Learning Linear Dependency Trees from Multivariate Time-series Data

Jarkko Tikka and Jaakko Hollmén
Helsinki University of Technology
Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland
tikka@mail.cis.hut.fi, Jaakko.Hollmen@hut.fi

Abstract

Representing interactions between variables in large data sets in an understandable way is usually important and hard task. This article presents a methodology how a linear dependency structure between variables can be constructed from multivariate data. The dependencies between the variables are specified by multiple linear regression models. A sparse regression algorithm and bootstrap based resampling are used in the estimation of models and in construction of a belief graph. The belief graph highlights the most important mutual dependencies between the variables. Thresholding and graph operations may be applied to the belief graph to obtain a final dependency structure, which is a tree or a forest. In the experimental section results of the proposed method using real-world data set were realistic and convincing.

1. Introduction

Large data sets are available from many different sources, for example from industrial processes, economy, mobile communications network, and environment. Deeper understanding of the underlying process can be achieved by exploring or analyzing the data. Economical or ecological benefits are a great motivation for the data analysis.

In this study, dependencies between the variables in data set are analyzed. The purpose is to estimate multiple linear regression models and learn a linear dependency tree or forest of the variables. The dependency structure clearly shows how a change in a value of one variable induces changes in values of other variables. This might be useful information in many cases, for instance, if values of some variable cannot be controlled directly.

The multiple linear regression models have a couple of advantages. The dependencies in linear models are easy to interpret. In addition, processes may be inherently linear or over short ranges many processes can be approximated by a linear model.

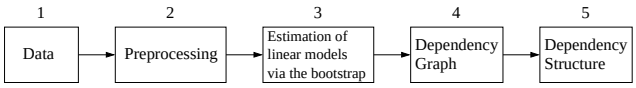

Figure 1. The flow chart of the proposed method.

A flow chart of the methodology proposed in this study is presented in Figure 1. The method consists of five phases. First, there should be some multivariate data available. The data do not necessarily have to be time-series data, although time-series data is used as an example in this study.

The second phase deals with the preprocessing of data. Some operations have to be usually performed on measurements before they can be analyzed mathematically. Some measurements may be missing or measurements can be noisy.

In the third phase, as many multiple linear regression models as there are variables in the data are estimated. Each variable is a dependent variable in turn and the rest of the variables are possible independent variables. The most significant independent variables for each model are selected using the bootstrap and a sparse regression algorithm. The relative weights of the regression coefficients are computed from the bootstrap replications. The relative weight of the regression coefficient measures a belief that the corresponding independent variable belongs to the estimated linear model.

In the fourth phase, a belief graph is constructed from the relative weights of the regression coefficients. The belief graph represents the strength of the dependencies between the variables. In the belief graph there are as many nodes as there are variables in the data. The relative weights define arcs of the belief graph. A predefined threshold value and a moralizing operation are applied to the belief graph

resulting a moral graph or a final dependency graph.

Finally, a dependency structure of the variables is calculated from the dependency graph. A set of variables, which forms a multiple linear regression model, belongs to a same maximal clique. However, the formulation of final dependency structure is restricted such that the dependencies cannot form circles in a final structure i.e. the variable cannot be dependent on itself through the other variables. Thus, the final dependency structure is a tree or a forest.

The rest of the article is organized as follows. In Section 2 a few other similar studies are briefly described. The multiple linear regression model and sparse regression algorithms are introduced in the beginning of Section 3, followed by the bootstrap and the computation of the relative weights of the regression coefficients. The construction of linear dependency tree or forest is proposed in Section 4. The proposed method is applied to real-world data set. The description of data and the results of experiments are shown in Section 5. The experiments mainly serve an illustrative example of the proposed methodology. Conclusion and final remarks are in Section 6.

2 Related work

To our knowledge, novelty of this work is in the sparse construction of linear models and the application of the bootstrap. Several studies about dependencies between the variables in multivariate data are accomplished, for example [3], [13], [11], [16], and [20].

Dependency trees are also used in [3]. A method which approximates optimally a d-dimensional probability distribution of the d variables is shown. Each variable can only be dependent on at most one variable in that model, when in this study one variable can be dependent on several variables.

Belief networks are discussed in [13]. The belief network induces a conditional probability distribution over its variables. The belief networks are directed and acyclic. Dependency networks which can be cyclic are presented in [11]. In both belief and dependency networks the variables are conditioned upon its parent variables. The directed dependency means that changes in the parent has effect on the child. The undirected dependency means that changes are induced into the both directions. In this study, continuous variables are only modeled, whereas the belief and the dependency network can be used with discrete variables.

Independent variable group analysis (IVGA) is proposed in [16]. In that approach the variables are clustered. The variables in one cluster are dependent on each other but they are independent on the variables which belong to other clusters. In IVGA, the dependencies between the groups or clusters are ignored and the dependencies in each group can be modeled in different ways.

Structural equation modeling (SEM) [20] is another technique to investigate relationships between the variables. SEM provides a methodology to test a plausibility of hypothesized models. The predefined dependencies between the variables are investigated using the SEM, when the dependencies are learned from the data using the method proposed in this study. Structural Equation models can consist of both observed and latent variables. The latent variables can be extracted from the observed ones using for example the factor analysis. Observed variables are only modeled in this study.

3 Methods

3.1 Multiple linear regression

The dependencies between the variables are modeled using the multiple linear regression. The model is

$$y_t = \beta_1 x_{t,1} + \beta_2 x_{t,2} + \ldots + \beta_k x_{t,k} + \epsilon_t, \qquad (1)$$

where y_t is the dependent variable, $x_{t,i}, i = 1, \ldots, k$ are the independent variables, $\beta_i, i = 1, \ldots, k$ are the corresponding regression coefficients, and ϵ_t is normally distributed random noise with zero mean and unknown variance $\epsilon_t \sim N(0, \sigma^2)$. The index $t = 1, \ldots, N$ represents the tth observation of the variables y and x_i and N is the sample size.

