Proximity-Based Anomaly Detection using Sparse Structure Learning

Tsuyoshi Idé
IBM Research,
Tokyo Research Laboratory
goodidea@jp.ibm.com

Aurelie C. Lozano Naoki Abe Yan Liu
IBM Research,
T. J. Watson Research Center
{aclozano, nabe, liuya}@us.ibm.com

Abstract

We consider the task of performing anomaly detection in highly noisy multivariate data. In many applications involving real-valued time-series data, such as physical sensor data and economic metrics, discovering changes and anomalies in the way variables depend on one another is of particular importance. Our goal is to robustly compute the "correlation anomaly" score of each variable by comparing the test data with reference data, even when some of the variables are highly correlated (and thus collinearity exists). To remove seeming dependencies introduced by noise, we focus on the most significant dependencies for each variable. We perform this "neighborhood selection" in an adaptive manner by fitting a sparse graphical Gaussian model. Instead of traditional covariance selection procedures, we solve this problem as maximum likelihood estimation of the precision matrix (inverse covariance matrix) under the L_1 penalty. Then the anomaly score for each variable is computed by evaluating the distances between the fitted conditional distributions within the Markov blanket for that variable, for the (two) data sets to be compared. Using real-world data, we demonstrate that our matrix-based sparse structure learning approach successfully detects correlation anomalies under collinearities and heavy noise.

1 Introduction

Knowledge discovery from networks and graphs is one of the most exciting topics in data mining. While most of the existing studies in network mining assume the knowledge of graph structure as input, estimating general dependency graphs given multivariate data is also an important problem. In contrast to binary sparse connectivities in e.g. coauthor networks, the dependency graph between the variables in a real-valued multivariate system is in general a complete graph since two variables are rarely totally independent. Since real-world data necessarily includes considerable amount of noise, however, some of the dependencies might very well be due to noise. This motivates us to learn a *sparse structure* from data representing an essential relationship between the variables that is not an artifact of noise.

We consider the task of performing graph-based anomaly detection in the following setting [16]: Given two dependency graphs of a multivariate system, identify the nodes (variables) which are most responsible for the difference between them, and compute the degree to which each node contributes to the difference. We call this problem *change analysis* to contrast with only detecting the overall change. Figure 1 summarizes our problem. As shown in the figure, we focus on dependency graphs typically produced by noisy sensor data, and anomalies that occur in the dependencies (we call them *correlation anomalies*). This is because detecting anomalies that occur only within the individual variables is often trivial, while detecting correlation anomalies is much harder and is practically important in fault analysis of complicated dynamic systems such as automobiles [16] and computer systems [15]. Starting with dense dependency graphs that might be contaminated by noise, we first attempt to find sparse structures by removing the unwanted effects of noise. Then we compare the learned structures to compute the anomaly score for each variable.

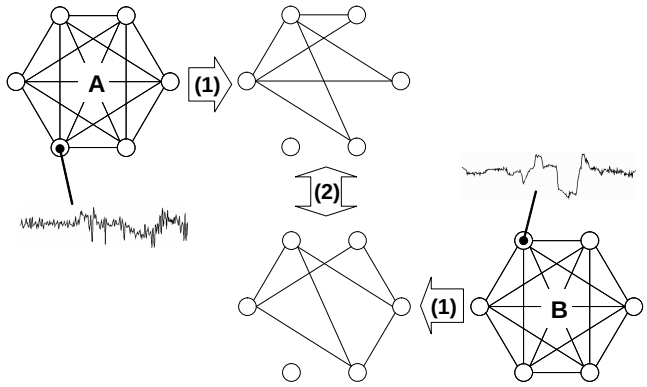

Figure 1: Problem setting. Given noisy sensor data sets A and B, (1) we first learn sparse structures based on given covariance matrices. (2) Next, the two sparse graphs are compared to give the anomaly score of *each* variable.

In statistics, the problem of structure learning from real-valued multivariate data has been treated within the framework of covariance selection, as originally proposed by Dempster [6]. As the name suggests, covariance selection

assumes the multivariate Gaussian as a generative model, and fits a precision matrix (the inverse of the covariance matrix) to the data under a sparsity constraint. Recently, several new sparse structure learning algorithms have been proposed [19, 20, 5, 3, 7, 9] to remedy drawbacks of the traditional covariance selection framework. In particular, Meinshausen and Bühlmann [20] formulated the task of sparse structure learning as neighborhood selection of each variable, and proposed a method based on lasso (L_1 penalized linear regression) where each variable is treated as the target and the rest as the predictors. Thanks to the L_1 regularizer, many of the linear regression coefficients become exactly zero, and predictors with nonzero coefficients are thought of as the neighbors of the target.

From a practical perspective, one of the most important merits of the method of [20] is that, in principle, it enables us to learn a sparse graph structure even when the number of variables (M) is comparable to the number of samples (N). This is different from the traditional covariance selection procedure, where the algorithm cannot give meaningful results when the sample covariance matrix is rank deficient, as is always the case when $M > N$ or when some of the variables are highly correlated.

In this paper, we apply L_1 penalized sparse structure learning to the task of scoring correlation anomalies. We assume that some of the variables are highly correlated in the data, which is quite common in sensor data analytics. We experimentally demonstrate that the algorithm of [20] is instable in such cases, while the graphical lasso algorithm [9] enjoys robustness against noise. Based on the framework of graphical Gaussian model, we propose a definition of "correlation anomaly score" in an information-theoretically consistent manner.

The rest of the paper is organized as follows. Section 2 illustrates our problem setting in more detail using a motivating example, and explains the essence of graphical Gaussian models. Section 3 surveys related work including comparison with the two-sample problem in statistics. Section 4 discusses sparse structure learning algorithms with particular emphasis on graphical lasso. Section 5 defines the correlation anomaly scores. Section 6 presents experimental results including those with actual car sensor data. Finally, Section 7 concludes the paper.

2 Preliminaries

In this section, we briefly recapitulate the graphical Gaussian model (GGM), and summarize our problem setting.

2.1 Motivating example and problem statement. In many real-valued time-series data such as physical sensor data and econometrics data, it is usual that some of the variables are highly correlated. Figure 2 shows such an example, where daily spot prices (foreign currency in dollars) are shown over the last 567 days in the *Actual spot rates* data [17]. The original data was collected over the 10 years from Oct. 9, 1986 through Aug. 9, 1996. Abbreviations used in the figure can be found in Table 1. As expected, European currencies such as BEF, FRF, DEM, and NLG are highly correlated. In fact, from Figure 3, which shows a pairwise scattering plot between the first four currencies, we see that BEF and FRF are almost perfectly correlated, while other squares show more complex trajectories.

Table 1: Abbreviations in *Actual spot rates* data.

AUD	Australian Dollar	NLG	Dutch Guilder
BEF	Belgian Franc	NZD	New Zealand Dollar
CAD	Canadian Dollar	ESP	Spanish Peseta
FRF	French Franc	SEK	Swedish Krone
DEM	German Mark	CHF	Swiss Franc
JPY	Japanese Yen	GBP	UK Pound

Table 2: Correlation coefficients for the data shown in Fig. 3. Values in the parenthesis correspond to the bottom plot.

	BEF	CAD	FRF
AUD	0.31 (-0.37)	0.91 (0.04)	0.26 (-0.23)
BEF		0.46 (0.19)	0.99 (0.97)
CAD			0.41(0.30)

This figure suggests an important starting point for correlation anomaly analysis. If a correlation is far from perfect, and if the data is noisy, generated by some complicated dynamics, trajectories in the pairwise scattering plots will be complex and instable. In fact, the corresponding correlation coefficients shown in Table 2 demonstrate considerable fluctuations in the pairs except for (BEF, FRF). Detecting anomalies under these circumstances can be challenging unless there is some detectable persistent property, motivating us to formulate the following assumption.

