Learning Non-parametric Markov Networks with Mutual Information

Janne Leppä-aho[1] JANNE.LEPPA-AHO@CS.HELSINKI.FI
Santeri Räisänen[2] J.S.RAISANEN@LSE.AC.UK
Xiao Yang[3] GRACEYX.SCUT@GMAIL.COM
Teemu Roos[1] TEEMU.ROOS@CS.HELSINKI.FI
[1]*University of Helsinki, Department of Computer Science / HIIT, Finland*
[2]*London School of Economics, Department of Philosophy, Logic and Scientific Method, UK*
[3]*Yale University, Department of Statistics and Data Science, USA*

Abstract

We propose a method for learning Markov network structures for continuous data without assuming any particular parametric distribution for the variables. The method makes use of previous work on a non-parametric estimator for mutual information which is used to create a non-parametric test for multivariate conditional independence. This independence test is then combined with an efficient constraint-based algorithm for learning the graph structure. The performance of the method is evaluated on several synthetic data sets and it is shown to learn more accurate structures than competing methods when the dependencies between the variables involve non-linearities.

Keywords: Graphical Models; Markov Networks; Structure Learning; Mutual Information.

1. Introduction

This paper addresses the problem of learning a Markov network structure from continuous data without assuming any particular parametric distribution. The large majority of the existing methods approach this problem by assuming that variables follow a multivariate normal distribution. This essentially reduces the problem of learning whether two variables are independent to deciding if they have a non-zero partial correlation. However, as the correlation measures only the strength of a linear dependence, the methods utilizing this might not be able to capture the dependence structure correctly when the relationships are non-linear or the data deviates from the multivariate Gaussian.

To remedy this, we opt to use conditional mutual information to measure the strength of association between the random variables. Like correlation, the mutual information equals zero for independent random variables which makes it possible to use it with Markov network structure learning algorithms based on independence testing, but unlike correlation, mutual information captures any kind of dependence and equals zero only if the variables are independent. In order to compute the mutual information without assumptions about the distributions of variables, we use the non-parametric estimators from previous work (Kozachenko and Leonenko, 1987; Kraskov et al., 2004; Vejmelka and Paluš, 2008) which are based on k-nearest neighbour statistics.

The literature on methods for non-parametric learning of Markov network structures in the continuous setting is scarce. Kernel methods provide tools for performing non-parametric tests of conditional independence, which makes these applicable to learning Markov networks through constraint based algorithms. One of the most commonly used tests is called Kernel-based Conditional Independence test (KCIT) (Zhang et al., 2011). However, this test becomes computationally demanding with large sample sizes as it scales cubically in the number of samples. Recently, Strobl et al. (2017) proposed two approximate versions of this test (randomized conditional independence test (RCIT) and other called randomized conditional correlation test) which both utilize random Fourier features to achieve linear scaling with the sample size. Also, Lafferty et al. (2012) proposed a method that does not make distributional assumptions but restricts the learned graph to be a forest. This forest graph is learned using Chow-Liu algorithm with pairwise mutual informations obtained using kernel density estimation.

One popular semiparametric approach to learning Markov networks is to assume that there exists univariate transformations for each variable after which the joint distribution of the transformed variables is multivariate normal. This then allows one to use all the machinery developed for Gaussian data. The resulting model class and the methods are termed *non-paranormal* or *Gaussian copulas* (Liu et al., 2009, 2012).

Our contribution is to provide a reliable way to learn Markov network structures based on a particular estimator of conditional mutual information without an exponential number of independence tests. Compared to the other approaches mentioned above, our method scales in the worst case quadratically in the sample size making it faster than the kernel-method (KCIT). Furthermore, its performance in our experiments is clearly superior compared to the approximate kernel-based and non-paranormal methods when the relationships between the variables involve non-linearities.

In Section 2, we will review how mutual information is estimated from continuous data based on k-nearest neighbour statistics, and how this estimator can be used for testing conditional independence. Section 3 goes through the constraint-based algorithm which we will use to learn the Markov network structures. In Section 4, we study the performance of our method with several synthetic data sets to illustrate the distinct behaviour of the proposed method especially when the data involves non-linearities.

2. Independence Testing Using Mutual Information

In this section we present the Kraskov estimator for mutual information and show how it can be used for independence testing.

