Mining Causal Relationships in Multidimensional Time Series

Yasser Mohammad and Toyoaki Nishida

Graduate School of Informatics,
Kyoto University

Abstract. Time series are ubiquitous in all domains of human endeavor. They are generated, stored, and manipulated during any kind of activity. The goal of this chapter is to introduce a novel approach to mine multidimensional time-series data for causal relationships. The main feature of the proposed system is supporting discovery of causal relations based on automatically discovered recurring patterns in the input time series. This is achieved by integrating a variety of data mining techniques.
The main insight of the proposed system is that causal relations can be found more easily and robustly by analyzing meaningful events in the time series rather than by analyzing the time series numerical values directly. The RSST (Robust Singular Spectrum Transform) algorithm is used to find interesting points in every time series that is further analyzed by a constrained motif discovery algorithm (if needed) to learn basic events of the time series. The Granger-causality test is extended and applied to the multidimensional time-series describing the occurrences of these basic events rather than to the raw time-series data.
The combined algorithm is evaluated using both synthetic and real world data. The real world application is to mine records of activities during a human-robot interaction experiment in which a human subject is guiding a robot to navigate using free hand gesture. The results show that the combined system can provide causality graphs representing the underlying relations between the human's actions and robot behavior that cannot be recovered using standard causal graph learning procedures.

Key words: Mining Time Series, Robust Singular Spectrum Transform, Granger-Causality, Mining Causal Relations

1 Introduction

In today's information societies, effective knowledge management became an important field of study. With advancements of sensing technologies, data are now collected from various kinds of natural, industrial and other human activities in various forms and stored in large databases. The need for discovering useful knowledge from such huge databases was behind the recent interest of many scholars and industrial entities in data mining.

Time series data is one of the most ubiquitous data formats that captures activities and information. Moreover many other complex types of data formats

can be converted to time series data which adds more importance for algorithms that can discover useful knowledge in time series databases. For example Wang et al. [1] developed an algorithm to convert shapes to time series while Inokuchi and Washio [2] proposed a graph-edit mechanism to convert series of graphs to multidimensional time series.

A variety of techniques have been proposed in literature to deal with multidimensional time series analysis including algorithms for efficient clustering [3], searching [4] and discovery of recurrent patterns [5]. We can distinguish two types of time series analysis algorithms. The first type is pattern-based algorithms including motif discovery [6], motif detection, pattern search [5] and some types of compression and discretization algorithms. These algorithms do not deal with the whole time series but with subcomponents of it called patterns, subsequences, or motifs. On the other hand there are holistic algorithms that deal with whole time series, including regression, and clustering[3]. Most causality inference algorithms available belong to the holistic algorithms and so they can discover relations like *smoking cause cancer*. In some cases we are more interested in discovering relations between specific patterns in the input time series rather than the relations between the time series themselves. For example we may like to check if a specific pattern of change in oil prices causes a specific kind of change in the stock market. Holistic approaches cannot answer this question and they can only tell whether or not there is a causal relation between oil prices and stock market in general. In this chapter we focus on unsupervised discovery of causal relations between recurring patterns in multidimensional time series. Given a set of n time series that are *roughly* synchronized, our goal is to first discover the recurring patterns on these time series and then to discover *causal* relations between these recurring patterns.

There are many possible industrial, engineering, and social applications of the proposed techniques. For example behavioral data – in the form of motion captured signals from humans during face to face interactions – can be used to discover the effects of one partner's verbal or nonverbal behavior patterns on the behavior of the partner(s).

The main insight of the proposed techniques in this chapter is to base the analysis of causality on recurring events in the input signals (time series) rather than on the instantaneous values of these series.

The rest of this chapter is organized as follows: Section 2 details the state of the art in causal relation discovery from time series. Section 3 gives a bird's eye view of the proposed technique which has two major phases described in sections 4 and 5. Section 6 reports preliminary evaluation of the system on synthetic data (section 6.1) and real world human-robot interaction data (section 6.2). The paper is then concluded.

2 Related Work

Causality was – and is still – a subject of great disagreement between philosophers. The study of causality can be traced back to Aristotle who defined four types of causal relations [7]:

1. Material Causality: This is the relation between a statue and the bronze it is made of.
2. Formal Causality: This is the relation between the statue and its shape.
3. Efficient Causality: This is the relation between what makes to what is made or what changes to what is changed.
4. Final Causality: This is the relation between the goal and whatever exists (or happens) to produce this goal.

From these four types, only efficient causality has survived as *causality* until today. In his *Treatise of Human Nature* (1739-1740), David Hume described causality in terms of regular succession. For Hume, causality is a regular succession of event-types: one thing invariably following another. His famous first definition of causality runs as follows: *We may define a CAUSE to be 'An object precedent and contiguous to another, and where all the objects resembling the former are placed in like relations of precedence and contiguity to those objects, that resemble the latter'.*

Some philosophers support the positivistic claims that causality is a metaphysical concept that has been replaced in science by nonmetaphysical concepts such as functional interdependence (Mach, 1883/1960), association or correlation (Pearson 1911), or functional relation (Schlick, 1932/1959). The aim of these positivists' efforts was to move science away from the ideas of necessary connection and determinism that they identified with their contemporaries' ideas of causality. On the other hand other philosophers [8] take causality for granted and even try to build mathematical models of causal inference. The only ingredient that *most* definitions of causality agree upon is that causation involves time. Simply causes *precede* effects. In fact this is the basis of the definition of Causal Linear Systems as used in electrical engineering literature.

Some views of causation do not highlight this time asymmetry between causes and effects explicitly even though in most cases the concept implicitly exists. For example, some theorists have equated causality with manipulability [9]. Under these theories, x causes y just in case one can change x in order to change y. Even though the assumption that a cause must precede its effect is not explicit in this case, it does exist implicitly in the definition because manipulation can only cause changes in the future (or present if infinite effect speed is allowed) but no manipulation can *change* the past values of y. Under this assumption about manipulation – which is consistent with both common sense and physical knowledge – the definition of causality through manipulation involves time and assumes that causes *precede* effects.

Another example is the definition of causality as a counterfactual which means that the statement "x causes y" is equivalent to "y would have happened if x". This kind of definition of causality is the basis of Pearl's formalization

of the Structure Equation Model [10]. In this definition also time asymmetry between causes and effects enters the definition through the connective *would have happened* which assumes that x should happen in the past or at least the present for y to happen.

In this chapter, we will not attempt answering these centuries old questions about the meaning of causation or its implications or even contribute in any way to the philosophical discussions about it. Our goal in this chapter is to propose a practical way to discover specific kinds of relations that can be treated as causal ones by simply rejecting relations that cannot satisfy the condition *causes precede effects in time*. This condition is a necessary but not a sufficient condition which means that the proposed technique may generate output relations that are not actually causal depending on the definition of *causality* used. Even though this is not satisfying from the theoretical point of view, it is the only practical solution given that an exact definition of causality is far from being agreed upon. The produced relations can be filtered using a stricter sufficient condition but this will invariably lead to rejection of authentic *causal* relations depending on some definitions of causality.

Discovering relations between variables from data is a subject of great importance to science and there are two main techniques currently available.

The first approach is Rule Discovery using algorithms like DBMiner [11]. In this case the input is a set of n variables $X_{1:n}$ and the output is a set of association rules in the form:

$$X_i = x_i \wedge X_j = x_j \wedge ..... \wedge X_k = x_k \Rightarrow X_l = x_l \tag{1}$$

The most frequently mentioned example is the supermarket scenario. $X_i(c)$ is a binary variable representing the existence of product i in the basket of customer c at the check-out. Determining which products customers are likely to buy together is considered valuable for marketing applications such as mailing or catalogue design, as well as store layout and customer segmentation. Rule Discovery can be classified to Classification Rule Discovery which tries to find only rules with high prediction value and Association Rule Discovery which tries to discover all rules covering the input data set. The former is more important for efficient decision making while the later is more useful for expanding our understanding of the domain. The strength of the discovered rule is described using two concepts: support and confidence. Support indicates the frequency that the association rule is true for the population being examined and can be defined as:

$$Support\,(A \Rightarrow B) = \frac{|\text{coverset}\,(A \cup B)|}{|X|} \tag{2}$$

where coverset (A) is the set of all records in X that satisfy A.

Confidence indicates the prediction strength of the rule and can be defined as:

$$Confidence\,(A \Rightarrow B) = \frac{|Support\,(A \cup B)|}{|\text{coverset}\,(A)|} \tag{3}$$

Examination of the definitions of *Support* and *Confidence* indicates that Rule Discovery algorithms – in this form – do not take into account the time at which A and B happens. Given the importance of time in the causality relation, Rule Discovery – either classification or association – can be dismissed as a way for discovering underlying causality models. In fact the same supermarket example clearly indicates that Rule Discovery algorithms do not even try to discover causal relations. For example if we found with 100% confidence that:

$$NeckTie \Rightarrow FullSuit \tag{4}$$

We can only infer that there is a hidden variable (SocialRule) that causes the people who buy neck ties to buy full suits (e.g. because neck ties are only used with full suits). This is exactly the kind of rule that an authentic causal relation discovery algorithm should reject because buying a neck tie was not the *cause* of buying a full suit.

