \text{(A.29)} \\
& + \cdots + && \text{(A.30)} \\
& + (\sum_{e(N_j+1)} n_{ijl})/n \cdot \xi^{N_j+1} \;] && \text{(A.31)} \\
= \; & \lim_{n\to\infty} [\sum_{z=0}^{N_j+1} (1-\xi)^{N_j+1-z} \cdot \xi^z (\sum_{e(z)} n_{ijl}) \;]/n, && \text{(A.32)}
\end{aligned}
$$

which proves the first part of the lemma. Moreover, clearly the limit in (A.32) belongs to the interval (0,1], since the counts in the numerator are always a positive fraction of n and the sum of all the counts is at most n.

We can repeat the same argument for the fuzzy counts s_{jl} with minimal changes (i.e. omitting all the factors containing i and lowering the total number of both non-noise and noise variables included correspondingly from $N_j + 1$ to N_j) to show that

$$
s_{jl}/n \overset{a.s.}{\to} \lim_{n\to\infty} [\sum_{z=0}^{N_j} (1-\xi)^{N_j-z} \cdot \xi^z (\sum_{e(z)} n_{jl}) \;]/n. \qquad \text{(A.33)}
$$

Combining (A.32) and (A.33) we get

$$\lim_{n\to\infty} \frac{s_{ijl}}{s_{jl}} = \lim_{n\to\infty} \frac{s_{ijl}/n}{s_{jl}/n} = \lim_{n\to\infty} \frac{\sum_{z=0}^{N_j+1}(1-\xi)^{N_j+1-z}\cdot\xi^z(\sum_{e(z)} n_{ijl})}{\sum_{z=0}^{N_j}(1-\xi)^{N_j-z}\cdot\xi^z(\sum_{e(z)} n_{jl})}, \tag{A.34}$$

as claimed. □

The next lemma affirms that QPL1 without noise is equivalent to MPL.

Lemma A.2.4. *For a Markov blanket of size N_j, repeating the assumptions made in subsection A.2.1 with the exception that the expected value for the noise variables $E(\epsilon_j) = \xi = 0\ \forall j$, and assuming suitable (i.e. non-zero) prior distributions, QPL1 is equivalent to MPL.*

Proof.

Since $\epsilon_j \in [0,1]$ and $E(\epsilon_j) = 0\ \forall j$, necessarily $\epsilon_{j,k} = 0$ for all $k = 1,\ldots,n$, and consequently QPL1 is trivially equivalent to MPL. □

The fourth lemma just states that the MPL estimator is consistent. This has already been shown elsewhere.

Lemma A.2.5. *The MPL estimator is consistent.*

Proof.

See the MPL consistency proof in [24]. □

The final lemma shows that given some model producing fuzzy counts, we can always find a corresponding new noiseless model with asymptotically the same counts.

Lemma A.2.6. *Given a model producing fuzzy counts s_{ijl}, s_{jl} constructed from the QPL1 responsibilities and assumptions made in A.2.1, there exists a noiseless model with counts n'_{ijl}, n'_{jl} that satisfies*

$$\lim_{n\to\infty} \frac{n'_{ijl}}{n'_{jl}} = \lim_{n\to\infty} \frac{s_{ijl}}{s_{jl}} \forall i,j,l. \tag{A.35}$$

Proof.

Let $(i, l_1, \ldots, l_{N_j})$ be an arbitrary configuration for node j and its MB. By lemma A.2.2, we know that the fuzzy counts behave as probabilities. In addition, from lemma A.2.3 we know how the fuzzy counts asymptotically behave in terms of the corresponding original noiseless counts n_{ijl}, n_{jl}. Consequently, with obvious notations, the new noiseless model is given simply by the probabilities

$$P(X'_j = i | X'_{mb(j,1)} = l'_1, \ldots, X'_{mb(j,N_j)} = l'_{N_j}) = \lim_{n\to\infty} \frac{n'_{ijl}}{n'_{jl}} \tag{A.36}$$

$$= \lim_{n\to\infty} \frac{\sum_{z=0}^{N_j+1} (1-\xi)^{N_j+1-z} \cdot \xi^z (\sum_{e(z)} n_{ijl})}{\sum_{z=0}^{N_j} (1-\xi)^{N_j-z} \cdot \xi^z (\sum_{e(z)} n_{jl})}. \tag{A.37}$$

□

A.2.4 Consistency check for QPL1

Taken together, lemmas A.2.2-A.2.6 mean that we can utilize the consistency criterion since we have shown that QPL scoring on the fuzzy counts, interpreted as the new noiseless counts, is consistent. To show that QPL1 does not generally learn the true dependence structure, it is therefore enough to show that it violates the consistency criterion A.2.1 in a specific case.

In this section, we modify the notation somewhat to keep it more readable. We write e.g. n_{itl} for the true counts where the variables I, T, L have the values i, t, l, and in the same spirit the probability $P(I = i, T = t, L = l)$ is written as $P(itl)$.

Theorem A.2.7. *With the assumptions made in A.2.1, the QPL1 responsibility does not generally preserve the dependence structure in the data, i.e. there exists a case where QPL1 does not give a consistent estimator for the true model structure.*

Proof. Assume our MRF consists of three variables, denoted I, T, L s.t. $I\&L$, and $L\&T$ are neighbours in the graph. Assume also that the general assumptions made in $A.2.1$ are true. The true MB for node I is then L, and consequently $I \perp\!\!\!\perp T|L$. Since we have a positive joint distribution that is faithful to the graph, we also know that no other independencies hold, specifically $I \not\perp\!\!\!\perp L$, $I \not\perp\!\!\!\perp T$ and $L \not\perp\!\!\!\perp T$.

Let i, t, l be an arbitrary configuration for the variables. Clearly, using the new notation we have

$$\lim_{n\to\infty} \frac{n_{itl}}{n_{tl}} = \lim_{n\to\infty} \frac{n_{i\neg tl}}{n_{\neg tl}} \tag{A.38}$$

$$\Leftrightarrow \frac{P(itl)}{P(tl)} = \frac{P(i\neg tl)}{P(\neg tl)} \tag{A.39}$$

$$\Leftrightarrow P(i|tl) = P(i|\neg tl) \tag{A.40}$$

$$\Leftrightarrow P(i|l) = P(i|l), \tag{A.41}$$

since the true counts are ML-estimates for the corresponding probabilities, i.e. in this case the l.h.s. of the consistency criterion A.2.1 is true, since $I \perp\!\!\!\perp T|L$. If QPL1 is consistent, the r.h.s. of the consistency criterion therefore has to be true also. From the r.h.s. of A.2.1 we have

$$\lim_{n\to\infty} \frac{s_{itl}}{s_{tl}} = \lim_{n\to\infty} \frac{s_{i\neg tl}}{s_{\neg tl}} \tag{A.42}$$

$$\Leftrightarrow \lim_{n\to\infty} s_{itl}/n \cdot \lim_{n\to\infty} s_{\neg tl}/n - \lim_{n\to\infty} s_{i\neg tl}/n \cdot \lim_{n\to\infty} s_{tl}/n = 0. \tag{A.43}$$

For ease of reading, we will use the following shorthand-notation for the true counts

$$\begin{aligned} a &= n_{itl} & f &= n_{i\neg t\neg l} \\ b &= n_{\neg itl} & g &= n_{\neg i\neg t\neg l} \\ c &= n_{i\neg tl} & h &= n_{it\neg l} \\ d &= n_{\neg i\neg tl} & k &= n_{\neg it\neg l}. \end{aligned}$$

The rest of the counts needed in the proof can then be written in terms of these, e.g. $n_{tl} = n_{itl} + n_{\neg itl} = a + b$ etc.

Using lemma A.2.3 and omitting the limits for clarity, we can calculate the products in (A.43) to get the summands:

$$\begin{gathered} (1-\xi)^5\{a(c+d) - c(a+b)\}, \\ (1-\xi)^4\xi\{(c+d)(b+c+h) + (a+b+f+g)a - [(a+b)(a+d+f) + (c+d+h+k)c]\}, \\ (1-\xi)^3\xi^2\{(c+d)(d+k+f) + (a+b+f+g)(b+c+h) + (h+k)a \dots \\ \cdots - [(a+b)(b+g+h) + (c+d+h+k)(a+d+f) + (f+g)c]\}, \\ (1-\xi)^2\xi^3\{(c+d)g + (a+b+f+g)(d+k+f) + (h+k)(b+c+h) \dots \\ \cdots - [(a+b)k + (c+d+h+k)(b+g+h) + (f+g)(a+d+f)]\}, \\ (1-\xi)\xi^4\{(a+b+f+g)g + (h+k)(d+k+f) - [(c+d+h+k)k + (f+g)(b+g+h)]\}, \\ \xi^5\{g(h+k) - k(f+g)\}. \end{gathered}$$

To continue, we collect the coefficient for each power of ξ and equate them to zero. This results in a system of 6 equations, i.e. one for each power including the constant terms. Since (A.43) defines a polynomial equation of degree 5 at most, for the given counts there can be at most 5 solutions for the equation that depend on ξ. In these cases the following argument does not need to hold. In other words, our solution holds for almost every value of ξ.

After some tedious calculations, we are left with five non-trivial equations:

$$ad - bc = 0, \tag{A.44}$$

$$ad - bc - ag + bf - dh + ck + gh - fk = 0, \tag{A.45}$$

$$12(bc - ad) + 8(ag - bf + dh - ck) + 4(fk - gh) = 0, \tag{A.46}$$

$$13(ad - bc) + 5(-ag + bf - dh + ck) + gh - fk = 0, \tag{A.47}$$

$$6(bc - ad) + ag - bf + dh - ck) = 0. \tag{A.48}$$

Equation (A.44) can be solved to get

$$\frac{a}{b} = \frac{c}{d} \Leftrightarrow I \perp\!\!\!\perp T | L = l. \tag{A.49}$$

Substituting this to (A.45) we get

$$fk - gh = bf - ag + ck - dh. \tag{A.50}$$

Continuing with (A.46) we are left with

$$ag + dh - bf - ck = 0, \tag{A.51}$$

which also satisfies (A.48). This means by the previous equation that

$$\frac{f}{g} = \frac{h}{k} \Leftrightarrow I \perp\!\!\!\perp T|L = \neg l. \tag{A.52}$$

Putting (A.49) & (A.52) together, we then have $I \perp\!\!\!\perp T|L$, which is all well and good. The problem, however, is equation (A.51).

We have

$$ag + dh = bf + ck \tag{A.53}$$

$$\Leftrightarrow \frac{ag}{h} + d = \frac{bf}{h} + \frac{ck}{h} \tag{A.54}$$

$$\Leftrightarrow \frac{ag}{h} + d = \frac{bg}{k} + \frac{cg}{f} \tag{A.55}$$

$$\Leftrightarrow \frac{a}{h} + \frac{d}{g} = \frac{b}{k} + \frac{c}{f}, \tag{A.56}$$

where (A.55) follows from (A.52). Since the true counts are ML-estimates, we can calculate the (omitted) limits to get the corresponding probabilities, which gives an equivalent equation

$$\frac{P(\neg itl)}{P(\neg it\neg l)} + \frac{P(i\neg tl)}{P(i\neg t\neg l)} = \frac{P(itl)}{P(it\neg l)} + \frac{P(\neg i\neg tl)}{P(\neg i\neg t\neg l)}$$

$$\Leftrightarrow$$

$$\frac{P(\neg i|tl)P(l|t)P(t)}{P(\neg i|t\neg l)P(\neg l|t)P(t)} + \frac{P(i|\neg tl)P(l|\neg t)P(\neg t)}{P(i|\neg t\neg l)P(\neg l|\neg t)P(\neg t)} = \frac{P(i|tl)P(l|t)P(t)}{P(i|t\neg l)P(\neg l|t)P(t)} + \frac{P(\neg i|\neg tl)P(l|\neg t)P(\neg t)}{P(\neg i|\neg t\neg l)P(\neg l|\neg t)P(\neg t)}.$$

Using the assumptions of conditional independency $I \perp\!\!\!\perp T|L$, we get

$$\Leftrightarrow$$

$$\frac{P(\neg i|l)P(l|t)}{P(\neg i|\neg l)P(\neg l|t)} + \frac{P(i|l)P(l|\neg t)}{P(i|\neg l)P(\neg l|\neg t)} = \frac{P(i|l)P(l|t)}{P(i|\neg l)P(\neg l|t)} + \frac{P(\neg i|l)P(l|\neg t)}{P(\neg i|\neg l)P(\neg l|\neg t)}$$

$$\Leftrightarrow$$

$$\frac{P(\neg i|l)}{P(\neg i|\neg l)}\{\frac{P(l|t)}{P(\neg l|t)} - \frac{P(l|\neg t)}{P(\neg l|\neg t)}\} + \frac{P(i|l)}{P(i|\neg l)}\{\frac{P(l|\neg t)}{P(\neg l|\neg t)} - \frac{P(l|t)}{P(\neg l|t)}\} = 0$$

$$\Leftrightarrow$$

$$\{\frac{P(l|t)}{P(\neg l|t)} - \frac{P(l|\neg t)}{P(\neg l|\neg t)}\} \cdot \{\frac{P(\neg i|l)}{P(\neg i|\neg l)} - \frac{P(i|l)}{P(i|\neg l)}\} = 0,$$

which corresponds to claiming that either $L \perp\!\!\!\perp T$ or $I \perp\!\!\!\perp L$. Since neither of these independencies holds with the assumed model structure, we conclude that (A.42) is not true, which means that QPL1 fails the consistency criterion A.2.1, and consequently, that QPL1 is inconsistent with almost all values of ξ in this case. □

Appendix B

Appendix: Test results

This appendix contains figures and numerical results from the tests described in section 4.3. The lines in the plots represent means of the Hamming distances w.r.t. the right dependence structures, calculated from 5 independent runs for each setting. The 95% CIs for the means are plotted as dashed lines. QPL results correspond to the hill-climbing solutions (see section 3.3.2).

B.1 First test setup results

Results for Ising models with varying temperature.

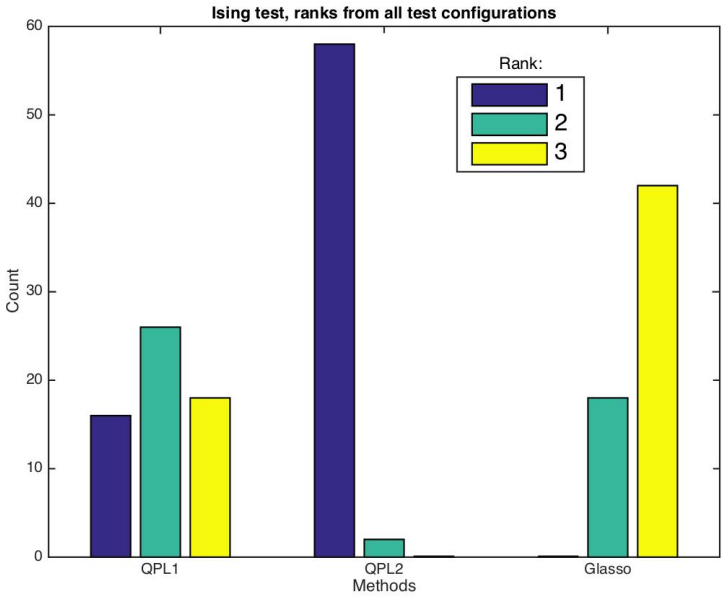

Figure B.1 – QPLs & Glasso: Number of different rankings for each method in test 1.

