DMD: Discovering Time-Variant Temporal Dependence from Time Series Using Predictive Codelength

Luofeng Zhou
Columbia University
New York, NY
luofeng.zhou@columbia.edu

Zixuan Cao
Wuhan University
Wuhan, China
zixuancao@whu.edu.cn

Bo Han*
Wuhan University
Wuhan, China
bhan@whu.edu.cn

ABSTRACT

Granger causality test, the most well-established data-driven approach for discovering temporal dependence, applies a vector autoregression (VAR) model to infer temporal dependence among series by regression. However, it is designed for stationary and time-invariant systems. We instead propose a novel method, named DMD, for discovering time-variant temporal dependence from long time series without assumptions on stationarity and data generating process. It measures temporal dependence through predictive codelength from the viewpoint of certainty. Through the difference between predictive codelength when information from another series is involved or not, we form a statistics for discovering time-variant temporal dependence. We conduct exclusive experiments to evaluate the empirical performance of the proposed method on various synthetic and real-world data sets. The encouraging results show that DMD is consistent to Granger causality test and other baseline models while temporal dependence is alleged to be time-invariant. In addition to its ***O(n)*** time-complexity, the essence of DMD enables the discovery of time-variant temporal dependence which prevails in real-life applications.

CCS CONCEPTS

• **Information systems → Data stream mining**; • **Applied computing → Law, social and behavioral sciences**.

KEYWORDS

Time-Variant Temporal Dependence, Time Series, Predictive Codelength, Sequential Normalized Maximum Likelihood

ACM Reference Format:
Luofeng Zhou, Zixuan Cao, and Bo Han. Working Paper. DMD: Discovering Time-Variant Temporal Dependence from Time Series Using Predictive Codelength. In ,. ACM, New York, NY, USA, 13 pages. https://doi.org/\

1 INTRODUCTION

It is frequently of high interest to discover temporal dependence, the causal relationships between time series, in many applications, such as financial economics, biomedical engineering, etc. For example, investors tend to measure the temporal dependence (especially the local one) between one of the foreign exchanges to another and thus construct their investigating strategies in a more plausible and efficient way.

The most well-known data-driven temporal dependence discovery model, Granger causality test (abbreviated as ***G-test***), is embedded in a vector autoregression (VAR) model within the framework of Granger causality. Due to the essence of regression, it is designed for stationary and time-invariant systems. However, time series in real life are often mixed up with exogenous shocks, whose dependence structure is highly ***time-variant***[1]. For these common data with dynamic nature of dependence structure, G-test cannot capture each local dependence pattern accurately and will result in spurious inference in a time-invariant viewpoint.

In order to solve these challenges, in this paper, we propose a novel method, named ***DMD***[2], for discovering time-variant temporal dependence without assumptions on data generating process (DGP), such as time-invariant and stationary assumptions. Beginning from the parsimonious setting, given two time series x^t and y^t, DMD postulates that y^t is likely temporal dependent to y^t if the past samples of both x^t and y^t help to significantly and sequentially improve the prediction of the current sample of x^t than we do prediction with just the past samples of x^t. Different from G-test, where predictability is measured in terms of variance of the regression error, DMD interprets predictability in terms of the codelength difference required to encode x^t. Our model sequentially scans x^t within a bounded window, efficiently compute sequential normalized maximum likelihood (sNML) by self-prediction with past records of x, or by cross-prediction with past records from both x^t and y^t, and thus compare the difference between these predictive codelengths according to information theory. Therefore, we infer temporal dependence between the past of y^t and the present of x^t if the predictive codelength has been significantly reduced. In each step of prediction, DMD select the ***most advantagious***[3] information between the two sequences. In addition, by visualizing the codelength difference as a measure of temporal dependence, DMD is capable for mining time-variant temporal dependence embedded in long sequences.

*Corresponding Author.

Preprint, Preprint

ACM ISBN …$
https://doi.org/\

[1] Time-variant temporal dependence here refers to the inconsistency between local and global temporal dependence. A more formal definition of time-variant temporal dependence and its solution given by DMD is discussed in the following part.
[2] DMD comes from the abbreviation of "Discovering Time-Variant Temporal Dependence", DTVTD whose interim three digits "TVT" is replaced by "M" in short.
[3] Intuitively, the selection conforms to minimize the prediction function and codelength and the more element exchanges occur, the more information one sequence uses from another. However, it seems that we should go through all combinations of past information but in fact enumerative algorithm is computational hard. Thus, the explicit form is derived and formal theorems are given in Section 3.

DMD, unlike G-test which assumes that all digits from a time series are generated through a single DGP, assumes each digit from a time series are generated independently but the parameters of their distribution functions are correlated temporally. The estimator of the distribution parameter of the next bit of x^t, $\hat{\theta}_t$, is given by sNML prediction strategy, which is a modified version of maximum likelihood strategy. Plus, the statistics of DMD is additive which has the potential to show the time-variant structure of temporal dependence.

The key advantages of DMD are its highly competitive performance in discovering time-variant temporal dependence without any assumptions on DGP, and fairly attractive computational efficiency. We conduct extensive experiments to evaluate the empirical performance of the proposed method on various synthetic and real-world data sets. The encouraging results show that DMD is consistent to Granger causality test and other baseline models while temporal dependence is alleged to be static. In addition to its ***O(n)*** time-complexity, the essence of DMD enables the discovery of time-variant temporal dependence which prevails in real-life applications.

Our contributions are mainly threefold:

- To our best knowledge, DMD is the first data-driven model that exploits predictive codelength to mine time-variant temporal dependence.
- DMD does not assume stationarity, which complements G-test on time-invariant temporal dependence discovery. In addition, DMD does not assume time series to be time-invariant, making it possible for analyzing time-variant dependence structure in theory.
- The empirical results show highly consistent to G-test and other benchmark methods when dependence structure is time-invariant and when dependence structure is time-variant, the results are promising. Plus, the time complexity of DMD is linear with respect to the length of each time series, suggesting that DMD is suitable for temporal dependence mining from time-variant systems in real-life applications.

The rest of this paper is organized as follows. The second section lists two motivation examples referring to the two challenges of the G-test. The third section describes our DMD model in details. The fourth section shows exclusive experimental results and analysis. The fifth section lists extensions and discussions for model settings and time-variant temporal dependence and its solution. The sixth section summarizes related works. The final section gives the concluding remarks.

2 MOTIVATION EXAMPLES

In this section, we use two representative examples to illustrate the situations where G-test is not applicable and thus motivate our work.

In Figure 1 (a), G-test infers a bidirectional temporal dependence with p-values are both 0.00. This type of cyclic series is hard to convert to stationary data by simply taking difference. Consequently, G-test is not applicable nor reliable to infer temporal dependence on these non-stationary data.

In Figure 1 (b), the dependence structure between two series are changing overtime. In real-world applications, the role of cause and effect could still be changing. In this case, G-test infers no temporal dependence between two series while it infers temporal dependence from series1 to series2 in intervals denoted by red line.

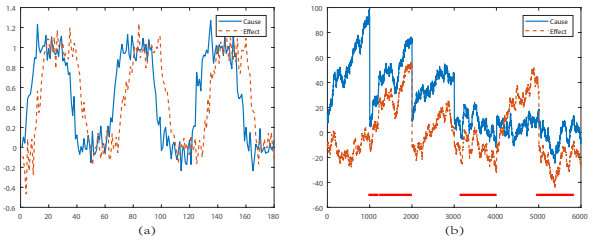

Figure 1: Motivation examples where G-test is inapplicable: (a) Non-stationarity; (b) Time-variant dependence structure.

From the above visualization analysis, we can see that it is quite difficult or even impossible for G-test to discover these two kinds of temporal dependence provided that it utilizes regression which assumes that each digit from a time series is generated from same DGP and series should be stationary. Thereby, is it possible to improve G-test performance by using another pattern discovery model to replace VAR and corresponding F-test? Inspired by this idea, we propose a novel temporal dependence discovery model based on information theory. Specifically, it leverages predictive codelength to construct a statistics to measure temporal dependence in the view of certainty[4]. By assuming each digit is generated independently while their DGPs are correlated temporally, an additive statistics is built which is caplable for analyzing time-variant temporal dependence.

3 DMD FRAMEWORK

3.1 Overview

Granger causality test explains temporal dependence as "significantly decrease in regression-based standard error". Though simple and natural, the usage of regression has some limitations as discussed above. In this paper, DMD explains temporal dependence as "significantly increase in certainty" through the perspective of predictive coding. Inspired by the idea, by sequentially scanning time series in a bounded window, we utilize a simple and widely robust prediction strategy, sequential normalized maximum likelihood (sNML) strategy, to compute the possibility of encoding next point of x^t by using past values of x^t itself (self-prediction) or by using past values from both x^t and y^t (cross-prediction). By information theory, we thus compute the encoding length difference (ELD) between self-prediction and cross-prediction as the measurement of temporal dependence at each temporal point. By tracing the difference between two ELD curves (x^t w.r.t. y^t and y^t w.r.t. x^t), DMD is capable to infer time-invariant as well as time-variant temporal dependence.