To overcome this difficulty, Das, Lin and Mannila [12] proposed an algorithm that can discover association rules in temporal time series that have this extended form:

$$A \xrightarrow{T} B \tag{5}$$

which means that when A happens at time step t, B happens at time step $t + T$. The method is based on discretizing the sequence $X(t)$ by methods resembling vector quantization. First subsequences are formed by sliding a window through the time series, and then these subsequences are clustered using a suitable measure of time-series similarity. The discretized version of the time series is obtained by concatenating the cluster identifiers corresponding to the subsequence. Once the time-series is discretized, simple rule finding methods are used to obtain rules from the sequence. Many researchers suggested improvements to the basic algorithm provided in this work to increase its speed [13], to utilize constraints in the search for rules [14] and to use parallel implementations rather than serial processing of the time series [15]. One common problem of all of these algorithms is that what the rule $A \xrightarrow{T} B$ actually tells us is a kind of association (this is why it is called Association Rule Mining) but not causal relation because A and B can still be caused by a hidden factor with different time delays T_A and T_B so that $T_B - T_A = T$.

The second approach to causal relation discovery can be called Bayesian Causal Network Induction approach and this approach can overcome the limitation of Rule Discovery concerning causal relation discovery. The goal of these algorithms is to build a Directed Graph (DG) describing the causal relations between variables. As with the Association Rule Discovery case, the input is a set of n variables $X_{1:n}$. If Feedback (e.g. X causes Y and Y causes X) is not supported then the graph is a DAG (Directed Acyclic Graph). There are three categories of Bayesian Causal Network Induction algorithms that utilize different aspects of causality to discover causal rules.

The first of these techniques can be called Causation from Perturbation Algorithms under this category use some form of change or perturbation of the

generating process to distinguish causal and non-causal correlations. This technique is rooted in the work of the economist Kevin Hoover [16] who inferred the direction of causation among correlated economical variables (e.g. employment and money supply) by observing changes that sudden modifications of the economy (e.g. tax reformation) induce on the statistics of these variables. The main assumption was that the conditional probability of an effect given its causes should not vary due to economy modifications while the conditional probability of a cause given its effect can vary with such modifications. Formally, if E is an effect, C is its cause, $P_a(x)$ is the probability distribution of x after the disturbance/modification of the generating process, $P_b(x)$ is the probability distribution of x before the disturbance, and $\Psi(P)$ is some statistic of P, then:

$$\Psi(P_a(E|C)) = \Psi(P_b(E|C)) \tag{6}$$

$$\Psi(P_a(C|E)) \neq \Psi(P_b(C|E)) \tag{7}$$

This technique is usually used for confirming or rejecting specific causal models rather than discovering them from scratch. One problem of this approach is the need for some system wide perturbation or modification but if this modification is already available it can augment available causal relation discovery algorithms.

Tian and Pearl [17] proposed an algorithm based on this approach and applied it to simulated data showing that it can be superior to other techniques that rely only on static conditions. It should be noted that fixing variables (or variable statistics) during experimental procedures corresponds to a perturbation of the underlying generation processes and so experimental scientific discovery can be related to this approach.

The second Bayesian Causal Network Induction approach can be called Causality Model Assessment. In this category of algorithms, first a set of hypothesized causal structures (*models*) is generated by the modeler and then the algorithm is used to find which of them has the highest fit to the data.

Structural Equation Modeling (SEM) [10] technique is an example of this approach. In SEM, the modeler first specifies the model and specifies the exogenous and endogenous variables. Exogenous variables are the variables that cause other variables while endogenous variables are ones that are caused by other variables. The same variable can be both an endogenous and exogenous variable in the same model. The model is specified as a set of structure equations where a structure equation takes the form:

$$E = \sum_{causes} a_i C_i + D \tag{8}$$

where E is an effect (endogenous variable), and C_i are its causes (exogenous variables). The coefficients a_i are called structural coefficients, while D represents unmodeled disturbance or noise in the system. Once the model is specified, its best parameters are estimated from the data using either multiple regression or a more sophisticated SEM specific estimation procedure like SPSS's AMOS

program. The estimated model parameters are used to predict the correlations or covariances between measured variables and the predicted correlations or covariances are compared to the observed correlations or covariances. There are many model fitting techniques that can be used including χ^2 fitting, Root Mean Square Error of Approximation, Comparative Fit Index etc. The main disadvantage of this approach is that it does not specify a way to construct (specify) the set of models to be tested in the first place. Most scientific discovery using the standard scientific procedure of generating hypotheses and testing them using experiments can be considered a combination of Causality Model Assessment and Causation By Perturbation.

The third approach to Bayesian Causal Network Induction can be called Causality from Data. In this category, the algorithm does not require a predefined causal model to test but it constructs its own model from the data directly. Usually more data points are required to generate robust models. Granger-causality tests fall under this category. Granger causality [18] is a technique for determining whether one time series is useful in forecasting another.

A Ganger-Causes B ($A_{GC} \rightarrow B$) *iff*, it can be shown, usually through a series of F-tests on lagged values of A, that those A values provide statistically significant information about future values of B.

There are many ways in which to implement a test of Granger causality. One particularly simple approach uses the autoregressive specification of a bivariate vector autoregression. First we assume a particular autoregressive lag length ρ, and estimate the following unrestricted equation by ordinary least squares (OLS):

$$\hat{A}(t) = \epsilon_1 + u(t) + \sum_{i=1}^{\rho} \alpha_i \hat{A}(t-i) + \sum_{i=1}^{\rho} \beta_i \hat{B}(t-i) \tag{9}$$

Second, we estimate the following restricted equation also by OLS:

$$\hat{A}(t) = \epsilon_2 + e(t) + \sum_{i=1}^{\rho} \lambda_i \hat{A}(t-i) \tag{10}$$

We then calculate the sum of squared residuals (SSR) in both cases:

$$\begin{aligned} SSR_1 &= \sum_{i=1}^{T} u^2(t) \\ SSR_0 &= \sum_{i=1}^{T} e^2(t) \end{aligned} \tag{11}$$

We then calculate the test statistic S_ρ as:

$$S_\rho = \frac{(SSR_0 - SSR_1)/\rho}{SSR_1/(T - 2\rho - 1)} \tag{12}$$

If S_ρ is larger than the specified critical value then reject the null hypothesis that A does not granger cause B. The p-value in this case is $1 - F_{\rho, T-2\rho-1}(S_\rho)$. As

presented here Ganger Causality can be used to deduce causal relations between two variables assuming a linear regression model. The test was extended to multiple variables and nonlinear relations (using radial basis functions) [19].

Another algorithm that falls under this category is the algorithm proposed by Kamiri and Hamilton [20] to distinguish causal and accusal temporal relations called Temporal Investigation Method for Enregistered Record Sequences (TIMERS). The main idea is to run time forward and backward while mining the relations from system data. If the reversed run provides better model fit then the relation is estimated as an accusal relation. If the best fit was in the forward direction, this means that it is a causal relation. A relation that relates variables at the same point of time (same time step) is called an instantaneous relation. The authors also designed TimeSleuth software that can be used to mine causal relations in time series data. This is the first approach that explicitly uses the direction of time to test for causation and in this it is similar to the approach presented in this paper. The main difference between this approach and our proposed approach is in the kind of knowledge it reveals about the system. While TIMERS only reveals the direction of causation, the proposed system discovers the exact patterns in every time series that cause changes in other time series.

Another more ambitious example of systems that fall under this category is the TETRAD II and III programs developed by Spirtes, Glymour and Scheines [21]. These programs aim at automatically discovering the causal structure of input data. The program has two inference engines PC and FCI. The choice of the engine depends on whether the data examined is causally sufficient for the population, that is, whether there exist unmeasured hidden or latent causal variables outside of X that explain spurious associations between variables in X. If data is causally sufficient, the PC algorithm is used. Otherwise, the Fast Causal Inference (FCI) algorithm is used. Instead of logical rules, the PC and FCI algorithms find different kinds of causal relationships between variables X_i and X_j in X, graphically represented as follows:

- X_i – X_j, meaning in PC that either X_i causes X_j or X_j causes X_i, but the direction is indeterminate. In FCI it means that there is a correlation between X_i and X_j but causation cannot be inferred
- $X_i \rightarrow X_j$, meaning that X_i causes X_j.
- $X_i \leftrightarrow X_j$, indicating a common hidden cause. In this case PC cannot be used.
- $X_i \bullet\!\!\rightarrow X_j$, indicating that common cause cannot be ruled out but cannot be confirmed (potentially causal relation).

The resulting graph represents a set of Markov equivalent Bayes networks. This causal inference algorithm makes strict assumptions about the data: Variables in X must satisfy the Markov Condition; i.e., variables can be organized into a directed acyclic graph so that any variable X_i in X conditioned on X_i's parents is independent of all sets of variables that do not include X_i or its descendants.

The following factors affect whether or not the Markov Condition is satisfied by data:

1. Values of one unit of the population must be independent of values in other units of the population (e.g. i.i.d. data).
2. Mixtures in populations that result in contradictory causal connections between two variables violate the Markov Condition.
3. Cyclical processes that reach equilibrium also violate the Markov condition. Consider the sequence: in time t the value of X_i affects the value of X_j. Then in time t+1 the value of X_j affects the value of X_i. In time t+2 the value of X_i affects the value of X_j. This circular relationship violates the Markov condition. For example supply and demand variables in economics frequently violate this condition.
4. The sample must be representative of the population.