Equation (1) can also be written in matrix form as follows

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon}. \qquad (2)$$

Here we assume that the variables are normalized to zero mean and thus, there is no need for a constant term in models (1) and (2).

The ordinary least squares (OLS) solution is

$$\boldsymbol{b}_{OLS} = (\boldsymbol{X}^T\boldsymbol{X})^{-1}\boldsymbol{X}^T\boldsymbol{y} \qquad (3)$$

where $\boldsymbol{b}_{OLS} = [b_1, \ldots, b_k]$ is the best linear unbiased estimate of the regression coefficients.

3.2 Linear sparse regression

The usual situation is that the available data are $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k, \boldsymbol{y})$ and the linear regression model should be estimated. The OLS estimates are calculated using all the independent variables. However, the OLS estimates may not always be satisfactory. The number of possible independent variables may be large and there are likely non-informative variables among them.

The OLS estimates have a low bias but a large variance. The large variance impairs the prediction accuracy. The prediction accuracy can sometimes be improved by shrinking

some regression coefficients toward zero, although at the same time the bias increases [4]. The models with too many independent variables are also difficult to interpret. Now, the objective is to find a smaller subset of independent variables that have the strongest effect in the regression model.

In the subset selection regression only a subset of the independent variables are included to the model, but it is an inefficient approach if the number of independent variables is large. The subset selection is not robust because small changes in the data can result in very different models. More stable result can be achieved using the nonnegative garrote [2]. The garrote also eliminates some variables and shrinks other coefficients by some positive values.

Ridge regression [12] and lasso [22] algorithms produce a sparse solution or at least shrink estimates of the regression coefficients toward zero. Both algorithms minimize a penalized residual sum of squares

$$\underset{\boldsymbol{\beta}}{\operatorname{argmin}} \Big\{ ||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}||^2 + \lambda \sum_{i=1}^{k} |\beta_i|^\gamma \Big\}, \tag{4}$$

where $\gamma = 2$ in ridge regression and $\gamma = 1$ in lasso. The tuning parameter λ controls the amount of shrinkage that is applied to the coefficients. The problem in Equation (4) can be represented equivalently as a constrained optimization problem. In that approach the residual sum of squares $||\boldsymbol{y} - \boldsymbol{X}\boldsymbol{b}||^2$ is minimized subject to

$$\sum_{i=1}^{k} |\beta_i|^\gamma \leq \tau, \tag{5}$$

where γ is the same as in Equation (4) and the constant τ controls the amount of the shrinkage. The parameters λ in Equation (4) and τ in Equation (5) are related to each other by a one-to-one mapping [10]. A large value of λ corresponds to a small value of τ.

The ridge regression solution is easy to calculate, because the penalty term is continuously differentiable. The solution is

$$\boldsymbol{b}_{RR} = (\boldsymbol{X}^T\boldsymbol{X} + \lambda\boldsymbol{I})^{-1}\boldsymbol{X}^T\boldsymbol{y}, \tag{6}$$

where $\boldsymbol{I}$ is an identity matrix, but it does not necessarily set any coefficients exactly to zero. Thus, the solution is still hard to interpret if the number of independent variables k is large. The lasso algorithm sets some coefficients to zero with a proper τ, but finding the lasso solution is more complicated due to absolute values in the penalty term. A quadratic programming algorithm has to be used to compute the solution. Also, the value of τ or λ which controls the shrinkage is strongly dependent on data. Therefore, seeking such a value may be difficult in many cases. The data-based techniques for estimation of the tuning parameter τ are presented in [22].

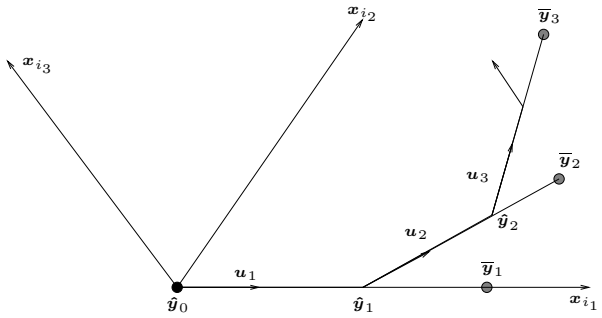

Figure 2. The progress of the LARS algorithm. The figure is reproduced from the original LARS article by Efron et al. [7].

The lasso algorithm is not applicable if the number of possible independent variables is large. Forward stagewise linear regression (FSLR) can be used instead of lasso in that case [10]. FSLR approximates the effect of the lasso penalty $\gamma = 1$ in Equation (4). New independent variables are added sequentially to the model in FSLR. Two constants δ and M have to be set before iterations. The regression coefficient that diminish most the current residual sum of squares, is adjusted by amount of δ at each successive iteration. The value of δ should be small and M should be a relatively large number of iterations.

All the estimates of coefficients $b_i, i = 1, \ldots, k$ are set to zero in the beginning. Many of the estimates b_i are possibly still zero after M iterations. It means that corresponding independent variables are not yet added to the regression model. The solution after the M iterations is almost similar than the lasso solution with some λ. They are even identical in some cases [10].

The preceding methods such as ridge regression, lasso, and FSLR are introduced as the historical precursors of the Least Angle Regression (LARS) model selection algorithm. We are mainly interested in the methods producing sparse models. Thus, lasso and FSLR could be applied, but they have deficiencies compared to the LARS algorithm. The parameters λ or τ in lasso and δ and M in FSLR have to be predefined, whereas LARS is completely parameter free and it is also computationally more efficient than lasso or FSLR. However, all the three methods produce nearly same solutions. Only LARS algorithm is applied to selection of the most significant independent variables in this study.