3.2 Preliminaries

3.2.1 General Notations. Without losing any generality, the two time series are denoted as x^t and y^t, where t is the total number

[4]According to information theory and compression theory, with shorter codelength, the time series is considered of more certainty in nature.

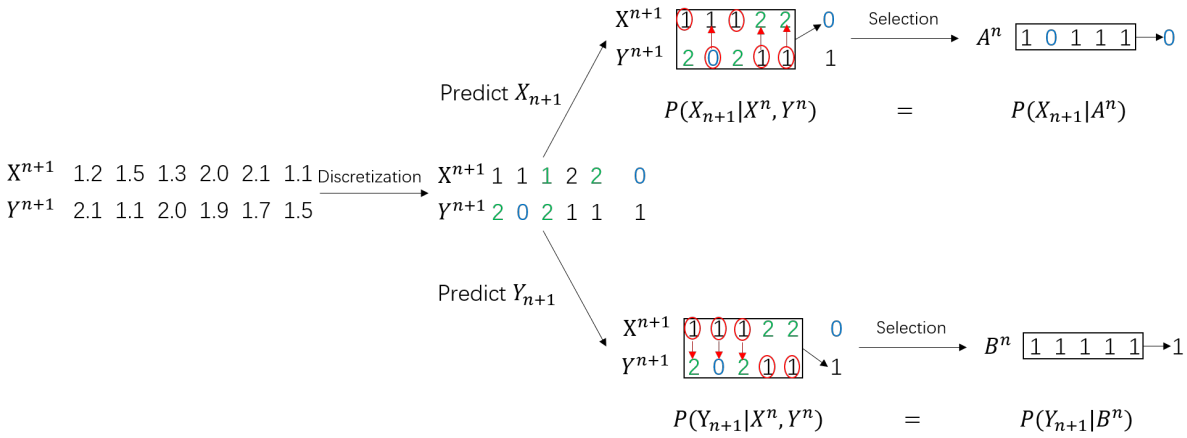

Figure 2: A visualization example of how to utilize information from another source to predict the next bit.

points in a series. The realization of x^t at time point n is denoted as x_n. A statistical measure is denoted as z_t, as one of the intermediate realization between x_t and y_t.

3.2.2 Temporal Dependence and Granger Causality. We follow the idea of [11, 25] to define temporal dependence within the framework of G-causality:

Definition 1. *(**G-causality**) A time series x^t **Granger-causes** another series y^t if the prediction function holds $P(y_{t+1}, \theta|x^t, y^t) > P(y_{t+1}, \theta|y^t)$. That is to say, y^t is **temporal dependent** to x^t.*

This definition concisely reveals the basic ideas of Granger causality: (1) cause precedes the effect in time sequence; (2) cause contains unique information for predicting the future values of effect.

It is worth of remarkable that G-causality is quite different from G-test. G-test implements G-causality by using an regression model over whole sequence and interprets predictability enhancement as the reduction of variance of regression Our DMD model follows G-causality rules, but implements G-causality by using predictive coding and explains predictability increase as significantly reducing predictive codelength.

3.2.3 Predictive Codelength. We measure predicatablity in temporal dependence by certainty. More precisely, we define temporal dependence in terms of the predictive codelength required to encode the time series within a bounded window. First, we define two types of predictive codelengths as below:

Definition 2. *(**Ideal Codelength by Self-Prediction**) According the information theory, the ideal codelength by self-prediction (unconditional probability) for current point x_t is given by: $len(x_t) = -logP_{\hat{\theta}}(x_t|x^{t-1})$[5] where $P_{\hat{\theta}}(x_t|x^{t-1})$ denotes a prediction of x_t using the past values of x^{t-1} and the estimated (by a certain prediction strategy) parameter $\hat{\theta}$. Thus the t-period total self-prediction codelength of x^t is given by: $\boldsymbol{L}(x^t) = \sum_{t=1}^{T} -logP_{\hat{\theta}}(x_t|x^{t-1})$.*

Definition 3. *(**Ideal Codelength by Cross-Prediction**) The ideal codelength by cross-prediction (conditional probability) of x_t on the information of y^t is given by: $len(x_t|y^{t-1}) = -logP_{\hat{\theta}}(x_t|x^{t-1}, y^{t-1})$, where $P_{\hat{\theta}}(x_t|x^{t-1}, y^{t-1})$ denotes a prediction of x_t under the information of x^{t-1}, y^{t-1} and the estimated (by a certain prediction strategy) parameter $\hat{\theta}$. The t-period total conditional-prediction codelength of x^t conditional on y^t is given by: $\boldsymbol{L}(x^t|y^{t-1}) = \sum_{t=1}^{T} -logP_{\hat{\theta}}(x_t\ |x^{t-1}, y^{t-1})$.*

The ideal codelength by self-prediction, $len(y_t)$, shows the certainty using the past realizations of itself and the ideal codelength by cross-prediction, $len(y_t|x^{t-1})$, involves the past realizations of both series. Hence, their difference is a natural measurement of the extra predictability of y_t uniquely contributed by the realizations of x^{t-1}. Specifically, the prediction strategy in this paper is referred to as sequential normalized maximum likelihood (sNML) prediction strategy.

3.2.4 Robust Prediction by Normalized Maximum Likelihood. When we consider the form of prediction function, we should first build our assumptions on the dependence structures.

Assumption 1. *(**Independent Generation**) For any two digits in a time series x^t, say x_i and x_j, they are generated independently from two different data-generating process.*

Note that this assumption is quite different from the assumption on which regression is based. In regression models, the generation of each digit is dependent on some past realizations.

Under this assumption, information of the past realizations is only about the parameters of distribution at present. The most natural prediction strategy is through maximum likelihood (ML) strategy. Specifically, for modeling the prediction function P, ML strategy for parameters is computed by solving a normal maximization problem with an assumed data distribution. However, if the assumption of data distribution is not consistent with the real case, ML strategy will result in arbitrarily bad outcomes [14]. Thereby, a robust prediction function should be constructed by minimizing the difference between the true data distribution and the assumed ones, or say, deriving the minimax of ***regret***, which measures the performance of a specific prediction strategy, say P, w.r.t a model class $\mathcal{P}$. Here comes a formal definition of the regret.

[5] All logs in this paper refer to log_2.

Definition 4. *(**Regret**) A collection of prediction strategies on time series x^t is defined as $\mathcal{P} = \{P_\theta : \theta \in \Theta\}$ where Θ indicates a k-dimensional parameter space, the performance of a specific prediction strategy P on x^n w.r.t a model class $\mathcal{P}$ is measured by:*

$$\begin{aligned}\mathcal{R}(P; x^n) &= \sum_{t=1}^{n} -logP(x_t|x^{t-1}) - \min_{P_\theta \in \mathcal{P}} \sum_{t=1}^{n} -logP_\theta(x_t|x^{t-1}) \\ &= -logP(x^n) - \min_{P_\theta \in \mathcal{P}} (-logP_\theta(x^n)).\end{aligned}$$

It can be seen that the regret is the additional bits required to encode the time series x^t using the mis-specified prediction strategy, P, instead of the best prediction strategy [3]. Due to the time dependence of the regret, our aim becomes to minimize the worst-case regret. Although the loss of ML strategy is the negative logarithm of the forecast density of the outcome, an amazingly simple modification of ML strategy, sequential normalized maximum likelihood (sNML) strategy, can achieve the minimax of regret just like a rather complex normalized maximum likelihood (NML) strategy [12, 14, 19].

In sNML strategy, the prediction function of x_t under x^{t-1} is:

$$P_{sNML}(x_t \,|x^{t-1}) = \frac{\mathcal{L}(\hat{\theta}_{x^{t-1},x_t}; x^{t-1}, x_t)}{\sum_x \mathcal{L}(\hat{\theta}_{x^{t-1},x}; x^{t-1}, x)}, \tag{1}$$

where $\hat{\theta}_{x^{t-1},x_t}$ denotes the MLE based on x^{t-1}, x_t and $\hat{\theta}_{x^{t-1},x}$ denotes the MLE based on x^{t-1}, x.

Since for some special distributions lying within the exponential family such as Bernoulli, binomial and Gaussian, sNML strategy has closed-form expression, in this paper, sNML strategies of discrete series are computed by binomial distributions without loss of generality.

3.2.5 Finite Memory Assumption and Window Approach. Since the data-generating processes are different at different timepoints, it makes no sense to assume that they are identical, which will lead to the assertion that the series x^t obeys an i.i.d. distribution without time series characteristics. But if, to another extreme, the DGPs change randomly, it is by no means that past realizations convey information of the next-bit's DGP. Here we come up with an assumption lying between.

Assumption 2. *(**Temporally Dependent in Generating Process**) For any two nearby digits in a time series x^t, say x_i and x_{i+k} if and only if k is relatively small, the parameters of their data-generating process is relavant. To be more specific, if series x^t is treated in continuous form $x(t)$, $\theta_t - \theta_{t'} \to 0$ when $t' \to t$.*

To be more percise, the conditional prediction function is assumed to have finite memory.