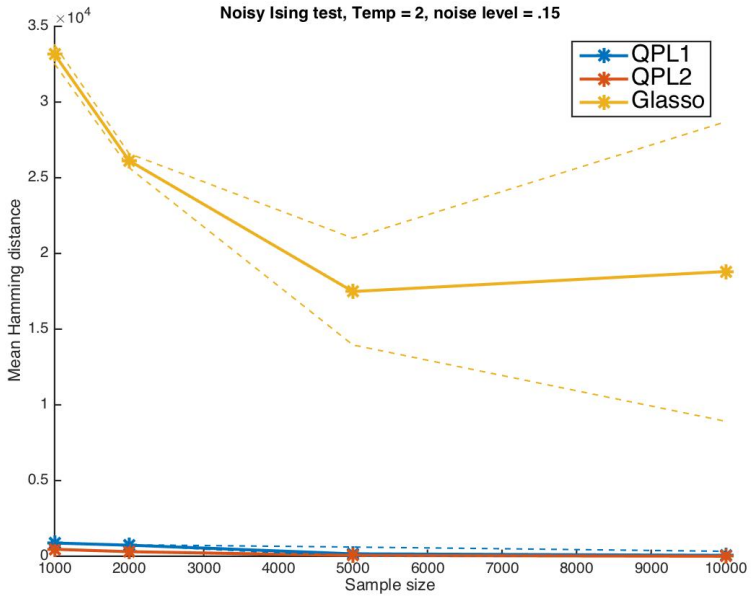

Figure B.2 – QPLs & Glasso: Temp=2.

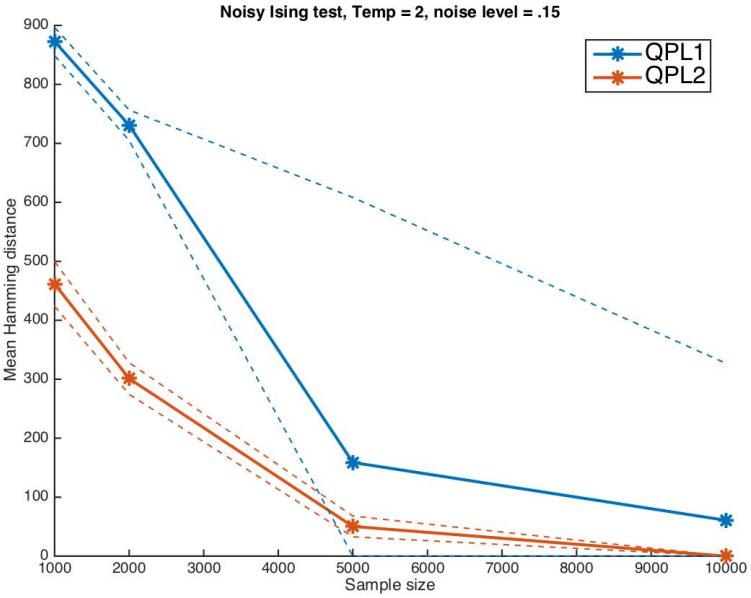

Figure B.3 – QPL1 & 2 : Temp=2.

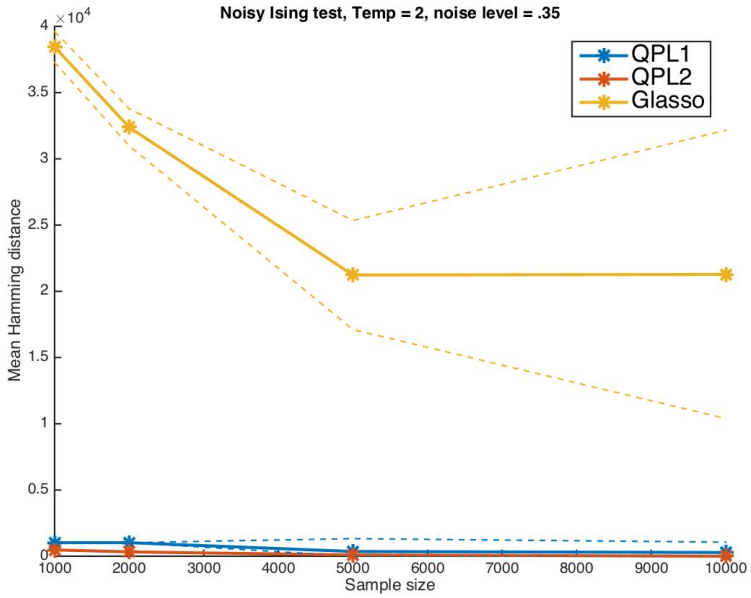

Figure B.4 – QPLs & Glasso: Temp=2.

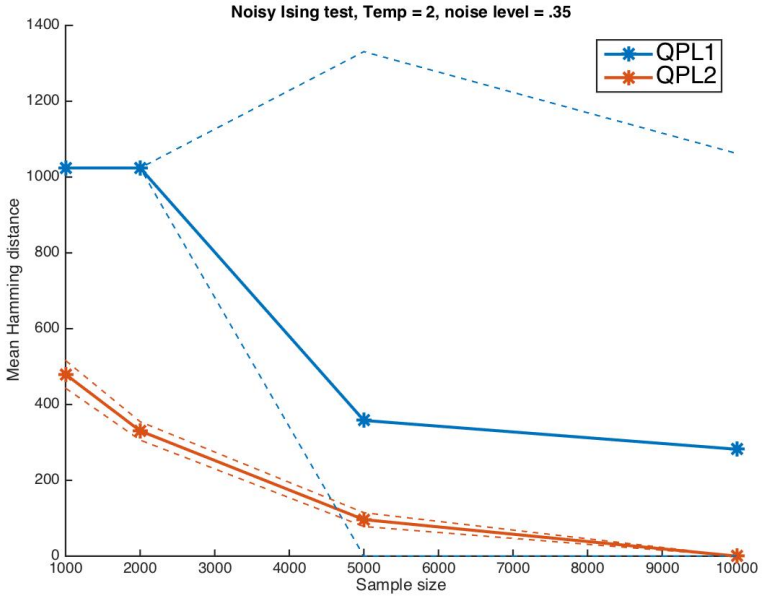

Figure B.5 – QPL1 & 2 : Temp=2.

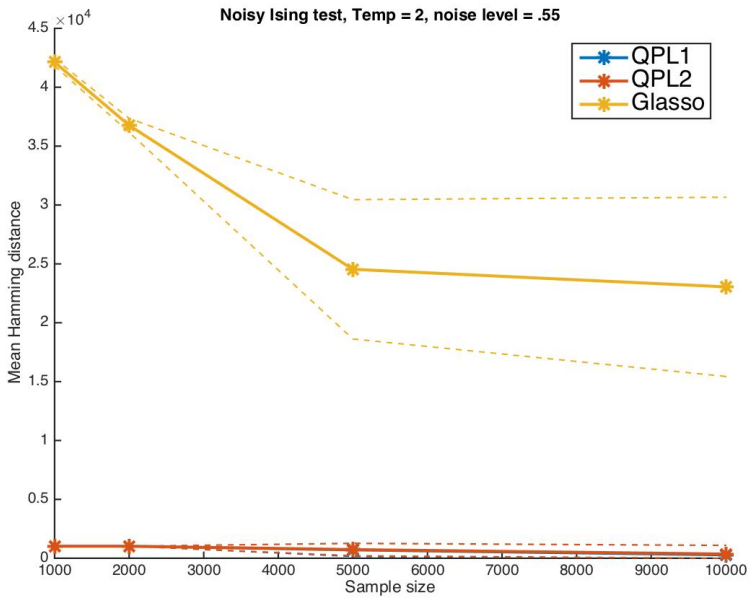

Figure B.6 – QPLs & Glasso: Temp=2.

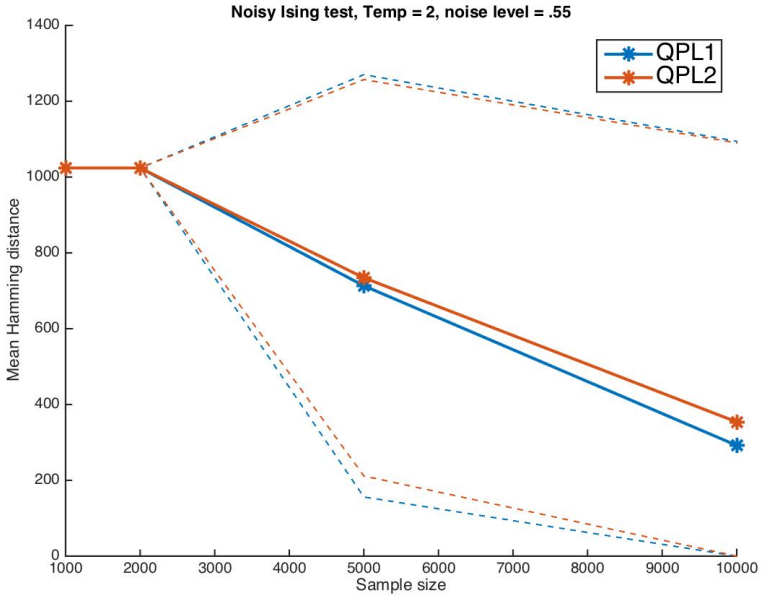

Figure B.7 – QPL1 & 2 : Temp=2.

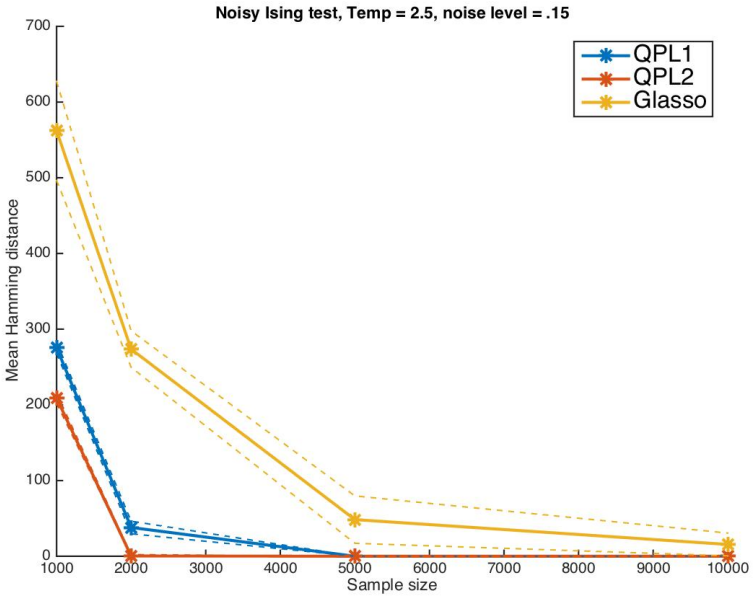

Figure B.8 – QPLs & Glasso: Temp=2.5.

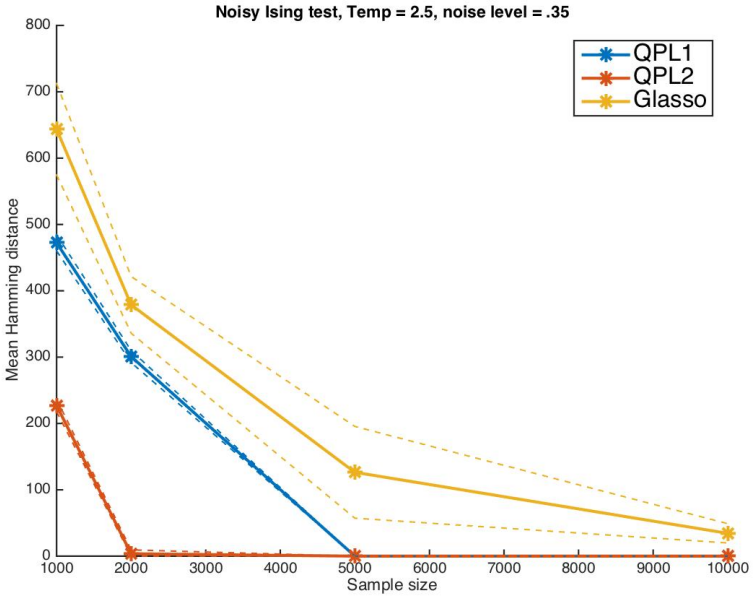

Figure B.9 – QPLs & Glasso: Temp=2.5.

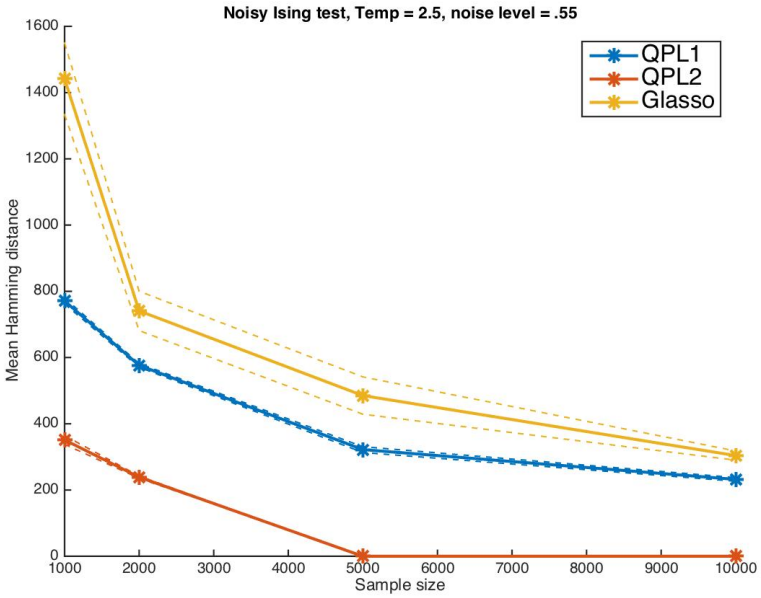

Figure B.10 – QPLs & Glasso: Temp=2.5.