In Figure 2 the progress of the LARS algorithm is visualized. All the variables are scaled to have zero mean and unit variance. One independent variable is added to the model in each step. First, all regression coefficients are set to zero. Then, the most correlated independent variable $\boldsymbol{x}_{i_1}$ with $\boldsymbol{y}$ is found. The largest possible step in the direction

of $\boldsymbol{u}_1$ is taken until some other variable $\boldsymbol{x}_{i_2}$ is as correlated with the current residuals as $\boldsymbol{x}_{i_1}$. That is the point $\hat{\boldsymbol{y}}_1$. At this point LARS differs from traditional Forward Selection, which would proceed to the point $\overline{\boldsymbol{y}}_1$, but the next step in LARS is taken in a direction $\boldsymbol{u}_2$ equiangular between $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$. LARS proceeds in this direction until a third variable $\boldsymbol{x}_{i_3}$ is as correlated with the current residuals as $\boldsymbol{x}_{i_1}$ and $\boldsymbol{x}_{i_2}$. Next step is taken in a direction $\boldsymbol{u}_3$ equiangular between $\boldsymbol{x}_{i_1}$, $\boldsymbol{x}_{i_2}$, and $\boldsymbol{x}_{i_3}$ until a fourth variable can be added to the model. This procedure is continued as long as there are still independent variables left. So, k steps are needed for the full set of solutions i.e. the result is k different multiple linear regression models. In Figure 2 $\overline{\boldsymbol{y}}_i$ represent the corresponding OLS estimates from kth step. LARS estimates $\hat{\boldsymbol{y}}_k$ approach but never reach OLS estimates $\overline{\boldsymbol{y}}_k$, except at the last step the LARS and OLS estimates are equivalent. The mathematical details of LARS algorithm are presented in [7].

The problem is to find the best solution from all the k possibilities which LARS returns i.e. a proper number of independent variables. This selection can be done according to the minimum description length (MDL) information criterion [9]. The variance σ^2 of ϵ_t is assumed to be unknown, thus, the MDL criterion is written in context of the linear regression, as presented in [9],

$$MDL(k) = \frac{N}{2}\log||\boldsymbol{y} - \hat{\boldsymbol{y}}||^2 + \frac{k}{2}\log N. \qquad (7)$$

$\boldsymbol{y}$ is the dependent variable, $\hat{\boldsymbol{y}}$ is the estimate of the dependent variable, N is the sample size and k is the number of added independent variables. The value of Equation (7) is calculated for all the solutions. The selected regression model minimizes Equation (7).

The Mallows C_p criterion [18], [19] is a common criterion in subset selection. However, C_p is not used, because it can select submodels of too high dimensionality [1]. A review of several other information criteria can be found from [21].

3.3 Bootstrap

The bootstrap is a statistical resampling method and it was introduced by Efron in [6]. The idea of bootstrap is to use sample data to estimate some statistics of the data. No assumptions are made about the forms of probability distributions in the bootstrap procedure. The statistic of interest and its distribution are computed by resampling the original data with replacement.

Bootstrapping a regression model can be done in two different ways. The methods are bootstrapping residuals and bootstrapping pairs [8]. The independent variables $(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$ are treated as fixed quantities in the bootstrapping residuals approach. That assumption is strong and it can fail even if Equation (1) for the regression model is correct. In the bootstrapping pairs approach weaker assumptions about validity of Equation (1) are made.

In the bootstrapping pairs, $\hat{F}$ is assumed to be an empirical distribution of the observed data vectors $(x_{t,1}, \ldots, x_{t,k}, y_t)$, where $t = 1, \ldots, N$. $\hat{F}$ puts probability mass of $1/N$ on each vector $(x_{t,1}, \ldots, x_{t,k}, y_t)$. A bootstrap sample is now a random sample of size N drawn with replacement from the population of N vectors $(x_{t,1}, \ldots, x_{t,k}, y_t)$.

B independent bootstrap samples $(\boldsymbol{X}^{*i}, \boldsymbol{y}^{*i}), i = 1, \ldots, B$ of the size N are drawn from the distribution $\hat{F}$. The bootstrap replications $\boldsymbol{b}^{*i}$ of the estimates $\boldsymbol{b}$ are computed using the LARS algorithm and the MDL information criterion. The statistic of interest or some other features of the parameters $\boldsymbol{b}$ can be calculated from these B bootstrap replications.

3.4 Computation of relative weights of regression model

In this study, relative weights of the coefficients of multiple linear regression model are computed. The relative weights are calculated from the bootstrap replications as follows

$$\boldsymbol{w} = \frac{1}{B}\sum_{i=1}^{B}\frac{|\boldsymbol{b}^{*i}|}{\mathbf{1}^T|\boldsymbol{b}^{*i}|}. \qquad (8)$$

B is the number of bootstrap replications and $\boldsymbol{b}^{*i}$ is the ith bootstrap replication of coefficients $\boldsymbol{b}$. The absolute values are taken over all the components of vector $\boldsymbol{b}^{*i}$. There is a sum of the absolute values of coefficients in the denominator. $\mathbf{1}$ is a vector of ones and the length of the vector is the same as the length of the vector $\boldsymbol{b}^{*i}$. All the components of vector $|\boldsymbol{b}^{*i}|$ are divided by the previous sum. These operations are done for every bootstrap replication and the scaled bootstrap replications are added together. This sum is divided by the number of bootstrap samples B. The result is a vector $\boldsymbol{w}$, which includes the relative weights of the coefficients $\boldsymbol{b}$.