2.1 Preliminaries

Let X and Y denote two continuous random variables with density functions f_X and f_Y. Mutual information (Cover and Thomas, 2006) measures the information that one random variable carries about the other and it can be expressed using entropies as

$$I(X;Y) = H(X) + H(Y) - H(X,Y), \tag{1}$$

where $H(\cdot)$ denotes (differential) entropy. Let Z be a random vector. We assume that Z has a joint density function denoted by f_Z. The conditional mutual information between X and Y given Z can be written as

$$I(X;Y \mid Z) = H(X,Z) + H(Y,Z) - H(X,Y,Z) - H(Z). \tag{2}$$

The mutual information equals zero iff the variables X and Y are independent. The same holds for the conditional mutual information:

$$I(X;Y \mid Z) = 0 \Leftrightarrow X \perp\!\!\!\perp Y \mid Z.$$

2.2 Estimating Mutual Information

Here, we review how the quantities $H(X), I(X, Y)$ and $I(X; Y \mid Z)$ can be estimated given the observed samples x_i, y_i and z_i, where $i = 1, \dots, n$. In this paper, x_i and y_i are scalars whereas z_i can be a vector.

The Kraskov estimator for mutual information builds on the previous entropy estimator by Kozachenko and Leonenko (1987). The derivation of this entropy estimator is presented in (Kraskov et al., 2004) and it starts from the definition of entropy of X, which can be interpreted as the expected value of the log-density, $-\log f_X$. This implies that if one has an unbiased estimator for $\log f_X$, then the unbiased estimate for entropy can be obtained as a sample average over local log-probability density estimates. Assuming that the probability density is constant in hyperspheres containing $(k-1)$-nearest neighbours of each data point, one arrives in the following formula:

$$\hat{H}(X) = \psi(n) - \psi(k) + \log c_d + \frac{d}{n} \sum_{i=1}^{n} \log \epsilon(i), \tag{3}$$

where $\epsilon(i)$ is twice the distance to the kth nearest neighbour of data point x_i, $\psi(\cdot)$ is the digamma function, d denotes the dimension of X and c_d is the volume of the unit ball w.r.t. the used norm. From now on, we assume that the maximum norm is used, implying $\log c_d = 0$.

Kraskov et al. expand this to mutual information estimation with help of the Eq. (1). Naively applying the estimate (3) for each of the entropies in (1) would induce errors due to the different length scales in spaces (X, Y), X and Y. Instead, the length scale is fixed by searching the k-nearest neighbours first in the joint space (X, Y). We let $\epsilon(i)/2$ to denote the distance to the kth nearest neighbour of the point (x_i, y_i). When computing the entropy estimate in the marginal space X, the following approximation is used:

$$\psi(k) = \frac{1}{n} \sum_{i=1}^{n} \psi(n_x(i) + 1),$$

where $n_x(i)$ is the number of points x_j such that $||x_i - x_j|| < \epsilon(i)/2,\ j \neq i$. The similar approximation is used in the Y space by replacing the x_i with y_i. This is motivated by the fact that Eq. (3) holds for any k, and $\epsilon(i)/2$ is the distance either to the $(n_x(i) + 1)$th neighbour of x_i or to the $(n_y(i) + 1)$th neighbour of y_i. Using equations (1) and (3) with the approximation in the marginal spaces leads to the cancellation of the $\epsilon(i)$ terms and we obtain the following formula for the mutual information:

$$\hat{I}(X; Y) = \psi(k) + \psi(n) - \frac{1}{n} \sum_{i=1}^{n} \Big(\psi(n_x(i) + 1) + \psi(n_y(i) + 1) \Big). \tag{4}$$

Using similar reasoning, Vejmelka and Paluš (2008) present the following formula for the conditional mutual information:

$$\hat{I}(X; Y \mid Z) = \psi(k) - \frac{1}{n} \sum_{i=1}^{n} \Big(\psi(n_{xz}(i) + 1) + \psi(n_{yz}(i) + 1) - \psi(n_z(i) + 1) \Big), \tag{5}$$

where the counts $n_z(i)$, $n_{yz}(i)$ and $n_{xz}(i)$ in the marginal spaces are defined in a similar fashion as in Eq. (4) using the k-nearest nearest neighbour distances found in the joint space.

The parameter k in these estimators controls the bias-variance trade-off: a small k means that the assumption about the constant density holds only in small regions, thus implying smaller bias, whereas large k decreases the variance as more data are used to obtain the local estimates.

2.3 Non-parametric Test for Conditional Independence

Due to statistical variation, the empirical joint distribution is hardly ever exactly equivalent to the product of the margins, just like an empirical correlation coefficient is hardly ever exactly zero. Hence, we need to consider a test that takes into account the statistical uncertainty of the mutual information estimator. To this end, we apply a permutation test to simulate the sampling distribution of the mutual information statistic under the null hypothesis of conditional independence.