Assumption 3. *(**Finite Memory**) Prediction function (conditional probability) of x_t has finite memory, that is, the prediction is irrelevant to realizations too far away from the most recent one, i.e. $P(x_t|x^{t-1}) = P(x_t|x_{t-1}, ..., x_{t-T})$, $T \in (0, t)$.*

Under this assumption, prediction concerns only the past realization within a small window. Thus, we call this method ***window approach***. The window length, denoted as $\boldsymbol{l}$, is usually supposed to be set through application domain theories or other prior knowledge. Note that when the window length is set to be the length of the whole sequence or infinity, it becomes the non-discount situation. In the rest of the paper, robustness of the window size in simple temporal dependence will be discussed.

3.2.6 Temporal Dependence based on Predictive Codelength. After computing codelength by self- and cross-prediction, if x^t contains more unique and useful information for y^t than vice versa, a temporal dependence is inferred from $x^t \to y^t$. So we define the temporal dependence as below.

Definition 5. *(**Temporal Dependence based on Predictive Codelength**) The **temporal dependences** from y_t to x_t and vice versa are given by:*

$$\Delta_{y^t \to x^t} = \boldsymbol{L}(x^t) - \boldsymbol{L}(x^t|y^{t-1}),$$

$$\Delta_{x^t \to y^t} = \boldsymbol{L}(y^t) - \boldsymbol{L}(y^t|x^{t-1}),$$

Following the framework of Granger causality, we form our belief of temporal dependence direction as following:

- If $\Delta_{y^t \to x^t} > \Delta_{x^t \to y^t}$, we infer $x^t \to y^t$.
- If $\Delta_{y^t \to x^t} < \Delta_{x^t \to y^t}$, we infer $y^t \to x^t$.
- If $\Delta_{y^t \to x^t} = \Delta_{x^t \to y^t}$, we are undecided.

3.2.7 Hypothesis Test. After the corresponding statistics is built, the main problem is how to test the null hypothesis. Its essence is to determine the probability of wrongly refusing of null hypothesis which is said to be no temporal dependence. [21] gives the answer for this problem.

In the context of our research, the null hypothesis is that there is no temporal dependence. When given the significant level α, the decision rule is specified as:

- If $\Delta_{y^t \to x^t} - \Delta_{x^t \to y^t} > -log\alpha$, we infer $x^t \to y^t$.
- If $\Delta_{x^t \to y^t} - \Delta_{y^t \to x^t} > -log\alpha$, we infer $y^t \to x^t$.
- If $|\Delta_{y^t \to x^t} - \Delta_{x^t \to y^t}| < -log\alpha$, we are undecided.

3.3 DMD Model

Following the principal lines discussed above, we state the whole algorithm of DMD model in **Algorithm 1**.

In **Algorithm 1**, we first discretize time series into ternary sequences with values $\{0,1,2\}$[6]. Next, we construct two FILO queues z_0^k, z_2^k to store the minimum and maximum values between two series up to a temporal point k within a maximum length. Their corresponding sum in the queue is recorded as $\Sigma_0 z^k$ and $\Sigma_2 z^k$ respectively. The imbalance between 2 and 0 for z_0^k is denoted as I_0, suggesting the difference between the number of 2 and that of 0 within a given window. And I_2, the imbalance between 2 and 0 for z_2^k, is calculated through the same procedure. The boolean indicator B shows whether there exists 0-1 pair or 1-2 pair for the same time in the past. Note that the temporal dependence statistics in DMD is additive, which means that the calculation is of linear complexity.

Since the sNML strategy minimizes the ***regret*** for mis-specification of data-generating process, we only prove the situation of binomial distribution with three states for simplicity. The reason why we

[6]The reason why we do not use continuous time series directly is twofold. First, the computational cost is much lower when using ternary data especially for a longer time window. Second, continuous data contain more noise. Tenary encoding omits some noisy oscillations and, in the meantime, allows state to be non-informative. In Section 4, the consistency before and after encoding is tested on synthetic data.

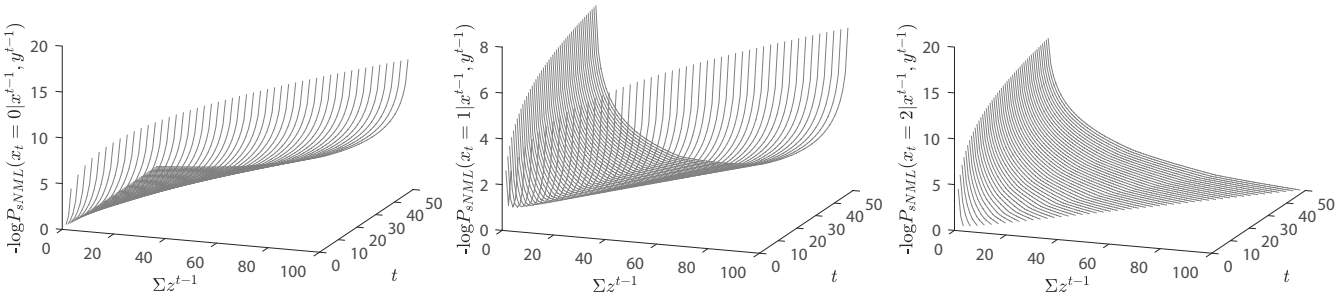

Figure 3: The cross-prediction sNML strategy for x^t conditional on y^t when the current x_t equals to 0 (a), 1 (b) and 2 (c). The x-axis is the sum of z_i where $z_i \in \{x_i, y_i\}$ and the y-axis is the past length.

Algorithm 1 DMD model

Input:
Two time series, x^t, y^t;
The window length, $\boldsymbol{l}$;
Output:
The inferred temporal dependence between x^t and y^t
Discretize time series into ternary data.
Initialize intermediate variables: $\Sigma_0 z^0 = \Sigma_2 z^0 = I_0 = I_2 = B = 0$;
Initialize intermediate series: $z_0^0 = z_2^0 = \{\}$;
Set the loop indicator k=0;
while $k < t$ **do**
  $k+ = 1$;
  Calculate self-prediction sNML strategies w.r.t. $\boldsymbol{l}$;
  Calculate cross-prediction sNML strategies w.r.t. $\boldsymbol{l}$;
  $z_0^k = z_0^{k-1}, min\{x_k, y_k\}; z_2^k = z_2^{k-1}, max\{x_k, y_k\}$;
  Update intermediate variables by variable definitions;
  Calculate temporal dependence statistics in this period;
end while
return The inferred temporal dependence drawn by the last-period statistics and the decision rule;

choose three states rather than more states partly lies on the little difference between mis-specified binomial distribution and the true data-generating process. Even though sNML minimizes this kind of difference and gives the best estimator within the support set, this gap is supposed to be greater when states grows larger. In general, the choice of the number of states is a trade-off between bad results brought by mis-specification of DGP and the bad fitting of more continuous-like time series.

DMD model allows any combination of subsequences within a window for prediction. It seems to result in expensive computation cost. To put it into practice, we then put forward an efficient algorithm to maximize the cross-prediction sNML strategy.

3.3.1 sNML Implementation. Because sNML strategy gives a robust prediction even if the DGP is mis-specified and its calculation is kind of complex, we consider the simplest binomial distributions for implementation $P_\theta(X = k) = C_n^k \theta^k (1-\theta)^{n-k}, k = 0, 1, ..., n$. We deduce **Theorem 1** and **Theorem 2** to compute self- and cross-prediction sNML strategies.

Theorem 1. *For binomial distribution $P_\theta(X = k) = C_n^k \theta^k (1-\theta)^{n-k}, k = 0, 1, ..., n$, the self-prediction sNML strategy is the function of $\sum_{j=1}^{t-1} x_j := \Sigma_{t-1}$ but irrelevant to the specific constituent or permutation. To be more specific, the self-prediction sNML strategy for binomial distribution with $P_\theta(X = k) = C_n^x \theta^x (1-\theta)^{n-k}, k = 0, 1, ..., n$ is specified as:*

$$P_{sNML}(x_t|x^{t-1}) = \frac{f_t^{x_t}}{\sum_{i=0}^{n} f_t^i}, \tag{2}$$

where for any i = 0, 1,.. .., n:

$$f_t^i = C_n^i (i + \Sigma_{t-L}^{t-1})^{(i+\Sigma_{t-L}^{t-1})} \cdot (nL + n - i - \Sigma_{t-L}^{t-1})^{(nL+n-i-\Sigma_{t-L}^{t-1})},$$
$$\Sigma_{t-L}^{t-1} = x_{t-L} + x_{t-L+1} + ... + x_{t-1}.$$

When implementing cross-prediction sNML strategy, we just prove the ternary situation although the core theorem is also applicable and valid when the number of states is greater than three.

Before theoretical proof is given, a numerical visualization helps to understand the incentive behind the corresponding theorem. Figure 3 plots the negative logarithm of prediction function given by cross-prediction sNML strategy for x^t conditional on y^t, which gives intuition showing that the maximum of prediction function is attained when the difference between mean of the past values of the intermediate series, z^{t-1}, and the value of next bit is minimized. A formal theorem which specifies the maximum of cross-prediction sNML strategy is described and detailed proof is in appendix.