There is a relative weight w_i for the each possible independent variable $\boldsymbol{x}_i, i = 1, \ldots, k$ in the vector $\boldsymbol{w}$. The value of each w_i is within the range $w_i \in [0, 1]$ and $\sum_i w_i = 1$. The relative weight of the independent variable is a measure of the belief that the independent variable belongs to the estimated linear model. The independent variable can be rejected from the estimated model if the value of w_i is zero or under a predefined threshold value. The most significant independent variables have the largest relative weights. In this study the variables are scaled to have unit variance, therefore, the regression coefficients are comparable to each other and their absolute values can be used as a measure of significance.

The vector of relative weights can also be regarded as a discrete probability distribution. From the probability distribution it can be seen which independent variables are likely to be included to the final linear sparse regression model.

4 Learning a linear dependency structure

4.1 Constructing a belief graph

Let us assume now that there are data D available, which have $k+1$ variables and N measurements for each variable. The objective is to find multiple linear regression models among the variables. Each variable is the dependent variable in turn and the rest of the variables are the possible independent variables. So, the following models have to be estimated.

$$
\begin{aligned}
\hat{\boldsymbol{x}}_1 &= b_2^1\boldsymbol{x}_2 + b_3^1\boldsymbol{x}_3 + \ldots + b_k^1\boldsymbol{x}_k + b_{k+1}^1\boldsymbol{x}_{k+1} \\
\hat{\boldsymbol{x}}_2 &= b_1^2\boldsymbol{x}_1 + b_3^2\boldsymbol{x}_3 + \ldots + b_k^2\boldsymbol{x}_k + b_{k+1}^2\boldsymbol{x}_{k+1} \\
&\vdots \\
\hat{\boldsymbol{x}}_j &= b_1^j\boldsymbol{x}_1 + \ldots + b_{j-1}^j\boldsymbol{x}_{j-1} + b_{j+1}^j\boldsymbol{x}_{j+1} + \ldots \\
&\quad\; b_{k+1}^j\boldsymbol{x}_{k+1} \\
&\vdots \\
\hat{\boldsymbol{x}}_k &= b_1^k\boldsymbol{x}_1 + b_2^k\boldsymbol{x}_2 + \ldots + b_{k-1}^k\boldsymbol{x}_{k-1} + b_{k+1}^k\boldsymbol{x}_{k+1} \\
\hat{\boldsymbol{x}}_{k+1} &= b_1^{k+1}\boldsymbol{x}_1 + b_2^{k+1}\boldsymbol{x}_2 + \ldots + b_{k-1}^{k+1}\boldsymbol{x}_{k-1} + b_k^{k+1}\boldsymbol{x}_k
\end{aligned}
$$

The relative weights of the regression coefficients are computed for all the above $k+1$ linear models as it is described in Sections 3.2-3.4. A belief graph G_b is constructed from these $k+1$ vectors of the relative weights.

Each variable of data D is presented as a node in the belief graph G_b. The weighted arcs between the nodes are obtained from the nonzero relative weights. Thus, the weights of arcs measure the strength of the belief that there exists a linear dependency between the corresponding two variables. The directions of dependencies are from the independent variable to the dependent variable.

The dependencies or the number of arcs in the belief graph can be reduced by setting some threshold value λ for the relative weights. The relative weight is set to zero if it is below the threshold and the rest of the relative weights are set to unity. This means that remaining dependencies are treated as equally important thereafter. The belief graph G_b becomes unweighted directed graph G_d after using the threshold λ. However, some dependencies may be bidirectional. The value of threshold is not estimated according to some defined principle. A suitable value for λ is decided by exploring the values of relative weights. The purpose is to find such value for λ that minor changes in λ would not cause major changes in the graph G_d.

The direct use of the full information in the belief graph will be studied further.

4.2 Constructing a moral graph

The following idea of constructing an undirected and a moral graph from the belief graph is adapted from [13]. Let $V_i, i = 1, \ldots, k$ stand for a node or a variable in the graphs. The directions of dependencies can be discarded from G_d and the result is an unweighted undirected graph G_u. It can be assumed now that two variables V_i and V_j belong potentially to the same linear model if they are connected by an arc in G_u. We mean that the variables belong possibly to the same set of variables which forms one linear sparse regression model. That is, the roles of the variables V_i and V_j either as independent variable or dependent variable are not specified yet.

Let us assume that a variable $\boldsymbol{x}_{j_3}$ is the actual dependent variable. A possible regression model is $\boldsymbol{x}_{j_3} = \beta_{j_1}\boldsymbol{x}_{j_1} + \beta_{j_2}\boldsymbol{x}_{j_2} + \phi$, where ϕ is a function of the other independent variables and noise. When variables $\boldsymbol{x}_{j_1}$ and $\boldsymbol{x}_{j_2}$ are considered as the dependent variables, it is possible that the dependency with the variable $\boldsymbol{x}_{j_3}$ is found, but the dependencies between $\boldsymbol{x}_{j_1}$ and $\boldsymbol{x}_{j_2}$ are ignored in both cases. However, all three variables $\boldsymbol{x}_{j_1}$, $\boldsymbol{x}_{j_2}$, and $\boldsymbol{x}_{j_3}$ belong potentially to the same linear model. An arc can be added to connect the corresponding nodes in G_u. The added arc is called a moral arc. A moral graph G_m is obtained when all moral arcs have been added to G_u. The moral arcs are added to G_u according to the following procedure.

- Create a directed graph G'_u from G_u. The directions of dependencies are set to graph G'_u such that there do not exist cycles. For each node V_i, find its parents $\mathcal{P}_{V_i}$ in G'_u. Connect each pair of nodes in $\mathcal{P}_{V_i}$ by adding undirected arcs between the corresponding nodes in G_u.

In this study, the graph G'_u is created from G_u such that the parent V_i has a smaller index than the child V_j i.e. $i < j$ in G'_u. This restriction confirms that the relationships can be interpreted correctly and the number of added moral arcs is reasonable. The parent and child relationships can be defined differently to G'_u as above and it likely results in a dissimilar moral graph and a final dependency structure. The final dependency structure can be constructed as well from G_u as from G_m. Basically, sparser models are obtained from G_u, but the moral arc addition can give additional useful information in some cases.