To test the conditional independence based on observed data $\boldsymbol{x}$, $\boldsymbol{y}$ and $\boldsymbol{z}$, we first set a significance level α, and compute the estimate $\hat{I}(\boldsymbol{x}; \boldsymbol{y} \mid \boldsymbol{z})$. Conditional independence is simulated by randomly permuting the samples $\boldsymbol{y} = (y_1, \ldots, y_n)$ to create a vector $\boldsymbol{y}_{perm}$, and then computing $\hat{I}(\boldsymbol{x}; \boldsymbol{y}_{perm} \mid \boldsymbol{z})$. This is repeated T times. After this, we count the number of permuted mutual information values that are greater than or equal to the initial estimate $\hat{I}(\boldsymbol{x}; \boldsymbol{y} \mid \boldsymbol{z})$. We let K to denote this number. This gives us an estimate for the p-value, $\hat{p} = (K+1)/(T+1)$, which is then compared to the significance level α. To ease the computational burden we skip this test in two cases: 1) when there are no conditioning variables and the correlation based test (we used the Fisher-z test) rejects independence, the algorithm returns ’False’ implying dependence, and 2) if the same (partial) correlation based test accepts independence and the estimated value for (conditional) mutual information is below 0.001 nats, the algorithm returns ’True’.

Algorithm 1 shows the pseudocode for the permutation test based conditional independence test discussed above.

Algorithm 1: Conditional Independence Test

```
Require:
    Significance level $\alpha$, number of iterations $T$
procedure CIT($\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}$)
    $estCMI \leftarrow \hat{I}(\boldsymbol{x}; \boldsymbol{y} \mid \boldsymbol{z})$
    $PermutedMI \leftarrow \emptyset$
    for $i \leftarrow 1, \ldots, T$ do
        $\boldsymbol{y}_{perm(i)} \leftarrow$ random permutation of $\boldsymbol{y}$
        $mi \leftarrow \hat{I}(\boldsymbol{x}; \boldsymbol{y}_{perm(i)} \mid \boldsymbol{z})$
        $PermutedMI \leftarrow PermutedMI \cup \{mi\}$
    $K \leftarrow \#\{\, i : \hat{I}(\boldsymbol{x}; \boldsymbol{y}_{perm(i)} \mid \boldsymbol{z}) \geq estCMI\}$
    if $(K+1)/(T+1) < \alpha$ then return False
    return True
```

Recently, Runge (2018) independently proposed a similar non-parametric independence test based on the Kraskov mutual information estimators. The test proposed by Runge differs from ours in how the permutation of $\boldsymbol{y}$ is created. Runge notes that simply permuting $\boldsymbol{y}$ destroys dependence between $\boldsymbol{x}$ and $\boldsymbol{y}$, but in addition, the dependence between $\boldsymbol{y}$ and the conditioning $\boldsymbol{z}$ is also lost, which is in principle wrong when we are testing for conditional independence. To remedy this, Runge uses a local permutation scheme in which we first identify k_{perm}-nearest neighbours of each point z_i. Then the ith component of the permuted vector, $\boldsymbol{y}_{perm}$, is randomly drawn from y_j corresponding to the neighbours of z_i. In this scheme, some y_j values might appear multiple times in $\boldsymbol{y}_{perm}$. In our experiments, we found that this strategy lead in most cases to inferior accuracy when used with the structure learning algorithm described in Section 3.2 (comparison shown in Appendix A). Therefore, we propose to use the simple permutation strategy which also avoids the choice of another hyperparameter k_{perm}.

3. Structure Learning of Markov Networks

In this section, we will go briefly through the basic concepts related to Markov networks and then present the structure learning algorithm which is combined with the presented non-parametric conditional independence test. For a more thorough treatment, we refer to Whittaker (1990); Lauritzen (1996); Koller and Friedman (2009).

3.1 Representation

Let $X = (X_1, \dots X_p)$ be a random vector and $G = (V, E)$ denote an undirected graph (UG), where $V = \{1, \dots, p\}$ is the set of nodes corresponding to elements of X and $E \subset V \times V$ the set of edges. Given an UG G, we define the Markov blanket of the node i to be the set containing its neighbouring nodes in the graph G, $mb(i) = \{j \in V | (i,j) \in E\}$, where $(i,j) = (j,i)$ is an undirected edge between nodes i and j. The graph G encodes a set of conditional independence assumptions that can be characterized via *Markov properties*: 1) if $(i,j) \notin E$, the variable X_i is independent of X_j given the remaining ones $V \setminus \{i,j\}$, 2) every variable $i \in V$ is conditionally independent of all the other variables given its Markov blanket, 3) for the disjoint subsets of variables, $A, B, C \subset V$, it holds that X_A is conditionally independent of X_B given X_C if C separates A and B in the graph. The notation X_A stands for the random vector containing the variables belonging to a set $A \subset V$. These properties are termed the pairwise, the local and the global Markov properties, respectively.