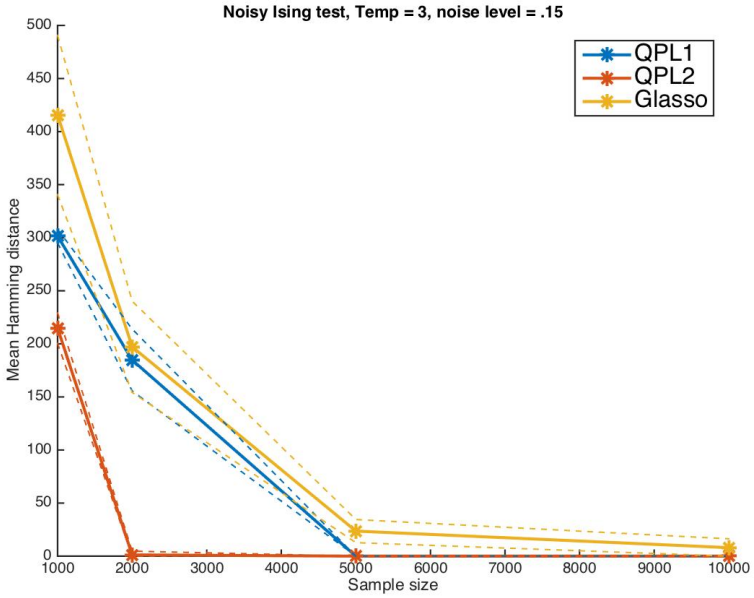

Figure B.11 – QPLs & Glasso: Temp=3.

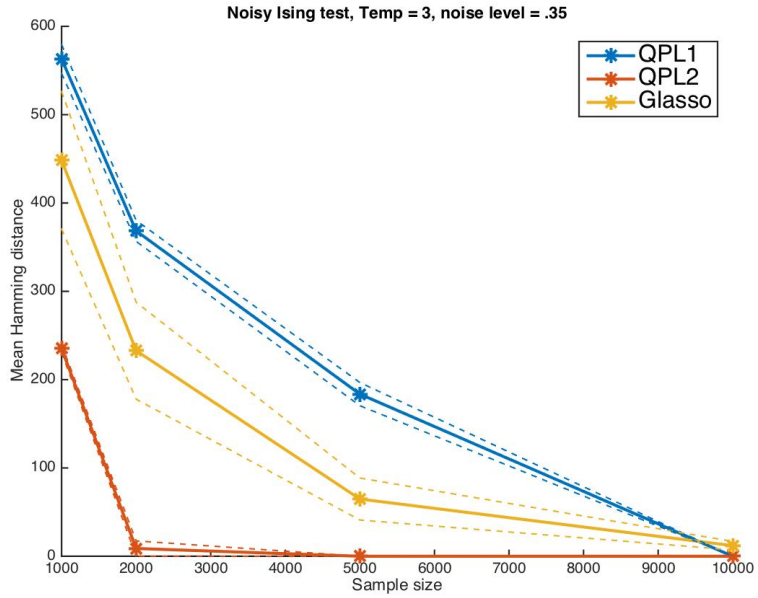

Figure B.12 – QPLs & Glasso: Temp=3.

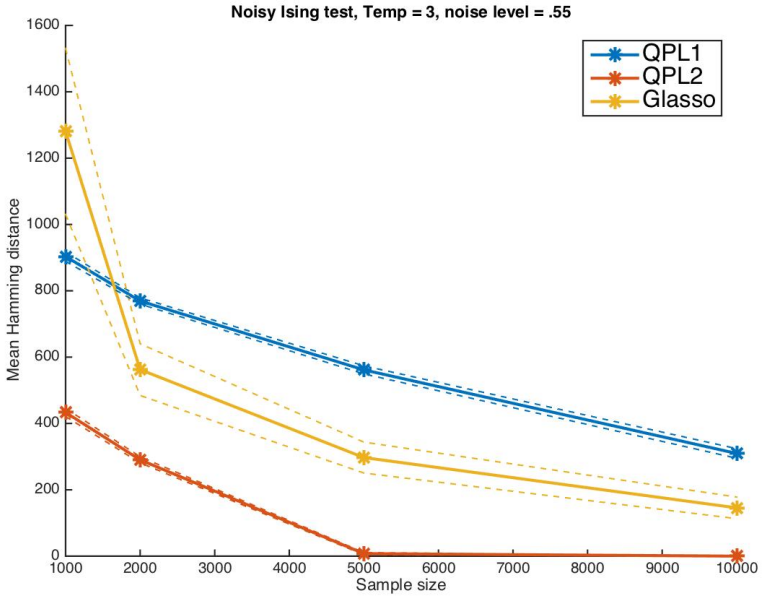

Figure B.13 – QPLs & Glasso: Temp=3.

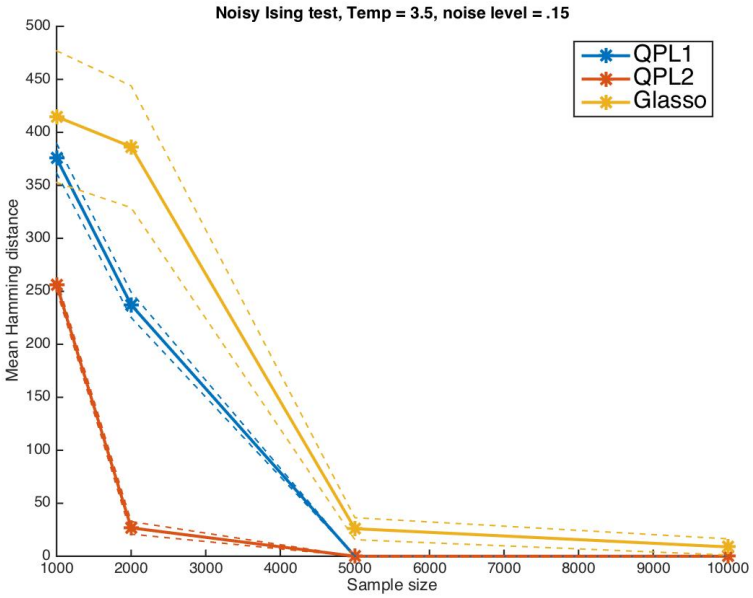

Figure B.14 – QPLs & Glasso: Temp=3.5.

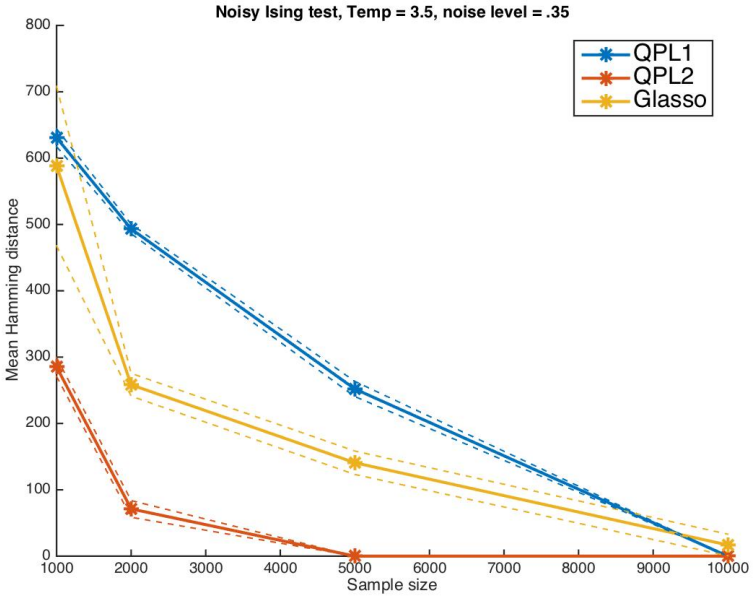

Figure B.15 – QPLs & Glasso: Temp=3.5.

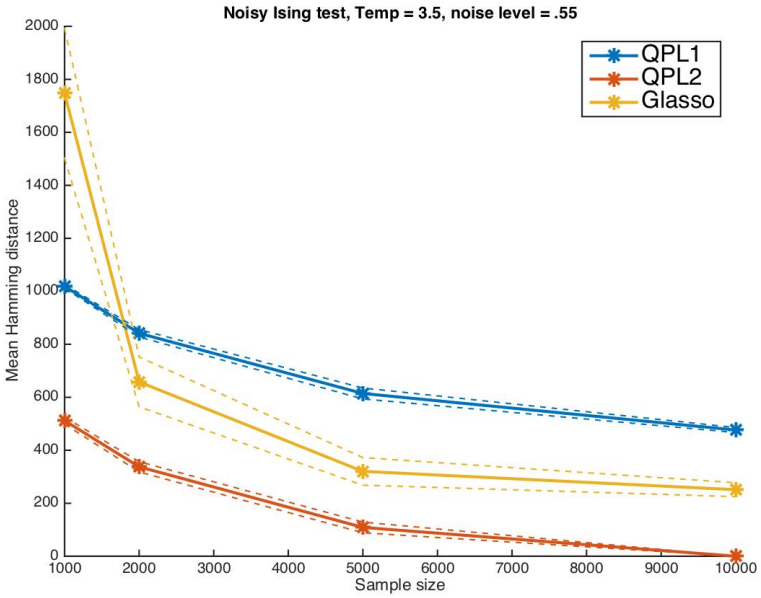

Figure B.16 – QPLs & Glasso: Temp=3.5.

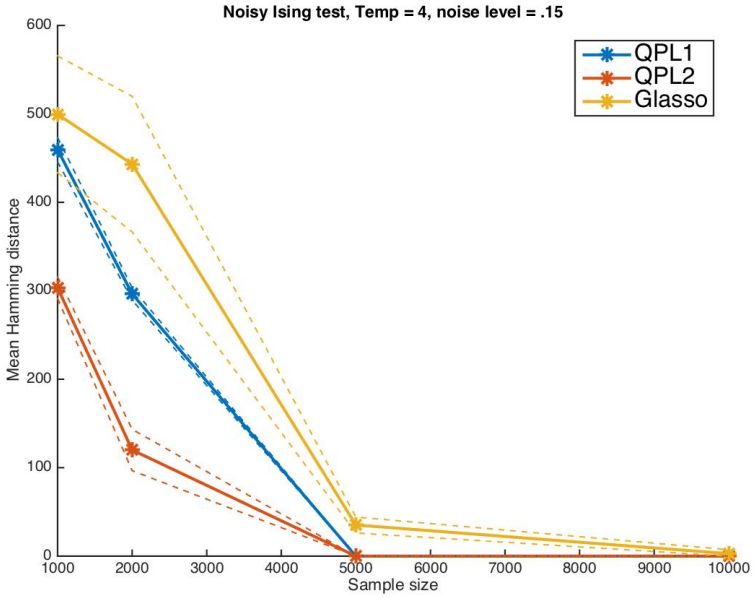

Figure B.17 – QPLs & Glasso: Temp=4.

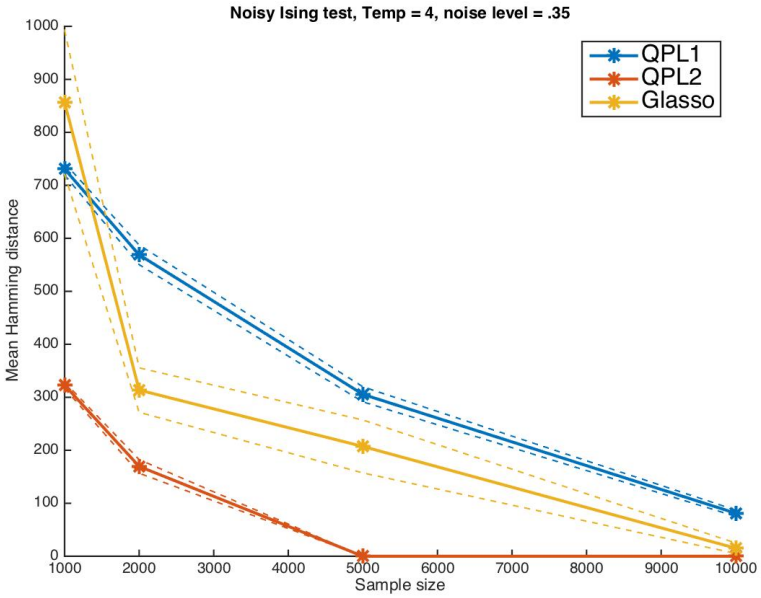

Figure B.18 – QPLs & Glasso: Temp=4.

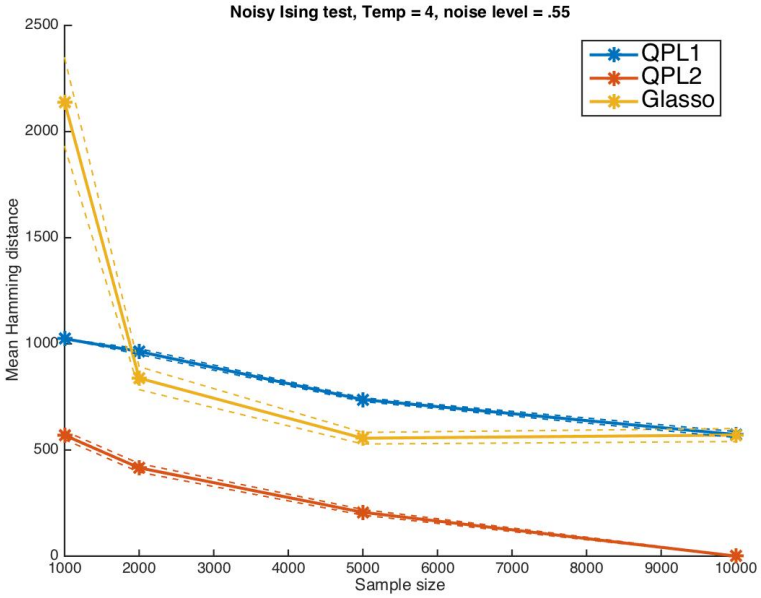

Figure B.19 – QPLs & Glasso: Temp=4.

Table B.1 – Mean Hamming distances & standard deviations, T= 2; Note: Glasso results in this table have to be multiplied by 10^3.