Theorem 2. *For ternary data, the maximum of cross-prediction sNML strategy $P_{sNML}(x_t = j|x^{t-1}, y^{t-1}), j = 0, 1, 2$ is attained by the series which satisfies:*

$$\Sigma_{i=t-L}^{t-1} z_i = L \cdot j + \min_{z_i \in \{x_i, y_i\}} \sum_{i=t-L}^{t-1} (z_i - j). \tag{3}$$

Proposition 1. *For ternary data, the cross-prediction sNML strategy for binomial distribution: $P_\theta(X = k) = C_2^k \theta^k (1-\theta)^{n-k}, k = 0, 1, 2$ is characterized as:*

$$P_{sNML}(x_t|x^{t-1}, y^{t-1}) = \frac{f_{x_t,t}^{x_t}}{\sum_i f_{x_t,t}^i}, \tag{4}$$

where for any i = 0, 1, 2:

$$f^i_{j,t} = C^i_2 (i + \Sigma^{t-1}_{i=t-L} z_i)^{(i+\Sigma^{t-1}_{i=t-L} z_i)} \cdot (2L + 2 - i - \Sigma^{t-1}_{i=t-L} z_i)^{(2L+2-i-\Sigma^{t-1}_{i=t-L} z_i)},$$

$$\Sigma^{t-1}_{i=t-L} z_i = L \cdot j + \min_{z_i \in \{x_i, y_i\}} \sum_{i=t-L}^{t-1} (z_i - j).$$

Since in practice the function f^i is always too large to be calculated forthright, a *log-sum-exp* is applied to solve the overflow exception.

Even if binomial is parsimonious for ternary data, when calculating the cross-prediction sNML strategy for ternary, things are difficult when compared with binary situations where only one time of scanning is needed to calculate the optimal cross-prediction sNML strategy through the calculation of XOR. When it comes to multi-state, say ternary, series, the corresponding calculate is totally a different story from binary sequences in which an XOR operator can easily solve this problem. However, if we use enumerative method, the computational cost is high especially when the window length is large. So the maximum likelihood of the intermediate situation should be well identified in an explicit form.

3.3.2 Realizing Cross-Prediction sNML strategy. An simple and efficient algorithm is designed for the implementation of cross-prediction sNML strategy. The core idea is to make full use of outcomes from previous steps to simplify the computation and obtain the ELD-based statistics in one sequential scanning over the time series.

Algorithm 2 sNML Strategy for Cross-Prediction

Input:
The next bit of the series, $x_{t_0} = i$;
The past discounted sum interval, $[\Sigma_0 z^{t_0-1}, \Sigma_2 z^{t_0-1}]$;
Imbalance between 2 and 0 for two series, I_0, I_2;
Boolean indicator, B;
Output:
$P_{sNML}(x_{t_0} = i | x^{t_0-1}, y^{t_0-1})$;

if i = 1 **then**
  if I_0 is even or B =1 **then**
    $\Sigma_1 z^{t_0-1} \leftarrow t_0 - 1$;
  else
    $\Sigma_1 z^{t_0-1} \leftarrow t_0 - 1 + 1$;
  end if
end if
Calculate $P_{sNML}(x_{t_0} = i|\cdot)$ by $\Sigma_i z^{t_0-1}$ through **Theorem 1**;
$\Sigma_0 z^{t_0} \leftarrow \Sigma_0 z^{t_0-1} + min\{x_{t_0}, y_{t_0}\}$;
$\Sigma_2 z^{t_0} \leftarrow \Sigma_2 z^{t_0-1} + max\{x_{t_0}, y_{t_0}\}$;
if $B = 0$ and $|x_{t_0} - y_{t_0}| = 1$ **then**
  $B \leftarrow 1$
end if
return $P_{sNML}(x_{t_0} = i | x^{t_0-1}, y^{t_0-1})$;

First, we focus on the simplest case where the discount function remains a constant. **Algorithm 2** gives specific actions to determine whether the optimal sum is attainable for any given $t_0 \in [0, t]$. If the feasible interval does not include the unconstrained optimal solution, then the optimal solution within the feasible interval is one of the corner points. However, even if when the unconstrained optimal solution falls into the feasible interval, it may not always be feasible too when the difference of lower-bound and the optimal solution is odd but all the feasible incremental chances give only an even increase. The algorithm is telling exactly the same story in a formal and logical way.

When it comes to the case where window approach is applied to DMD model, it can be seen as a certain step in non-discount situation. If the length of window is chosen as $\boldsymbol{l}$, then each step can be regarded as the $\boldsymbol{l}^{th}$ step in the case described above.

3.4 Time Complexity Analysis

To put the DMD model into practice, we should calculate self- and cross-prediction codelength respectively. According to **Theorem 1**, we calculate self-prediction codelength $\boldsymbol{L}(x^t)$ in linear time if we know the sum within the given window or the whole previous series, which is realized through temporal variables and these variables are updated for each step. Thereby, the self-prediction codelength can be completed in ***O(n)*** time. According to **Algorithm 2**, we can compute cross-prediction codelength $\boldsymbol{L}(x^t|y^{t-1})$. If all the inputs are known, we can calculate it in linear time. In the same way, we can use extra variables to store these values and arrays and update them every time the current step ends. Therefore, we can compute cross-prediction codelength in ***O(n)*** time. It is worth mentioning that the complexity of our algorithm is independent of the choice of window size.

Figure 3 illustrates the time complexity of DMD model for the three different selections of window length. The time complexity of DMD is linear and independent to the window length, which can be easily seen in the description of **Algorithm 2**.

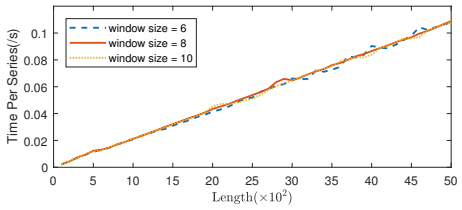

Figure 4: Time complexity of DMD model

3.5 Time-Variant Temporal Dependence

It is often the case when a new policy comes into being, the pattern of the markets changes a lot. If the direction of battery of a circuit changes, temporal dependence is reversed. In real-world situations, exogenous, or say out-of-model, shocks are not always easily to observe. When the dependence structure changes, temporal dependence is alleged to be ***time-variant***. This is why regression-based model fails to explain a couple of this kind of situation in real world since regression models assume the dependence structure (DGP) is time-invariant. In general, regression-based models assume that data from a time series share a single time-invariant structure and what we need to do is to estimate, unbiasedly or consistently, the coefficients of the assumed data-generating process while our model

assumes that each timepoint is independently generated but their dependence structure are correlated, leaving it possible to analyze time-variant temporal dependence.

To begin with our analysis, we try to give a general definition of time-variant temporal dependence.

Definition 6. *(**Time-Variant Temporal Dependence**) The dependence structure between time series x^t and y^t is **time-variant** if for a continuous period of time, y^t is temporal dependent to x^t and for another period of time, y^t is temporal independent to x^t or x^t is temporal dependent to y^t.*

As is stated before, when temporal dependence is time-variant, G-test is at best "half correct". Although it seems plausible that we can perform G-test for every segments, the essence of F-statistics is not additive. Since the exact changing point of temporal dependence is always unknown, enumerative method can be the only choice, which is too costly in investigating time-variant temporal dependence. Our statistics, however, is additive, which has the potential to analyze time-variant temporal dependence. Here comes an example illustrating how DMD deal with time-variant temporal dependence.

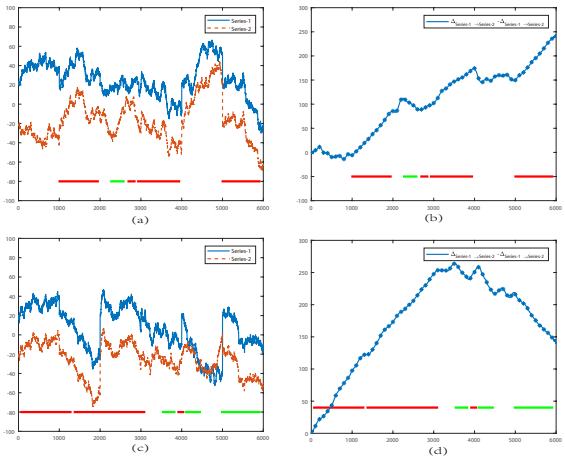

Figure 5: Two examples (curves of the series on the left and statistics of DMD on the right) on discovering time-variant temporal dependence: (a, b) Temporal dependence from series 1 to series 2 mingled with temporal independence; (c, d) Changing directions on temporal dependence.

The red line indicates that DMD believes there exists temporal dependence from series 1 to series 2 and green line indicates another direction, corresponding to the increasing trend in statistics and decreasing trend respectively.

3.5.1 Algorithm for Automated Evaluation. To evaluate the soundness of DMD when mining time-variant temporal dependence, an automated evaluation algorithm should be designed. There are various ideas about how to design this algorithm, such as machine learning method with a carefully designed loss function and convolutional neural network in the field of computer vision. However, in this paper, our main focus is not the ***piecewise linear fitting problem***. We implement a simple and heuristic algorithm to evaluate our results on time-variant temporal dependence.