One limitation of TETRAD is that it does not allow the user to incorporate temporal ordering so a time series is actually treated as a list of unordered values. Another limitation of this system is that for continuous variables it supports only normally distributed variables. A third limitation is that it cannot allow Feedback which will result in a cycle in the causal graph.

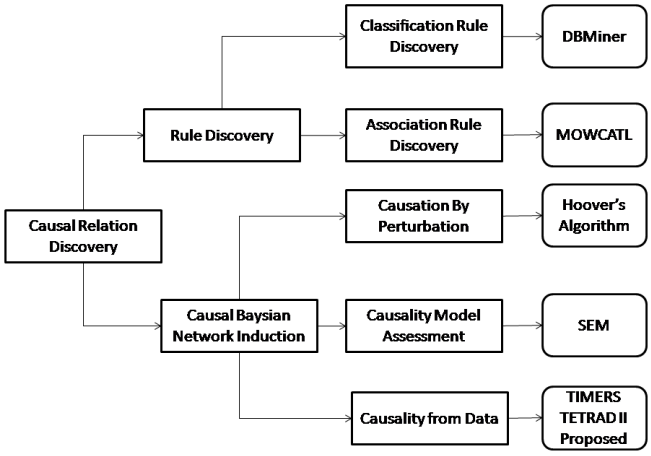
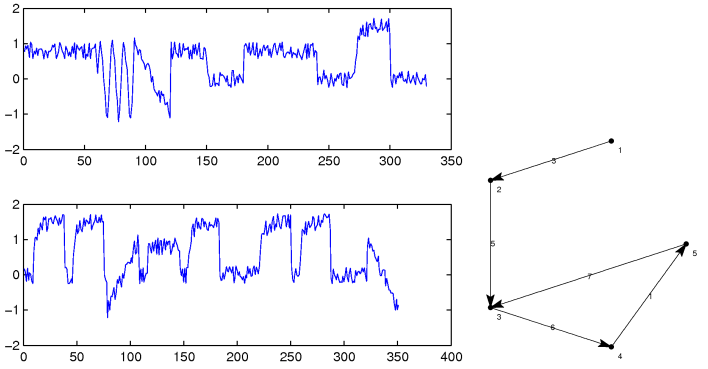


Fig. 1. A taxonomy of causal relation discovery/assessment techniques

Fig. 1 shows a summary of the reviewed approaches to causal relation discovery with examples of each approach. The algorithm proposed in this paper falls under the causality from data category of causal Bayesian induction algorithms. The main contributions of the proposed algorithm are:

– It does not search for causal relations in the instantaneous values of the input but in the time series generated by finding interesting events in the

input. This allows the proposed approach to handle inputs that cannot be handled using any of the reviewed algorithms. Fig. 2 shows an example of such system.
- It combines Granger-Causality's statistical testing with TIMERS-like reversed time check of causal connections
- It supports Feedback (finding causal relations when there is a cycle in the causal graph).
- It can find subdimensional patterns and so does not require patterns to appear in all dimensions of the variable. For example if one of the variables represents acceleration data from a moving object, the system can find patterns that exist in any one, two or the three spatial dimensions even if *distractive* noise exists in the other dimensions.
- It can find the response time (or dead time) between the activation of the cause and the beginning of the effect. This value is found for every causal relationship separately and is represented as a weight on the causal graph.



(a) Time Series X_1 and X_2 (b) Generating Causal Graph

Fig. 2. Example of two variables X_1 and X_2 where patterns in X_1 cause patterns in X_2. This kind of causal relation is not discoverable by any of the reviewed techniques. The proposed algorithm can handle this kind of signals

Fig. 2(a) shows an example of two variables X_1 and X_2. In this example both X_1 and X_2 has five patterns. Activation of some pattern in X_1 causes an activation of another pattern in X_2. The causal graph used to generate this data is shown in Fig. 2(b). The reviewed techniques cannot discover this causal structure because the instantaneous values of the signals are not what matter – and they do not contain any information useful for causality discovery – but the pattern in which these values happen.

There are many real world examples of systems that generate time series of the form shown in X_1 and X_2. For example X_1 could represent movements of

a robotic head while X_2 represents the respiration pattern of its human partner during a face to face interaction and we would like to know if the head motion of the robot has some effect on the respiration pattern of the human which in turn is correlated with the stress level of him/her. In this case the exact value of robot head angles is not of any importance in most cases but the pattern of movement (e.g. looking to the subject, looking to unexpected location etc) is what can affect the respiration rate. Another example is analyzing the relation between two frequency modulated signals to discover if one of them is causing the other which may inform us about the internal workings of an unknown device. A third example is analyzing the relation between a bee's movement pattern and the movement patterns of other bees to determine if there is some form of communication happening (communication entails a causal relation between the message sent and the message received).

3 Proposed System Architecture

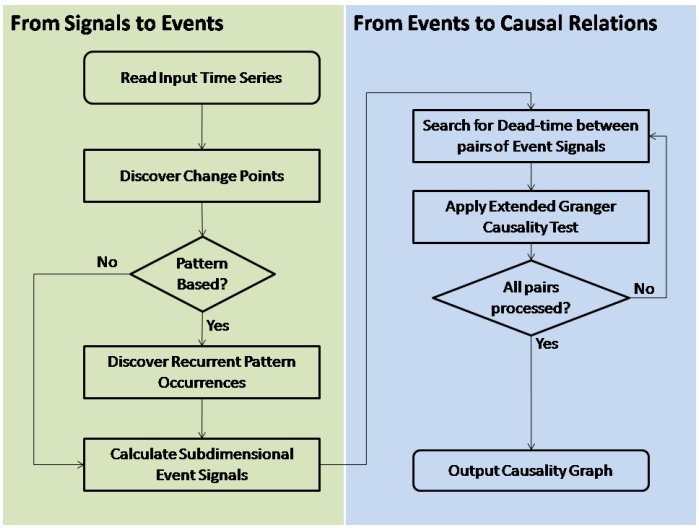


Fig. 3. Overview of the proposed causal relation mining technique. The system proceeds in two main steps: discovering interesting events in the time series followed by discovery of the underlying causal structure.

Fig. 3 summarizes the steps of the proposed approach to discover causal relations in time series. The inputs to the system are a set of n synchronized time series $X_{1:n}$ and a parameter specifying if causality should be discovered in the instantaneous values of the time series or in unknown patterns inside the time series. The algorithm proceeds in two phases. The aim of the first phase is to

discover a set of important events that are useful for discovering the underlying causal structure using the algorithm described in section 4. The second phase takes the occurrence times of these events as input and estimates the causal structure used to generate the original time series in the form of a weighted directed graph (WDG) where the weights correspond to the response time/dead time between the activation of the cause and the start of the effect. This phase is described in section 5.

In the remaining of this paper we use the following operational definition of causality:

Definition: $A \xrightarrow{T} B$ – which means A causes B after T time units – if and only if:

1. $A(t)_{EGC} \rightarrow B(t+T)$. The condition means that A causes B based on the extended granger causality test described in section 5.
2. There is no variable X in the set of tested variables so that $A(t)_{EGC} \rightarrow X(t+T_A)$ and $X(t)_{EGC} \rightarrow B(t+T_B)$ and $T = T_A + T_B$.

4 From Signals to Events

The first phase of the proposed algorithm aims at finding interesting events in the time series that can be utilized to reliably discover the underlying causal structure. The input to this phase is the set of input time series $X_{1:n}(t)$ as well as a Boolean parameter (*pattern-based*) that takes the value $FALSE$ if causal relations are to be found between input dimensions $1:n$ using instantaneous values of the input ($X_{1:n}$).The output of this phase is a set of m time series $E_{1:m}(t)$ where $0 \leq E_{1:m}(t) \leq 1$ called *event signals*. The way to calculate *event signals* depends on the value of *pattern-based* parameter.

In all cases we start by finding the change points of the input time series (applied to every dimension of the input separately) using a Change Point Discovery algorithm. The result of this step is another set of n time series $\hat{X}_{1:n}$ where $0 \leq \hat{X}(t) \leq 1$. Semantically $\hat{X}(t)$ is an estimation of the amount of change in the generating dynamics at time step t. Section 4.1 details the change point discovery algorithm used in this research.

If the causal relations are to be found using the instantaneous values of the input time series (*pattern-based* equals $FALSE$); then the *event signals* are equal to $\hat{X}_{1:n}$ and m is equal to n.

If the causal relations are to be found using recurring patterns in the input time series (*pattern-based* equals $TRUE$); then the *event signals* are calculated by first discovering recurring patterns in every dimension using the algorithm described in section 4.2 which also finds the time steps at which each one of these recurring patterns begins and ends (also called motifs in this chapter). The total number of patterns found is m and determines the number of *event signals* outputted from this phase. Every *event signal* is then calculated using the technique described in section 4.

4.1 Change Point Discovery

The Change Point Discovery problem can be defined as: given any n points real-valued time series $X = \{x(1), x(2), ..., x(n)\}$ find another n real valued time series $X_s(t) = \{x_s(1), x_s(1), ..., x_s(n)\}$ where $0 \leq x_s(t) \leq 1$ and the value of $x_s(t)$ represents the difference between $x(t-p) : x(t)$ and $x(t) : x(t+f)$ for an appropriate difference measure D.

The research in change point (CP) discovery problem have resulted in many techniques including CUMSUM [22], wavelet analysis [23], inflection point search [24], autoregressive modeling [25], Discrete Cosine Transform, and Singular Spectrum Analysis SST [26].