ASSUMPTION 1. (NEIGHBORHOOD PRESERVATION) *If the system is working normally, the neighborhood graph of each node is almost invariant against the fluctuations of experimental conditions.*

(The definition of a neighborhood graph will be given in the next subsection, Definition 2). Thus, in the present paper, we equate the problem of correlation anomaly detection with that of detecting significant changes in the neighborhood graph of the variables involved. We believe that this assumption is reasonable in many practical situations, including those in which data are generated by a system with complicated dynamics, and this is the main motivation for our graph-based approach to anomaly detection.

Now we state the problem addressed in this paper.

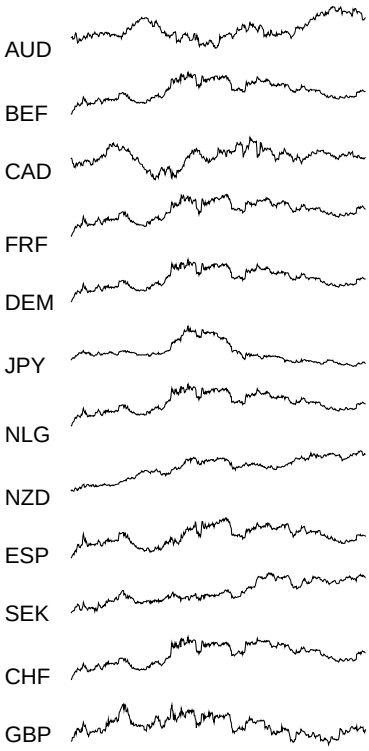

Figure 2: *Actual spot rates* data over the 567 days until Aug. 9, 1996.

Suppose that we are given two data sets

$$\mathcal{D}_{\mathrm{A}} \equiv \{\boldsymbol{x}_{\mathrm{A}}^{(n)} | \boldsymbol{x}_{\mathrm{A}}^{(n)} \in \mathbb{R}^M, n = 1, 2, ..., N_{\mathrm{A}}\}$$
$$\mathcal{D}_{\mathrm{B}} \equiv \{\boldsymbol{x}_{\mathrm{B}}^{(n)} | \boldsymbol{x}_{\mathrm{B}}^{(n)} \in \mathbb{R}^M, n = 1, 2, ..., N_{\mathrm{B}}\}.$$

Since we are interested mainly in noisy sensor data, the index n typically runs over discrete values of time. In $\mathcal{D}_{\mathrm{A}}$ and $\mathcal{D}_{\mathrm{B}}$, the number of measurements, N_{A} and N_{B}, can be different, but the number of variables and the identity of each variable must be the same. For example, if the first dimension of $\boldsymbol{x}_A^{(n)}$ in $\mathcal{D}_{\mathrm{A}}$ measures atmospheric pressure, the first dimension of $\boldsymbol{x}_B^{(n)}$ in $\mathcal{D}_{\mathrm{B}}$ also measures the same quantity in some different situation.

Now our problem is stated as follows.

DEFINITION 1. (CHANGE ANALYSIS PROBLEM) *Given $\mathcal{D}_{\mathrm{A}}$ and $\mathcal{D}_{\mathrm{B}}$, compute the anomaly score of each variable representing how much each variable contributes to the difference between the data.*

This problem, as formulated, shares some similarity with the "two-sample problem" in statistics. We will discuss the relationship between the two problems in the next section.

2.2 Graphical Gaussian model. For an M-dimensional random variable $\boldsymbol{x} \in \mathbb{R}^M$, the GGM assumes M-dimensional Gaussian distribution

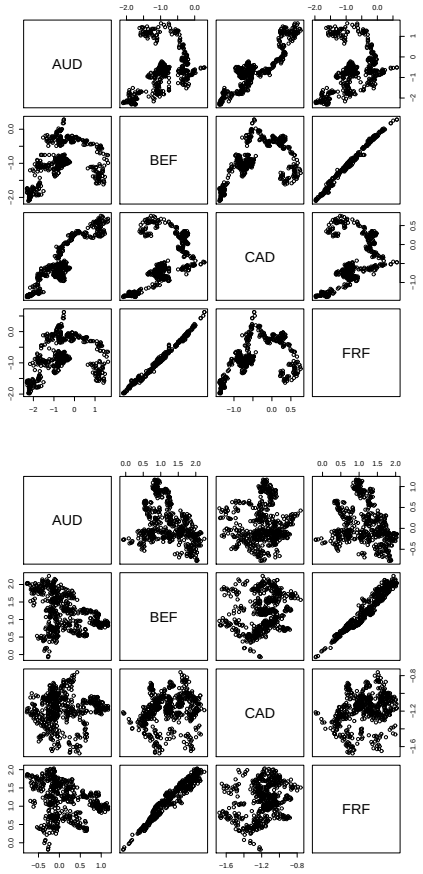

Figure 3: Pairwise scattering plot of four countries in *Actual spot rates* data. Top: first 500 days. Bottom: last 567 days.

$$\mathcal{N}(\boldsymbol{x}|\boldsymbol{0}, \Lambda^{-1}) = \frac{\det(\Lambda)^{1/2}}{(2\pi)^{M/2}} \exp\left(-\frac{1}{2}\boldsymbol{x}^\top \Lambda \boldsymbol{x}\right), \tag{2.1}$$

where det represents matrix determinant, and $\Lambda \in \mathbb{R}^{M \times M}$ denotes a precision matrix. We denote by $\mathcal{N}(\cdot|\boldsymbol{\mu}, \Sigma)$ a Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix Σ. As mentioned above, a precision matrix is the inverse of a covariance matrix, and vice versa.

In the GGM, a Gaussian distribution is associated with a graph (V, E), where V is the set of nodes containing all the M variables, and E is a set of edges. The edge between x_i and x_j is absent if and only if they are independent conditioned on all the other variables. Under the Gaussian assumption, this condition is represented as

$$\Lambda_{i,j} = 0 \quad \Rightarrow \quad x_i \perp\!\!\!\perp x_j \mid \text{other variables}, \tag{2.2}$$

where $\perp\!\!\!\perp$ denotes statistical independence. Stating formally, we give the definition of neighborhood as follows.

DEFINITION 2. (NEIGHBORHOOD) *We say that a node x_i is a neighborhood of x_j, if and only if $\Lambda_{i,j} \neq 0$. A neighborhood graph of x_i is the graph that contains x_i and its neighbors, connected with edges between the x_i and its neighbors. Neighborhood selection is the task to enumerate all neighbors of each node.*

The condition (2.2) can be easily understood by explicitly writing down the conditional distribution. Let us denote $(x_i, x_j)^\top$ by $\boldsymbol{x}_a$, and the rest of the variables by $\boldsymbol{x}_b$. For centered data, a standard partitioning formula of Gaussian (see, e.g. [4], Sec. 2.3) gives the conditional distribution as

$$p(\boldsymbol{x}_a|\boldsymbol{x}_b) = \mathcal{N}(\boldsymbol{x}_a| - \Lambda_{aa}^{-1}\Lambda_{ab}\boldsymbol{x}_b, \Lambda_{aa}^{-1}), \tag{2.3}$$

where, corresponding to the partitioning between $\boldsymbol{x}_a$ and $\boldsymbol{x}_b$, we put

$$\Lambda = \begin{pmatrix} \Lambda_{aa} & \Lambda_{ab} \\ \Lambda_{ba} & \Lambda_{bb} \end{pmatrix}. \tag{2.4}$$

In this case, Λ_{aa} is 2×2, so the inverse can be analytically calculated, giving the off-diagonal element proportional to $\Lambda_{i,j}$. Thus if $\Lambda_{i,j} = 0$, x_i and x_j are statistically independent conditioned on the rest of the variables.