The heuristic idea is that if x^t is time-dependent on y^t, the statistics should be increasing in time and when y^t is time-dependent on x^t, the statistics should be decreasing. When we consider a noisy series to be increasing, we believe that the ***increasing points*** should be much greater than ***decreasing points***. When an increasing series changes to a decreasing one, we expect that the nearest digits have more decreasing points than ***increasing points*** while the earlier points have more ***increasing points***.

Algorithm 3 describes the pseudocode for the heuristic algorithm above. Note that the process in determining the present trend is set as: if the past realizations in the window w has decreasing points more than $thres$, the present trend is estimated as decreasing; if the past realizations in the window w has greater increasing points than $thres$, the present trend is estimated as increasing; otherwise the present trend is unclear.

Algorithm 3 A Heuristic Algorithm for Automated Evaluation

Input:
The list of $\Delta_{x^t \to y^t} - \Delta_{y^t \to x^t}$ at each step, $values$;
Minimum length of each dependent fragment, min_length;
The Threshold of judging, $thres$;
The size of the window, w
Output:
The list of temporal dependent fragments, td_list;
The list of corresponding directions, $td\_direction$;

```
Define values to be increasing or decreasing, Δvalues;
Declare the initial trend of wth bit according to thres, trend;
Declare the beginning point of the potential trend, begin;
for i ∈ [1, len(values) − w] do
    Update the trend according to thres, trend_new;
    if trend_new ≠ trend then
        if trend ∈ {pos, neg} and i − begin > min_length then
            td_list.append([begin, i]);
            td_direction.append(trend);
        end if
        begin ← i;
        trend ← trend_new;
    end if
end for
Estimate the statistics for every fragments in td_list and delete
    insignificant ones in both td_list and td_direction;
return td_list, td_direction
```

3.5.2 Forward-Backward Modification. Although in essence, our statistics is additive and it should be able to analyze the time-variant dependence structure naturally, statistics around the junction points between two patterns are not easy to recognize, both by machine and human beings, due to the noisy environment. See Figure 6 for example. In Figure 6.(a), the red line indicates the temporal dependence from series 1 to series 2 in the forward series.

In order to increase the reliability when conducting temporal dependence discovery under real world datasets, we follow [5] to introduce forward-backward modification (abbreviated as **FB**

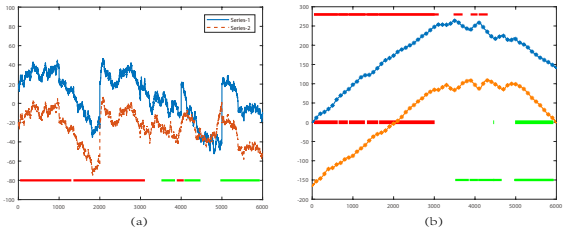

Figure 6: An example of how FB modification is implemented on DMD: (a) Curves of two series; (b) Statistics of forward and backward statistics.

modification) to DMD. The underlying of this FB modification is simple: if the forward temporal dependence, say from x^t to y^t, is alleged to be reliable, the backward temporal dependence from the opposite direction, say from y^t to x^t, should also be tested. In the previous synthetic tests, we do not implement this forward-backward modification since our aim is to test the priority of DMD. Actually, when we are trying to reveal temporal dependence among variables in practice, forward-backward modification makes this data-based inference more reliable and cautious. When forward temporal dependence is consistent to backward one, we are more confident to examine this problem in theory while when they are not consistent, we are more careful about these somewhat fragile findings. In Figure 6.(b), the lines on the top and on the bottom are the outer join of temporal dependent from forward and backward series and the line in the middle is the inner join. The inner join (the middle line) is 100% in precision while the recall is relatively lower than result from forward series only.

4 EXPERIMENTAL RESULTS

Before showing how DMD, in practice, is capable for time-variant temporal dependence discovery, we conduct exlusive experiments on both synthetic and real-world datasets to examine whether DMD is consistent to benchmarks in time-invariant cases and DMD is capable in mining time-variant dependence structure.

4.1 Experimental Settings

4.1.1 General Settings. Our source code and datasets are public and details are included in appendix. We implemented DMD model in Python and provide the source code for research purposes along with the datasets and synthetic dataset generator. All experiments were executed on a laptop computer with an Intel Core i7 1.8GHz CPU and 16 GB main memory.

We compare DMD model with ***G-test*** and ***DMD-2*** (which contains only two states rather than three in ordinary DMD model). For Granger causality test, the lag operator in VAR is selected through prior knowledge for testing accuracy towards synthetic cause-effect and AIC is applied otherwise. For DMD-2 model, it is an extension of CUTE in [3] which originally intends to discover causality on event sequence. Although CUTE receives stable results in binary data, if it is applied without a window discount to binary data transferred from continuous sequences, CUTE cannot achieve good results because the autocorrelation effect has not been taken into consideration. So for the rest of this paper, we use window approach to modify CUTE model to become an DMD-2, a simpler version of our DMD model.

In order to control false discovery rate for our repeated experiments in DMD and DMD-2 model, we follow [3] and apply the most widely used *Benjamini-Hochberg procedure* [2]. Let $H_1, ..., H_m$ be the null hypotheses tested, and their corresponding p-values $p_1, ..., p_m$. First sort these p-values in ascending order. For a significance level of α, we reject the null hypothesis or all H_i, where $i = 1, ..., k$ for the largest k that $p_k \leq k/m \cdot \alpha$.

4.1.2 Window Length. Just like G-test, the effect on the choice of window length is unclear for many empirical datasets. G-test always follows the standard information criteria such as Bayesian Information Criteria to choose a VAR specification, although different information criteriun may contradict with each other.

Intuitively, when the window length is smaller than the true lags of information transmission, DMD is unable to unearth the dependence structure. While the window length is relatively large, the extra bits within the window has little negative effect on the aimming temporal dependence[7]. Through this belief, we should choose a suitable window length a little bit larger than the true lags of information transmission.

4.1.3 Encoding Method. DMD gives a third intermediate state between upward state and downward state. Take DMD-2 model for example, the encoding method is obviously problematic since it cannot distinguish a drastic increase (say 100%) and a non-informative slight increase (say 0.01%). G-test, to another extreme, considers the temporal dependence linearly. Our DMD model falls in between. Although we admit that slight oscillation could be non-informative, the drastic changes tell a rather different story. According to this belief, the encoding process should base on the distribution of the dataset and what kind of impact is concerned informative or non-informative[8].

However, due to the little prior knowledge of the true DGP, we perform ***percentile encoding***[9]. We first choose a threshold, say a%, and encode the first a% to be 0 and the last a% to be 2. The rest encodes 1. Unless otherwise stated, we choose percentile encoding and its parameter a% to be 25%.

In order to ascertain the coding method of our model does not change the temporal dependence of the original time series, we do the following simulation experiments. Unless otherwise stated, we choose the window size from $[6, 9]$ and the significance level as 0.05. For each synthetic experiment, we generate 1000 samples whose true lag falls in $[1, 3]$. In the case of different lengths, we perform G-test before and after the encoding process and the corresponding model (DMD for ternary and DMD-2 for binary) with the synthetic data of the most obvious case (linear, noise 0). In addition, although **equation (1)** guarantees the probability with null hypothesis falsely rejected to be no more than any given significance level,

[7] The detailed discussion of window length will be shown in **Section 5.1**.
[8] As for why we choose three states rather than more, through Theorem 2, the unconstrained optima for intermediate situations are much easier to acheive and are not decisive when compared to extreme states. Thus, three-state discretization has no essencial difference from other multi-state discretization.
[9] Percentile encoding is a widely used encoding method from continuous data to ternary series. It symmetrically encodes the first a% to be 0 and last a% to be 2. Another encoding method, binomial encoding, is to fit binomial distribution. The robustness of encoding method is discussed in **Section 5.1**.