Most of these methods with the exception of SST either discover a single kind of change (e.g. CUMSUM discovers only mean shifts), require ad-hoc tuning for every time series (e.g. wavelet analysis), or assumes a restricted generation process (e.g. Gaussian mixtures). The main disadvantages of SST though are the sensitivity to noise and the need to specify five different parameters.

The main idea of SST is to use PCA to discover the degree of 'difference' between the past and future signal pattern around every point in the time series and use the difference as the change score for this point.

Many researchers suggested improvements to traditional SST even though most of these suggestions targeted increasing the speed of the algorithm not its accuracy. [27] introduced online SVD and [28] proposed Krylov Subspace Learning. [28] also proposed using the angle between the subspaces associated with the major PCA components of the past and the future to calculate the change score at every point. The main problem of this proposal is the assumption that all Eigen vectors are equal in importance which can be an inaccurate assumption if the distribution of the top Eigen values is not nearly uniform (a condition that happens most of the time in our experience with real world time series). In this paper we propose a different approach to utilize the information of the Eigen values as well as the Eigen vectors for finding the change score.

The authors designed the Robust Singular Spectrum Transform (RSST) [29] for discovering change points in time series which reduces the number of parameters required into two parameters rather than five and dramatically increases the specificity of the traditional SST without decreasing its sensitivity. RSST is linear in time and space requirements as SST and adds very small constant increase in the processing time. Moreover speedup techniques like the use of Krylov Subspace Learning suggested in [28] can be directly utilized with RSST. Extensive comparisons between SST and RSST on synthetic data support the superiority of RSST in both synthetic and real world data [29]. In this work we use RSST for the discovery of change points in all dimensions of the input time series. A brief explanation of the algorithm is given in section 4.1

Robust Singular Spectrum Transform The essence of the *RSST* transform is to find for every point $x(i)$ the difference between a representation of the dynamics of the few points before it (i.e. $x(i-p) : x(i)$) and the few points

Algorithm 1 Robust Singular Spectrum Transform (RSST)

Inputs: $X = \{x(1), x(2), ..., x(T)\}$.
parameters: w, n.
Outputs: $X_s(t) \leftarrow \{x_s(1), x_s(1), ..., x_s(T)\}$ where $0 \leq x_s(i) \leq 1$ for $i \leftarrow 1 : T$.
procedure SST ▷ Calculates the SST Transform
  for $t \leftarrow w \times n : T - w \times n$ **do**
    1. Extract past pattern:
    1.1. $m \leftarrow n$ and $g \leftarrow 0$
    1.2. Calculate Subsequences $SEQ(t)$
    1.3. Find the Herkel Matrix $H(t)$ at point t (Eq. 13)
    1.4. Find the SVD decomposition of $H(t)$ by solving 14.
    1.5. Estimate the optimal value for $l(t)$
    1.6. $U_l(t) \leftarrow$ first $l(t)$ left singular vectors
    2. Extract future pattern:
    2.1. Calculate subsequences $R(t)$
    2.2. Find the Herkel Matrix of the future $G(t)$ (Eq. 16)
    2.3. Find the optimal value for $l_f(t)$
    2.4. Find the largest $l_f(t)$ Eigen vector of $G(t)$ ($\beta_{1:f}(t)$)
    3. Calculate Change Score Guess:
    calculate $\hat{x}_s(t)$ using 23
  end for
  4. Remove Noise Effect:
  4.1. Estimate x_s^{min} and x_s^{max}
  for $t \leftarrow w \times n : T - w \times n$ **do**
    4.2. Find the average and variance of the w points subsequences before and after point t
    4.3. Modify $\hat{x}_s(t)$ using Eq. 28
  end for
  4.4. Keep only local maximums of $x_s(t)$
  4.5. Normalize $x_s(t)$
end procedure

after it (i.e. $x(i+g) : x(i+f)$). This difference is normalized to have a value between zero and one and named $x_s(i)$.

The dynamics of the points before and after the current point are represented using the Herkel matrix which is calculated as:

$$H(t) = [seq(t-n), ..., seq(t-1)] \tag{13}$$

where $seq(t) = \{x(t-w+1), ..., x(t)\}^T$

Singular Value Decomposition (SVD) is then used to find the singular values and vectors of the Herkel Matrix by solving:

$$H(t) = U(t) S(t) V(t)^T \tag{14}$$

where $S(i+1, i+1) \leq S(i,i) \leq (i-1, i-1)$.

In RSST the value of $l(t)$ is allowed to change from point to point in the time series depending on the complexity of the signal before it. To calculate a

sensible value for l we first sort the singular values of $H(t)$ and find the corner of the accumulated sum of them ($l(t)$) [the point at which the tangent to the curve has an angle of $\pi/4$]. The singular vectors with singular values higher than this value are assumed to be caused by the genuine dynamics of the signal while the other directions encode the effect of noise. This dynamic setting of l reduces the effect of noise on the final results as will be shown in the following section.

A similar procedure is used to find the direction of largest change in the dynamics for the future of the signal by concatenating m overlapping windows of size w starting g points after t according to:

$$r(t+g) = \{x(t+g), ..., x(t+g+w-1)\}^T \tag{15}$$

$$G(t) = [r(t+g), ..., r(t+g+m-1)] \tag{16}$$

The Eigen vector $\beta(t)$ corresponding to the direction of maximum change in the future of the signal is found by solving:

$$G(t) G(t)^T u^g = \mu u^g \tag{17}$$

$$\beta(t) = u^g_m \tag{18}$$

where $m = \arg\max_i (\mu_i)$

To find a first guess of the change score around every point, RSST uses the $l_f(t)$ Eigen vectors of $G(t) G(t)^T$ with highest corresponding Eigen values ($\lambda_{1:l_f}$) rather than only the first Eigen vector used in SST. The value of $l_f(t)$ is selected using the same algorithm for selecting $l(t)$.

$$G(t) G(t)^T u^g = \mu u^g \tag{19}$$

$$\beta_i(t) = u^g_i,\ i \leq l_f\ and \lambda_{j+1} \leq \lambda_j \leq \lambda_{j-1}\ for\ 1 \leq j \leq w \tag{20}$$

Each one of these l_f directions are then projected onto the hyperplane defined by $U_l(t)$

The projection of $\beta_i(t)$s and the hyperplane defined by $U_l(t)$ is then found using:

$$\alpha_i(t) = \frac{U_l^T \beta_i(t)}{\left\| U_l^T \beta_i(t) \right\|}, i \leq l_f \tag{21}$$

The change scores defined by $\beta_i(t)$s and $\alpha_i(t)$s are then calculated as:

$$cs_i(t) = 1 - \alpha_i(t)^T \beta_i(t) \tag{22}$$

The first guess of the change score at the point t is then calculated as the weighted sum of these change point scores where the Eigen values of the matrix $G(t)$ are used as weights.

$$\hat{x}(t) = \frac{\sum_{i=1}^{l_f} \lambda_i \times cs_i}{\sum_{i=1}^{l_f} \lambda_i} \tag{23}$$

After applying the aforementioned steps we get a first estimate $\hat{x}(t)$ of the change score at every point t of the time series. RSST then applies a filtering step to attenuate the effect of noise on the final scores. The main insight of this filter is that the reduction of SST specificity in noisy signals happens in the sections in which noise takes over the original signal in the time series. The response of SST at these sections can be modeled by a random walk around a high average for uncorrelated white noise. The filter used by RSST discovers these sections in which the average and the variance of $\hat{x}(t)$ remains nearly constant and attenuates them. The filter first calculates the average and variance of the signal before and after each point using a subwindow of size w:

$$\mu_b(t) = \frac{\sum_{i=0}^{w-1} \hat{x}(t-i)}{w} \tag{24}$$

$$\sigma_b(t) = \frac{\sum_{i=1}^{w} (\hat{x}(t-i) - \mu_b(t))^2}{w-1} \tag{25}$$

$$\mu_a(t) = \frac{\sum_{i=1}^{w} \hat{x}(t+i)}{w} \tag{26}$$

$$\sigma_a(t) = \frac{\sum_{i=1}^{w} (\hat{x}(t+i) - \mu_a(t))^2}{w-1} \tag{27}$$

The guess of the change score at every point is then updated by:

$$\tilde{x}(t) = \hat{x}(t) \times |\mu_a(t) - \mu_b(t)| \times \left| \sqrt{\sigma_a(t)} - \sqrt{\sigma_b(t)} \right| \tag{28}$$

where μ_a and σ_a are the mean and variance of $\hat{x}(t)$ in a subsequence of length w before the point t while μ_b and σ_b are the mean and variance of $\hat{x}(t)$ in a subsequence of length w after the point t

RSST then keeps only the local maxima of $\tilde{x}(t)$ and normalizes the resulting time series by dividing with its maximum. This normalized signal $x(t)$ represents the final change score of RSST.

4.2 Pattern/Motif Discovery

If the causal relations are to be found in recurring patterns in the input streams rather than in the instantaneous values of these streams; a reliable algorithm to extract these patterns and their occurrences is needed. This problem is known in data mining literature as motif discovery in time series. The research in unsupervised motif discovery have led to many techniques including the PROJECTIONS algorithm [30], PERUSE [31], Gemoda [32] among many others ([33]). These algorithms try to find all recurring motifs with lengths between l_{min} and l_{max} time steps where l_{min} and l_{max} are input parameters to the algorithm (or at least one of them).