Our first goal is to find a sparse Λ, whose entries are nonzero for essentially coupled pairs, and are zero for weakly correlated pairs that might be induced just by the noise. Such a sparse Λ will represent an essential dependency structure not due to noise, and thus should be useful for detecting correlation anomalies. In real noisy data, however, every entry in the sample covariance matrix S will be nonzero, and the precision matrix Λ cannot be sparse in general. Moreover, if there are highly correlated variables, S becomes rank deficient, and Λ does not even exist. If S is full rank in theory, it is sometimes the case that matrix inversion is numerically unstable when M is more than several tens. This is an essential difficulty in traditional covariance selection procedures [6], where small entries in Λ are set to be zero step by step. Since our assumption is that the data include some highly correlated variables, which holds very generally in sensor data, such approaches are of little use in our context. This motivates us to use an L_1 penalized maximum likelihood approach, as discussed later.

3 Related work

3.1 Anomaly detection from graphs. Anomaly or change detection from sequences of graphs [15] is of particular importance in practice. Sun et al. [23] proposed a method for identifying anomalous nodes by computing the proximities between nodes. Their task is similar to ours, but differs in that they consider a single bi-partite graph rather than comparing two graphs. Sun et al. [22] also studied change detection from a sequence of graphs based on a clustering technique for graph nodes. Node clustering is similar to neighborhood selection, but it is generally hard to be applied to dense graphs. Tong et al. [24] addressed a similar change detection task, but their goal is not to give the anomaly score to the individual nodes. Xuan and Murphy [25] addressed the task of segmenting multivariate time series. While their task differs from ours, they used L_1 penalized maximum likelihood for structure learning, which is the same approach as ours.

3.2 Structure learning. Covariance selection [6] is a standard approach to sparse structure learning. However, it is known that it has several drawbacks in practice such as a high computational cost and a suboptimality in terms of statistical tests. Drton and Perlman [7] addressed mainly the second issue, and proposed an algorithm named SIN, although it does not lift the requirement that the sample covariance matrix must be full rank. Since we are interested in the situation where measurement systems have some redundancy, and hence some of the variables are highly correlated, SIN is less useful to our problem.

In such cases, L_1-penalized regression approaches [20, 5, 3, 9] and a Bayesian sparse learning strategy [19] are promising. There is no consensus in the literature, however, on what is the best method to use to learn sparse structures in these situations. For the task of causal modeling, Arnold et al. [2] compares a number of structure learning algorithms including SIN [7] and lasso [20], although their goal is to learn causal graphs itself, and did not address the issue of data with highly correlated variables.

3.3 Two-sample tests. The two-sample test is a statistical test whose goal is to detect the difference between two data sets. Formally, it attempts to decide whether $p_{\mathrm{A}} = p_{\mathrm{B}}$ or $p_{\mathrm{A}} \neq p_{\mathrm{B}}$, where p_{A} and p_{B} are probability distributions learned from $\mathcal{D}_{\mathrm{A}}$ and $\mathcal{D}_{\mathrm{B}}$, respectively. The two-sample test has a long history in statistics, and a variety of methods have been proposed so far, such as the Kolmogorov-Smirnov test [10] and nearest neighbor test [14]. Although the two-sample problem is similar to ours, but different in that the goal is just to tell how much p_{A} and p_{B} are different, rather than scoring individual variables.

Related to the two-sample test, kernel-based tests for independence have attracted attention in recent years. Gretton et al. [12, 13] proposed kernel-based metrics for the two-sample test and an independence test. Fukumizu et al. [11] proposed to use a covariance operator defined on reproducing kernel Hilbert spaces to test conditional dependence. Their approach can be viewed as an extension of the GGM, and is potentially useful for structure learning from the data having complex correlations as shown in Fig. 3. However, it is still an open problem how to generalize Assumption 1, which is implicitly based on the notion of linear correlation, in accordance with the generalized notion of independence.

This would be interesting future work that is not covered in the present paper.

4 Sparse structure learning

This section considers step (1) in Fig. 1. That is, we consider how to learn a sparse structure from the data. Since this step is common to both the data A and B, we omit the subscript showing A or B for now, and write either of the data as $\mathcal{D} = \{\boldsymbol{x}^{(n)} | n = 1, ..., N\}$. We assume that $\mathcal{D}$ has been standardized to have zero mean and unit variance. Then the sample covariance matrix S is given by

$$\mathsf{S}_{i,j} \equiv \frac{1}{N}\sum_{n=1}^{N} x_i^{(n)} x_j^{(n)}, \tag{4.5}$$

which is the same as the correlation coefficient matrix of this data.

4.1 Penalized maximum likelihood. In the GGM, structure learning is reduced to finding a precision matrix Λ of the multivariate Gaussian (Eq. (2.1)). If we ignore the regularization penalty for sparsity for now, we can get Λ by maximizing the log-likelihood

$$\ln \prod_{t=1}^{N} \mathcal{N}(\boldsymbol{x}^{(t)} | \boldsymbol{0}, \Lambda^{-1}) = \text{const.} + \frac{N}{2}\left\{\ln\det(\Lambda) - \mathrm{tr}(\mathsf{S}\Lambda)\right\},$$

where tr represents the matrix trace (sum over the diagonal elements), and we used a well-known identity $\boldsymbol{x}^{(t)\top}\boldsymbol{x}^{(t)} = \mathrm{tr}(\boldsymbol{x}^{(t)}\boldsymbol{x}^{(t)\top})$ and (4.5). If we use the well-known formulas on matrix derivative

$$\frac{\partial}{\partial\Lambda}\ln\det(\Lambda) = \Lambda^{-1}, \quad \frac{\partial}{\partial\Lambda}\mathrm{tr}(\mathsf{S}\Lambda) = \mathsf{S}, \tag{4.6}$$

we readily get the formal solution $\Lambda = \mathsf{S}^{-1}$. However, as mentioned before, this produces less practical information on the structure of the system, since the sample covariance matrix is often rank deficient and the resulting precision matrix will not be sparse in general.

Therefore, instead of the standard maximum likelihood estimation, we solve an L_1-regularized version of maximum likelihood:

$$\Lambda^* = \arg\max_{\Lambda} f(\Lambda; \mathsf{S}, \rho), \tag{4.7}$$

$$f(\Lambda; \mathsf{S}, \rho) \equiv \ln\det\Lambda - \mathrm{tr}(\mathsf{S}\Lambda) - \rho||\Lambda||_1, \tag{4.8}$$

where $||\Lambda||_1$ is defined by $\sum_{i,j=1}^{M}|\Lambda_{i,j}|$. Thanks to the penalty term, many of the entries in Λ will be exactly zero. The penalty weight ρ is an input parameter, which works as a threshold below which correlation coefficients are thought of as zero, as discussed later.

4.2 Graphical lasso algorithm. Since Eq. (4.7) is a convex optimization problem [3], one can use subgradient methods for solving this. Recently, Friedman, Hastie and Tibshirani [9] proposed an efficient subgradient algorithm named graphical lasso. We recapitulate it in this subsection.

The graphical lasso algorithm first reduces the problem Eq. (4.7) to a series of related L_1 regularized regression problem by utilizing a block coordinate descent technique [3, 8]. Using the formula Eq. (4.6), we see that the gradient of Eq. (4.7) is given by

$$\frac{\partial f}{\partial\Lambda} = \Lambda^{-1} - \mathsf{S} - \rho\,\mathrm{sign}(\Lambda), \tag{4.9}$$

where the sign function is defined so that the (i,j) element of the matrix $\mathrm{sign}(\Lambda)$ is given by $\mathrm{sign}(\Lambda_{i,j})$ for $\Lambda_{i,j} \neq 0$, and a value $\in [-1, 1]$ for $\Lambda_{i,j} = 0$.