Table 1: Accuracies on synthetic time-variant cause-effect pairs (%)

Type		Simple Linear				Nonlinear Monotonic: tanh				Nonlinear Monotonic: exp				Nonlinear Non-Monotonic: $x + 0.5x^3$				Nonlinear Non-Monotonic: $x + x^3$			
Model	Noise	t=200	t=300	t=400	t=500	t=200	t=300	t=400	t=500	t=200	t=300	t=400	t=500	t=200	t=300	t=400	t=500	t=200	t=300	t=400	t=500
G-test	0	94.2	94.4	94.4	94.7	94	94.7	95.3	94.9	95.4	95.3	94.9	95.6	95.6	95.4	95.2	95.1	95.5	95.7	94.6	96.1
DMD-2	0	99.9	100	100	100	99.9	100	100	100	99.5	100	100	100	100	100	100	100	99.9	100	100	100
DMD	0	100	100	100	100	99.9	100	100	100	99.8	100	100	100	99.9	100	100	100	100	100	100	100
G-test	0.1	94.9	94.7	96.5	94.6	95.5	94.5	95.9	95.3	94	96.3	95.2	94.8	95.1	95.5	95.6	95.1	95.8	94.9	95.5	95.5
DMD-2	0.1	99.7	99.9	100	99.2	100	100	100	100	98.5	100	100	100	99.2	100	100	100	99.4	99.9	100	100
DMD	0.1	98.2	99.5	99.7	100	97.7	99.9	100	100	98.5	99.9	100	100	99.4	100	100	100	99.7	99.9	100	100
G-test	0.2	96.1	94.8	94.7	94.1	94.9	95.7	95.3	94.2	95.3	95.1	94.9	95.1	95.7	95.2	94.8	94.7	95.8	96.2	95.7	95.2
DMD-2	0.2	98.2	99.5	99.7	100	94.2	99.7	99.8	99.9	95	99.2	99.7	99.8	96.5	99.5	99.8	99.9	96.3	98.6	99.9	99.9
DMD	0.2	98	99.6	100	100	93.8	99.7	99.8	99.9	92.9	99.8	100	100	97.9	99.9	100	100	97.6	99.8	100	100
G-test	0.3	95.3	95.5	95.3	96	95.2	95	95.9	92.8	95.5	95.8	93.5	94.2	96.1	94.7	94.9	96.0	96.6	94.9	95.1	94.6
DMD-2	0.3	94.1	95.9	98.8	99.5	87.4	95.4	98.2	99.3	88.1	97.6	98.5	99.1	90.7	96.3	98.8	99.6	91.5	96.0	99.1	98.9
DMD	0.3	94.7	99.4	99.9	100	86.6	99.1	99.7	99.7	86.6	98.8	99.9	99.8	94.8	98.7	99.7	100	94.1	98.6	99.8	100

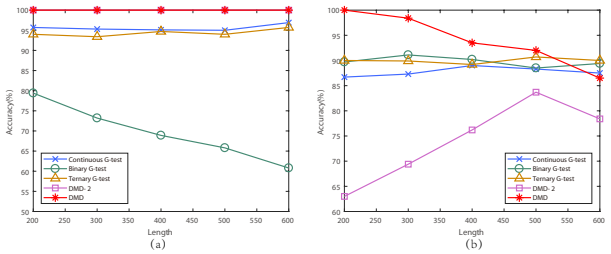

Figure 7: The consistency before and after encoding. (a) cause-effect pairs. (b) non-cause series.

the difference from continuous distributions and discrete distributions could also bias the effectiveness. So the robustness towards no temporal dependence series is also tested. Results are shown in Figure 5.

The results show that the G-test is consistent before and after ternary coding, which shows that the coding method does not change the intrinsic temporal dependence. Plus, the consistency between G-test and DMD model is also salient. However, binary coding method for DMD-2 model is not that promising for series without dependence. This is because that when the series is longer, thougth the expected value of statistics remains unchanged, its variance grows through time, thus leading a lower accuracy.

4.2 Results on Time-Invariant Cases

4.2.1 Synthetic Data. In general, we generate synthetic simple linear cause-effect pairs through a single standard Gaussian distribution. The cause series, x^t, is just the copy of the random generated standard Gaussian distribution with a certain length and the effect series, y^t, is the generated by $y_t = x_{t+L}$ where L is an exogenous scalar indicating the lag operator. The first L digits of the effect series are randomly chosen through standard Gaussian distribution.

When it comes to nonlinear monotonic cause-effect pairs, we test two kinds of nonlinear and monotonic function $f(x) = tanh(x)$ and $g(x) = e^x$ which is widely used in many subjects. In each case, the nonlinear monotonic function is applied to the effect series only and the temporal dependence is tested under three models.

As for nonlinear non-monotonic cases, we follow [5] and choose cubic function for analysis. Specifically, we examine three models under different degree of nonlinearity: $y = x + x^3$ and $y = x + 0.5x^3$.

We test 3 models with length 200, 300, 400 and 500. At each length, the noise-free case is tested and noises of 0.1, 0.2 and 0.3 are applied to the time series respectively.

The results are shown in Table 1. It is clear that our model and DMD-2 perform similarly and better than G-test in longer time series. But G-test performs better with shorter and noisier time series. The results are not surprising due to the different construction of temporal dependence statistics. While in G-test, high noise leads to higher the standard error but little change in coefficient, in DMD and DMD-2, high noise is alleged to cause the significance level harder to reach.

4.2.2 Real-World Data. We apply three data sources of cause-effect pairs to examine the performance of DMD model[10].

River data is first applied in [3]. This dataset collects water level data for several monitoring points in the two rivers of Saar and Rhein, and the ground truth is *upstream* → *downstream*.

Meteorology and Environment data are retrieved from Tübingen cause-effect benchmark pairs[11], in which datasets are tagged with the corresponding ground truth. We use 4 cause-effect pairs to test the DMD model. Three pairs are about meteorology and the ground truth is *temperature* → *ozone*. Another pair is about environment and the ground truth is *outdoor* → *indoor*.

Table 2 gives an overview of all the empirical results on real-world data for each model.

Table 2: Empirical results on real-world datasets

Application Field	Samples	G-test	CUTE	DMD-2	DMD
River	9	4/9√	8/9√	8/9√	8/9√
Meteorology	3	2/3√	1/3√	1/3√	√
Environment	1	×	√	√	√

River Data. It can be seen that the G-test incorrectly identifies the direction of temporal dependence, and our model is completely consistent with the results of the CUTE model. This proves the consistency of our model with the CUTE model.

Meteorology Data. DMD model correctly tells the temporal dependence from temperature to ozone. G-test makes two correct

[10] In reality, theoretical causality from time series data is hard to determine due to, for example, the existence of colliders or endogeneity. On the other hand, data-driven temporal dependences are not suitable for reference as *ground truth*.
[11] http://webdav.tuebingen.mpg.de/cause-effect/

judgments while CUTE and DMD-2 tell the correct direction for only one cases.

Environment Data. Only G-test indicates wrong temporal dependence between indoor and outdoor temperature.

4.3 Results on Time-Variant Cases

4.3.1 Piecewise Linear Fitting. Since this question is not among our main concerns in this paper, the detailed experimental evaluations are omitted. Our algorithm, trying to solve the piecewise linear fitting problem, is just one of the heuristic solutions. In fact, there are many other heuristic ideas which have same and even better precision and recall than ours. Our algorithm results in relatively lower recall (not tabulated) when signal-to-noise ratio is lower since we have no punishment on the numbver of linear piece.

4.3.2 Synthetic Data. We use the following synthetic experiments to demonstrate the effectiveness of DMD when dealing with time-variant dependence structure.

We generate two time series with time-variant dependence. For each dependence structure, the generated length is 1000. We concatenate 6 pieces of dependent time series pair and form a total 6000 length for the two input time series. The dependence structure is randomly choosen from independence, x^t temporally dependent on y^t and y^t temporally dependent on x^t.

We record the statistics from each step and judge the time-variant temporal dependence by **Algorithm 3**.

Figure 8 shows detailed experimental results. Through FB modification, our accuracy maintains in a relatively high level, and the median of recalls are greater than 80% while the variance is relatively lagre. This might be due to the instability of the heuristic algorithm in piecewise linear fitting since when the environment is noiser, the more linear pieces the algorithm will return, and of course many of them are not satisfied to the minimum length and significance level. After all, the FB modification gains higher precision at the cost of lower recall, and the overall F1 scores are higher (not tabulated).

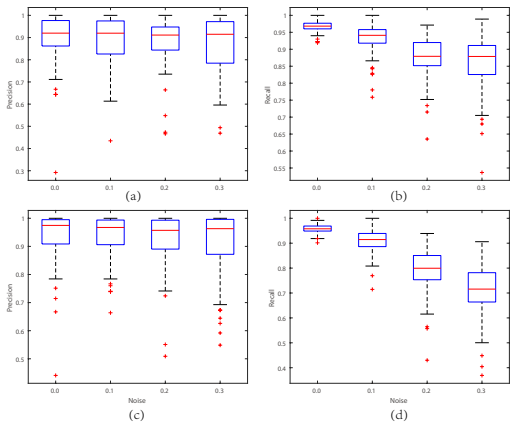

Figure 8: DMD Performance on Synthetic Time-Variant Dependence Dataset: (a) Precision before FB modification; (b) Recall before FB modification; (c) Precision after FB modification; (d) Recall after FB modification.

4.3.3 Real-World Data. Since the ground truth of the temporal dependence in a time-variant system is almost impossible to determine, we try to use the random combination of the datasets used in **Section 4.2.2**. The data source and direction in combination are chosen randomly.

[GRAPH]

5 DISCUSSIONS ON HYPERPARAMETERS

In Section 3, we put forward our DMD framework and in Section 4, both experimental and real-world datasets are tested. However, there are some remainings when applying DMD into practice. In this section, we first discuss the robustness of encoding threshold and window length. Then we propose another main concern, time-variant temporal dependence, that G-test is difficult to handle while DMD is capable in a natural way.

5.1 Robustness of Window Length

It is quite possible for an empirical dataset, different selection of window length draws different or even contradictory inference. In this section we show that when temporal dependence is static, the sensitivity of window length is scarce.