With the exception of Gemoda which is quadratic in time and space complexities, these algorithms aim to achieve sub-quadratic time complexity by first looking for candidate motif stems using some heuristic method and then doing exhaustive motif detection instead of motif discovery which is linear in time.

The most used method for finding these stems is the PROJECTIONS algorithm [30] which requires discretization of the data using the SAX [5] algorithm. A major drawback of all methods relying on discretization of the data is the need to specify a word length and a vocabulary size that are difficult to decide in real world situations.

Another problem of most of the previously proposed techniques is the need to specify an exact or at least roughly correct motif length.

A third problem is deciding when to stop searching for new motifs ([33] suggested using density estimation for this purpose). One common problem to all the algorithms based on the PROJECTIONS algorithm [33, 30] is the need to construct and keep the collision matrix which is in general quadratic in the length of the SAX word describing the time series. Given that the optimal word size can be very short for short motifs in signals with high frequency components, the size of this matrix can grow quadratic with the length of the time series.

Catalano et al. [34] suggested a very efficient algorithm for locating variable length patterns in data series using random sampling that allows it to run in linear time and constant memory. The main problem of this approach is that it relies on random sampling which can lead to poor performance for long time series with infrequent embedded motifs including long records of human activities.

Most of the methods proposed for motif discovery that we are aware of assume no prior knowledge of the probable locations of the motifs which leads to this explosion in the processing time or space needed. In response to that, the authors designed two variations of Catalano et al.'s algorithm called MCInc and MCFull [6] that can utilize prior information about the locations of motif occurrences to speed up the operation of discovery to linear and in some cases sublinear both in time and space. Moreover, the output of any change point discovery algorithm (e.g. RSST described in section 4.1 can be used as the input constraint to limit the search to the points of the time series around the points of high change score based on the assumption that interesting patterns happen near the points in which there is a change in the underlying dynamics.

Catalano et al.'s algorithm [34] works by processing a fixed sized set of candidate and comparison windows randomly sampled from the time series. The input to the algorithm are minimum and maximum motif lengths ($\hat{w}$,w) as well as the number of candidate windows (n_c) which determines the space requirements of the algorithm. The steps of the algorithm given a single dimension input time series $X(t)$ can be summarized as:

1. Select a subsequence s^w of length $w \geq l_{max}$.
2. Select w values randomly from X and concatenate them to form a noise sequence n^w
3. select a set of n_c comparison subsequences of X ($\{{c^w}_i\}$) each of length w.
4. Find the set $S^{\hat{w}}$ of subsequences of length $\hat{w}$ where $\hat{w} \leq l_{min}$ for s^w,${c^w}_i$, and n^w. Then normalize all of the resulting subsequences to have unit mean square.
5. For the candidate subsequence of s^w ($s_k^{\hat{w}}$) do the following
   (a) Randomly select $w-\hat{w}-1$ subsequences from the set of all subsequences of the comparison windows (${c^w}_i$). call this set the comparison set $\hat{c}_j^{\hat{w}}$
   (b) Find the distances $d_{kj} = d\left(s_k^{\hat{w}}, {\hat{c}^w}_j\right)$.
   (c) Group the set $\hat{c}_j^{\hat{w}}$ with their parent subsequence ${c^w}_i$ and for every group select $\hat{c}_j^{\hat{w}}$ that has the minimum distance d_{kj}. This leads to a set of R subsequences $\tilde{c}_r^{\hat{w}}$ where $R \leq n_c$
   (d) Keep only the $\hat{R}$ subsequences of $\tilde{c}_r^{\hat{w}}$ with least d_{kr}.
   (e) Repeat the previous three steps for subsequences of the noise subsequence n^w. This leads to another set of $\hat{R}$ subsequences called $\tilde{n}_r^{\hat{w}}$
6. Remove all candidate subsequences $s_k^{\hat{w}}$ that have similar average distance with both $\tilde{n}_r^{\hat{w}}$ and $\tilde{c}_r^{\hat{w}}$ then repeat the steps above using this reduced set. Repeat this reduction for n_r times.
7. If the final set $s_k^{\hat{w}}$ is not empty output each of them as a motif seed M_s after concatenating any continuous subset of them.

Given the change point score ($\tilde{X}(t)$) calculated using RSST (see section 4.1) we can speed up Catalano algorithm by modifying the first three steps as follows:

1. Apply a Gaussian smoothing filter ($N(0,\sigma^2)$) to the original $\tilde{X}$ constraint which results on the smoothed constraint $\tilde{P}$.
2. Normalize $\tilde{P}$ so that $\sum_{t=1}^{n} \hat{p}(t) = 1$ and $0 \leq \hat{p}(t) \leq 1$.
3. Randomly select a subsequence s^w of length $w \geq l_{max}$ using $\hat{P}$ as the probability distribution.
4. Randomly select w values from X and concatenate them to form a noise sequence n^w using $1-\hat{P}$ as the probability distribution.
5. Randomly select a set of n_c comparison subsequences of X ($\{{c^w}_i\}$) each of length w using $\hat{P}$ as the probability distribution.

For more details about this algorithm and extensive evaluation of its performance on synthetic and real world data please refer to [6].

4.3 Calculating Subdimensional Event Signals

The output of this phase is a set of m time series $E_{1:m}(t)$ where $0 \leq E_{1:m}(t) \leq 1$ called *event signals*. The way to calculate *event signals* depends on the value of *pattern-based* parameter.

If the causal relations are to be found using the instantaneous values of the input time series (*pattern-based* equals $FALSE$); then the *event signals* are equal to $\hat{X}_{1:n}$ and m equals to n.

If the causal relations are to be found using recurring patterns in the input time series (*pattern-based* equals $TRUE$); then the *event signals* are calculated by first discovering recurring patterns in every dimension using the algorithm described in section 4.2 which also finds the time steps at which each one of these recurring patterns begins and ends (also called motifs in this chapter) appear. The total number of patterns found is m_p and determines the number of *event signals* outputted from this phase. Every *event signal* is then calculated from the locations of recurring pattern occurrences using the following equation:

$$\hat{E}_i(t) = \begin{cases} t/\Delta_i + 1 - L_{ij}/\Delta_i & if\ max(L_{ij} - \Delta_i, B_{ij}) \leqslant t \leqslant L_{ij} \\ -t/\Delta_i + 1 + L_{ij}/\Delta_i & if\ L_{ij} < t \leqslant min(L_{ij} + \Delta_i, End_{ij}) \\ 0 & otherwise \end{cases} \tag{29}$$

where B_{ij}, L_{ij}, and End_{ij} are the beginning middle, and end time step of the occurrence number j of pattern i, Δ_i is a parameter specifying the confidence of the pattern detector. In all of the experiments done in this chapter we fixed Δ_i to ∞. This equation represents two lines intersecting at the value 1 in the center of the discovered occurrence. $E_i(t)$ then decreases monotonically with rate $1/\Delta_i$ until the two extremes of the discovered occurrence.

The final step is to combine event signals that are estimated to correspond to the same pattern but in multiple dimensions. First the correlation of every pair of *event signals* is calculated using Pearson correlation. If the correlation coefficient is over a predefined threshold and is statistically significant the two *event signals* are combined in a single *subdimensional event signal* using:

$$E_k(t) = \hat{E}_i(t) \times \hat{E}_j(t) \tag{30}$$

Finally if some *event signal* ($\hat{E}_l$) was not combined with any other *event signal*; it is added to the final *subdimensional event signals* (E). This procedure is iterated until no *event signals* can be combined anymore. It should be noted that this procedure cannot distinguish *event signals* that correspond to occurrences of the same multidimensional pattern and immediate causal relations. If immediate causal relations are possible in the system then *event signals* should not be combined and in this case $E = \hat{E}$.

5 From Events to Causal Relations

The output of the previous phase is a set of m time series called the *subdimensional event signals* ($E_{1:m}(t)$). Each one of these signals represents one variable

and the goal of this final phase of the algorithm is to discover the causal relations between these variables. In this section we present our proposed algorithm for this phase.