To use a block coordinate descent algorithm for solving $\partial f/\partial\Lambda = 0$, we focus on a particular single variable x_i, and partition Λ and its inverse as

$$\Lambda = \begin{pmatrix} L & \boldsymbol{l} \\ \boldsymbol{l}^\top & \lambda \end{pmatrix}, \quad \Sigma \equiv \Lambda^{-1} = \begin{pmatrix} W & \boldsymbol{w} \\ \boldsymbol{w}^\top & \sigma \end{pmatrix}, \tag{4.10}$$

where we assume that rows and columns are always arranged so that the x_i-related entries are located in the last row and column. In these expressions, $W, L \in \mathbb{R}^{(M-1)\times(M-1)}$, $\lambda, \sigma \in \mathbb{R}$, and $\boldsymbol{w}, \boldsymbol{l} \in \mathbb{R}^{M-1}$. Corresponding to this x_i-based partition, we also partition the sample covariance matrix S in the same way, and write as

$$\mathsf{S} = \begin{pmatrix} S^{\setminus i} & \boldsymbol{s} \\ \boldsymbol{s}^\top & s_{i,i} \end{pmatrix}. \tag{4.11}$$

Now let us find the solution of the equation $\partial f/\partial\Lambda = 0$. Since Λ must be positive definite, the diagonal element must be strictly positive. Thus, for the diagonal element, the condition of vanishing gradient leads to

$$\sigma = s_{i,i} + \rho. \tag{4.12}$$

For the off-diagonal entries represented by $\boldsymbol{w}$ and $\boldsymbol{l}$, the optimal solution under which all the other variables are kept constant is obtained by solving

$$\min_{\boldsymbol{\beta}}\left\{\frac{1}{2}||W^{\frac{1}{2}}\boldsymbol{\beta} - \boldsymbol{b}||^2 + \rho\,||\boldsymbol{\beta}||_1\right\} = 0, \tag{4.13}$$

where $\boldsymbol{\beta} \equiv W^{-1}\boldsymbol{w}$, $\boldsymbol{b} \equiv W^{-1/2}\boldsymbol{s}$, and $||\boldsymbol{\beta}||_1 \equiv \sum_l |\beta_l|$. For the proof, see Appendix A.1. This is an L_1-regiularized quadratic programming problem, and again can be solved efficiently with a coordinate-wise subgradient method [9]. The algorithm is sketched in Appendix B.

Now to obtain the final solution Λ^*, we repeat solving Eq. (4.13) for $x_1, x_2, ..., x_M, x_1, ...$ until convergence. Note that the matrix W is full rank due to Eq. (4.12). This suggests a numerical stability of the algorithm. In fact, as shown later, it gives a stable and reasonable solution even when some of the variables are highly correlated.

4.3 Connection to Lasso. The coordinate-wise optimization problem (Eq. (4.13)) derived by the graphical lasso algorithm has clear similarity to the lasso-based structure learning algorithm. The algorithm of Ref. [20] solves separate lasso regression problems for each x_i:

$$\min_{\boldsymbol{\beta}} \left\{ \frac{1}{2} ||\mathsf{Z}_i \boldsymbol{\beta} - \boldsymbol{y}_i||^2 + \mu ||\boldsymbol{\beta}||_1 \right\}, \tag{4.14}$$

where we defined $\boldsymbol{y}_i \equiv (x_i^{(1)}, ..., x_i^{(N)})^\top$, and a data matrix $\mathsf{Z}_i \equiv [\boldsymbol{z}_i^{(1)}, ..., \boldsymbol{z}_i^{(N)}]^\top$ with

$$\boldsymbol{z}_i^{(n)} \equiv (x_1^{(n)}, .., x_{i-1}^{(n)}, x_{i+1}^{(n)}, ..., x_M^{(n)})^\top \in \mathbb{R}^{M-1}.$$

Using the definition of S (Eq. (4.5)), it is easy to see that this problem is equivalent to Eq. (4.13), when

$$W = S^{\backslash i} \quad \text{and} \quad \rho \propto \mu \tag{4.15}$$

are satisfied. Since W is a principal submatrix of Λ^{-1}, we see that there is a correspondence between W and $S^{\backslash i}$ when ρ is small. It will never be satisfied for $\rho > 0$, however. In this sense, the graphical lasso algorithm solves an optimization problem similar to but different from the one in [20]. This fact motivates us to empirically study the difference between the two algorithms as shown in the next section.

4.4 Choosing ρ. We have treated the penalty parameter ρ as a given constant so far. In many regularization-based machine learning methods, how to choose the penalty parameter is a subtle problem. In the present context, however, ρ should be treated as an input parameter since our goal is not to find the "true" structure but to reasonably select the neighborhood.

To get insights on how to relate ρ with the neighborhood size, we note the following result:

PROPOSITION 1. *If we consider a 2×2 problem defined only by two variables x_i and x_j ($i \neq j$), the off-diagonal element of the optimal Λ as the solution to Eq.* (4.7) *is given by*

$$\Lambda_{i,j} = \begin{cases} -\frac{\mathrm{sign}(r)(|r|-\rho)}{(1+\rho)^2-(|r|-\rho)^2} & \text{for } |r| > \rho \\ 0 & \text{for } |r| \leq \rho, \end{cases}$$

where r is the correlation coefficient between the two variables.

For the proof, see Appendix A.2.

Although this is not the solution to the full system, it gives us a useful guide about how to choose ρ. For example, if a user wishes to think of dependencies corresponding to absolute correlation coefficients less than 0.5 as noise, then the input ρ should be less than the intended threshold, and possibly a value around $\rho = 0.3$ would work. If ρ is close to 1, resulting neighborhood graphs will be very small, while a value close to 0 leads to an almost complete graph where all the variables are thought of as being connected.

This property is very useful given the neighborhood preservation assumption. In Section 2, we saw that the correlation coefficients are subject to strong fluctuations in many highly dynamic systems unless their magnitude is close to 1. It is evident, however, that simply letting some entries be zero with a threshold does not maintain the mathematical consistency as a graphical Gaussian model. In addition, derived results can be sensitive to the threshold value. Sparse structure learning allows us to reduce the undesired effects of noise by fitting a sparse model in a theoretically consistent fashion.

We should also note that sparse structure learning allows us to conduct neighborhood selection in an adaptive manner. If a variable is isolated with almost no dependencies on others, the number of selected neighbors will be zero. Also, we naturally expect that variables in a tightly-connected cluster would select the cluster members as their neighbors. We will see, however, that the situations when there are highly correlated variables are much trickier than it seems.

5 Scoring Correlation Anomalies

Suppose that based on the algorithm in the previous section, we have obtained two sparse GGMs, $p_{\mathrm{A}}(\boldsymbol{x})$ and $p_{\mathrm{B}}(\boldsymbol{x})$. In this section, we discus how to define the anomaly score for each variable, given these models.

5.1 Expected Kullback-Leibler divergence. Our final goal is to quantify how much each variable contributes to the difference between $\mathcal{D}_{\mathrm{A}}$ and $\mathcal{D}_{\mathrm{B}}$ in terms of each variable. Given the probabilistic models, the most natural difference measure is the Kullback-Leibler (KL) divergence. Let us focus on a variable x_i for a while, and consider the following quantity

$$d_i^{\mathrm{AB}} \equiv \int \mathrm{d}\boldsymbol{z}_i \; p_{\mathrm{A}}(\boldsymbol{z}_i) \int \mathrm{d}x_i \; p_{\mathrm{A}}(x_i|\boldsymbol{z}_i) \ln \frac{p_{\mathrm{A}}(x_i|\boldsymbol{z}_i)}{p_{\mathrm{B}}(x_i|\boldsymbol{z}_i)}. \tag{5.16}$$