4.3 Constructing final linear models

The objective is to find multiple linear regression models among the variables in the data D. The linear models or

the sets of variables are sought from the unweighted undirected graph G_u or from the moral graph G_m. The variables, which are interpreted to belong to the same model, are parts of the same maximal clique. A subgraph of G_u or G_m is called a clique if the subgraph is complete and maximal. A subgraph is complete, if every pair of nodes in the subgraph is connected by an arc. The clique is maximal, if it is not a subgraph of the larger complete subgraph [13].

An algorithm, which can be used to generate all maximal cliques from an arbitrary undirected graph, is presented in detail in [15]. A short description of the algorithm is given in the next two paragraphs.

Let C_n stand for a list of all cliques which include n nodes. The algorithm starts by forming all 2-cliques. All pairs of nodes which are connected by an arc are 2-cliques. There exists 3-clique if two 2-cliques have one node in common and two sole nodes are connected. For example, if there are cliques $\{V_1, V_2\}, \{V_1, V_3\}$ and $\{V_2, V_3\}$ in the graph, then there exists 3-clique $\{V_1, V_2, V_3\}$. All 3-cliques are collected to the list C_3.

All $(n+1)$-cliques can be constructed from the list C_n. Two n-cliques c_n^1 and c_n^2, which have already $(n-1)$ nodes in common, are tested if they could form a new $(n+1)$-clique c_{n+1}. There has to exist n-clique c_n^3, which has $(n-2)$ nodes in common with cliques c_n^1 and c_n^2, in the list C_n. Additionally, $(n-1)$th node of c_n^3 has to be equivalent to nth node of c_n^1 and nth node of c_n^3 has to be equivalent to nth node of c_n^2, then there is $(n+1)$-clique c_{n+1} in the graph. For example, if there exist cliques $c_4^1 = \{V_1, V_2, V_3, V_4\}$, $c_4^2 = \{V_1, V_2, V_3, V_5\}$ and $c_4^3 = \{V_1, V_2, V_4, V_5\}$, then there exist 5-clique $c_5 = \{V_1, V_2, V_3, V_4, V_5\}$ in the graph. This procedure is repeated as long as new cliques can be constructed. All lists $C_i, i = 1, \ldots, n_{max}$ are tested in the end, that any n-clique is not a subclique of $(n+m)$-clique, $m > 0$. If there exist subcliques they can be eliminated. In the end, the dependent variable in all the cliques is selected such that the coefficient of determination is maximized.

The problem to find all maximal cliques is known to be NP-hard [14]. This means that the computational time for a solution is nondeterministic and the number of cliques can increase exponentially. Several other algorithms for solving the clique problem are introduced and analyzed in [14]. Computationally, the task is feasible, if the number of variables in the data D is not large. The number of variables can be a few hundred. The number of arcs in the graph also affects on the computational efficiency.

The number of found complete and maximal cliques can be large, but there are additional criteria how final cliques are selected. Firstly, two cliques can have only one variable or node in common. Secondly, the common variable cannot be a dependent variable in both cliques. Finally, cycles are not allowed in the dependency structure. Therefore, the dependency structure is a dependency tree or a forest under these restrictions. The independent variables are the parents of the dependent variable in the final dependency structure. The construction of the dependency structure starts from the linear model which has the highest coefficient of determination. After that, the linear models are added such that the coefficients of determination are as good as possible and the previous restrictions are not violated.

5 Experiments

5.1 Data

A real-world data set which is used in this study is called the System data. The System data consist of nine measurements from a single computer which is connected to a network. The computer is used for example to edit programs or publications and to calculate computationally intensive tasks [23].

Four of the variables describe the network traffic. Rest of the variables are measurements from the central processing unit (CPU). All the variables are in relative measures in the data set. The variables are 1. `blks/s` (read blocks per second (network)), 2. `wblks/s` (written blocks per second (network)), 3. `usr` (time spent in user processes (CPU)), 4. `sys` (time spent in system processes (CPU)), 5. `intr` (time spent handling interrupts (CPU)), 6. `wio` (CPU was idle while waiting for I/O (CPU)), 7. `idle` (CPU was idle and not waiting for anything (CPU)), 8. `ipkts` (the number of input packets (network)), and 9. `opkts` (the number of output packets (network)).

The System data is collected during one week of computer operation. The first measurement is done in the morning on Monday and the last one is done in the evening on Friday. The measurements are done every two minutes during the day and every five minutes during the night. The measurements are done from every nine variables each time. There are missing values in all the variables. A more detailed description of the System data set is found from [23].

5.2 Preprocessing of the data

In general, processes are usually in different states during the measurements. It is possible that dissimilar linear dependency structures are needed to describe the operation of computer during the week, for example one structure during the day and another during the night. The variable `blks/s` can vary depending on if someone is working with the computer.

In this study, the similar states of the process are sought using the variable `blks/s`, which is, thus, a reference variable. The reference variable can be any of the variables in the data set depending on which feature is wanted to be explored. The reference variable is plotted in Figure 3.

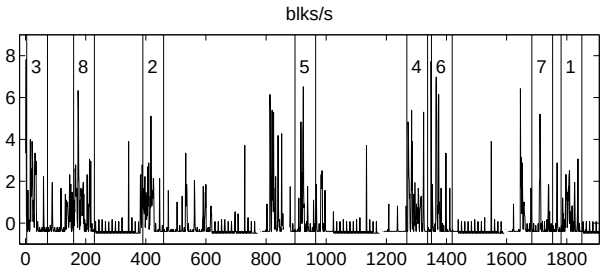

Figure 3. The reference variable `blks/s` and the selected windows.