The first step in this phase is to search for pairwise dead-time for every pair of *subdimensional event signals*. The dead-time between any cause and its effect is defined as the time in time-steps required between the start of the cause and the start of the effect. To find the dead-time between a pair of signals E_i and E_j, we calculate the Pearson correlation coefficient between $E_i(t)$ and $E_j(t+\tau)$ for different values of τ and select the dead-time between variables i and j as:

$$d_{i\to j} = \arg\max_{\tau} \left(\rho\left(E_i(t), E_j(t+\tau)\right)\right)$$

$$d_{j\to i} = \arg\max_{\tau} \left(\rho\left(E_j(t), E_i(t+\tau)\right)\right)$$

After obtaining the dead-time ($d_{i\to j}$) between variables i and j we apply an extended version of Granger-Causality test as follows: First, we estimate the following equation also by OLS

$$\hat{X}_j(t) = \varepsilon_1 + u(t) + \sum_{l=1}^{m}\sum_{k=1}^{\rho} \alpha_{lk}\hat{X}_l(t-k-d_{l\to j}) \tag{31}$$

Second, we estimate the following restricted equation also by OLS:

$$\hat{X}_j(t) = \varepsilon_2 + e(t) + \sum_{l\neq i}\sum_{k=1}^{\rho} \beta_{lk}\hat{X}_l(t-k-d_{l\to j}) \tag{32}$$

We then calculate the sum of squared residuals (SSR) in both cases:

$$\begin{aligned} SSR_1 &= \sum_{i=1}^{T} u^2(t) \\ SSR_0 &= \sum_{i=1}^{T} e^2(t) \end{aligned} \tag{33}$$

We then calculate the test statistic S_ρ as:

$$S_\rho = \frac{(SSR_0 - SSR_1)/\rho}{SSR_1/(T-2\rho-1)} \tag{34}$$

If S_ρ is larger than the specified critical value then we estimate the following equation also by OLS:

$$\hat{X}_j(t) = \varepsilon_3 + f(t) + \sum_{l\neq i}\sum_{k=1}^{\rho} \lambda_{lk}\hat{X}_l(t-k-d_{l\to j}) + \sum_{k=1}^{\rho} \gamma_k\hat{X}_i(t+k+d_{j\to i}) \tag{35}$$

We then calculate the sum of squared residuals (SSR) in this case:

$$SSR_2 = \sum_{i=1}^{T} f^2(t) \tag{36}$$

We then calculate the test statistic S_ρ^f as:

$$S_\rho^f = \frac{(SSR_0 - SSR_2)/\rho}{SSR_2/(T - 2\rho - 1)} \tag{37}$$

Now if S_ρ^f is less than the specified critical value then we reject the null hypothesis that E_i does not cause E_j after $d_{i\to j}$ steps and in this case $X_i \xrightarrow{d_{i\to j}} X_j$. The basis of this decision is that having S_ρ greater than the specified critical value while S_ρ^f smaller than it indicates that the past of E_i gives information about E_j while the future of E_i does not give extra information. This means that the relation between E_i and E_j satisfies the necessary condition that a cause must precede its effect.

If S_ρ^f is greater than the specified critical value then we have a more complex situation as both past and future values of E_i are useful for predicting E_j. This can happen if there is a hidden variable that causes both E_i and E_j but with variable delays so that sometimes E_i precedes E_j and vice versa. In this case we add a hidden variable H and assume that $H \to E_j$ and $H \to E_i$.

After processing all pairs of *subdimensional event signals* ($2m\,(m-1)$ operations), the complete causal graph is built and outputted.

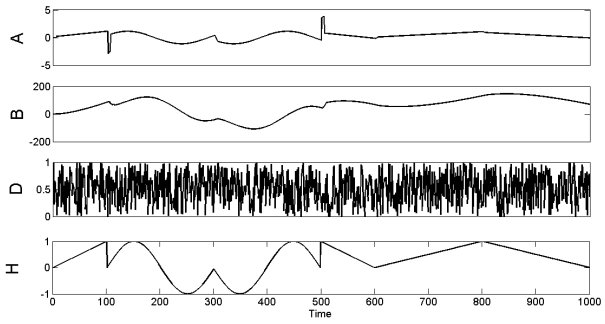


Fig. 4. Time series used to evaluate the effectiveness of the proposed extended Granger causality test. A seams to cause B if H is not taken into account using standard Granger Causality test. Using the proposed extended version the system infers the existence of H and correctly concludes that H causes both A and B.

To illustrate the usefulness of the extra steps of the proposed extended Granger-Causality check, consider the signals presented in Fig. 4. These signals were generated using the following linear systems:

$$A\,(t) = 4H\,(t-1) - 3H\,(t-6) + 0.5A(t-1)$$
$$B\,(t) = 8H\,(t-1-\kappa) - 6H\,(t-6-\kappa) + 0.99B\,(t-1)$$

where κ is a random number uniformly distributed between 0 and 9.

H is a sequence of linear and sinusoidal signals while D is a random signal that is not correlated with anything else.

Applying standard Granger Causality test (as explained in equations 9:12) to A and B to the data set containing (A, B, and D) results in falsely accepting that A causes B. On the other hand applying the proposed extended method to the same data detects that both A and B are caused by a common hidden variable that is not in the data set.

6 Evaluation

Evaluation of causal rule discovery algorithms is complicated by the difficulty to recover the causal structure of the domain at hand. Synthetic data evaluation is one way around this problem as the evaluator can design the underlying causal structure (even though some philosophical views may not accept the relation between computer variables to be causal in a strict sense [35]). The other way around this is to use a domain in which a tremendous amount of knowledge supports some known causal model (e.g. use a mechanical system known to be governed by Newtonian mechanics). The third possibility is to utilize a controlled experimental situation in which the causal structure is fixed by the designer.

In this paper we evaluate the proposed system using synthetic data in section 6.1 and using a controlled experiment in section 6.2. It should be noted that using a controlled experiment in this case does not entail any *manipulation* of the causal structure as we do not intend to discover the causal structure of the system in the normal (uncontrolled) case before the experiment but to discover the causal structure that was fixed by the experiment itself. This leaves the proposed approach under the category of *causality from data* rather than *causation from perturbation*.

6.1 Synthetic Data

First preliminary evaluation of the proposed algorithm was done using synthetic data that was generated using a deterministic model. The model consists of two signals X_1 and X_2 each of which is a concatenation of the outputs from five different patterns. The patterns constituting X_1 and X_2 are the same (generated from same processes). Fig. 5 shows the five patterns generated by these processes. Fig. 2 shows an example short sequence of these patterns in X_1 and X_2 along with the causal model used to generate them. In all cases the active process in X_1 *causes* the activation of one of these five processes in X_2 after some delay that is selected randomly using a uniform distribution.

To evaluate the algorithm we applied it using the two signals generated from 16 randomly generated causal models each was used to generate 100 different time series pairs of 1000 points each. In all cases the system was able to discover the five patterns correctly and to regenerate the same causality graph with a mean error of 6% in the discovered delays and standard deviation of 3%. It should

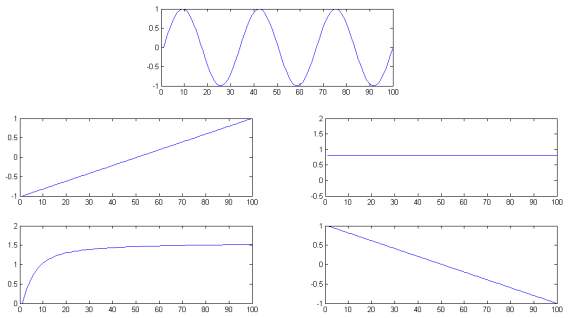


Fig. 5. The five patterns used in the generation of synthetic data.

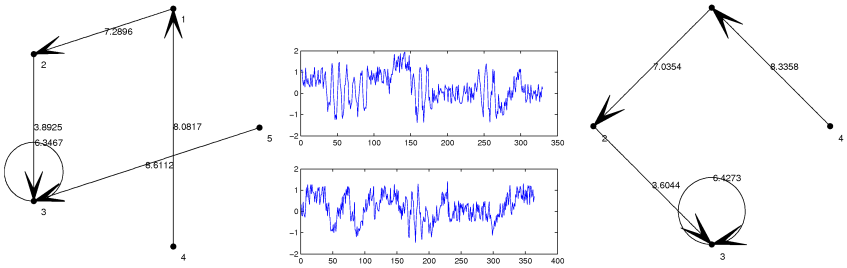
be noted that the output of the algorithm in this example is not five patterns but twelve corresponding to the five patterns in the two input time series. To simplify the display we combined corresponding patterns (using simple Euclidean distance measure) together both in the generating and discovered graphs. This perfect discovery results just show that the algorithm can *in principle* work but gives no information about its accuracy as the patterns used are easily sparable and there was no noise, complex feedback, or unknown hidden factors in this experiment.

The second experiment using synthetic data targeted evaluating the accuracy of the system in the presence of noise in the input time series. The same generation process was used as in the previous experiment except that uniformly distributed noise of variable peak-to-peak value was added to the signal X_i and X_j before passing them to the algorithm. Fig. 6 shows five examples of the operation of the system at P-P noise level of 50% of the P-P values of the original signal. At this high noise level the maximum number of missed causal relations was a single one, the false alarm rate (taking a non-causal relation as causal) was still 0% while the false rejection rate (missing a causal relation as a non-causal) was 8.9% (detection accuracy is 91.1%).

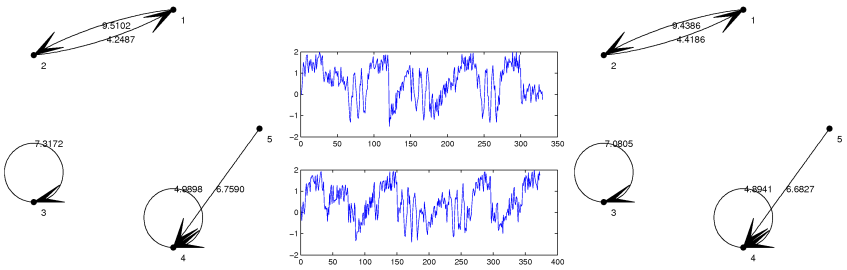
6.2 Mining Human-Robot Interaction Data

This section presents a feasibility study to assess the applicability of the proposed approach in learning the causal structure in a controlled experiment.