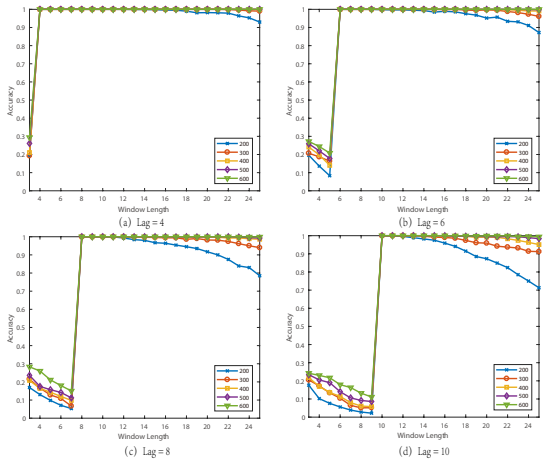

Figure 9: Robustness of window length

5.2 Robustness of Encoding Method

The feasible interval of the encoding threshold, a%, is (0%, 50%). If a% is close to 0%, all bits of series becomes 1 which means the whole series is always non-informative, not to mention temporal dependence. If a% is close to 50%, DMD model reduces to DMD-2 model which has no intermediate state.

6 RELATED WORKS

Granger Causality Test. [25] and [11] introduced the basic notion of statistical-based causality, also known as Granger causality or temporal dependence, to study the cause-effect relationships between time series. [23] began to use the lag of one series to predict another time series and [6] extended its idea and defined the Granger causality to vector autoregressive (VAR) regression models.

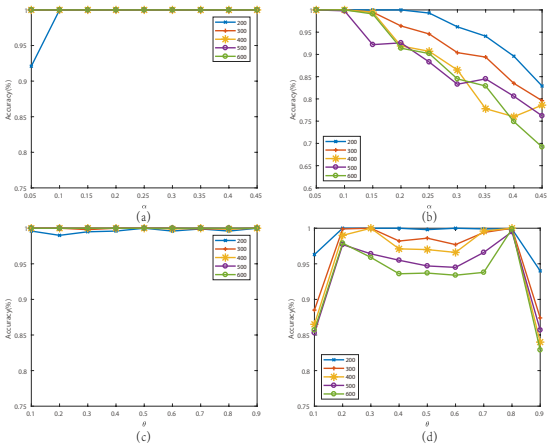

Figure 10: Robustness of encoding methods

Although some recent theoretical works focus on nonlinear and nonparametric test for Granger causality [15–17, 24], there are also a lot of causal discovery research using linear specification in VAR forms. For example, the temporally subsampled [8] and aggregated [9] time series and mixed frequency time series [7, 10] are investigated in VAR regression models; [5] introduces a forward-backward modification which can apply not only on G-test but also on other temporal dependence discovery model; in order to resolve inconsistency or high computational complexity, penalty regressions (such as lasso-regression) have been studied [1]. Our study falls in the Granger causality framework, but applies predictive codelength to replace VAR regression model.

Codelength and Prediction. Codelength is widely utilized in compression theory. Data compression which encodes information into fewer bits is closely related to data prediction. [4] draw techniques from data compression to form a theoretical basis for a prediction task. [20] addresses the problem of online prediction for time series by compression-based methods. Recently, [22] summarized that how data compressors can be used for solving prediction problems in time series. These researches reveal compression techniques can work for prediction. In fact, many compressors consist of predictors. [13] show that a gaze predictor can help to improve the sequence compression effects. [18] improve compression rates by using adaptive block differential prediction. Thereby, there exists intimacy between compression and prediction. The compression methods can apply multiple coding or prediction method to increase the compression ratios and improve predictability. They can dynamically capture temporal patterns along sequence and provide more flexible and accurate prediction than a VAR regression model with a fixed lag window.

Temporal Dependence Discovery based on Codelength. [3] put forward a model CUTE for inferring temporal dependence through ideal encoded (compressed) length to analyze temporal dependence on event sequence. Although his idea is novel, CUTE model is only designed for binary relationships. To our best knowledge, our proposed DMD model is among the very few works that embed codelength into Granger causality framework on time-variant temporal dependence discovery from continuous time series, which is closer to real-world temporal dependence discovery nature.

7 CONCLUSION

In this paper, we propose a novel model, named DMD, which is a data-driven approach to discovery time-variant temporal dependence discovery for time series datasets. When temporal dependence is time-invariant, DMD is consistent with Granger causality test and other benchmarks; when temporal dependence is time-variant, DMD is also capable to mine the dependence structure as a function of time. Plus, the time complexity is linear of time, making it possible for handling longer time series. With a simple forward-backward modification, it can be applied in many real-world issues where the structure of temporal dependence is time-variant.

REFERENCES

[1] Andrew Arnold, Yan Liu, and Naoki Abe. 2007. Temporal causal modeling with graphical granger methods. In *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 66–75.
[2] Yoav Benjamini and Yosef Hochberg. 1995. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the royal statistical society. Series B (Methodological)* (1995), 289–300.
[3] Kailash Budhathoki and Jilles Vreeken. 2018. Causal Inference on Event Sequences. In *Proceedings of the 2018 SIAM International Conference on Data Mining*. SIAM, 55–63.
[4] I-Cheng K Chen, John T Coffey, and Trevor N Mudge. 1996. Analysis of branch prediction via data compression. *ACM SIGPLAN Notices* 31, 9 (1996), 128–137.
[5] Dehua Cheng, Mohammad Taha Bahadori, and Yan Liu. 2014. FBLG: a simple and effective approach for temporal dependence discovery from time series data. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 382–391.
[6] Jean-Marie Dufour and Eric Renault. 1998. Short run and long run causality in time series: theory. *Econometrica* (1998), 1099–1125.
[7] Eric Ghysels, Jonathan B Hill, and Kaiji Motegi. 2016. Testing for Granger causality with mixed frequency data. *Journal of Econometrics* 192, 1 (2016), 207–230.
[8] Mingming Gong, Kun Zhang, Bernhard Schoelkopf, Dacheng Tao, and Philipp Geiger. 2015. Discovering temporal causal relations from subsampled data. In *International Conference on Machine Learning*. 1898–1906.
[9] Mingming Gong, Kun Zhang, Bernhard Schölkopf, Clark Glymour, and Dacheng Tao. 2017. Causal discovery from temporally aggregated time series. In *Uncertainty in artificial intelligence: proceedings of the... conference. Conference on Uncertainty in Artificial Intelligence*, Vol. 2017. NIH Public Access.
[10] Thomas B Götz, Alain Hecq, and Stephan Smeekes. 2016. Testing for Granger causality in large mixed-frequency VARs. *Journal of Econometrics* 193, 2 (2016), 418–432.
[11] Clive WJ Granger. 1969. Investigating causal relations by econometric models and cross-spectral methods. *Econometrica: Journal of the Econometric Society* (1969), 424–438.
[12] Fares Hedayati and Peter L Bartlett. 2017. Exchangeability Characterizes Optimality of Sequential Normalized Maximum Likelihood and Bayesian Prediction. *IEEE Transactions on Information Theory* 63, 10 (2017), 6767–6773.
[13] Sigrid Klerke, Yoav Goldberg, and Anders Søgaard. 2016. Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357* (2016).
[14] Wojciech Kotlowski and Peter Grunwald. 2012. Sequential normalized maximum likelihood in log-loss prediction. In *Information Theory Workshop (ITW), 2012 IEEE*. IEEE, 547–551.
[15] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. 2006. Nonlinear parametric model for Granger causality of time series. *Physical Review E* 73, 6 (2006), 066216.
[16] Daniele Marinazzo, Mario Pellicoro, and Sebastiano Stramaglia. 2008. Kernel method for nonlinear Granger causality. *Physical Review Letters* 100, 14 (2008), 144103.
[17] Yoshihiko Nishiyama, Kohtaro Hitomi, Yoshinori Kawasaki, and Kiho Jeong. 2011. A consistent nonparametric test for nonlinear causality Specification in time series regression. *Journal of Econometrics* 165, 1 (2011), 112–127.
[18] Cristian Perra. 2015. Lossless plenoptic image compression using adaptive block differential prediction. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 1231–1234.

[19] Teemu Roos and Jorma Rissanen. 2008. On sequentially normalized maximum likelihood models. In *Workshop on Information Theoretic Methods in Science & Engineering*.
[20] Boris Ryabko. 2009. Compression-based methods for nonparametric prediction and estimation of some characteristics of time series. *IEEE Transactions on Information Theory* 55, 9 (2009), 4309–4315.
[21] Boris Ryabko and Jaakko Astola. 2005. Application of data compression methods to hypothesis testing for ergodic and stationary processes. In *International Conference on Analysis of Algorithms DMTCS proc. AD*, Vol. 399. 408.
[22] Boris Ryabko, Jaakko Astola, and Mikhail Malyutov. 2016. *Compression-based methods of statistical analysis and prediction of time series.* Springer.
[23] Christopher A Sims. 1972. Money, Income, and Causality. *The American Economic Review* 62, 4 (1972), 540–552.
[24] Abderrahim Taamouti, Taoufik Bouezmarni, and Anouar El Ghouch. 2014. Nonparametric estimation and inference for conditional density based Granger causality measures. *Journal of Econometrics* 180, 2 (2014), 251–264.
[25] Norbert Wiener. 1956. The theory of prediction. *Modern Mathematics for Engineers* (1956).