This is the expected KL divergence between $p_{\mathrm{A}}(x_i|\boldsymbol{z}_i)$ and $p_{\mathrm{B}}(x_i|\boldsymbol{z}_i)$, integrated over the distribution $p_{\mathrm{A}}(\boldsymbol{z}_i)$. By replacing A with B in Eq. (5.16), we also obtain the definition of d_i^{AB}. Since we are working with Gaussians, the integral can be analytically performed. The result is

$$\begin{aligned} d_i^{\mathrm{AB}} &= \boldsymbol{w}_{\mathrm{A}}^\top (\boldsymbol{l}_{\mathrm{B}} - \boldsymbol{l}_{\mathrm{A}}) \\ &+ \frac{1}{2} \left\{ \frac{\boldsymbol{l}_{\mathrm{B}}^\top \mathsf{W}_{\mathrm{A}} \boldsymbol{l}_{\mathrm{B}}}{\lambda_{\mathrm{B}}} - \frac{\boldsymbol{l}_{\mathrm{A}}^\top \mathsf{W}_{\mathrm{A}} \boldsymbol{l}_{\mathrm{A}}}{\lambda_{\mathrm{A}}} \right\} \\ &+ \frac{1}{2} \left\{ \ln \frac{\lambda_{\mathrm{A}}}{\lambda_{\mathrm{B}}} + \sigma_{\mathrm{A}}(\lambda_{\mathrm{B}} - \lambda_{\mathrm{A}}) \right\}, \end{aligned} \tag{5.17}$$

where we partitioned $\Lambda_{\rm A}$ and its inverse $\Sigma_{\rm A}$ as

(5.18)
$$\Lambda_{\rm A} = \begin{pmatrix} L_{\rm A} & \boldsymbol{l}_{\rm A} \\ \boldsymbol{l}_{\rm A}^\top & \lambda_{\rm A} \end{pmatrix},\ \Sigma_{\rm A} \equiv \Lambda_{\rm A}^{-1} = \begin{pmatrix} W_{\rm A} & \boldsymbol{w}_{\rm A} \\ \boldsymbol{w}_{\rm A}^\top & \sigma_{\rm A} \end{pmatrix},$$

respectively (see Eq. (4.10)). A similar partition is also applied to $\Lambda_{\rm B}$ and $\Sigma_{\rm B}$. The definition of $d_i^{\rm BA}$ is obtained by replacing A with B in the above. The derivation of Eq. (5.17) is straightforward if the standard partitioning formula (see Eq. (2.3)) is used.

The definition (5.17) has a clear interpretation. By definition of GGMs, the number of nonzero entries in $\boldsymbol{l}_{\rm A}$ is the same as the degree of the node x_i. In this sense, $\boldsymbol{l}_{\rm A}$ contains the information on the neighborhood graph of x_i. Thus the first term mainly detects the change of the degree. The second term corresponds to the difference in the "tightness" of the neighborhood graph. Specifically, if x_i has a single link to j, this term is proportional to the difference between corresponding correlation coefficient, normalized by the single variable precisions $\lambda_{\rm A}$ and $\lambda_{\rm B}$. The third term is related to the change in single variable precisions (or variances).

5.2 Anomaly score. $d_i^{\rm AB}$ and $d_i^{\rm BA}$ are quantities that measure the change in the neighborhood graph of the i-th node. The greater these quantities are, the greater change we have concerning x_i. Thus, given the assumption of neighborhood preservation, it is reasonable to define the anomaly score of the i-th variable as

(5.19)
$$a_i \equiv \max\{d_i^{\rm AB},\ d_i^{\rm BA}\}$$

This definition is a natural extension of a prior proposal of [16]. One of the drawbacks of that approach is that it simply uses the k-NN strategy for neighborhood selection. Also, due to a heuristic definition of the dissimilarity, it cannot detect anomalies caused by sign changes such as $x_i \to -x_i$. In the present study, we propose an information-theoretic definition of the anomaly score, which detect any type of anomaly that affects the probability distribution, in principle.

5.3 Algorithm summary. Our method for scoring correlation anomalies consists of two steps. The first step learns a sparse structure, and the second step is to compute the anomaly score of each variable.

1. Input:
   - Reference and target data sets $\mathcal{D}_{\rm A}$ and $\mathcal{D}_{\rm A}$.
   - Penalty parameter ρ.
2. Output: Individual anomaly scores $a_1, ..., a_M$.
3. Algorithm:
   (a) Compute correlation matrices $\mathsf{S}_{\rm A}$ and $\mathsf{S}_{\rm B}$ using Eq. (4.5).
   (b) Use graphical lasso to obtain precision matrices $\Lambda_{\rm A}$ and $\Lambda_{\rm B}$, and also obtain their inverse $\Sigma_{\rm A}$ and $\Sigma_{\rm B}$ as side products.
   (c) Compute discrepancies $d_i^{\rm AB}$ and $d_i^{\rm BA}$ using Eq. (5.17) to obtain anomaly score a_i for $i = 1, ..., M$.

Finally, we briefly examine at the complexity of the algorithm. As shown in Eq. (4.5), the cost to compute the covariance matrix is $O(M^2N)$. For computing the precision matrix, the graphical lasso algorithm needs $O(M^3)$ cost in the worst case. While the behavior of the algorithm is still not fully understood, it is known in practice that the cost can be sub-cubic in the sparse case [9]. Systematic analysis on the complexity of structure learning algorithms would be an interesting future work.

6 Experiments

In this section, we first compare different structure learning algorithms with particular emphasis on the stability under collinearities. Then we test the proposed correlation anomaly metric using real-world car sensor data.

6.1 Comparing structure learning algorithms. Considering the fact that the traditional covariance selection procedures face evident difficulty with data having highly correlated variables, studying the stability of L_1-penalized learning algorithm is of particular interest. We compared the graphical lasso algorithm (denoted by `Glasso`) with two other structure learning algorithms.

The first alternative (denoted by `Lasso`) is the method due to Meinshausen and Bühlmann [20], where lasso regression is done for each variable using the others as predictors. They showed that their approach satisfies a form of statistical consistency. In practice, however, it is known that their algorithm tends to over-select neighbors [21, 5]. As an alternative, it has been proposed [5] to use an adaptive lasso algorithm [26] for sparse structure learning. Adaptive lasso, denoted by `AdaLasso`, is a two-stage regression algorithm where the results of the first regression is used to improve the second stage lasso regression. Here we use a method which uses lasso also in the first stage, as suggested in Ref. [5]. Since we are interested in the situations where some of the variables are highly correlated, and hence S is rank deficient, traditional types of approaches [18, 7] based on direct estimation of the precision matrix are out of our scope.

Data and evaluation measure. We tested the stability of structure learning algorithms by comparing learned structures before and after adding white noise. The data used was *Actual spot rates* data as explained in Section 2. We generated 25 subsets of the data by using non-overlapping win-

dows containing 100 consecutive days, and applied the three algorithms to each one, changing the value of the penalty parameter. We then computed the sparsity defined by

$$(\text{sparsity}) \equiv \frac{N_0}{M(M-1)},$$

where N_0 is the number of zeros in the off-diagonal elements of Λ.

After the first learning, we added zero-mean Gaussian noise to the data as $x_i \leftarrow x_i + \epsilon_i$, where ϵ_i is independent and identically distributed Gaussian noise. Then we computed the probability of edge flip formally defined by

$$(\text{flip probability}) \equiv N_1/N_0,$$

where N_1 is the number of edges that are flipped (i.e. either appeared or disappeared) by the noise.

Results. Figure 4 shows the result, where the flip probability is shown as a function of the sparsity. We used white noise with the standard deviation of only 0.1 (applied after standardization of the entire data). From the figure, we see that there are considerable instabilities in `Lasso` and `AdaLasso`. With these algorithms, the flip probability is on the order of 50% at sparsity of 0.5. On the other hand, `Glasso` is much more stable under noise. The instability of `Lasso` and `AdaLasso` can be understood from the general tendency that lasso tends to select only one of the correlated features. In the *Actual spot rates* data, European currencies such as BEF, FRF, DEM, and NLG are highly correlated, and which one is selected as neighbors is almost determined by chance. Although this kind of parsimonious behavior is quite useful in regression in terms of generalization ability, it is really tricky in structure learning.

To conclude this subsection, the separated regression strategy adopted in `Lasso` and `AdaLasso` cannot reproduce stable structures when there are correlated variables in the data. In contrast, `Glasso` gives reasonably stable structures.