QPL1	873	731	159	61	1024	1024	358	282	1024	1024	713	292
QPL2	462	301	50	0	480	330	96	0	1024	1024	734	354
Glasso	33.2	26.1	17.5	18.8	38.4	32.4	21.2	21.3	42.1	36.8	24.5	23.0
QPL1	12.2	13.2	229.3	136.0	0	0	496.3	398.1	0	0	284.2	409.4
QPL2	19.4	13.7	9.0	0	18.6	12.4	9.3	0	0	0	266.8	375.6
Glasso	0.32	0.24	1.80	5.05	0.60	0.72	2.10	5.55	0.156	0.30	3.02	3.88

Table B.2 – Mean Hamming distances & standard deviations, T= 2.5

QPL1	276	38	0	0	472	301	0	0	771	576	322	232
QPL2	209	0	0	0	227	4	0	0	351	239	0	0
Glasso	562	274	48	16	644	379	126	34	1442	741	485	304
QPL1	3.8	4.2	0	0	6.2	4.6	0	0	4.6	3.0	4.5	2.8
QPL2	3.3	0.9	0	0	4.8	3.0	0	0	6.7	1.8	0	0
Glasso	33.4	12.3	16.0	7.7	35.1	21.8	35.3	7.4	55.3	30.3	28.8	7.1

Table B.3 – Mean Hamming distances & standard deviations, T= 3

QPL1	302	185	0	0	563	368	184	0	903	769	562	310
QPL2	215	1	0	0	235	9	0	0	433	290	8	0
Glasso	416	197	24	8	448	233	65	12	1281	562	297	146
QPL1	3.3	14.9	0	0	8.3	6.1	6.7	0	7.3	4.8	6.2	7.4
QPL2	7.3	1.8	0	0	2.3	4.4	0	0	6.3	4.8	1.7	0
Glasso	38.3	21.9	5.5	4.2	39.7	28.0	12.1	2.4	127.7	39.8	23.8	16.6

Table B.4 – Mean Hamming distances & standard deviations, T= 3.5

QPL1	376	237	0	0	631	493	252	0	1019	842	614	476
QPL2	256	27	0	0	285	71	0	0	512	338	109	0
Glasso	415	386	26	9	588	258	141	17	1747	658	320	251
QPL1	7.1	6.1	0	0	6.9	3.6	5.8	0	3.6	7.1	10.5	4.8
QPL2	2.6	3.0	0	0	7.3	6.4	0	0	6.6	9.8	10.3	0
Glasso	31.7	29.3	5.3	3.9	61.3	8.8	9.0	8.3	124.9	47.9	26.6	13.4

Table B.5 – Mean Hamming distances & standard deviations, T= 4

QPL1	460	296	0	0	731	569	306	81	1024	964	737	573
QPL2	304	120	0	0	323	170	0	0	570	416	206	0
Glasso	500	443	35	3	857	314	207	15	2138	838	556	570
QPL1	7.0	3.6	0	0	5.8	9.4	7.3	3.0	0	6.9	4.6	7.2
QPL2	6.2	11.9	0	0	3.0	6.5	0	0	10.2	10.4	6.8	0.9
Glasso	33.7	39.1	4.6	2.3	69.3	21.4	25.5	5.0	106.7	27.6	14.0	15.6

B.2 Second test setup results

Results for Sherrington-Kirkpatrick models with varying temperature.

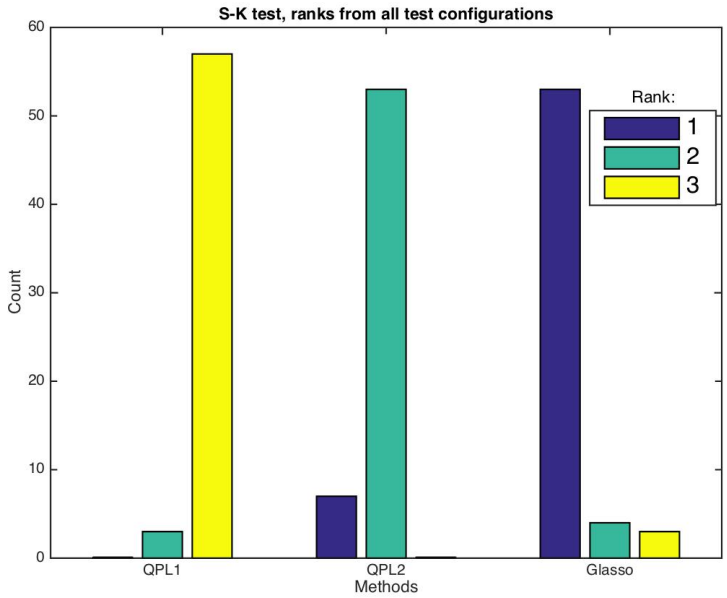

Figure B.20 – QPLs & Glasso: Number of different rankings for each method in test 2.

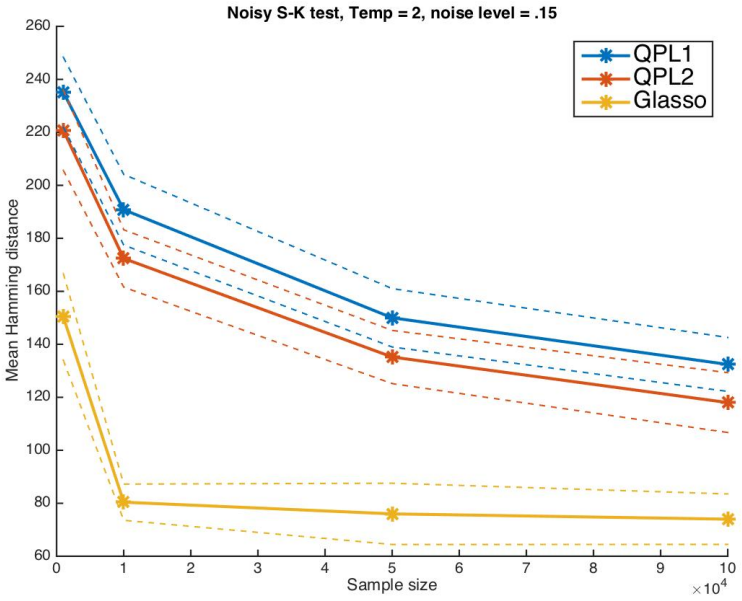

Figure B.21 – QPLs & Glasso: Temp=2.

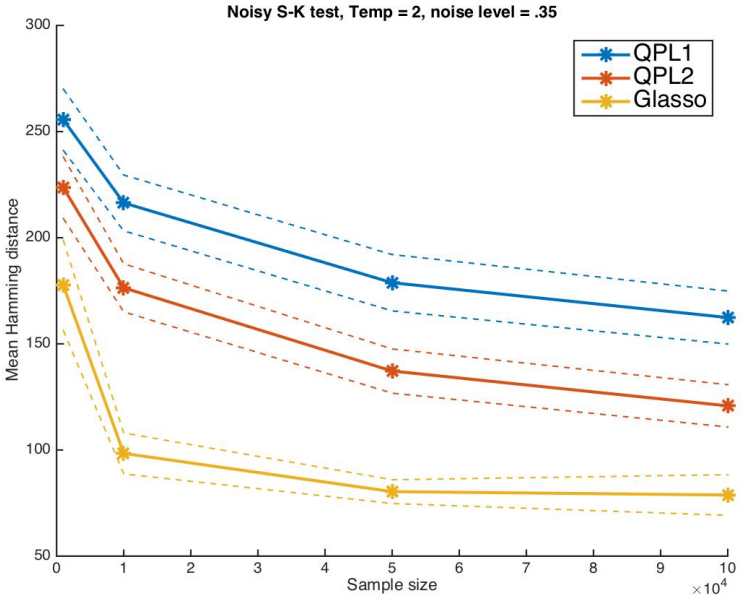

Figure B.22 – QPLs & Glasso: Temp=2.

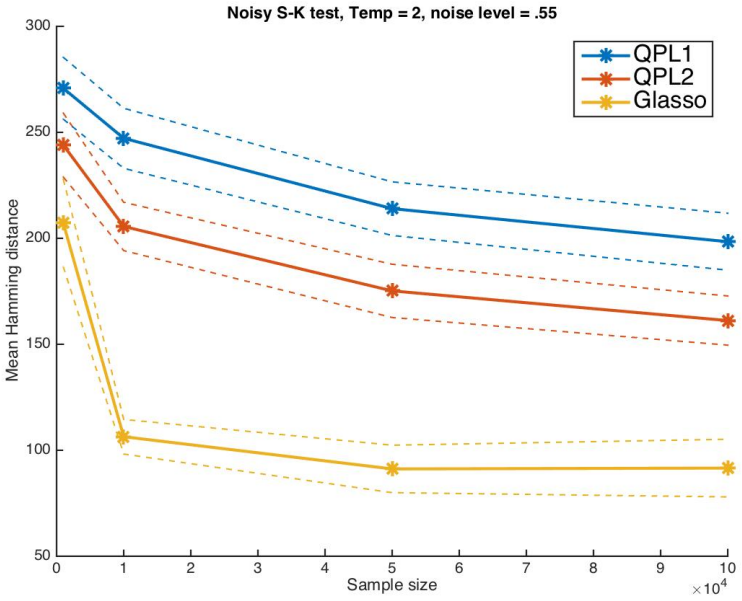

Figure B.23 – QPLs & Glasso: Temp=2.

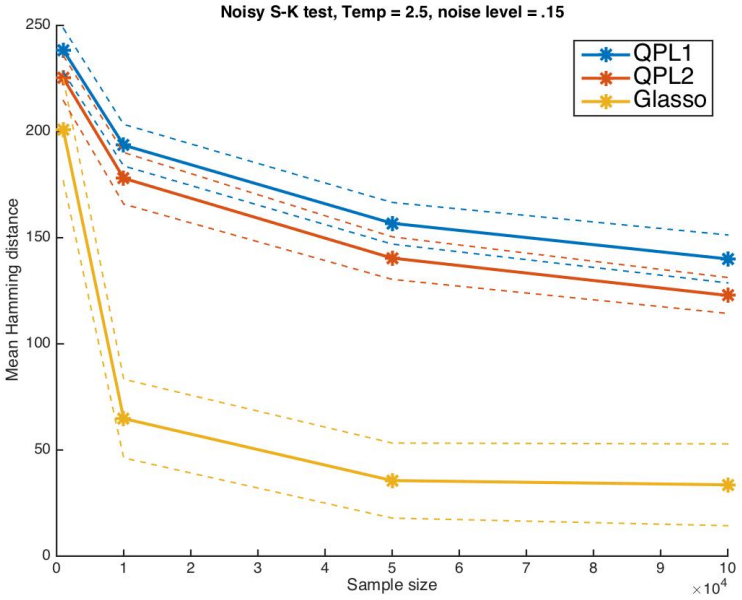

Figure B.24 – QPLs & Glasso: Temp=2.5.

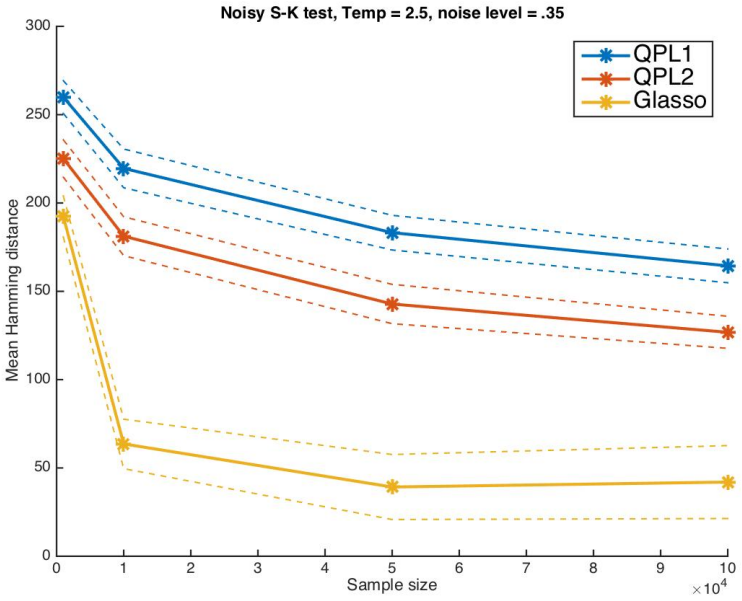

Figure B.25 – QPLs & Glasso: Temp=2.5.

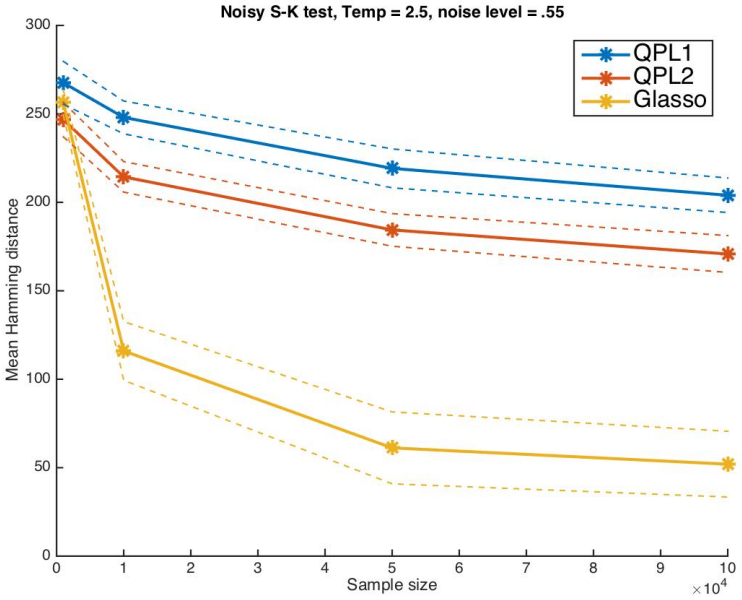

Figure B.26 – QPLs & Glasso: Temp=2.5.

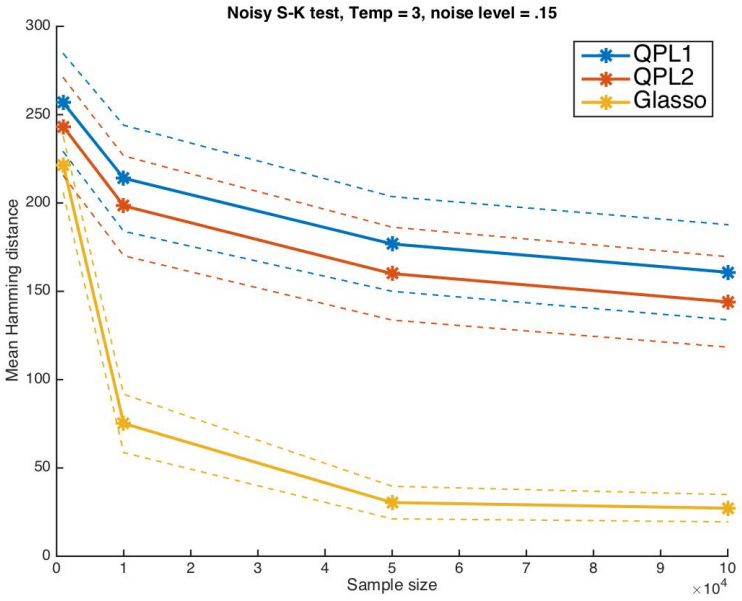

Figure B.27 – QPLs & Glasso: Temp=3.