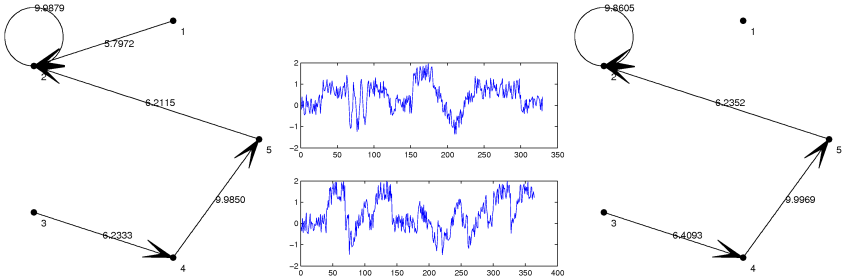
The evaluation experiment was designed as a Wizard of Ooz (WOZ) experiment in which an untrained novice human operator is asked to use hand gestures to guide the robot shown in Fig. 7 along the two paths in two consecutive sessions. The subject is told that the robot is autonomous and can understand any gesture (s)he will do. A hidden human operator was sitting behind a magic mirror and was translating the gestures of the operator into the basic primitive



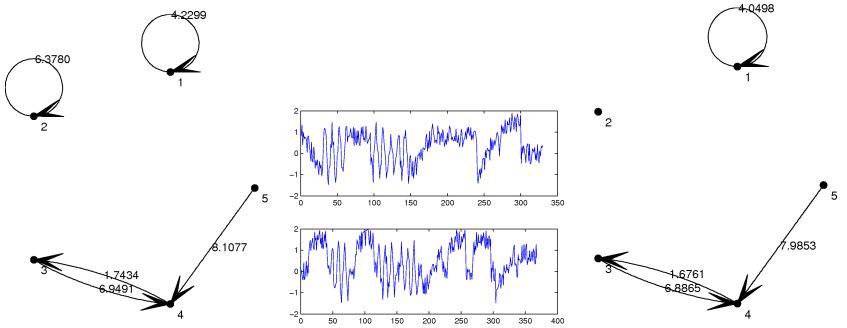
(a) Model (b) Time Series (c) Discovered Model



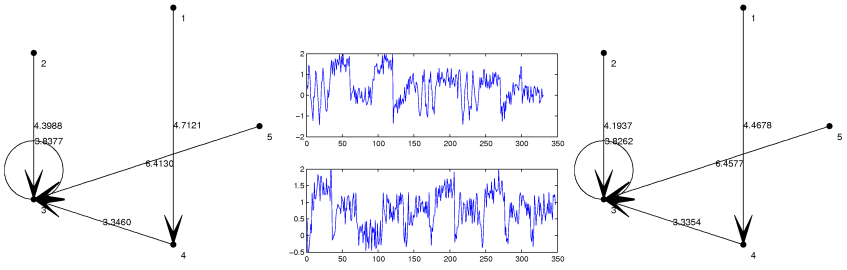
(d) Model (e) Time Series (f) Discovered Model



(g) Model (h) Time Series (i) Discovered Model



(j) Model (k) Time Series (l) Discovered Model



(m) Model (n) Time Series (o) Discovered Model

Fig. 6. Five example models used in the second experiment (after adding noise) with a part of the generated time series and the discovered model using the proposed algorithm

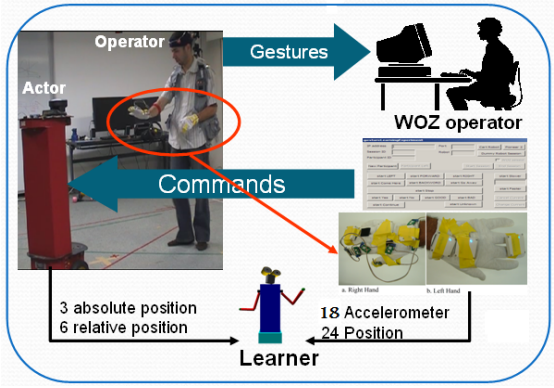


Fig. 7. The experiment setup. The actor is a WOZ operated robot. Command stream is 48 dimensions accelerometer and position sensor signals, and the action stream is 6 dimensional absolute and operator-relative location of the robot

actions of the WOZ robot that were decided based on an earlier study of the gestures used during navigation guidance [36].

In this design the movement of the robot is known to be *caused* by the commands sent by the WOZ operator which in turn is partially *caused* by the gestures of the participant. This can be formally explained as:

$$G_i \xrightarrow{T_1} W_j$$
$$W_j \xrightarrow{T_2} M_k$$

where G_i represent some gesture done by the participant, W_j represent some action done by the WOZ operator (e.g. pressing a button on the GUI of the control software), and M_k represents some pattern in the movement of the robot (e.g. moving toward the participant, stopping, etc).

Eight participants of ages 21 to 34 (all males) that have no prior knowledge of robots and do not study in the field of engineering were selected for this experiment. The total number of sessions conducted was 16 sessions with durations ranging from 5:34 minutes to 16:53 minutes.

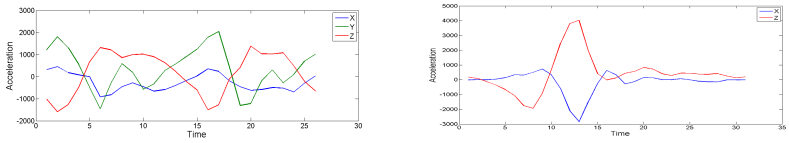
The motion of the subject's hands (G) was measured by six B-Pack ([37]) sensors attached to both hands as shown in Fig. 7 generating 18 channels of data. The PhaseSpace motion capture system ([38]) was also used to capture the location and direction of the robot using eight infrared markers. The location and direction of the subject was also captured by the motion capture system using six markers attached to the head of the subject (three on the forehead and three on the back). 8 more motion capture markers were attached to the thumb and index of the right hand of the operator.

The following four feature channels were used as representation of robot's action (M):

- The directional speed of the robot in the XZ (horizontal) plane in the direction the robot is facing (by its cameras) (2 dimensions).
- The direction of the robot in the XZ plane as measured by the angle it makes with the X axis (1 dimension).
- The relative angle between the robot and the actor (1 dimension).
- The distance between the operator and the actor (1 dimension).

The time of button presses in the GUI used by the WOZ operator was collected in synchrony with both participant gestures and robot actions. The interface had seven buttons (related to robot motion) each of which can be toggled on and off and a single button can be on at any point of time. The WOZ operator's actions were represented by a single input dimension giving the ID of the currently active button (from 1 to 7).

This leads to a total of 67 input dimensions. The generating causal model (ground truth) consists of seven gestures, seven button press configuration and corresponding seven robot actions (21 total patterns).



(a) The Come-Here Gesture (b) The Stop Gesture

Fig. 8. Examples of discovered recurring patterns in the gestures of the operator (G). The first gesture was found in 3 dimensions of the 48 gesture input dimensions while the second one was found in only two dimensions

Fig. 8 shows two example gestures discovered by the algorithm. The first gesture – corresponding to the command come-here needs – uses only three out of the 48 input gesture dimensions (corresponding to the sensor attached to the tip of the right hand). The second gesture – corresponding to the stop command – needs even less dimensions (only 2) because the acceleration in the Y direction does not matter for this command. This ability of the algorithm to discover patterns in a subset of the dimensions reduces its sensitivity to noise in the other distracting dimensions.

Fig. 9 shows the accuracy of the algorithm in discovering gestures and gesture occurrences. As shown in the Figure, the algorithm discovered all the types of gestures used by the subjects except the *go away* gesture which was mostly confused with the *Backward* gesture. The algorithm also made four false positive errors. The algorithm can then detect 85.7% (6 out of 7) of the gesture types and in average it finds 82% of the occurrences of discovered gestures.

The algorithm correctly discovered all the recurring patterns (corresponding to the seven actions) in the WOZ operator's time series data.

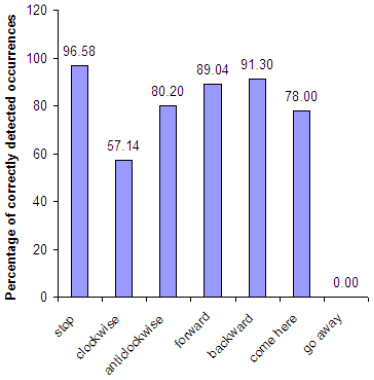
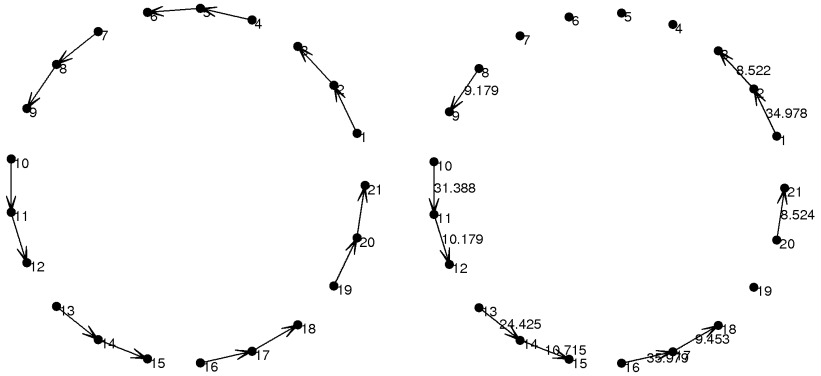


Fig. 9. The detection accuracy for the gestures used in the experiment.