3.2 Structure Learning

The main problem we are focusing on here is learning the graph structure G based on the observed data $\boldsymbol{X} = (\boldsymbol{x}_1, \dots, \boldsymbol{x}_n)$, where $\boldsymbol{x}_i \in \mathbb{R}^p$ is i.i.d sample from the distribution $p(X)$. The methods addressing this problem are usually either score- or constraint-based ones. The first mentioned approach is based on a data-dependent scoring function which evaluates the goodness of different structures whereas the constraint-based methods make use of the Markov properties and perform a series of conditional independence tests to infer the network structure. Here, we will adopt this latter approach. More in detail, we will use the Incremental Association Markov Blanket (IAMB) algorithm (Tsamardinos et al., 2003) to learn the Markov blanket for each of the nodes. The algorithm uses some measure of (conditional) dependence which is used to determine the order in which variables are considered to be entered in the blanket. The algorithm starts with an empty blanket, and then adds variable (with the highest dependence) if it is found to be conditionally dependent given the current variables in the blanket. This is repeated until no variables can be added. Addition phase is then followed by a step where variables are removed if they are conditionally independent of the target node given the remaining variables in blanket. The algorithm is guaranteed to return the correct Markov blanket assuming faithfulness and correctness of the independence tests (Tsamardinos et al., 2003; Peña et al., 2007). For any finite sample size n, the found Markov blankets are not necessarily coherent in a sense that $i \in mb(j)$ would imply that j was also found to belong to Markov blanket of i. To overcome this, we define the estimated undirected graph using conservative AND-rule, meaning that there is an undirected edge between i and j if $i \in \widehat{mb}(j)$ and $j \in \widehat{mb}(i)$.

Implementing this algorithm with the conditional independence test described in Section 2 and the measure of dependence given by the estimate of conditional mutual information yields our proposed method, which will be henceforth referred to as `knnMI`.

3.3 On Computational Complexity

The computational cost of our proposed approach is dominated by the nearest neighbours searches which become costly when the sample size and dimension of the data grow. In the concrete implementation of the algorithm we use kd-tree (Bentley, 1975) to perform these queries. Let us analyse the steps needed to compute the estimate for conditional mutual information (Eq. (5) in Sec. 2.2).

Let n be the number of observations and d denote the dimension of the joint space (X, Y, Z). The brute force approach of finding the k-nearest neighbour for each data point would scale as $O(n^2kd)$. However, when the dimension of the data (the size of largest considered Markov blanket) is fairly low, we can usually obtain significant saves by using the kd-tree:

1. Index construction of the tree for joint and marginal spaces takes $O(dn \log n)$ time.
2. For each data point (x_i, y_i, z_i), we need to find the k-nearest neighbour in the joint space and record the distance $\epsilon_i/2$. For a fixed d, finding one neighbour has expected running time of $O(\log n)$ (Friedman et al., 1977), which yields a total running time of $O(kn \log n)$. With respect to dimension d the worst case complexity is exponential. However, assuming the sizes of considered Markov blankets are fairly small, this does not cause difficulties in practice.
3. Using the found distances, we count for each data point the number of points whose distance is less than $\epsilon_i/2$. This is done in spaces (X, Z), (Y, Z) and Z. With fixed d this would naively take $O(n^2)$ time. In practice, when the dimension is fixed, the expected running times for single nearest-neighbour and radius queries in kd-trees could be significantly smaller, even a constant time operations (Bentley, 1990).

The number of independence tests and measure of dependence computations (estimate of conditional mutual information) performed by IAMB when searching for a single Markov blanket is in the worst case of order $O(p^2)$ (Tsamardinos et al., 2003). However, the authors state they experimentally observed an average case order of $O(p|mb(i)|)$ tests, where $|mb(i)|$ refers to the size of the Markov blanket for variable i. This implies that in the worst case finding the graph takes $O(p^3)$ tests but if the Markov blankets are relatively small, the complexity is considerably lower.

4. Experiments

In this section we evaluate the performance of the proposed approach and compare it to other methods by creating synthetic data from various Markov network structures where the dependencies between the variables are not necessarily linear or the distribution close to multivariate normal[1].