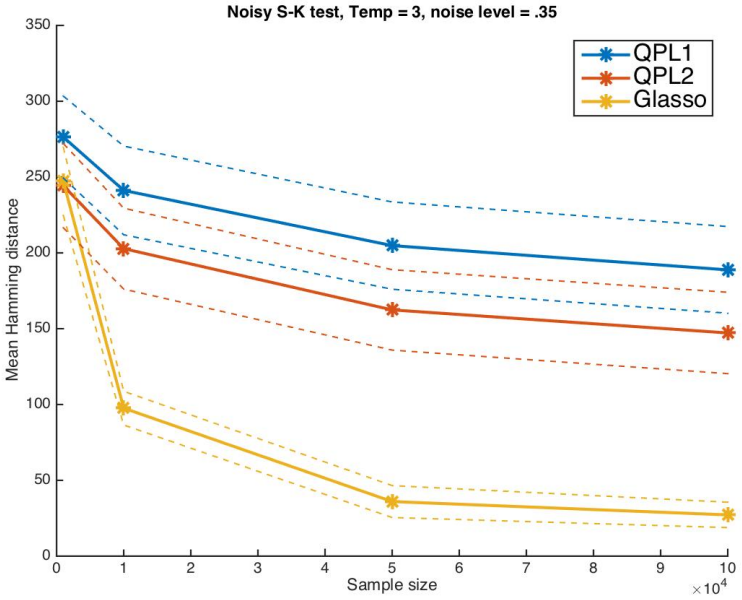

Figure B.28 – QPLs & Glasso: Temp=3.

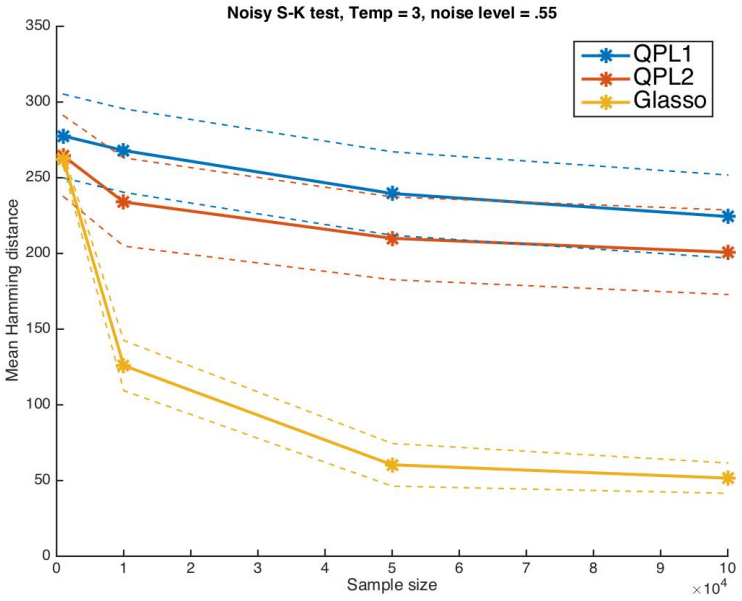

Figure B.29 – QPLs & Glasso: Temp=3.

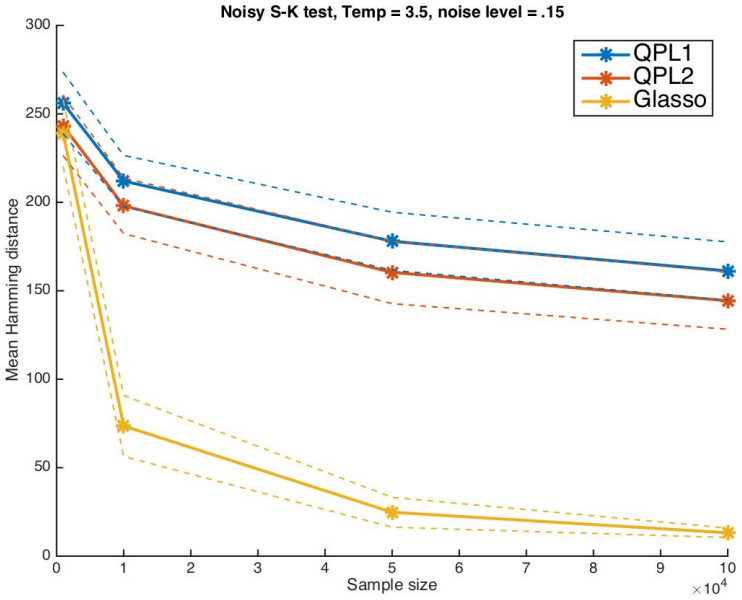

Figure B.30 – QPLs & Glasso: Temp=3.5.

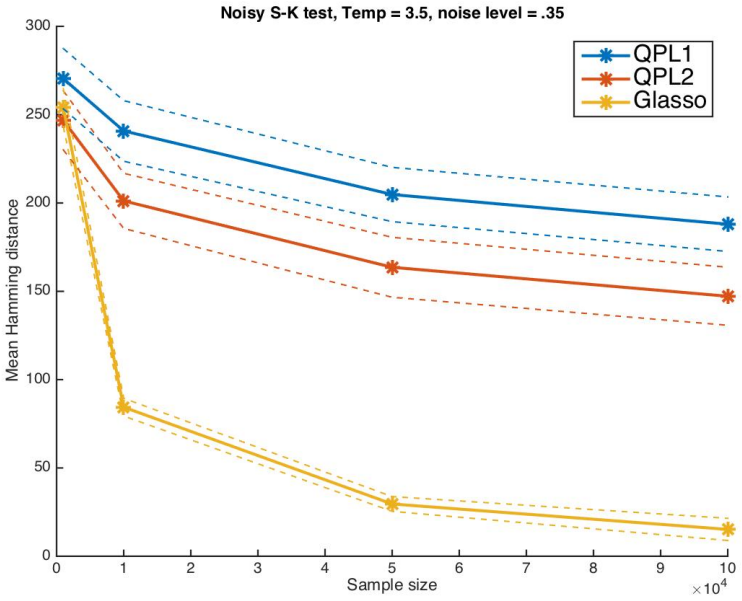

Figure B.31 – QPLs & Glasso: Temp=3.5.

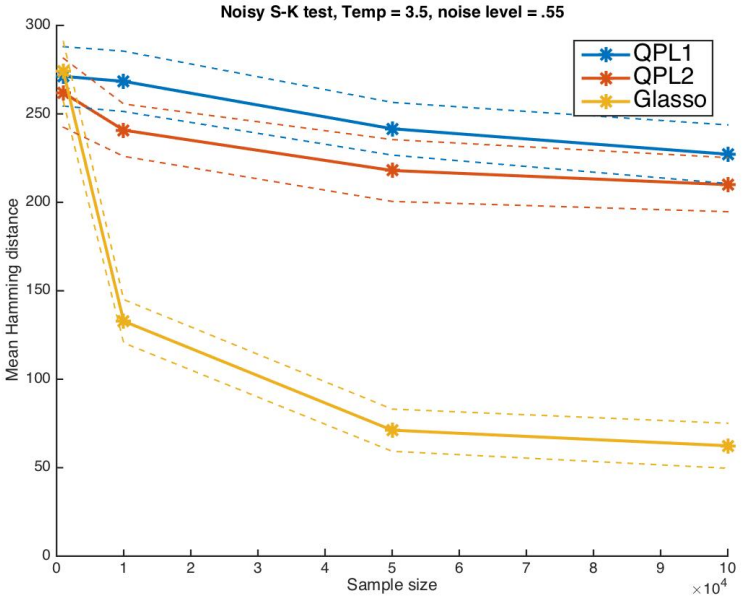

Figure B.32 – QPLs & Glasso: Temp=3.5.

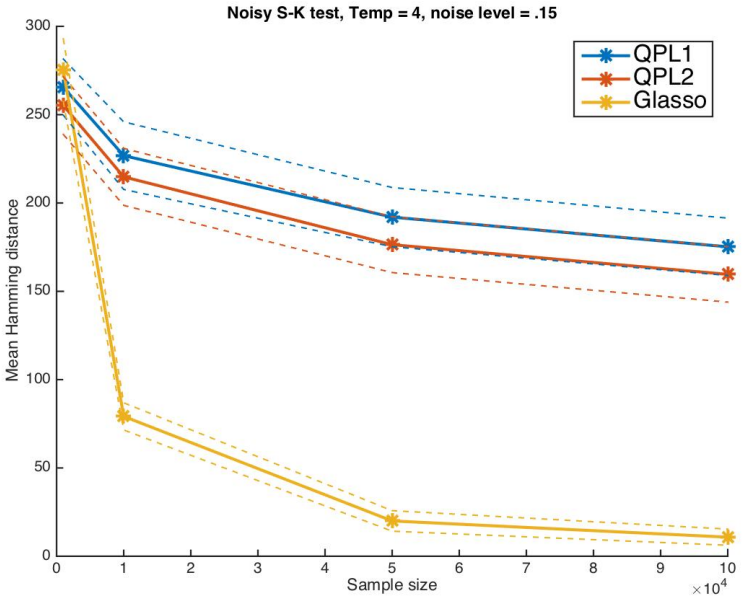

Figure B.33 – QPLs & Glasso: Temp=4.

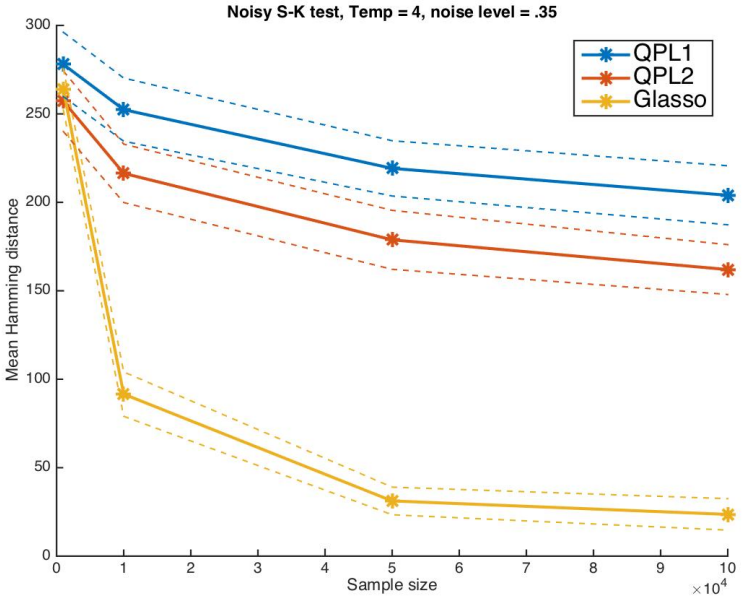

Figure B.34 – QPLs & Glasso: Temp=4.

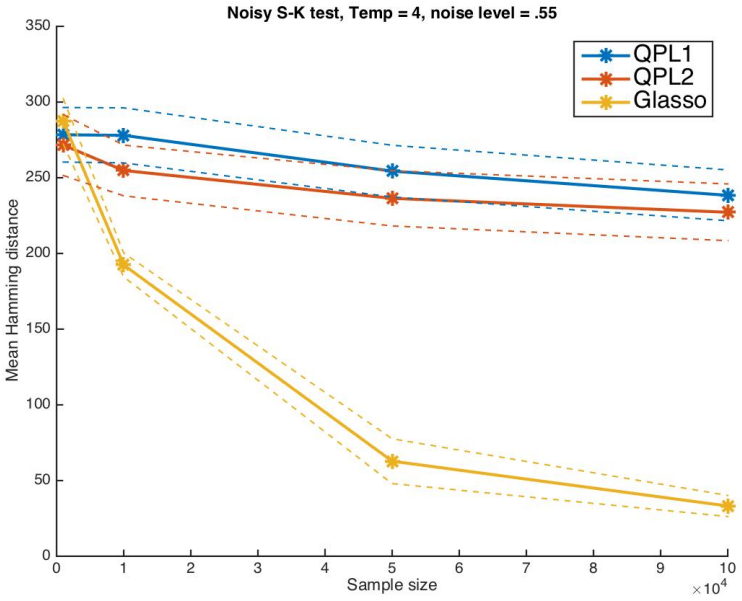

Figure B.35 – QPLs & Glasso: Temp=4.

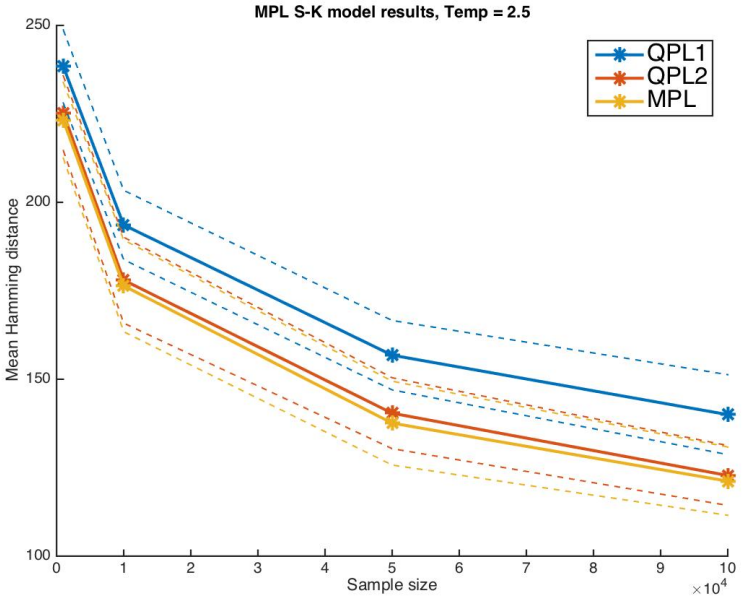

Figure B.36 – QPLs & MPL: Temp=2.5, noise level = 0.15.

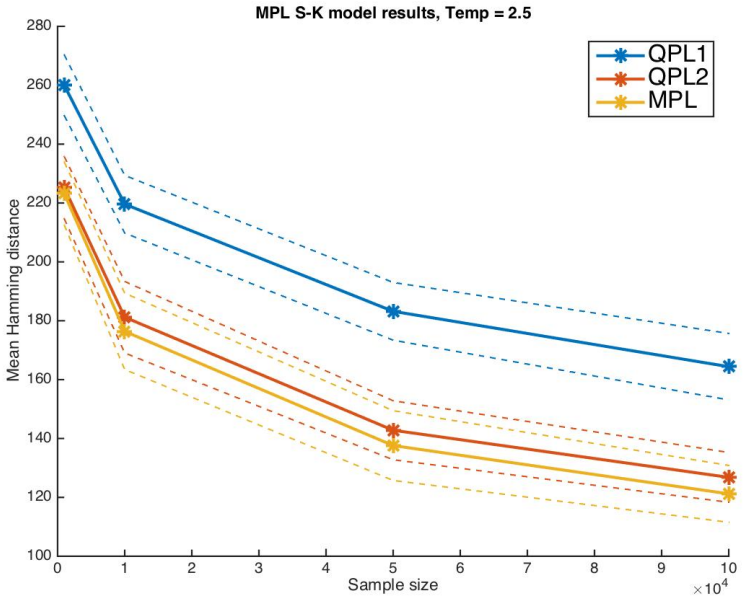

Figure B.37 – QPLs & MPL: Temp=2.5, noise level = 0.35.