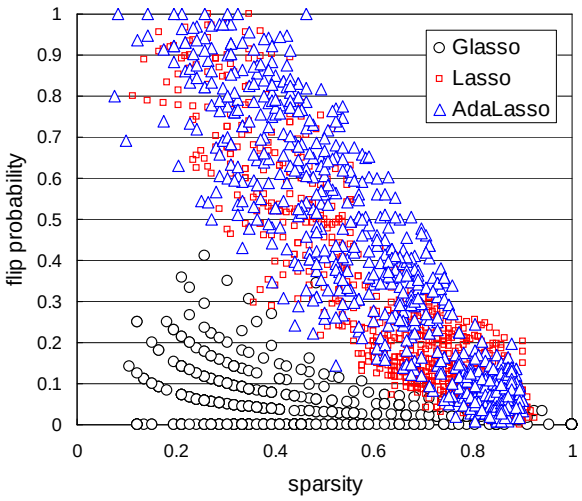

Figure 4: Edge flip probability as a function of sparsity, showing considerable instabilities in `Lasso` and `AdaLasso`.

Table 3: Compared anomaly metrics and their best AUC values.

symbol	neighborhood	metric	best AUC
`KL`	`Glasso`	Eq. (5.17)	**0.96** ($\rho = 0.3$)
`SNG`	`Glasso`	Eq. (6.20)	0.93 ($\rho = 0.7$)
`SNN`	k-NN	Eq. (6.20)	0.87 ($k = 2$)
`LR`	`Glasso`	Eq. (6.21)	0.81 ($\rho = 0.5$)

6.2 Comparing anomaly scores. We compared four different definitions of anomaly scores. The fist one is our proposed metric, the expected conditional KL divergence (denoted by `KL`). The second and third one are based on a correlation anomaly score in Ref. [16], which can be written as

$$d_i^{\mathrm{AB}} = \left| \frac{\tilde{\boldsymbol{l}}_{\mathrm{A}}^\top (\boldsymbol{s}_{\mathrm{A}} - \boldsymbol{s}_{\mathrm{B}})}{(1 + \tilde{\boldsymbol{l}}_{\mathrm{A}}^\top \boldsymbol{s}_{\mathrm{A}})(1 + \boldsymbol{s}_{\mathrm{B}}^\top \tilde{\boldsymbol{l}}_{\mathrm{A}})} \right|, \tag{6.20}$$

where $\tilde{\boldsymbol{l}}_{\mathrm{A}}$ represents the indicator vector whose element corresponding to x_j is 1 if x_j is a neighbor of x_i, 0 otherwise. Also, we assumed the same partition as Eq. (4.11) for the sample covariance matrix of $\mathcal{D}_{\mathrm{A}}$. To obtain the indicator vector, the second metric denote by `SNG` (stochastic neighborhood + `Glasso`) uses `Glasso`, and set each element to be 1 if nonzero, while the third metric denoted by `SNN` (stochastic neighborhood + k-NN) simply uses the k-NN method according to the absolute values of the correlation coefficients, as proposed in Ref. [16].

Finally, the fourth definition of anomaly score is based on the likelihood ratio, and defined as

$$d_i^{\mathrm{AB}} = 1 - \prod_{n=1}^{N_{\mathrm{A}}} \frac{p_{\mathrm{A}}(x_{Ai}^{(n)} | \boldsymbol{z}_{Ai}^{(n)})}{p_{\mathrm{B}}(x_{Ai}^{(n)} | \boldsymbol{z}_{Ai}^{(n)})}. \tag{6.21}$$

If the data $\mathcal{D}_{\mathrm{A}}$ perfectly fits to both p_{A} and p_{B}, d_i^{AB} will be 0. Otherwise, it takes a value less than 1. In the above definitions, d_i^{BA} is obtained by replacing A with B, and the final score is defined by $\max\{d_i^{\mathrm{AB}}, d_i^{\mathrm{BA}}\}$, as in Eq. (5.16).

Data. To demonstrate the utility of our approach, we used *sensor_error* data,[1] which are based on many experimental runs with prototype cars. The experiments were originally designed to check the behaviors when a driver suddenly brakes, and thus the signals are highly nonstationary.

[1] Correlation coefficient matrices generated from the raw data are available on request.

The data are preprocessed to have zero mean, unit variance, a 0.1 second interval, and no monotonic trends. Our observations showed that correlations along the time axis are not considerable, thus time-series modeling is less useful.

This *sensor_error* data includes 79 experimental runs under normal system operation, and 20 runs in a faulty state. Each run contains about $N = 150$ points of $M = 44$ variables. Since N is on the same order as M, traditional asymptotic theories in statistics are hard to be applied. Anomalies included in the faulty runs are due to sensor miswiring errors, and we arranged the data so that each of the faulty runs includes two faulty sensors at x_{24} and x_{25}. In general, miswiring errors are very hard to detect since the individual sensors are healthy.

Figure 5 shows examples of pairwise scatter plots, where only four variables out of the $M = 44$ variables were chosen from particular runs as described in the caption. In this example, x_{24} is one of the error variables. This is suggested by the disappearance of linear correlations. However, considering the heavy fluctuations of the pairwise trajectories, detecting anomalies of this kind is very hard with existing methods such as statistical tests on correlation coefficients based on the Wishart distribution theory [1].

Evaluation measures. In our problem setting, there are $20 \times 79 = 1580$ possible tests between the reference and faulty runs. To summarize the results, we use the ROC (Receiver Operating Characteristic) curve, which represents the averaged relationship between the detection rate (how many truly faulty variables are picked up) and the data coverage (how many variables are looked at). In this case, a ROC curve is plotted by counting the number of detected faulty variables at each value of the data coverage, $0, \frac{1}{M}, \frac{2}{M}, ..., 1$. We also use AUC (Area Under Curve) to compare the goodness of different ROC curves.

Results. Figures 6-8 show ROC curves for $\rho = 0.3, 0.5$ and 0.7, where the dashed line is also plotted to represent a random selection. Regarding SNN, we plotted the same curve with $k = 2$ in the figures, which gave the best AUC value. Comparing four metrics, we first see that the LR score is much worse than the others. This can be explained by the fact that LR uses the data in computing the score as well as in building the model. Since the data are extremely noisy, this strategy will be more sensitive to the unwanted effects of the noise.

Table 3 summarizes the best AUC values for each definition of the score. We see that KL and SNG outperform SNN, demonstrating the utility of the adaptive neighborhood selection. At the value of $\rho = 0.3$, our observation shows that the links that have the absolute correlation coefficients less than about 0.6 were pruned in this data (the mean sparsity was about 0.90 for the reference runs). Considering the neighborhood preservation assumption and the heavy fluctuation as shown in Fig. 5, this looks a reasonable thresholding. It

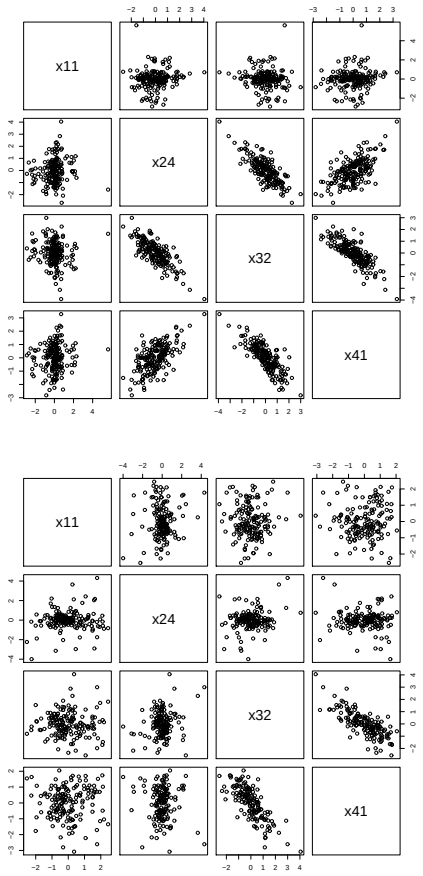

Figure 5: Pairwise scattering plot of *sensor_error* data. Top: The 10th reference run. Bottom: The third faulty run.

is interesting to see that SNG gets better than KL when ρ is more than about 0.5, where learned structures are very sparse (the mean sparsity was about 0.98 for the reference runs at $\rho = 0.7$). In this regime, the contribution of the individual variances represented by σ_{A} and λ_{B} etc. are relatively important. Since SNG uses a simple definition without individual variance terms, it is more robust to the variations of the individual signals. However, further theoretical and empirical analysis is left to the future work.