4.1 Considered Methods

We compare the performance of `knnMI` to a method that uses exactly the same structure learning algorithm but with an independence test based on Fisher's z-transformed sample partial correla-

[1] The code to reproduce all the experiments is available at `https://github.com/janleppa/graph_learn_mi`

tions, see, for instance, Kalisch and Bühlmann (2007). We will refer to this method as `fisherZ`. In addition, we combine KCIT and RCIT kernel tests with this algorithm. The corresponding methods are referred as `KCIT` and `RCIT`. Aforementioned tests utilize the p-value of the respective test as the measure of dependence which determines the order in which variables tested for entering the Markov blanket. For the hyperparameters of the kernel methods, we used the default values in R-package 'RCIT' [2]: The RBF kernel widths were chosen using the heuristic based on the median distance of the first 500 data points. `RCIT` method requires choosing the numbers of used random Fourier features for X, Y and Z. We chose the same values as used by Strobl et al. (2017): 5, 5 and 25, respectively. `KCIT` test uses bootstrapping to estimate null distribution of the test statistic whereas `RCIT` approximates the null distribution by moment matching (Lindsay-Pilla-Basak method). For more details on the hyperparameters, see Strobl et al. (2017).

Other methods we compare against include graphical lasso (glasso) (Friedman et al., 2008) and neighbourhood-selection method (mb) (Meinshausen and Bühlmann, 2006). As we mainly study non-Gaussian data, all the input data are put through a non-paranormal transformation based on a shrunken empirical cumulative distribution function (ECDF) (Liu et al., 2009, 2012) before applying glasso and mb.

The glasso method learns the graph by estimating the inverse of covariance which is done by optimizing an objective function comprising of ℓ_1-penalized Gaussian log-likelihood. The mb estimates the graph by conducting ℓ_1-penalized linear regression independently for each variable to find their Markov blankets. We will use the similar AND-rule as mentioned before to construct the graph from the estimated Markov blankets. As the output of glasso and mb depends on the tuning parameter $\lambda > 0$ which controls the amount of ℓ_1-regularization, we computed graphs for 20 tuning parameter values, starting from the tuning parameter value λ_{max} that resulted in an empty graph and then decreased it to a value $\lambda_{min} = 0.01\lambda_{max}$. The densest model had always more edges than the true generating network structure. The best model was chosen according to the StARS criterion (Liu et al., 2010). We refer to these methods as `NPN_glasso` and `NPN_mb` to emphasize that methods are non-paranormal. With mb, we tried also choosing the parameter automatically as proposed by the authors to be $\lambda = (n^{-1/2})\Phi^{-1}(1 - \alpha/(2p^2))$, where $\alpha = 0.05$ and $\Phi(\cdot)$ denotes the c.d.f. of a standard normal random variable. We will refer to this method as `NPN_mb_auto`. In the experiments, we used the implementations of glasso and mb found in R-package 'huge'[3].

In all the conditional independence tests, we set the significance level to be 0.05. With `knnMI` we set $k = 5$ and do $T = 200$ permutations of data when testing for independence. The performance of `knnMI` seemed quite robust when small values of k were used. Comparisons involving different choices k are presented in Appendix A. To compare the methods, we measure the average Hamming distance (the sum of false positive and false negative edges) between the estimated graph and the ground truth graph. All the presented values are averages from 25 repetitions.

4.2 Small Network

First, we consider a small network consisting of seven nodes and eight edges. In this example, the considered graph is decomposable, implying that we can represent it equally well as a directed acyclic graph which simplifies the data generation. With the network structure fixed, we considered six different data generating schemes. The dependencies between the child variable and the parents

[2] https://github.com/ericstrobl/RCIT

[3] https://CRAN.R-project.org/package=huge

	Linear	Non-linear
X_1	ϵ_1	ϵ_1
X_2	$0.2X_1 + \epsilon_2$	$2\cos(X_1) + \epsilon_2$
X_3	$0.5X_2 + \epsilon_3$	$2\sin(\pi X_2) + \epsilon_3$
X_4	$0.25X_3 + \epsilon_4$	$3\cos(X_3) + \epsilon_4$
X_5	$0.35X_2 + 0.55X_3 + \epsilon_5$	$0.75X_2X_3 + \epsilon_5$
X_6	$0.65X_5 + \epsilon_6$	$2.5X_5 + \epsilon_6$
X_7	$0.9X_3 + 0.25X_5 + \epsilon_7$	$3\cos(0.2X_3) + \log\lvert X_5\rvert + \epsilon_7$

Table 1: Data generating model.

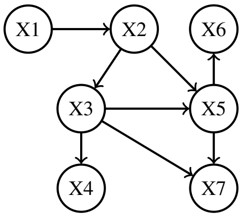

Figure 1: Ground truth graph.

were either linear or non-linear with an additive noise term. We also include three different noise distributions: standard Gaussian, uniform $[-1, 1]$, and standard t with two degrees of freedom. The data generating mechanism and the ground truth graph are presented in Table 1 and Figure 1, respectively. We use ϵ_i to denote the noise term which follows one of the aforementioned distributions. The non-linear data generating mechanism was inspired by the example in Tillman (2009).