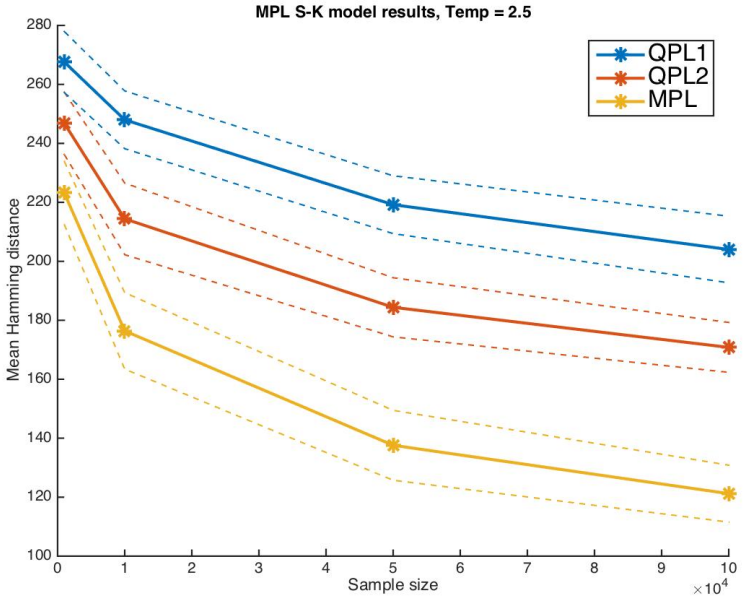

Figure B.38 – QPLs & MPL: Temp=2.5, noise level = 0.55.

Table B.6 – Mean Hamming distances & standard deviations, noisy S-K model, T= 2

QPL1	235.2	190.8	150.0	132.4	255.6	216.4	178.8	162.4	270.8	247.2	214.0	198.4
QPL2	220.8	172.4	135.2	118.0	223.6	176.4	137.2	120.8	244.0	205.6	175.2	161.2
Glasso	150.4	80.4	76.0	74.0	177.6	98.4	80.4	78.8	207.2	106.4	91.2	91.6
QPL1	15.1	15.2	12.6	11.6	16.5	15.0	15.1	14.2	16.6	16.2	14.4	15.3
QPL2	17.2	12.4	11.5	12.9	16.5	12.9	11.9	11.4	17.2	13.0	14.3	13.2
Glasso	18.7	7.8	13.2	10.9	24.2	11.0	6.4	10.9	23.6	9.3	12.8	15.5

Table B.7 – Mean Hamming distances & standard deviations, noisy S-K model, T= 2.5

QPL1	238.4	193.6	156.8	140.0	260.0	219.6	183.2	164.4	267.6	248.0	219.2	204.0
QPL2	225.2	178.0	140.4	122.8	225.2	181.2	142.8	126.8	246.8	214.4	184.4	170.8
Glasso	200.8	64.8	35.6	33.6	192.4	63.6	39.2	42.0	256.4	116.0	61.2	52.0
QPL1	11.8	11.2	11.2	12.9	10.6	12.5	11.2	10.9	13.8	10.6	12.5	11.1
QPL2	12.0	13.9	11.4	9.7	12.0	12.5	12.7	10.4	11.1	9.8	10.5	11.9
Glasso	27.4	21.1	20.2	22.0	13.2	16.0	21.0	23.5	7.9	18.9	23.2	21.2

Table B.8 – Mean Hamming distances & standard deviations, noisy S-K model, T= 3

QPL1	256.8	214.0	176.8	160.8	276.4	241.2	204.8	188.8	277.6	268.0	239.6	224.4
QPL2	243.2	198.4	160.0	144.0	244.4	202.8	162.4	147.2	264.4	234.0	210.0	200.8
Glasso	221.6	75.2	30.4	27.2	247.2	97.6	36.0	27.2	262.0	126.0	60.4	51.6
QPL1	31.7	34.3	30.6	30.7	30.8	33.4	32.8	32.6	31.5	31.5	31.4	31.3
QPL2	31.6	32.2	30.0	29.2	31.7	30.4	30.2	30.6	30.5	33.3	31.2	31.8
Glasso	18.2	18.8	10.5	8.8	25.6	12.7	12.0	9.5	5.1	18.9	16.1	11.4

Table B.9 – Mean Hamming distances & standard deviations, noisy S-K model, T= 3.5

QPL1	256.0	212.0	178.0	161.2	270.4	240.8	204.8	188.0	271.2	268.4	241.6	227.2
QPL2	243.2	198.0	160.4	144.4	246.8	201.2	163.6	147.2	262.0	240.8	218.0	210.0
Glasso	238.8	73.6	24.8	13.2	254.0	84.4	29.6	15.2	274.0	132.8	71.2	62.4
QPL1	19.8	16.6	18.7	18.7	19.5	19.6	17.5	17.6	19.2	19.5	17.0	18.9
QPL2	19.4	17.9	20.2	18.4	19.1	17.8	19.4	18.7	22.3	16.9	19.9	17.4
Glasso	21.3	19.7	9.7	3.0	12.2	5.7	4.8	7.2	19.3	14.0	13.6	14.5

Table B.10 – Mean Hamming distances & standard deviations, noisy S-K model, T= 4

QPL1	265.6	226.8	192.0	175.2	278.0	252.4	219.2	204.0	278.4	278.0	254.4	238.4
QPL2	255.2	214.8	176.4	159.6	257.2	216.4	178.8	162.0	271.6	254.8	236.4	227.2
Glasso	275.2	79.2	20.0	10.8	264.0	91.6	31.2	23.6	287.2	192.4	62.8	33.2
QPL1	18.1	21.8	19.1	18.5	20.7	20.4	17.8	19.0	20.6	20.7	19.5	19.2
QPL2	18.6	18.4	18.0	17.9	19.5	18.8	19.0	16.1	23.0	19.1	20.8	21.4
Glasso	20.3	8.8	6.6	5.2	13.3	14.3	8.9	10.1	17.1	9.0	16.9	7.9

B.3 Third test setup results

Results for fixed 2D Ising topology, random interactions model with temperature $T = 3$.

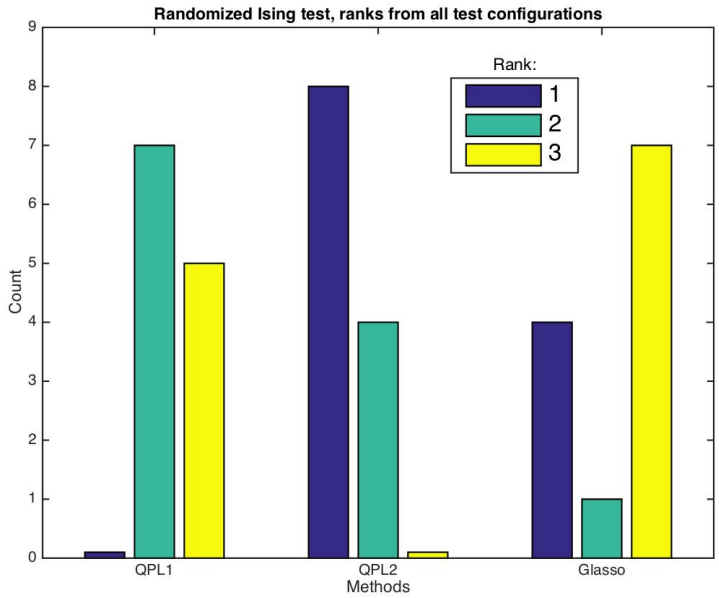

Figure B.39 – QPLs & Glasso: Number of different rankings for each method in test 3.

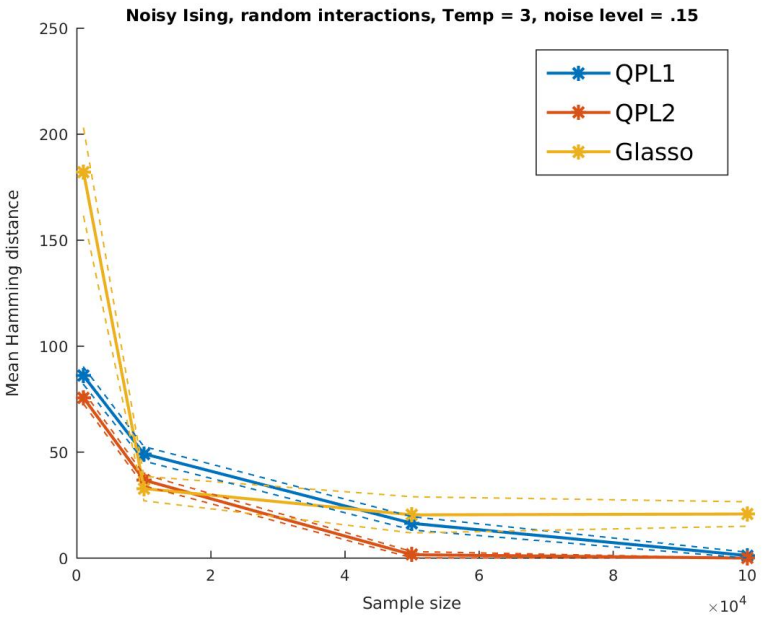

Figure B.40 – QPLs & Glasso: Temp=3.

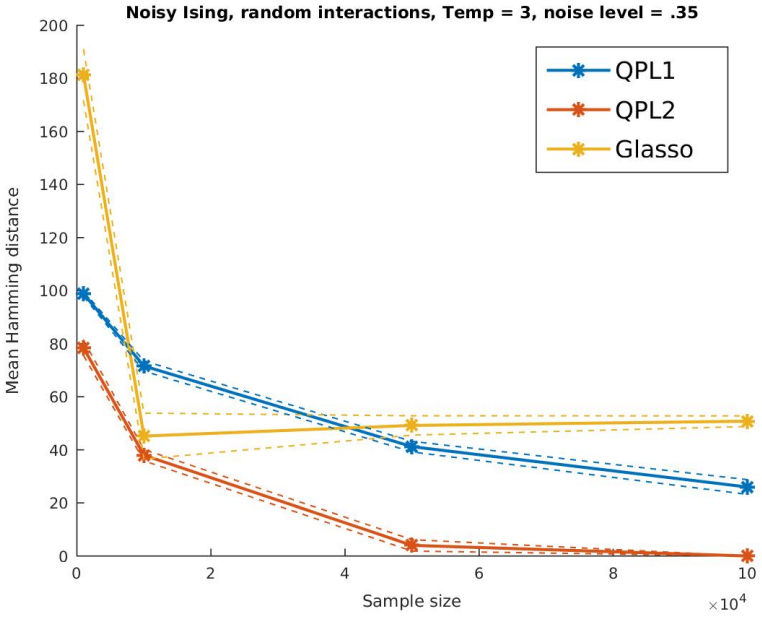

Figure B.41 – QPLs & Glasso: Temp=3.

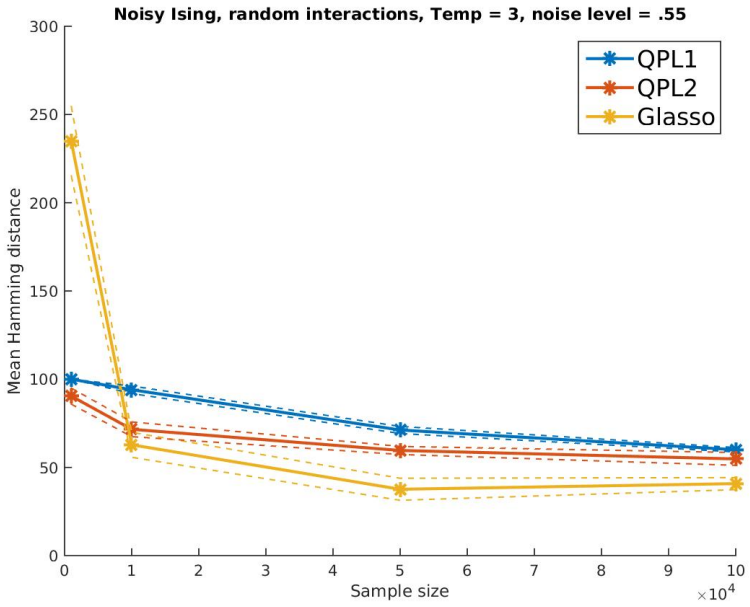

Figure B.42 – QPLs & Glasso: Temp=3.

Table B.11 – Mean Hamming distances & standard deviations, T= 3

QPL1	86.0	49.2	16.4	1.2	98.8	71.6	41.2	26.0	100.0	94.0	71.2	60.0
QPL2	75.6	36.8	1.6	0.0	78.4	38.0	4.0	0.0	90.4	71.6	59.6	54.8
Glasso	182.0	32.8	20.4	20.8	181.2	45.2	49.2	50.8	234.8	62.8	37.6	40.8
QPL1	4.7	3.9	3.6	1.8	1.1	2.2	2.3	3.2	0.0	2.4	2.3	1.4
QPL2	3.3	3.0	1.7	0.0	3.3	2.4	2.4	0.0	5.2	4.8	2.6	4.1
Glasso	23.7	6.6	9.7	6.6	11.0	9.9	4.1	2.3	22.4	8.1	7.1	3.9

B.4 Fourth test setup results

Results for fixed 2D Ising topology, random interactions model with temperature $T = 3$, varying prior (equivalent sample size) strength. The confidence intervals have been suppressed from the plots for clarity. Glasso results from test 3 have been added to ease comparisons.

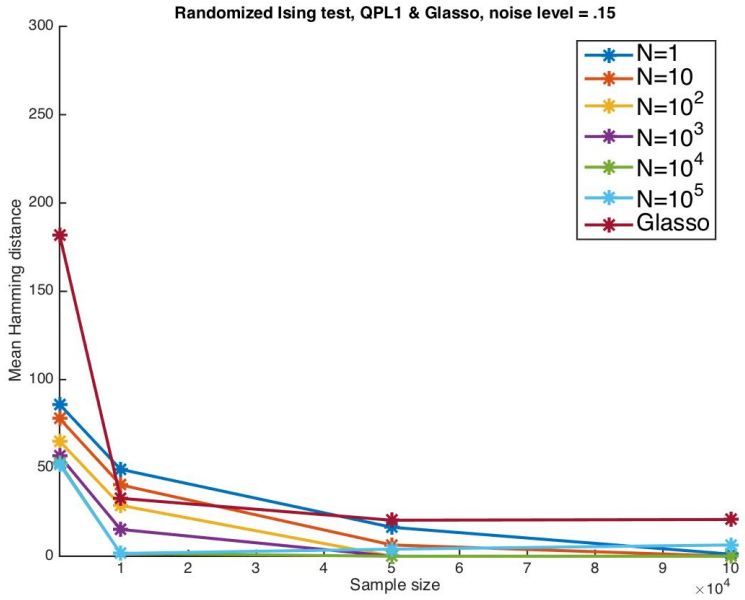

Figure B.43 – QPL1 with varying prior strength & Glasso.