7 Conclusion

We have proposed a framework that applies sparse structure learning to anomaly detection. Our task was to compute the anomaly scores of individual variables, rather than simply detecting that two data sets are different. To the best of our knowledge, this is the first work that tackles this task using sparse structure learning.

We demonstrated that recently proposed sparse structure learning methods are highly instable when collinearities exist in the data. Therefore, those methods are of limited use for

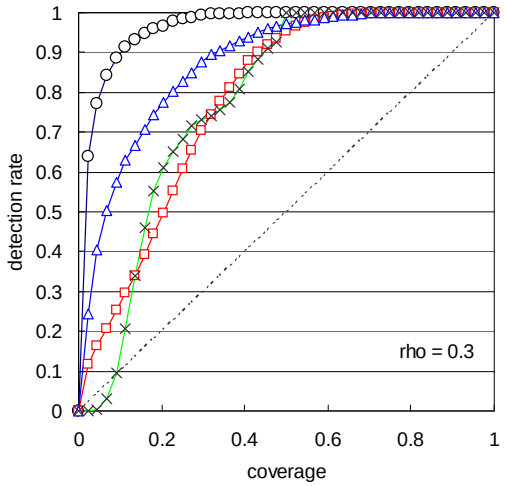

Figure 6: ROC curves for $\rho = 0.3$, comparing KL ($\circ$), SNG ($\square$), SNN ($\triangle$), and LR ($\times$).

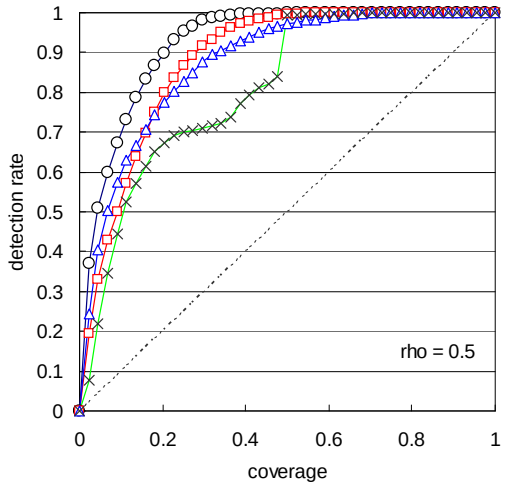

Figure 7: ROC curves for $\rho = 0.5$, comparing KL ($\circ$), SNG ($\square$), SNN ($\triangle$), and LR ($\times$).

real-valued sensor data in many cases. Our experimental results showed, however, that the graphical lasso algorithm successfully avoids this serious difficulty.

We compared a number of different metrics for scoring correlation anomalies using a real-world automotive sensor data set, and showed that the proposed conditional KL divergence metric significantly improves the performance over existing metrics.

APPENDIX

A Proofs

A.1 Proof of Eq. (4.13)**.** Based on the partitioning in Eq.(4.10), the upper right part of the equation $\partial f/\partial \Lambda = 0$ is readily written as

$$\boldsymbol{w} - \boldsymbol{s} - \rho\,\mathrm{sign}(\boldsymbol{l}) = 0. \tag{B-1}$$

Since $\Sigma\Lambda = \mathsf{I}_M$, we have

$$\Sigma\Lambda = \begin{pmatrix} WL + \boldsymbol{w}\boldsymbol{l}^\top & W\boldsymbol{l} + \lambda\boldsymbol{w} \\ \boldsymbol{l}^\top W + \lambda\boldsymbol{w}^\top & \boldsymbol{w}^\top\boldsymbol{l} + \sigma\lambda \end{pmatrix} = \begin{pmatrix} \mathsf{I}_{M-1} & \boldsymbol{0} \\ \boldsymbol{0}^\top & 1 \end{pmatrix}. \tag{B-2}$$

If we use the upper right part of this identity, we see

$$\boldsymbol{l} = -\lambda W^{-1}\boldsymbol{w} = -\lambda\boldsymbol{\beta}, \tag{B-3}$$

where we defined $\boldsymbol{\beta} \equiv W^{-1}\boldsymbol{w}$. Since Λ is positive definite, λ must be positive. Thus $\mathrm{sign}(\boldsymbol{l}) = -\mathrm{sign}(\boldsymbol{\beta})$ holds. Using this, we see that Eq. (B-1) is equivalent to

$$\frac{\partial}{\partial\boldsymbol{\beta}}\left\{\frac{1}{2}\boldsymbol{\beta}^\top W\boldsymbol{\beta} - \boldsymbol{\beta}^\top\boldsymbol{s} + \rho\,||\boldsymbol{\beta}||\right\} = 0. \tag{B-4}$$

If we let $W^{-1/2}\boldsymbol{\beta}$ be $\boldsymbol{b}$, it is evident that this is equivalent to Eq. (4.13).

If we got the solution $\boldsymbol{\beta}$ anyway, we update the corresponding columns of Λ by

$$\lambda = \frac{1}{\sigma - \boldsymbol{\beta}^\top W\boldsymbol{\beta}}, \quad \boldsymbol{l} = -\frac{\boldsymbol{\beta}}{\sigma - \boldsymbol{\beta}^\top W\boldsymbol{\beta},}, \tag{B-5}$$

where we used the lower right part of Eq. (B-2) $\boldsymbol{w}^\top\boldsymbol{l} + \sigma\lambda = 1$ and Eq. (B-3). Also, using the upper right part of Eq. (B-2), we update $\boldsymbol{w}$ as

$$\boldsymbol{w} = -W\boldsymbol{l}/\lambda.$$

Note that σ is kept constant because of Eq. (4.12). Therefore, in the graphical lasso algorithm, $\Sigma = \Lambda^{-1}$ is given as a side product of Λ, without making any explicit inversion.

A.2 Proof of Proposition 1. If $M = 2$, the objective function Eq. (4.7) is explicitly written as

$$\begin{aligned} f(\Lambda; \mathsf{S}, \rho) &= \ln(\lambda_{11}\lambda_{22} - \lambda_{12}^2) \\ &- (1+\rho)(\lambda_{11} + \lambda_{22}) - 2(r\lambda_{12} + \rho|\lambda_{12}|), \end{aligned}$$

where r is the correlation coefficient (or the off-diagonal element of S), and λ_{ij} is the (i,j) element of Λ. From equations $\partial f/\partial\lambda_{11} = 0$ and $\partial f/\partial\lambda_{22} = 0$, we easily see that

$$\lambda_{11} = \lambda_{22} = \frac{1}{2}\left\{\frac{1}{1+\rho} + \sqrt{\frac{1}{(1+\rho)^2} + 4\lambda_{12}}\right\}. \tag{B-6}$$

From the other condition $\partial f/\partial\lambda_{12} = 0$, we have

$$-\frac{1}{2}\frac{\partial f}{\partial\lambda_{12}} = (1+\rho)\frac{\lambda_{12}}{\lambda_{11}} + r + \rho\,\mathrm{sign}(\lambda_{12}) = 0, \tag{B-7}$$

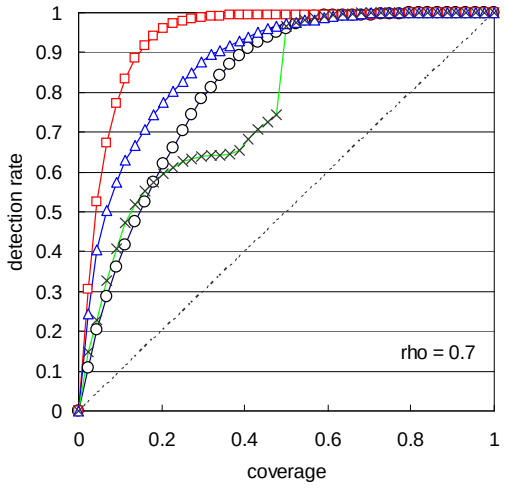

Figure 8: ROC curves for $\rho = 0.7$, comparing KL ($\circ$), SNG ($\square$), SNN ($\triangle$), and LR ($\times$).

where we used $\partial f/\partial\lambda_{11} = 0$ to simplify the first term. From this equation, we see that if $\lambda_{12} > 0$ then $-1 < r < -\rho$, while if $\lambda_{12} < 0$ then $\rho < r < 1$. Noting this, and solving simultaneous equations Eqs. (B-6) and (B-7) with respect to $|\lambda_{12}|$, we obtain Proposition 1 after some algebra.