We created multiple data sets with sample sizes ranging from 125 to 2000. The average Hamming distances to the true graph for each method are presented in Figure 2. In the Hamming distance figures, errors bars show the standard error of the mean.

In the linear case, `fisherZ` and `KCIT` are the most accurate regardless the noise distribution. It is also somewhat surprising how the performance of `fisherZ` did not seem to deteriorate at all when the assumption about normally distributed noise was violated. As maybe expected, our method can learn the structure the best in cases where the dependencies are non-linear. The only method with comparable performance is `KCIT`. In these cases, `knnMI` and `KCIT` are the only methods that steadily improve their performance as the sample size increases, recovering the true generating structure almost correctly when sample size $n = 2000$.

4.3 Larger Networks

Next, we generated non-paranormal data from randomly generated graph structures. The graphs were first created by randomly adding an edge between variables with a probability of $3/p$, where p is the number of variables. This implies that the expected number of edges is $3(p-1)/2$. The multivariate normal data was sampled using the R-package 'huge', and the non-paranormal data was created from this by applying a power transformation $X_i \mapsto X_i^3$ to each variable. The sample sizes of created data sets ranged from 125 to 2000. The results are shown in Figure 3. We consider dimension $p = 10$ (left in Figure 3) and $p = 20$ (center). Looking at the results, we can see that `NPN_mb` and `NPN_glasso` perform the best when $p = 20$ but generally worse than others in the lower-dimensional case. With the smallest sample sizes, `KCIT` and `knnMI` perform quite similarly. Both methods are close to finding the true generating graph on the largest sample sizes. However, `KCIT` seems to converge faster in the non-paranormal setting.

In the last setting, we consider a larger network with non-linear dependencies between the variables. The graph is created by combining three seven nodes graphs described in the previous section as disconnected components to form a larger 21 node graph. In each of these independent sub-graphs, data is generated according to non-linear mechanism, as explained in Section 4.2. The

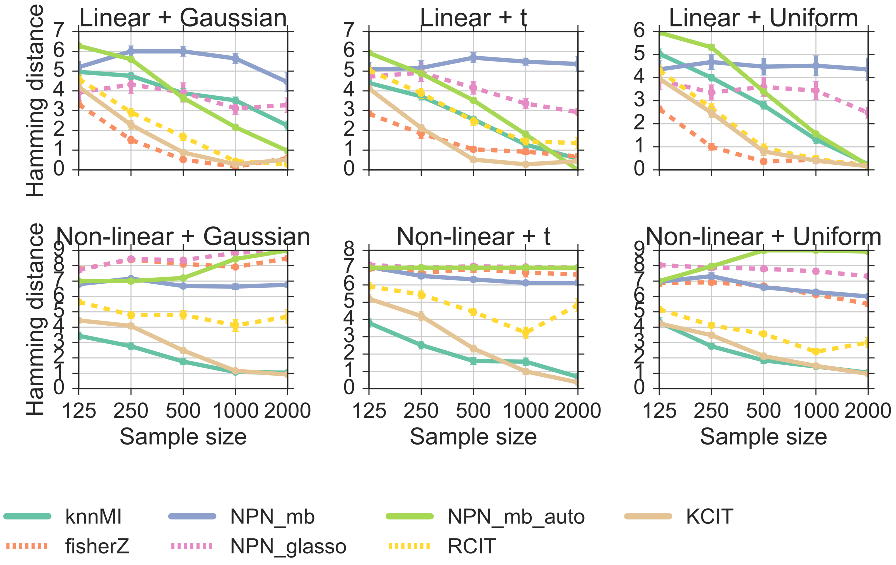

Figure 2: Hamming distances for the small network with different noise distributions. Considered methods: knnMI (proposed method); NPN_mb, NPN_glasso and NPN_mb_auto (non-paranomal methods); KCIT and RCIT (kernel methods) and fisherZ (Gaussian method).

results are shown right in Figure 3. We can see that here knnMI achieves the best performance with KCIT obtaining similar results only at the largest sample sizes. Other methods do not seem to be able to improve their performance as the sample size gets larger.

The performance of RCIT seems a bit unstable in the experiments. With the different choices of the hyperparameters, the performance would likely be closer to KCIT. However, this demonstrates that the computational efficiency of RCIT comes with a trade-off, requiring the user to pay a closer attention to the choice of the hyperparameters.