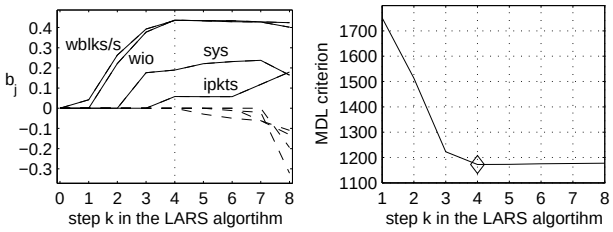

Figure 4. Development of coefficient values in the LARS algorithm (a). Values of MDL criterion for different models (b). The vertical line in (a) and the diamond in (b) represents the minimum value of MDL criterion.

A query window is selected from the reference variable. The query window of reference variable should include information or the measurements of the feature i.e. the interesting state of the time-series, which is under exploration. The selected query window is the window number one in Figure 3. The measurements in the query window are done in the afternoon on Friday.

The similar states of the reference variable can be located mathematically in many ways. In this study, the sum of squares of differences between the query window and a candidate window is minimized. The candidate window is a part of the reference variable which is as long as the query window. The candidate windows are not allowed to overlap with each other or with the query window. The candidate windows which have the smallest sum of squares of differences between the query window are chosen.

The sum of squares of differences between the query window and the candidate window i.e. the Euclidean distance between them is calculated as follows

$$E_c = \sum_{i=1}^{M} (y_{q,i} - y_{c,i})^2, \tag{9}$$

where y_q is the query window, y_c is the candidate window, and M is a number of the measurements which are included in the query window.

The number of chosen candidate windows can be decided, for example, by setting a threshold value to Equation (9). Another option is to select so many windows that there are enough data points in further calculations. In Figure 3, windows $2-8$ are the chosen candidate windows. Smaller numbers of candidate windows refer to smaller values of Equation (9). The measurements in all the chosen candidate windows are done during the working hours.

The data in the chosen candidate windows and in the query window from the reference variable are chosen and the rest of the measurements are excluded from further calculations. The parts, which have the same time label as chosen candidate windows and query window, are also selected from the rest of the time-series. All the selected windows are scaled to have zero mean and unit variance. There are 70 data points in each selected window so in the further calculations there are $N = 560$ data points in total from every variable.

New time-series are acquired when the selected windows of the original variables are put one after another. The original measurements often include noise. The level of the noise may be disturbingly high. In that case, noise reduction techniques can be applied to the selected windows, for example techniques based on the wavelet transform [5], [17].

This kind of similarity search in high dimensions, i.e with very long time windows, should be approached with caution. The Euclidean distance between the windows may not work because of the curse of dimensionality. This selection of windows, however, is not central to this work, but it can be used if only the certain parts of the time-series are interesting and wanted to be explored.

5.3 An example of sparse regression

The operation of LARS algorithm is illustrated with an example. The variable `blks/s` is the dependent variable and rest of the variables are the possible independent variables.

The independent variables are added to the regression model in the following order `wblks/s`, `wio`, `sys`, `ipkts`, `intr`, `idle`, `opkts`, and `usr`. Development of regression coefficients are plotted in the left panel of Figure 4. The values of MDL criterion are plotted in the right panel of the same figure. The minimum value is achieved by step four i.e. the first four added independent variables are included to the regression model.

The sparse regression model is

$$\begin{aligned}\hat{\boldsymbol{y}}_{\texttt{blks/s}} = {} & 0.44\boldsymbol{x}_{\texttt{wblks/s}} + 0.45\boldsymbol{x}_{\texttt{wio}} \\ & +0.19\boldsymbol{x}_{\texttt{sys}} + 0.07\boldsymbol{x}_{\texttt{ipkts}}.\end{aligned} \tag{10}$$

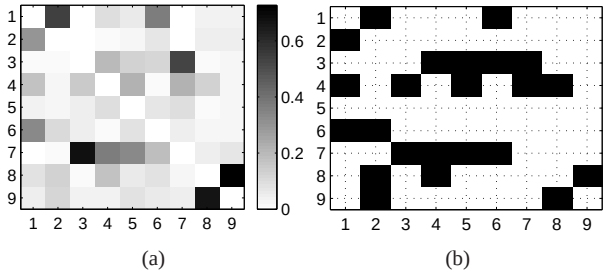

Figure 5. The adjacency matrices of the belief graph G_b (a) and the unweighted directed graph G_d (b).

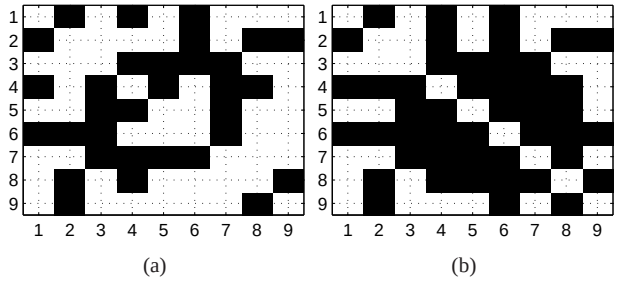

Figure 6. The adjacency matrices of the unweighted undirected graph G_u (a) and the moral graph G_m (b).

The coefficients of determination of the sparse model (10) and the full model are 0.89 and 0.90, respectively. Thus, the excluded variables can be considered as non-informative and dropping out them improve the interpretability of dependencies between the variables.

5.4 The dependency structure of System data

The objective is to find the best multiple linear regression models from the preprocessed data set. The process proceeds as it is described in Section 4. The first task is to construct the belief graph G_b.