B Subgradient algorithm for Eq. (4.13)

In this Appendix, we explain how to solve the L_1-regiularized quadratic programming problem defined in Eq. (4.13) using a coordinate-wise subgradient method.

Instead of Eq. (4.13), consider the equivalent expression of Eq. (B-4). Differentiating with respect to β_i, we have

$$\sum_m W_{i,m}\beta_m - s_i + \rho\,\mathrm{sign}(\beta_i) = 0.$$

For $\beta_i > 0$, a formal solution to this equation is given by

$$\beta_i = \frac{1}{W_{i,i}}(A_i - \rho),$$

where we defined

$$A_i \equiv s_i - \sum_{m\neq i} W_{i,m}\beta_m. \tag{B-8}$$

Since $W_{i,i} > 0$, this solution must satisfy $A_i > \rho$. If this condition does not satisfied, the minimum of the objective function is at $\beta_i = 0$, since its gradient is positive in this case. Similarly, considering also the $\beta_i < 0$ case, we have an updated equation as

$$\beta_i \leftarrow \begin{cases} (A_i - \rho)/W_{i,i} & \text{for } A_i > \rho \\ 0 & \text{for } -\rho < A_i < \rho \\ (A_i + \rho)/W_{i,i} & \text{for } A_i < -\rho \end{cases}$$

for each i. This is repeated until convergence.

C Lasso-based structure learning algorithms

In this appendix, we recapitulate the methods compared to the proposed approach.

C.1 Lasso. In Lasso [20], we build an L_1-regularized regression model to each variable, using the others as predictors. Specifically, for a variable x_i, solve Eq. (4.14) to get the coefficient $\boldsymbol{\beta}$. Since this coefficient predict the target variable x_i as $\boldsymbol{\beta}^\top \boldsymbol{z}_i$, comparison with the partioning formula of Gaussian (as Eq. (2.3)) gives one column of the precision matrix (see Eq. (4.10))

$$\lambda = 1/\tilde{\sigma_i}^2$$
$$\boldsymbol{l} = -\boldsymbol{\beta}/\tilde{\sigma_i}^2,$$

where $\tilde{\sigma_i}^2$ is the estimated predictive variance. If one uses a maximum likelihood estimator, this is given by

$$\tilde{\sigma_i}^2 = \frac{1}{N}\sum_{n=1}^{N}(x_i^{(n)} - \boldsymbol{\beta}^\top \boldsymbol{z}_i^{(n)})^2.$$

By repeating regression like this, we can build the whole precision matrix.

C.2 Adaptive lasso. In AdaLasso [26, 5], we proceed as follows.

1. Find a lasso regression coefficient vector $\boldsymbol{\beta}$ for a variable x_i.
2. Modify the predictor $\boldsymbol{z}_i^{(n)}$ by making element-wise product between $\boldsymbol{z}_i^{(n)}$ and $\boldsymbol{\beta}$.
3. Solve Eq. (4.14) based on the modified data matrix.
4. Build a precision matrix in the same way as above.
5. Proceed to another i.

References

[1] T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley-Interscience, 3rd. edition, 2003.

[2] A. Arnold, Y. Liu, and N. Abe. Temporal causal modeling with graphical Granger methods. In *Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 66–75, 2007.

[3] O. Banerjee, L. E. Ghaoui, and G. Natsoulis. Convex optimization techniques for fitting sparse Gaussian graphical models. In *Proc. Intl. Conf. Machine Learning*, pages 89–96. Press, 2006.

[4] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag, 2006.

[5] P. Bühlmann. Variable selection for high-dimensional data: with applications in molecular biology. 2007.

[6] A. P. Dempster. Covariance selection. *Biometrics*, 28(1):157–175, 1972.

[7] M. Drton and M. D. Perlman. A SINful approach to Gaussian graphical model selection. *Journal of Statistical Planning and Inference*, 138(4):1179–1200, 2008.

[8] J. Friedman, T. Hastie, H. Höfling, and R. Tibshirani. Pathwise coordinate optimization. *Annals of Applied Statistics*, 1(2):302–332, 2007.

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

[10] J. H. Friedman and L. C. Rafsky. Multivariate generalizations of the Wolfowitz and Smirnov two-sample tests. *Annals of Statistics*, 7:697–717, 1979.

[11] K. Fukumizu, A. Gretton, X. Sun, and B. Schölkopf. Kernel measures of conditional dependence. In *Advances in Neural Information Processing Systems 20*.

[12] A. Gretton, K. Borgwardt, M. Rasch, B. Schölkopf, and A. J. Smola. A kernel method for the two-sample-problem. In *Advances in Neural Information Processing Systems 19*, pages 513–520. MIT Press, 2007.

[13] A. Gretton, K. Fukumizu, C. H. Teo, L. Song, B. Schölkopf, and A. Smola. A kernel statistical test of independence. In *Advances in Neural Information Processing Systems 20*, pages 585–592, 2008.

[14] Z. Henze. A multivariate two-sample test based on the number of nearest neighbor type coincidences. *Annals of Statistics*, 16:772–783, 1988.

[15] T. Idé and H. Kashima. Eigenspace-based anomaly detection in computer systems. In *Proc. ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 440–449, 2004.

[16] T. Idé, S. Papadimitriou, and M. Vlachos. Computing correlation anomaly scores using stochastic nearest neighbors. In *Proc. IEEE Intl. Conf. Data Mining*, pages 523–528, 2007.

[17] E. Keogh and T. Folias. The UCR time series data mining archive [`http://www.cs.ucr.edu/~eamonn/TSDMA/index.html`]. 2002.

[18] S. L. Lauritzen. *Graphical Models*. Oxford, 1996.

[19] F. Li and Y. Yang. Using modified lasso regression to learn large undirected graphs in a probabilistic framework. In *Proc. National Conf. Artificial Intelligence*, pages 801–806, 2005.

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

[21] R. Opgen-Rhein and K. Strimmer. Learning causal networks from systems biology time course data: an effective model selection procedure for the vector autoregressive process. *BMC Bioinformatics*, 8(Suppl.2):S3, 2007.

[22] J. Sun, C. Faloutsos, S. Papadimitriou, and P. S. Yu. GraphScope: parameter-free mining of large time-evolving graphs. In *Proc. the 13th ACM SIGKDD Intl. Conf. Knowledge Discovery and Data Mining*, pages 687–696, 2007.

[23] J. Sun, H. Qu, D. Chakrabarti, and C. Faloutsos. Neighborhood formation and anomaly detection in bipartite graphs. In *Proc. IEEE Intl. Conf. Data Mining*, pages 418–425, 2005.

[24] H. Tong, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Proximity tracking on time-evolving bipartite graphs. In *Proc. 2008 SIAM Intl. Conf. Data Mining*, pages 704–715, 2008.

[25] X. Xuan and K. Murphy. Modeling changing dependency structure in multivariate time series. In *Proc. the 24th Intl. Conf. Machine Learning*, pages 1055–1062, 2007.

[26] H. Zou. The adaptive lasso and its oracle properties. *Journal of the American Statistical Association*, 101(476):1418–1429, 2006.