5. Conclusions

We have presented an algorithm for distribution free learning of Markov network structures. The algorithm combines previous work on non-parametric estimation of mutual information to an efficient structure learning algorithm in a novel way. The knnMI algorithm consistently outperforms other tested algorithms in structure learning in the case of strongly non-linear dependencies and its performance is robust to non-Gaussian noise. In these settings, KCIT is the only method capable of achieving nearly as good performance, which, however, comes with a higher computational cost.

Even though the Markov blanket searches and permutation tests can be computed in parallel, the computational cost of knnMI algorithm is greater than that of other tested algorithms except for KCIT. The nearest-neighbour search is a costly operation, which, especially in the high dimensional case, uses the largest proportion of computation time, even while using efficient metric

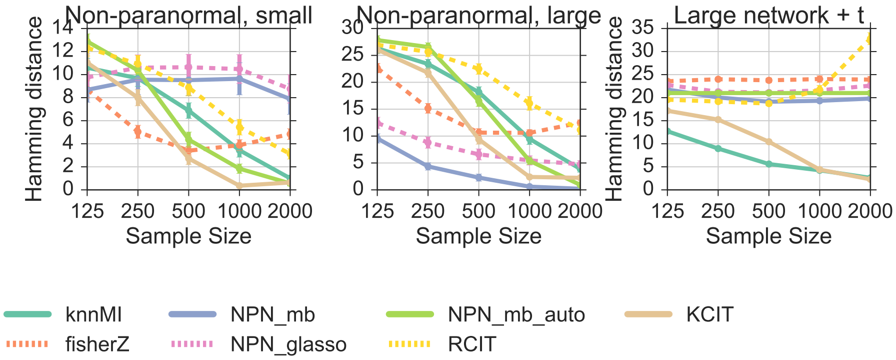

Figure 3: Averaged Hamming distances for the larger networks.

tree structures. A clear direction for future research is to study if approximate nearest-neighbour searches could by utilized to improve the efficiency while still maintaining the consistent estimation of mutual information.

Acknowledgments

J.L. and T.R. were supported by the Academy of Finland (COIN CoE and Project TensorML). J.L. was supported by the DoCS doctoral programme of the University of Helsinki.

Appendix A. Comparison with Local Permutation Approach

Figure 4 summarises results from experiments studying the effect of k and the impact of the local permutation strategy (Runge, 2018). Data is generated from the small network as explained in Section 4.2 of the main paper. The sample sizes in tests range from 125 to 1000. We consider the following values $k = 3, 5, 0.01n, 0.1n$ with local and simple permutation schemes. The values $0.1n$ and $0.01n$ for k depend on the sample size (in case of the latter, k was always at least 3). For the local permutation scheme, we set $k_{perm} = 5$, as suggested by Runge (2018). In the Figure, different choices of k correspond to different colors while solid lines correspond to local permutations and broken lines the simple. The shown results are averages computed from 25 repetitions.

Looking at the results, we see that in most of the cases simple permutation results in better Hamming distance than the local permutation strategy. Exception is the value $k = 0.1n$ with non-linear dependencies, where the opposite holds and the simple permutations scheme does not seem to converge to the ground truth graph. This suggests that one should prefer smaller values of k with the simple permutation scheme to obtain consistent results.

Runge (2018) argues that local permutation results in the better calibrated null-distribution for the conditional mutual information. He shows experimentally how this can cause higher false positive detection rate for the simple permutation based test. We also checked the false positive (FP) and the true positive (TP) edge rates in these experiments and found a similar pattern: local permutation based tests had in most cases slightly lower FP rate but the TP rate was also a bit lower (results not