The adjacency matrix of the belief graph G_b is presented in the left panel of Figure 5. The relative weights of the regression coefficients of the ith model are in the ith column in the adjacency matrix. The ith variable has been the dependent variable in the ith model and rest of the variables have been the possible independent variables. The relative weights for the variables $2, \ldots, 8$, when the variable 1 is the dependent variable, are presented in the first column of the adjacency matrix of G_b. The other columns are constructed in a corresponding way. Dark colors refer to a strong belief that these variables are significant in the multiple linear regression model. For example, in the first column or in the first regression model the variables 2 (`wblks/s`), 4 (`sys`), and 6 (`wio`) are the most significant independent variables. The number of bootstrap replications was $B = 1000$ in each of the nine cases. The relative weights of the coefficients were calculated according to Equation (8).

The directed graph G_d is computed from G_b using the threshold $\lambda = 0.1$ i.e the dependencies whose relative weight is under 0.1 are ignored and rest of the weights are set to unity. The adjacency matrix of G_d is in the right panel of Figure 5. The unweighted undirected graph G_u is obtained from G_d by ignoring the directions of dependencies and the moral graph G_m is calculated from G_u as it is described in Section 4.2. The adjacency matrices of G_u and G_m are drawn in Figure 6.

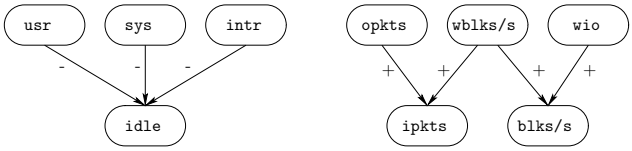

Figure 7. The dependency forest from the graph G_u.

The final linear models are sought from the undirected graph G_u and from the moral graph G_m. The variables which belong to the same multiple linear regression model are part of the same maximal clique in the graphs G_u or G_m. The maximal cliques are found using the algorithm which is presented in Section 4.3.

Three maximal cliques $c_4^1 = \{\texttt{idle}, \texttt{usr}, \texttt{sys}, \texttt{intr}\}$, $c_3^1 = \{\texttt{ipkts}, \texttt{opkts}, \texttt{wblks/s}\}$ and $c_3^2 = \{\texttt{blks/s}, \texttt{wblks/s}, \texttt{wio}\}$ were found from the graph G_u. The best models are achieved if the variables `idle`, `ipkts` and `blks/s` are chosen to be the dependent variables. The coefficients of determination are then 0.95, 0.94 and 0.82. The dependency forest of these linear models is in Figure 7.

All variables in clique c_4^1 are measurements from the CPU. All regression coefficients were negative in this model. If there is a positive change in some independent variable, the value of the dependent variable `idle` will decrease.

Cliques c_3^1 and c_3^2 are dependent on each other through the variable `wblks/s`, which is one of the independent variables in both models. When a positive change occurs in the variable `wblks/s` also the values of the dependent variables `ipkts` and `blks/s` increase. All variables in the clique c_3^1 are the measurements from the network traffic. In the clique c_3^2, the variable `blks/s` is the measurement from the network traffic and the variable `wio` is the measurement from the CPU.

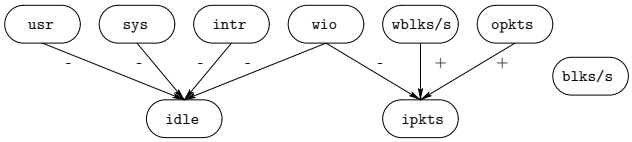

Figure 8. The dependency tree from the graph G_m.

An alternative dependency structure is computed from the graph G_m. There were two maximal cliques and the variable `blks/s` was left alone. Cliques are $c_5^1 = \{\texttt{idle}, \texttt{usr}, \texttt{sys}, \texttt{intr}, \texttt{wio}\}$ and $c_4^1 = \{\texttt{ipkts}, \texttt{wio}, \texttt{wblks/s}, \texttt{opkts}\}$. The dependency structure is plotted in Figure 8.

All the variables in the clique c_5^1 are measurements from the CPU. The best model is obtained, when `idle` is the dependent variable. Then the coefficient of determination is 0.96, which indicates that the linear model describes the dependencies between the variables very well. The first linear sparse regression model is

$$\begin{aligned}\hat{\boldsymbol{y}}_{\texttt{idle}} = & -0.64\boldsymbol{x}_{\texttt{usr}} - 0.30\boldsymbol{x}_{\texttt{sys}} \\ & -0.13\boldsymbol{x}_{\texttt{intr}} - 0.08\boldsymbol{x}_{\texttt{wio}}.\end{aligned} \tag{11}$$

The independent variables describe how much of the CPU power is spent to different activities and the dependent variable describes how much of the CPU power is unused at the moment. According to the estimated linear model the value of `idle` decreases when the values of `usr`, `sys`, `intr` and `wio` increase. This is very intuitive result because the processes which need CPU power obviously diminish the available CPU power.

The clique c_4^1 formulates another multiple linear regression model. When the variable `ipkts` is selected to the dependent variable, the best coefficient of determination (0.94) is achieved. The second model is

$$\begin{aligned}\hat{\boldsymbol{y}}_{\texttt{ipkts}} = & -0.01\boldsymbol{x}_{\texttt{wio}} + 0.11\boldsymbol{x}_{\texttt{wblks/s}} \\ & +0.93\boldsymbol{x}_{\texttt{opkts}}.\end{aligned} \tag{12}$$

The variable `ipkts` consists of measurements from the network traffic. The variables `wblks/s` and `opkts` describes also the network traffic and `wio` is the same measurement from the CPU as in model (11). When the number of written blocks per second and the number of output packets increase the number of input packets also increases according to Equation (12). This is a natural situation in the bidirectional network traffic. The packets are sent to both directions when for example a file is downloaded.

Models (11) and (12) are dependent on each other through the variable `wio`. Changes in `wio` has effect on both dependent variables `idle` and `ipkts`. A positive change in `wio` decreases the values of `idle` and `ipkts`. However, the variable `wio` could be possibly excluded from model (12), since the value of its regression coefficient is negligible and it has not much effect on the coefficient of determination.