For robot's actions, the total number of basic actions discovered by the algorithm is eight actions. 87.5% of these actions correspond to authentic actions that were implemented in the interface of the WOZ software (7 out of 8). The accuracy of detecting the discovered actions was 88.3%.



(a) The generation model (b) The Discovered Model

Fig. 10. Comparison between the generation model (based on experimental setup) and discovered causal graph using the proposed algorithm

Fig. 10 shows the generating model and the final discovered model. To simplify the display the nodes for patterns were numbered so that gestures correspond to nodes with numbers dividable by seven, WOZ operator actions corresponded to nodes with numbers that have a remainder of one when divided by seven, and Robot movement patterns corresponded to nodes with numbers that

have a remainder of two when divided by seven. This way, the generating model is a seven cliques graph as shown in Fig. 10(a). In the discovered model, nodes that correspond to false patterns where removed from the graph as none of them had any causal connection to any other node in the system. As Fig. 10(b) shows, the discovered model contains no false alarms (non-causal relationships assumed causal) but it has four false rejections (causal relationships assumed non-causal) with a false-rejection rate of 28.57%.

7 Conclusion

In this chapter we reviewed the available techniques for discovering causal relations from data. The chapter also described in details a novel causal relation discovery algorithm that does not work directly on the instantaneous values of the input time series but discovers interesting events in these time series either as points of change in the underlying dynamics or recurring pattern occurrences.

The algorithm works in two phases. In the first phase it discovers interesting *events* in the input multidimensional data and then in the second phase it discovers the causal structure based on the relative locations of these *events* in time. During event discovery the algorithm makes use of a robust change point discovery algorithm developed by the author called the Robust Singular Spectrum Transform (RSST) as well as a recurrent pattern discovery algorithm that utilizes RSST output to work in linear time and space. During causal relation discovery the algorithm uses an extended Granger-Causality test to assess causal relations between the occurrences of events. The main difference between the proposed extended test over traditional Granger-Causality tests is the utilization of an extra statistic to confirm that the past of the cause contains information that helps in predicting its effect while the future of the cause does not contain equivalent information. Using this kind of confirmation it is possible for the proposed algorithm to discover the existence of hidden common causes.

The proposed system was evaluated using both synthetic and real world data. Synthetic evaluation showed that even with 50% noise level the system can still recover the causal structure with an accuracy of 91.1%. For real world evaluation we used controlled experiment that involved a human participant controlling a robot using free hand gestures through a hidden operator. Analysis of the result of this experiment shows that the system can recover the causal structure of the underlying system with accuracy 71.4% (10 out of 14 relations).

Extended evaluation of the proposed system using data from domains with known causal structure (e.g. mechanical systems, electrical systems etc) will be done in the future to confirm the applicability of the proposed system to general data. One direction of future research is to allow the system to utilize domain knowledge to reduce the errors in the discovered causal graph. Another direction is to extend the system to work in multiple time scales to produce a hierarchical model of causal structure at different time scales of the input time series.

References

1. Wang, X., Ye, L., Keogh, E., Shelton, C.: Annotating historical archives of images. In: JCDL '08: Proceedings of the 8th ACM/IEEE-CS joint conference on Digital libraries, New York, NY, USA, ACM (2008) 341–350
2. Inokuchi, A., Washio, T.: Feasibility of graph sequence mining based on admissibility constraints. In: Thid International Workshop on Data Mining and Statistical Science. (2008) 1–4
3. Wang, X., Smith, K.A., Hyndman, R.J.: Dimension reduction for clustering time series using global characteristics. Computational Science ICCS 2005 **3516** (2005) 792–795
4. Kulic, D., Takano, W., Nakamura, Y.: Incremental on-line hierarchical clustering of whole body motion patterns. RO-MAN 2007.
5. Keogh, E., Lin, J., Fu, A.: Hot sax: efficiently finding the most unusual time series subsequence. Data Mining, Fifth IEEE International Conference on (Nov. 2005) 8 pp.–
6. Mohammad, Y., Nishida, T.: Constrained motif discovery. In: International Workshop on Data Mining and Statistical Science (DMSS2008). (September 2008)
7. Aristotle: Metaphysics Book V Part 1
8. Glymour, C.: Learning,predictionandcausalbayesnets. TRENDS in Cognitive Sciences **7**(1) (Jan. 2003) 43 – 48
9. Menzies, P., Price, H.: Causation as a secondary quality. British Journal for the Philosophy of Science **4** (1993) 187–203
10. Pearl, J.: Causality: Models, Reasoning, and Inferenc. Cambridge University Press (2000)
11. Han, J., Fu, Y., Wang, W., Chiang, J., Gong, W., Koperski, K., Li, D., Lu, Y., Rajan, A., Stefanovic, N., Xia, B., Zaiane, O.R.: Dbminer: A system for mining knowledge in large relational databases. In: In Proc. 1996 Int'l Conf. on Data Mining and Knowledge Discovery (KDD'96, AAAI Press (1996) 250–255
12. Das, K., Lin, I., Mannila, H., Renganathan, G., Smyth, P.: Rule discovery from time series. In: the 4th International Conference of Knowledge Discovery and Data Mining, AAAI Press (1998) 16–22
13. Hipp, J., Güntzer, U., Nakhaeizadeh, G.: Algorithms for association rule mining — a general survey and comparison. SIGKDD Explor. Newsl. **2**(1) (2000) 58–64
14. Lee, A.J., chuen Lin, W., sheng Wang, C.: Mining association rules with multidimensional constraints. Journal of Systems and Software **79** (2006) 79–92
15. Sarker, B.K., Hirata, T., Uehara, K., Bhavsar, V.C.: Mining Association Rules from Multi-stream Time Series Data on Multiprocessor Systems. In: Parallel and Distributed Processing and Applications. Springer (2005) 662–667
16. Hoover, K.: The logic of causal inference. Economics and Philosophy **6** (1990) 207–234
17. Tian, J., Pearl, J.: Causal discovery from changes. In: In: Proceedings of UAI 2001, Morgan Kaufmann (2001) 512–521
18. Gelper, S., Croux, C.: Multivariate out-of-sample tests for granger causality. Comput. Stat. Data Anal. **51**(7) (2007) 3319–3329
19. Ding, M., Chen, Y., Bressler, S. In: Granger causality: Basic theory and application to neuroscience. Wiley (2006)
20. Karimi, K., Hamilton, H.J.: Distinguishing Causal and Acausal Temporal Relations. In: Advances in Knowledge Discovery and Data Mining. Springer (2003) 569

21. Spirtes, P., Glymour, C.N., Scheines, R.: Causation, prediction, and search. MIT Press (2001)
22. m. Basseville, Kikiforov, I.: Detection of Abrupt Changes. Printice Hall, Englewood Cliffs, New Jersy (1993)
23. Kadambe, S., Boudreaux-Bartels, G.: Application of the wavelet transform for pitch detection of speech signals. Information Theory, IEEE Transactions on **38**(2) (Mar 1992) 917–924
24. Hirano, S., Tsumoto, S.: Mining similar temporal patterns in long time-series data and its application to medicine. In: ICDM '02: Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02), Washington, DC, USA, IEEE Computer Society (2002) 219
25. Gombay, E.: Change detection in autoregressive time series. J. Multivar. Anal. **99**(3) (2008) 451–464
26. Ide, T., Inoue, K.: Knowledge discovery from heterogeneous dynamic systems using change-point correlations. In: Proc. SIAM Intl. Conf. Data Mining. (2005)
27. Zha, H., Simon, H.D.: On updating problems in latent semantic indexing. SIAM Journal on Scientific Computing **21**(2) (1999) 782–791
28. Ide, T., , Tsuda, K.: Change-point detection using krylov subspace learning. In: Proceedings of the SIAM Internations Conference on Data Mining. (2007)
29. Mohammad, Y., Nishida, T.: Robust singular spectrum transform. In: IEA/AIE 2009. (2009) accepted.
30. Chiu, B., Keogh, E., Lonardi, S.: Probabilistic discovery of time series motifs. In: KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, New York, NY, USA, ACM (2003) 493–498
31. Oates, T.: Peruse: An unsupervised algorithm for finding recurring patterns in time series. In: International Conference on Data Mining. (2002) 330–337
32. Jensen, K.L., Styczynxki, M.P., Rigoutsos, I., Stephanopoulos, G.N.: A generic motif discovery algorithm for sequenctial data. BioInformatics **22**(1) (2006) 21–28
33. Minnen, D., Starner, T., Essa, I., Isbell, C.: Improving activity discovery with automatic neighborhood estimation. In: Int. Joint Conf. on Artificial Intelligence. (2007)
34. Catalano, J., Armstrong, T., Oates, T.: Discovering patterns in real-valued time series. In: Knowledge Discovery in Databases: PKDD 2006. (2007) 462–469
35. Freedman, D., Humphreys, P.: Are there algorithms that discover causal structure? Synthese **121**(1-2) (2004) 29–54
36. Mohammad, Y., Nishida, T.: Human adaptation to a miniature robot: Precursors of mutual adaptation. In: The 17th IEEE International Symposium on Robot and Human Interactive Communication, 2008. RO-MAN 2008. (2008) 124–129
37. Ohmura, R., Naya, F., Noma, H., Kogure, K.: B-pack: a bluetooth-based wearable sensing device for nursing activity recognition. Wireless Pervasive Computing, 2006 1st International Symposium on (Jan. 2006) 6 pp.–
38. : www.phasespace.com.