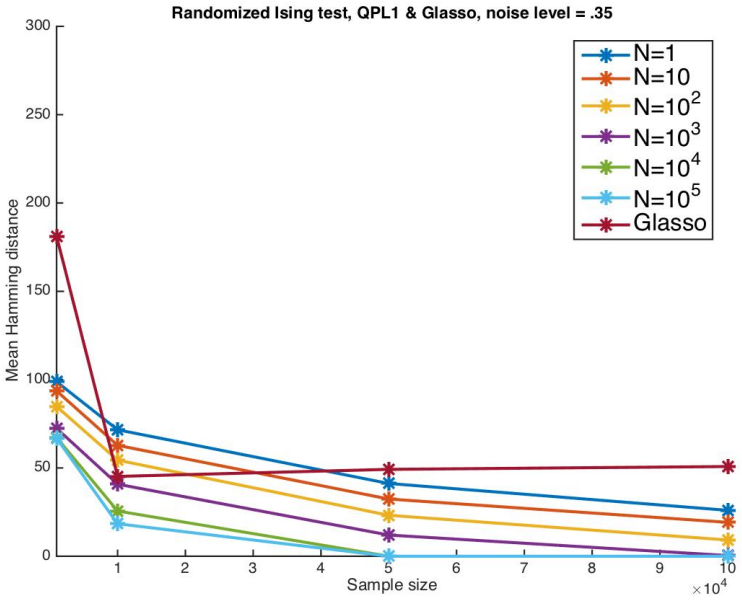

Figure B.44 – QPL1 with varying prior strength & Glasso.

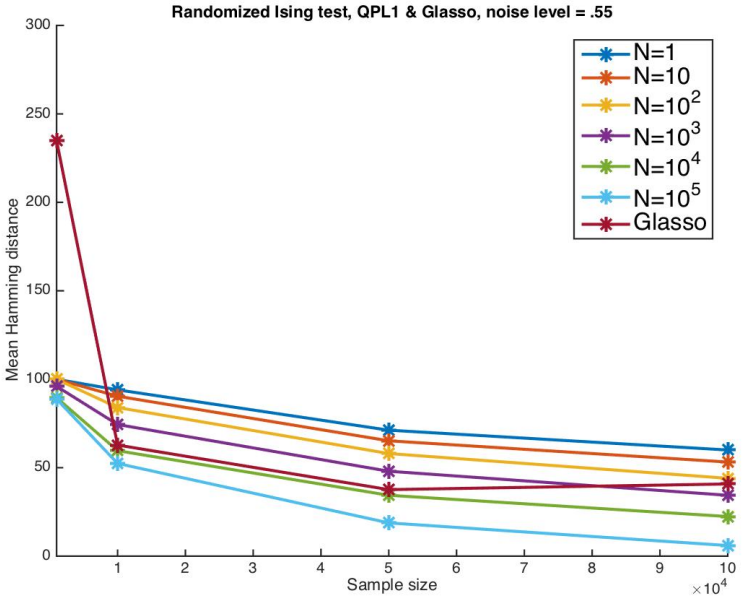

Figure B.45 – QPL1 with varying prior strength & Glasso.

Table B.12 – Mean Hamming distances & standard deviations, QPL1, T=3

$N = 1$	86.0	49.2	16.4	1.2	98.8	71.6	41.2	26.0	100.0	94.0	71.2	60.0
$N = 10$	78.0	40.4	6.4	0.0	93.6	62.8	32.4	19.2	100.0	90.4	65.2	53.2
$N = 10^2$	65.2	28.8	0.0	0.0	84.8	54.4	23.2	9.2	100.0	84.0	58.0	44.0
$N = 10^3$	56.8	15.2	0.0	0.0	72.4	40.8	12.0	0.4	96.0	74.4	48.0	34.4
$N = 10^4$	52.8	1.6	0.0	0.0	67.2	25.6	0.0	0.0	89.6	59.6	34.4	22.4
$N = 10^5$	52.0	1.6	4.0	6.4	66.8	18.4	0.0	0.0	88.8	52.4	18.8	6.0
$N = 1$	4.7	3.9	3.6	1.8	1.1	2.2	2.3	3.2	0.0	2.4	2.3	1.4
$N = 10$	2.8	3.0	2.6	0.0	4.1	1.1	3.8	1.8	0.0	4.1	3.6	2.3
$N = 10^2$	2.3	2.3	0.0	0.0	4.1	2.6	1.1	3.0	0.0	2.4	1.4	3.2
$N = 10^3$	5.4	2.7	0.0	0.0	5.2	2.3	4.7	0.9	1.4	3.0	3.2	3.0
$N = 10^4$	7.6	2.6	0.0	0.0	4.1	3.3	0.0	0.0	3.3	4.3	3.6	2.2
$N = 10^5$	8.1	2.6	4.0	4.3	3.3	1.7	0.0	0.0	3.0	3.8	4.4	3.2

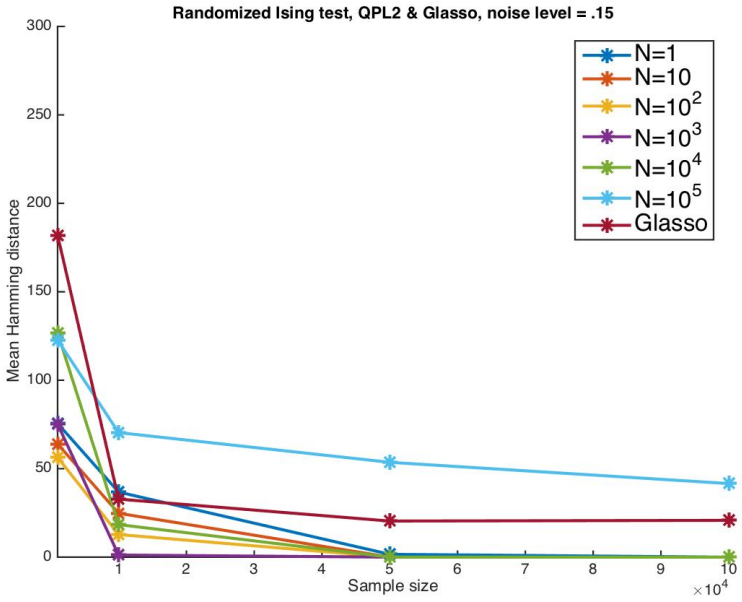

Figure B.46 – QPL2 with varying prior strength & Glasso.

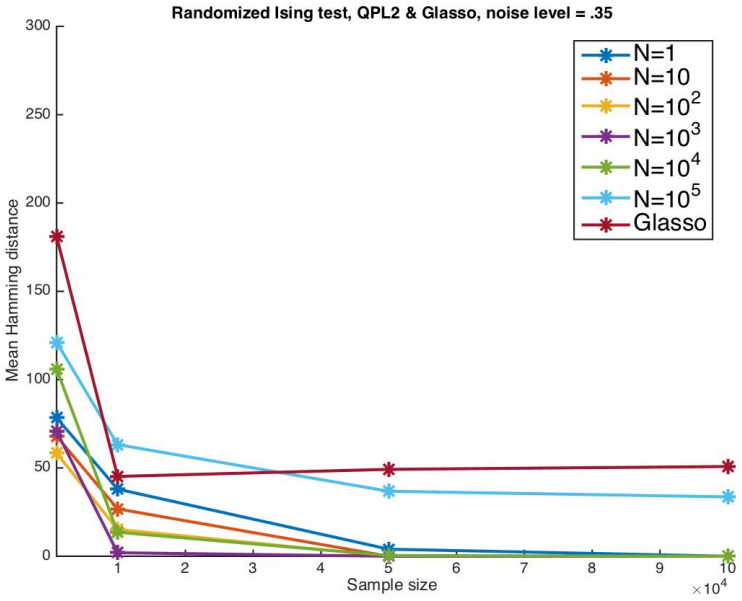

Figure B.47 – QPL2 varying prior strength & Glasso.

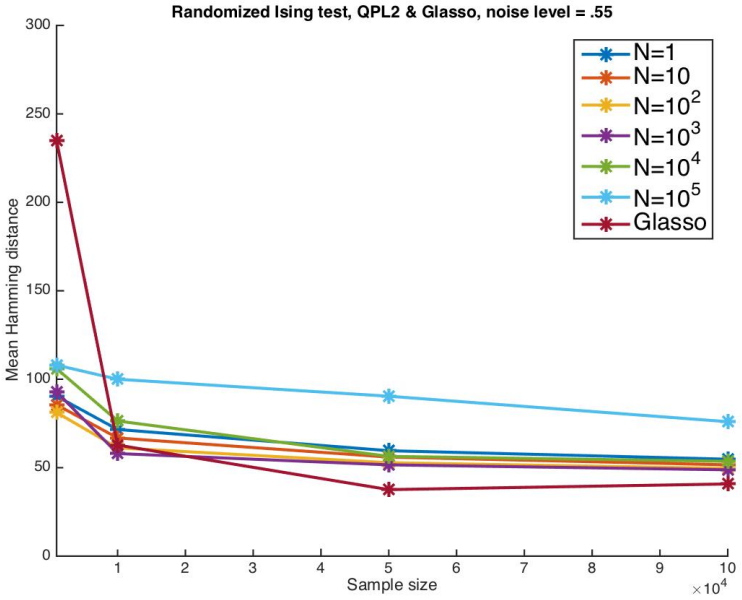

Figure B.48 – QPL2 varying prior strength & Glasso.

Table B.13 – Mean Hamming distances & standard deviations, QPL2, T=3

$N = 1$	75.6	36.8	1.6	0.0	78.4	38.0	4.0	0.0	90.4	71.6	59.6	54.8
$N = 10$	64.0	24.8	0.0	0.0	68.0	26.8	0.0	0.0	85.6	66.8	56.0	51.6
$N = 10^2$	56.4	12.8	0.0	0.0	58.4	15.2	0.0	0.0	81.2	61.2	52.8	49.6
$N = 10^3$	75.2	1.2	0.0	0.0	70.8	2.0	0.0	0.0	92.8	58.0	51.6	48.8
$N = 10^4$	126.8	18.4	0.0	0.0	106.0	13.6	0.4	0.0	106.0	76.4	56.4	53.6
$N = 10^5$	122.8	70.4	53.6	41.6	121.2	63.2	36.8	33.6	108.0	100.0	90.4	76.0
$N = 1$	3.3	3.0	1.7	0.0	3.3	2.4	2.4	0.0	5.2	4.8	2.6	4.1
$N = 10$	2.8	3.0	0.0	0.0	2.4	2.3	0.0	0.0	6.4	2.3	4.9	4.3
$N = 10^2$	4.3	3.3	0.0	0.0	5.2	1.8	0.0	0.0	6.7	3.0	3.9	3.6
$N = 10^3$	10.9	1.8	0.0	0.0	7.6	3.5	0.0	0.0	7.0	3.2	4.1	3.6
$N = 10^4$	5.4	3.8	0.0	0.0	8.9	4.3	0.9	0.0	6.6	6.8	4.3	4.1
$N = 10^5$	9.1	5.5	5.2	1.7	7.7	6.9	5.8	2.6	8.1	0.0	5.2	4.7

B.5 Fifth test setup results

Results for S-K model with temperature $T = 3$, varying prior (equivalent sample size) strength. The confidence intervals have been suppressed from the plots for clarity, although the standard deviations are noticeable. The corresponding Glasso results from test 2 have been added to the plots to ease comparisons.

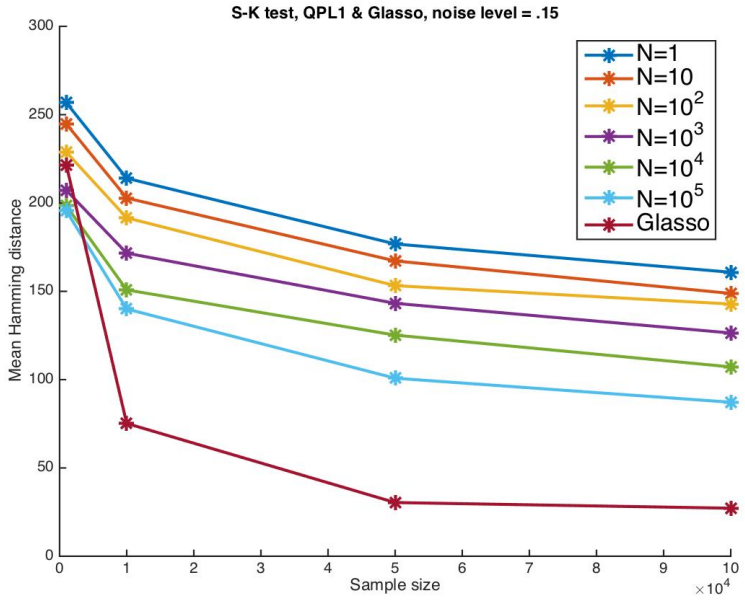

Figure B.49 – QPL1 with varying priors & corresponding Glasso, noise level = .15.

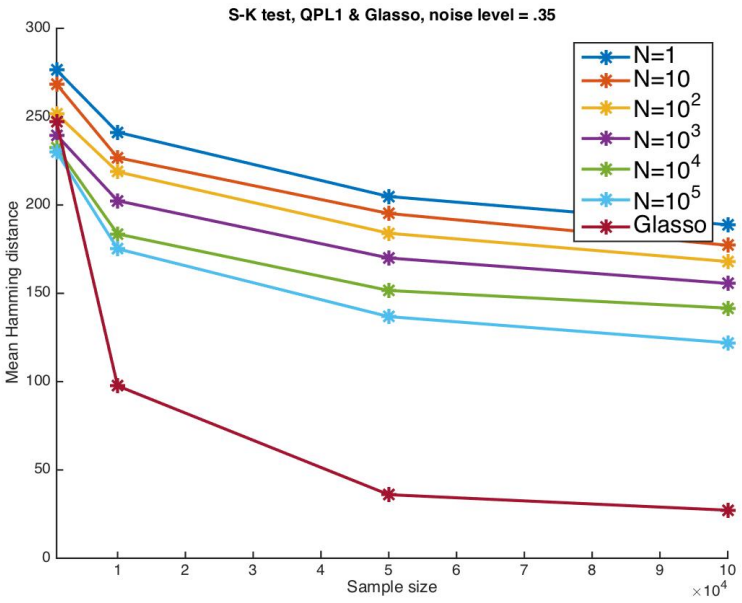

Figure B.50 – QPL1 with varying priors & corresponding Glasso, noise level = .35.