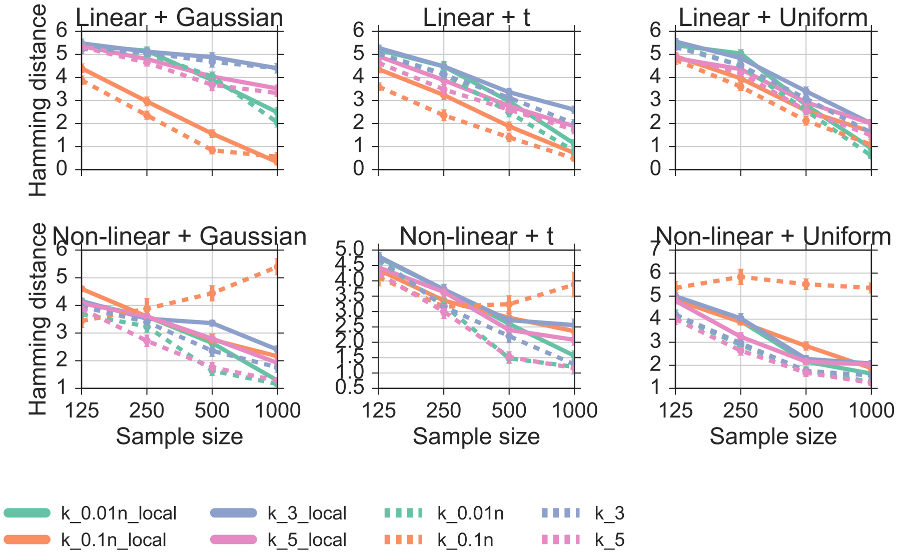

Figure 4: Hamming distances for the small network with different noise distributions

shown). In terms of Hamming distance, the simple permutation scheme seemed still better. This can be also explained by the used conservative AND-rule for combining the found Markov blankets into an undirected graph which itself helps to lower the possibility of adding a false edge in the final graph. To conclude, if false positive edges are highly unwanted, adopting the local permutation strategy might be beneficial.

References

J. L. Bentley. Multidimensional binary search trees used for associative searching. *Commun. ACM*, 18(9):509–517, 1975.

J. L. Bentley. K-d trees for semidynamic point sets. In *Proceedings of the Sixth Annual Symposium on Computational Geometry*, SCG '90, pages 187–197, 1990.

T. M. Cover and J. A. Thomas. *Elements of Information Theory 2nd Edition*. Wiley-Interscience, 2006.

J. Friedman, T. Hastie, and R. Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, 2008.

J. H. Friedman, J. L. Bentley, and R. A. Finkel. An algorithm for finding best matches in logarithmic expected time. *ACM Trans. Math. Softw.*, 3(3):209–226, 1977.

M. Kalisch and P. Bühlmann. Estimating high-dimensional directed acyclic graphs with the pc-algorithm. *JMLR*, 8:613–636, 2007.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques*. MIT press, 2009.

L. Kozachenko and N. N. Leonenko. Sample estimate of the entropy of a random vector. *Problemy Peredachi Informatsii*, 23(2):9–16, 1987.

A. Kraskov, H. Stögbauer, and P. Grassberger. Estimating mutual information. *Phys. Rev. E*, 69(6), 2004.

J. Lafferty, H. Liu, and L. Wasserman. Sparse nonparametric graphical models. *Statistical Science*, 27(4):519–537, 2012.

S. Lauritzen. *Graphical Models*. Clarendon Press, 1996.

H. Liu, J. Lafferty, and L. Wasserman. The nonparanormal: Semiparametric estimation of high dimensional undirected graphs. *JMLR*, 10:2295–2328, 2009.

H. Liu, K. Roeder, and L. Wasserman. Stability approach to regularization selection (stars) for high dimensional graphical models. In *Proceedings of the 23rd International Conference on Neural Information Processing Systems*, pages 1432–1440, 2010.

H. Liu, F. Han, M. Yuan, J. Lafferty, and L. Wasserman. High-dimensional semiparametric Gaussian copula graphical models. *Ann. Statist.*, 40(4):2293–2326, 2012.

N. Meinshausen and P. Bühlmann. High-dimensional graphs and variable selection with the lasso. *The Annals of Statistics*, 34(3):1436–1462, 2006.

J. M. Peña, R. Nilsson, J. Björkegren, and J. Tegnér. Towards scalable and data efficient learning of Markov boundaries. *International Journal of Approximate Reasoning*, 45(2):211 – 232, 2007.

J. Runge. Conditional independence testing based on a nearest-neighbor estimator of conditional mutual information. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, pages 938–947, 2018.

E. V. Strobl, K. Zhang, and S. Visweswaran. Approximate kernel-based conditional independence tests for fast non-parametric causal discovery. *ArXiv e-prints*, 2017.

R. E. Tillman. Learning directed graphical models from nonlinear and non-Gaussian data, 2009. URL `https://www.ml.cmu.edu/research/dap-papers/tillman_dap.pdf`. Data Analysis Project for Master of Science in Machine Learning.

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

M. Vejmelka and M. Paluš. Inferring the directionality of coupling with conditional mutual information. *Phys. Rev. E*, 77, 2008.

J. Whittaker. *Graphical Models in Applied Multivariate Statistics*. John Wiley & Sons, 1990.

K. Zhang, J. Peters, D. Janzing, and B. Schölkopf. Kernel-based conditional independence test and application in causal discovery. In *Proceedings of the Twenty-Seventh Conference on Uncertainty in Artificial Intelligence*, pages 804–813, 2011.