6 Summary and conclusion

In this study, the method for analyzing linear dependencies in multivariate data is proposed. The result of the method is a linear dependency tree or a forest. The dependencies of the variables can be clearly seen from the final dependency structure. Thus, it is possible to get deeper understanding of the underlying process. The linear dependencies are modeled by multiple linear regression models.

Similar states of time-series are selected using the Euclidean distance between a reference variable. A single regression model is constructed to model that selected state. It may be difficult or even impossible to construct a single regression model to time-series, which consists of many different states. Every state would require a model of its own.

This study proposes how the relative weights of the regression coefficients can be calculated from the bootstrap replications. The relative weight of the regression coefficient is a measure of belief that the corresponding independent variable belongs to a certain regression model. In addition, the dependent variable or variables are selected during the execution of the algorithm.

In the experiments it was shown that the most significant variables have the highest relative weights. The relative weights seem to be appropriate to measure significance of the independent variables.

The final dependency structure was constructed from the belief graph. The belief graph represents the variables and the relative strength of the dependencies between the variables. A threshold value was used to reduce the dependencies in the belief graph. The chosen threshold value has a strong impact on the final dependency structure. A minor change in the threshold value can cause a major changes in the final dependency structure. Thus, special attention to the threshold value should be paid. It would be beneficial to automate the selection of the threshold value. One possibility might be to include it somehow in the moralizing operation. Another possibility could be to learn the final dependency structure from the belief graph ignoring the weakest dependencies by some data based method such that a tree or a forest structure is achieved.

The proposed method was tested using a real world-data set. The constructed dependency structures were convincing. Approximately 95% of the variation of dependent vari-

ables was explained by the regression models which were constructed from the System data, although no assumptions were made about the number of linear models. The resulting dependency structure was almost similar to one shown in Figure 8, when the whole data set was used in the construction of the dependency structure. When models (11) and (12) were tested with the excluded data the coefficients of determination were nearly as good as with the used data.

The final dependency structure highlights the dependencies of the variables in an intelligible way. On the other hand, it is difficult to measure the level of interpretability. The goodness of the models may be hard to justify if coefficient of determinations are moderate, but the dependency structure can still give additional and unexpected information in co-operation with someone who has specific knowledge of the underlying process.

7 Acknowledgements

The authors thank Dr. Esa Alhoniemi for the idea of selecting similar windows from time-series. We also thank for the insightful comments from the reviewers of the paper.

References

[1] L. Breiman. The little bootstrap and other methods for dimensionality selection in regression: X-fixed prediction error. *Journal of the American Statistical Association*, 87(419):738–754, September 1992.

[2] L. Breiman. Better subset regression using the nonnegative garrote. *Technometrics*, 37(4):373–384, November 1995.

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

[4] J. Copas. Regression, prediction and shrinkage. *Journal of the Royal Statistical Society (Series B)*, 45(3):311–354, 1983.

[5] D. L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995.

[6] B. Efron. Bootstrap methods: Another look at the Jackknife. *The Annals of Statistics*, 7(1):1–26, January 1979.

[7] B. Efron, T. Hastie, I. Johnstone, and R. Tibshirani. Least angle regression. *The Annals of Statistics*, 32(2):407–499, April 2004.

[8] B. Efron and R. J. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, 1993.

[9] M. H. Hansen and B. Yu. Model selection and the principle of minimum description length. *Journal of the American Statistical Association*, 96(454):746–774, June 2001.

[10] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[11] D. Heckerman, D. M. Chickering, C. Meek, R. Rounthwaite, and C. Kadie. Dependency networks for inference, collaborative filtering, and data visualization. *Journal of Machine Learning Research*, 1:49–75, October 2000.

[12] A. E. Hoerl and R. W. Kennard. Ridge regression: Biased estimation for nonorthogonal problems. *Technometrics*, 12(1):55–67, February 1970.

[13] C. Huang and A. Darwiche. Inference in belief networks: A procedural guide. *International Journal of Approximate Reasoning*, 15(3):225–263, October 1996.

[14] I. Koch. Enumerating all connected maximal common subgraphs in two graphs. *Theoretical Computer Science*, 250(1-2):1–30, January 2001.

[15] F. Kose, W. Weckwerth, T. Linke, and O. Fiehn. Visualizing plant metabolomic correlation networks using clique-metabolite matrices. *Bioinformatics*, 17(12):1198–1208, December 2001.

[16] K. Lagus, E. Alhoniemi, and H. Valpola. Independent variable group analysis. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Proceedings of the International Conference on Artificial Neural Networks*, number 2130 in Lecture Notes in Computer Science, pages 203–210. Springer, 2001.

[17] M. Lang, H. Guo, J. E. Odegard, C. S. Burrus, and R. O. Wells. Noise reduction using an undecimated discrete wavelet transform. *IEEE Signal Processing Letters*, 3(1):10–12, January 1996.

[18] C. L. Mallows. Some comments on C_p. *Technometrics*, 15(4):661–675, November 1973.

[19] C. L. Mallows. More comments on C_p. *Technometrics*, 37(4):362–372, November 1995.

[20] G. M. Maruyama. *Basics of Structural Equation Modeling*. SAGE Publications, Inc., 1997.

[21] P. Stoica and Y. Selen. Model-order selection. *IEEE Signal Processing Magazine*, 21(4):36–47, July 2004.

[22] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.

[23] J. Vesanto and J. Hollmén. An automated report generation tool for the data understanding phase. In A. Abraham, L. Jain, and B. J. van der Zwaag, editors, *Innovations in Intelligent Systems: Design, Management and Applications*, Studies in Fuzziness and Soft Computing. Springer (Physica) Verlag, 2003.