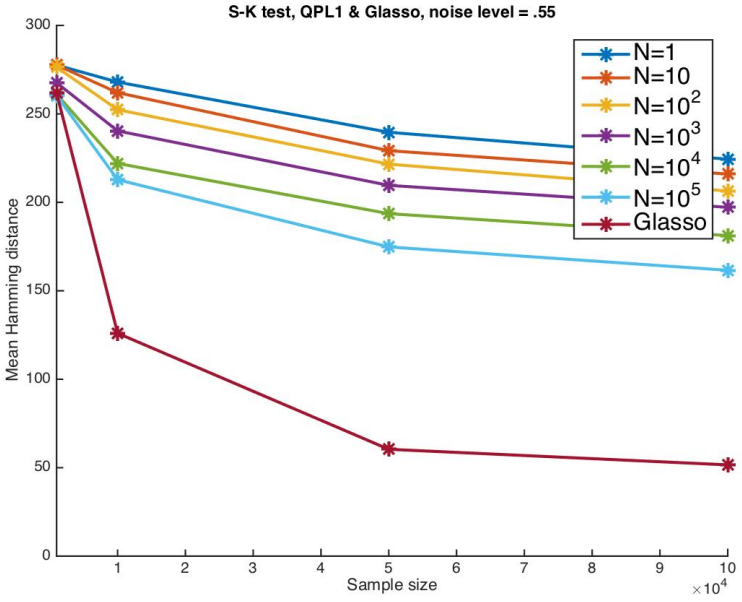

Figure B.51 – QPL1 with varying priors & corresponding Glasso, noise level = .55.

Table B.14 – Mean Hamming distances & standard deviations, QPL1, T=3

$N = 1$	256.8	214.0	176.8	160.8	276.4	241.2	204.8	188.8	277.6	268.0	239.6	224.4
$N = 10$	244.8	202.8	167.2	148.8	268.4	226.8	195.2	177.2	277.6	262.0	229.2	216.0
$N = 10^2$	228.8	191.6	153.2	142.8	251.6	218.8	184.0	168.0	276.4	252.4	221.6	206.4
$N = 10^3$	207.2	171.6	143.2	126.4	239.6	202.4	170.0	155.6	267.6	240.4	209.6	197.2
$N = 10^4$	198.4	150.8	125.2	107.2	232.4	183.6	151.6	141.6	261.2	222.0	193.6	181.2
$N = 10^5$	195.6	140.0	100.8	87.2	230.0	175.2	136.8	122.0	260.4	212.8	174.8	161.6
$N = 1$	31.7	34.3	30.6	30.7	30.8	33.4	32.8	32.6	31.5	31.5	31.4	31.3
$N = 10$	32.0	30.2	30.7	30.0	31.7	31.3	30.7	29.2	31.5	32.2	30.6	29.7
$N = 10^2$	31.9	30.8	30.7	30.7	31.9	30.5	30.5	30.5	31.5	29.9	32.0	31.9
$N = 10^3$	31.0	30.8	29.3	27.8	30.3	30.0	32.0	31.1	29.5	31.3	32.7	32.0
$N = 10^4$	29.3	30.4	29.1	27.9	31.1	30.3	29.2	28.7	30.2	33.5	32.1	30.6
$N = 10^5$	28.0	28.2	25.5	23.6	30.3	30.3	27.9	26.3	31.0	30.7	32.1	30.7

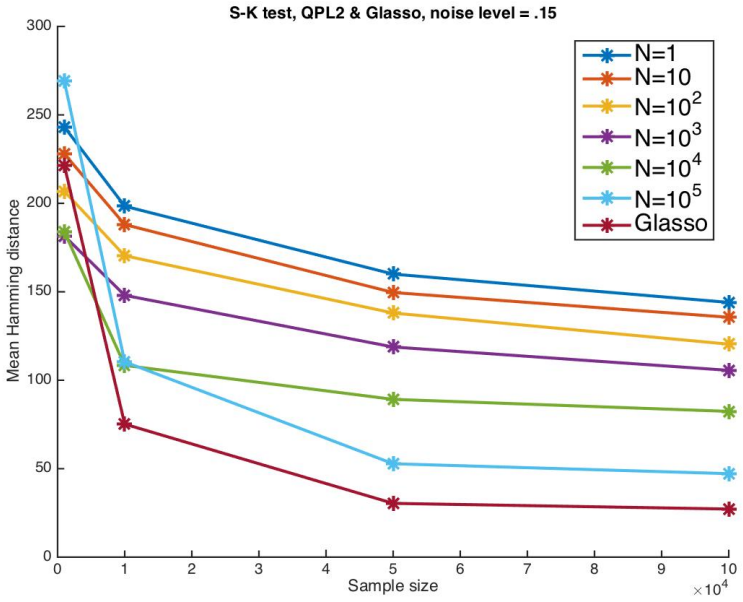

Figure B.52 – QPL2 with varying priors & corresponding Glasso, noise level = .15.

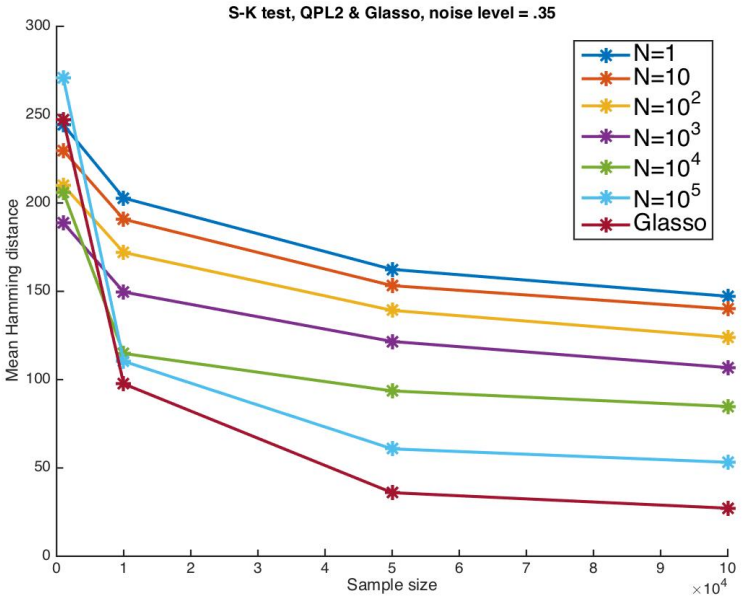

Figure B.53 – QPL2 with varying priors & corresponding Glasso, noise level = .35.

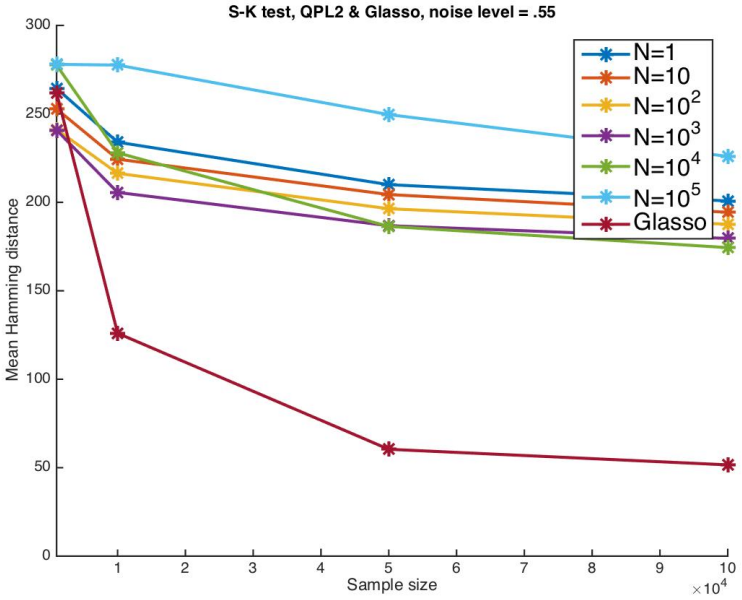

Figure B.54 – QPL2 with varying priors & corresponding Glasso, noise level = .55.

Table B.15 – Mean Hamming distances & standard deviations, QPL2, T=3

$N = 1$	243.2	198.4	160.0	144.0	244.4	202.8	162.4	147.2	264.4	234.0	210.0	200.8
$N = 10$	228.0	188.0	149.6	135.6	229.6	190.8	153.2	140.0	252.8	224.4	204.4	194.4
$N = 10^2$	206.8	170.4	138.0	120.4	210.0	172.0	139.2	124.0	241.2	216.4	196.4	187.6
$N = 10^3$	181.6	148.0	118.8	105.6	188.8	149.6	121.6	106.8	240.8	205.6	186.8	179.6
$N = 10^4$	184.0	108.4	89.2	82.4	206.0	114.8	93.6	84.8	277.6	228.0	186.4	174.4
$N = 10^5$	269.2	110.4	52.8	47.2	270.8	110.4	60.8	53.2	278.0	277.6	249.6	226.0
$N = 1$	31.6	32.2	30.0	29.2	31.7	30.4	30.2	30.6	30.5	33.3	31.2	31.8
$N = 10$	32.4	30.6	29.3	28.4	31.6	30.1	29.2	28.4	32.5	32.6	32.7	31.2
$N = 10^2$	31.6	30.4	29.2	28.4	32.2	31.2	27.7	27.7	34.0	31.9	31.9	30.5
$N = 10^3$	27.6	30.0	29.9	27.1	26.4	27.7	28.5	27.8	29.8	32.0	30.5	31.9
$N = 10^4$	22.5	25.0	25.5	24.3	27.6	27.3	26.2	25.0	33.1	31.3	31.2	28.5
$N = 10^5$	25.6	12.9	18.3	14.0	13.8	8.9	18.1	16.5	32.2	31.5	30.6	32.9

Bibliography

[1] Ash, Robert B.: *Basic Probability Theory.* Dover, 2008.

[2] Aurell, Erik & Magnus Ekeberg. Inverse Ising inference using all the data. In *arXiv:1107.3536v3*, 2012.

[3] Bernardo, José, M. & Adrian F. M. Smith: *Bayesian Theory,* fifth printing. Wiley, 1994.

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

[5] Buntine, W.: Theory Refinement on Bayesian Networks. In *Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence*, p.52-60. Morgan Kaufmann, 1991.

[6] Cormen, Thomas H.: Charles E. Leiserson; Ronald L. Rivest & Clifford Stein: *Introduction to Algorithms,* third edition. MIT Press, 2009.

[7] Darroch, J. N. , S. L. Lauritzen & T. P. Speed: Markov Fields and Log-Linear Interaction Models for Contingency Tables. In *Annals of Statistics*, Vol. 8, No. 3, p.522-539, 1980.

[8] DasGupta, Anirban: *Asymptotic Theory of Statistics and Probability.* Springer, 2008.

[9] Dikmen, Onur: Inferring Structures of Continuous Markov Random Fields using Quasi-Pseudolikelihood. Not yet published.

[10] Friedman, J., Hastie, T., and Tibshirani, R.: Sparse inverse covariance estimation with the graphical lasso. In *Biostatistics*, 9(3), p.432-441, 2008.

[11] Gibbs, J. Willard.: *Elementary Principles of Statistical Mechanics.* Charles Scribner's Sons, 1902.

[12] Givens, Geof H., Jennifer A. Hoeting: *Computational Statistics.* Wiley, 2012.

[13] Hastings, W.K.: Monte Carlo sampling methods using Markov chains and their applications. In *Biometrika*, 57, p.97-109, 1970.

[14] Heckerman, D.; D. Geiger & D. M. Chickering: Learning Bayesian networks: The combination of knowledge and statistical data. In *Machine Learning*, 20, p.197-243, 1995.

[15] Isaacson, Dean, L. & Richard W. Madsen: *Markov Chains: Theory and Applications.* Wiley, 1976.

[16] Koller, D. & N. Freedman: *Probabilistic Graphical Models: Principles and Techniques.* MIT Press, 2009.

[17] Lauritzen, S. L. & D. J. Spiegelhalter: Local computations with probabilities on graphical structures and their application to expert systems. In *Journal of the Royal Statistical Society, Series B*, Vol.50, No. 2, p.157-224, 1988.

[18] Lenz, Wilhelm: Beiträge zum Verständnis der magnetischen Eigenschaften in festen Körpern. In *Physikalische Zeitschrift*, 21, p.613-615, 1920.

[19] MacKay, David J. C.: *Information Theory, Inference, and Learning Algorithms.* Cambridge University Press, 2003.

[20] Marttinen, Pekka; Jing Tang, Bernard De Baets, Peter Dawyndt & Jukka Corander: Bayesian Clustering of Fuzzy Feature Vectors Using a Quasi-Likelihood Approach. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 31, No.1, p.74-85, 2009.

[21] McCullagh, P. & J. A. Nelder: *Generalized Linear Models.*, 2.ed., Chapman and Hall, 1989.

[22] Metropolis, N., A.W. Rosenbluth, M.N. Rosenbluth, A.H.Teller & E.Teller: Equation of state calculation by fast computing machines. In *Journal of Chemical Physics*, 21, p.1087-1091, 1953.

[23] Pearl, Judea: *Probabilistic Reasoning in Intelligent Systems.* Morgan Kauffman, 1988.

[24] Pensar, Johan; Henrik Nyman; Juha Niiranen & Jukka Corander 2014: Marginal Pseudo-Likelihood Learning of discrete Markov Network Structures. In *Bayesian analysis*, (in press) 2016. *ArXiv:1401:4988.*

[25] Ross, Sheldon M.: *Introduction to Probability Models*, 11th edition. Elsevier, 2003.

[26] Sherrington, David & Scott Kirkpatrick: Solvable Model of a Spin-Glass. In *Physical Review Letters*, Vol. 35, No. 26, p.1792-1796, 1975.

[27] Silander, T.; P. Kontkanen & P. Myllymäki. On sensitivity of the MAP Bayesian network structure to the equivalent sample size parameter. In R. Parr and L. van der Gaag, editors, *Proceedings of the 23rd Conference on Uncertainty in Artificial Intelligence*, p.360-367. UAI Press, 2007.

[28] Wedderburn, R. W. M.: Quasi-Likelihood Functions, Generalized Linear Models, and the Gauss-Newton Method. In *Biometrika*, Vol. 61, No. 3, p.439-447, 1974.

[29] Whittaker, Joe: *Graphical Models in Applied Multivariate Statistics*. John Wiley & sons, 1990.