A REPRODUCIBILITY

Our source code has been uploaded to a GitHub account without any personal information. The experimental datasets are also included.

The hyper-link is: https://github.com/EZnlp3aUSfR8R7N y/Causal-Discovery-from-Continuous-Time-Series-by-Compr ession.

See README.md for more details.

B PROOF OF CORE THEOREMS

B.1 Proof of Theorem 1

Proof. For any x = 0, 1,. .., n, the likelihood function of $x_{t-L}, ..., x_{t-1}, x$ is :

$$\mathcal{L}(\theta; x_{t-L}, ..., x_{t-1}, x) = \prod_{i=0}^{n} [C_n^i \theta^i (1-\theta)^{n-i}]^{t_i} \cdot C_n^x \theta^x (1-\theta)^{n-x},$$

where t_i denotes the number of the bit, i, appearing within the given window, $x_{t-L}, ..., x_{t-1}$.

To maximize this likelihood function, the first order condition is applied and the following equation is drawn as:

$$\frac{x + \sum_{i=0}^{n} i t_i}{\hat{\theta}_{x_{t-L}, ..., x_{t-1}}} = \frac{n - x + \sum_{i=0}^{n} (n-i) t_i}{1 - \hat{\theta}_{x_{t-L}, ..., x_{t-1}}}.$$

To simplify the above equation,

$$\hat{\theta}_{x_{t-L}, ..., x_{t-1}} = \frac{x + \sum_{i=0}^{n} i t_i}{nL},$$

Then the likelihood function in sNML is:

$$\begin{aligned} P_{sNML}(x|x^{t-1}) &= P_{sNML}(x|x_{t-L}, ..., x_{t-1}) \\ &= \prod_{i=0}^{n} [C_n^i \hat{\theta}^i (1-\hat{\theta})^{n-i}]^{t_i} \cdot C_n^x \hat{\theta}^x (1-\hat{\theta})^{n-x} \\ &= \frac{\prod_{i=0}^{n} [C_n^i]^{t_i}}{(nL)^n} C_n^x (x+\Sigma)^{x+\Sigma} (nL - x - \Sigma)^{nL-x-\Sigma} \\ &= \frac{\prod_{i=0}^{n} [C_n^i]^{t_i}}{(nL)^n} f^x. \end{aligned}$$

Now the unconditional sNML strategy can be calculated directly through **equation (1)**:

$$\begin{aligned} P_{sNML}(x_t \,|x_{t-L}, ..., x_{t-1}) &= \frac{\mathcal{L}(\hat{\theta}_{x_{t-L}, ..., x_{t-1}, x_t}; x_t)}{\sum_x \mathcal{L}(\hat{\theta}_{x_{t-L}, ..., x_{t-1}, x}; x)} \\ &= \frac{f^{x_t}}{\sum_{i=0}^{n} f^i}. \end{aligned}$$

□

B.2 Proof of Theorem 2

Proof. First we rewrite the conditional-prediction sNML strategy as a function of history sum within a window, $\Sigma := \Sigma_{i=t-L}^{t-1} z_i$, and the next bit, $i \in \{0, 1, 2\}$:

$$P_{sNML}(x_t = i|x^{t-1}, y^{t-1}) = \max_{\Sigma \in [0, 2L-2]} \frac{f_t^i(\Sigma)}{\sum_{j=0}^{2} f_t^j(\Sigma)}, \tag{5}$$

Notice that:

$$f_t^i(\Sigma) = f_t^0(\Sigma + i). \tag{6}$$

Then:

$$P_{sNML}(x_t = i|x^{t-1}, y^{t-1}) = \max_{\Sigma\in[0,2L-2]} \frac{f_t^1(\Sigma+i-1)}{\sum_{j=0}^{2} f_t^1(\Sigma+j-1)} \tag{7}$$

And:

$$\begin{aligned}\arg\max_{\Sigma\in[0,2L]} &\frac{f_t^1(\Sigma+i)}{\sum_{j=0}^{2} f_t^1(\Sigma+j)} \\ &= \arg\min_{\Sigma\in[0,2L]} \frac{\sum_{j=0}^{2} f_t^1(\Sigma+j)}{f_t^1(\Sigma+i)}.\end{aligned} \tag{8}$$

In order to simplify the calculation and proof, we denotes:

$$g(L,\lambda) := f_t^1(L+1+\lambda). \tag{9}$$

Then the minimization problem changes to:

$$\begin{aligned}\arg\min_{\Sigma\in[0,2L]} &\frac{\sum_{j=0}^{2} f_t^1(\Sigma+j)}{f_t^1(\Sigma+i)} \\ &= \arg\min_{\lambda\in[-L,L]} \frac{\sum_{j=0}^{2} g(L,\lambda+j-1)}{g(L,\lambda+i-1)}.\end{aligned} \tag{10}$$

For further discussion, we classify the three states into two classes: i) i = 1; ii) i = 0, 2.

Class 1. *i=1. We denote $h(\lambda) = g(L,\lambda)$, then we have:*

$$\frac{d}{d\lambda} lnh(\lambda) = ln(L+\lambda) - ln(L-\lambda),$$

$$\frac{d^2}{d\lambda^2} lnh(\lambda) = \frac{2L}{L^2-\lambda^2} > 0.$$

Here our aim is equivalent to minimize:

$$G_1(\lambda) := \frac{h(\lambda+1)}{h(\lambda)} + \frac{h(\lambda-1)}{h(\lambda)}.$$

where $\lambda \in [-L, L]$.

Before we analyse the derivative of the aim function, it is useful to consider the two conponents respectively.

$$\begin{aligned} g_1(\lambda) &=: \frac{d}{d\lambda}\frac{h(\lambda+1)}{h(\lambda)} \\ &= \frac{h'(\lambda+1)}{h(\lambda)} - \frac{h'(\lambda)\cdot h(\lambda+1)}{(h(\lambda))^2} \\ &= \frac{h(\lambda+1)}{h(\lambda)}\left(\frac{d}{d\lambda} lnh(\lambda+1) - \frac{d}{d\lambda} lnh(\lambda)\right) > 0. \end{aligned}$$

Then, the derivative of another component is:

$$\begin{aligned} g_{-1}(\lambda) &:= \frac{d}{d\lambda}\frac{h(\lambda-1)}{h(\lambda)} \\ &= \frac{h(\lambda-1)}{h(\lambda)}\left(\frac{d}{d\lambda} lnh(\lambda-1) - \frac{d}{d\lambda} lnh(\lambda)\right) < 0. \end{aligned}$$

When $\lambda = 0$:

$$\begin{aligned} g_1(0) + g_{-1}(0) = &\frac{h(1)}{h(0)}\left(\frac{d}{d\lambda} lnh(1) - \frac{d}{d\lambda} lnh(0)\right) \\ &+ \frac{h(-1)}{h(0)}\left(\frac{d}{d\lambda} lnh(-1) - \frac{d}{d\lambda} lnh(0)\right) = 0 \end{aligned}$$

When $\lambda > 0$, since:

$$\frac{h(\lambda+1)}{h(\lambda)} - \frac{h(\lambda-1)}{h(\lambda)} > 0,$$

$$\frac{d}{d\lambda} lnh(\lambda+1) - \frac{d}{d\lambda} lnh(\lambda) > 0,$$

We have:

$$\begin{aligned} g_1(\lambda) + g_{-1}(\lambda) > &\frac{h(\lambda-1)}{h(\lambda)}\left(\frac{d}{d\lambda} lnh(\lambda+1)\right. \\ &\left. - \frac{d}{d\lambda} lnh(\lambda) + \frac{d}{d\lambda} lnh(\lambda-1) - \frac{d}{d\lambda} lnh(\lambda)\right) > 0. \end{aligned}$$

When $\lambda < 0$, since:

$$\frac{h(\lambda+1)}{h(\lambda)} - \frac{h(\lambda-1)}{h(\lambda)} < 0,$$

$$\frac{d}{d\lambda} lnh(\lambda-1) - \frac{d}{d\lambda} lnh(\lambda) < 0,$$

We have:

$$\begin{aligned} g_1(\lambda) + g_{-1}(\lambda) < &\frac{h(\lambda+1)}{h(\lambda)}\left(\frac{d}{d\lambda} lnh(\lambda+1)\right. \\ &\left. - \frac{d}{d\lambda} lnh(\lambda) + \frac{d}{d\lambda} lnh(\lambda-1) - \frac{d}{d\lambda} lnh(\lambda)\right) < 0. \end{aligned}$$

*Until now we can conclude that for i = 1, the assertion of **Theorem 2** is proven.*

Class 2. *i = 0, 2. We only consider i = 0 due to the duality. Here the optimization problem is changed into minimizing:*

$$G_0(\lambda) = 2\cdot\frac{h(\lambda)}{h(\lambda-1)} + \frac{h(\lambda+1)}{h(\lambda-1)}.$$

where $\lambda \in [-L, L]$.

$$g_2(\lambda) =: \frac{d}{d\lambda}\frac{h(\lambda+2)}{h(\lambda)} > 0,$$

Hence:

$$G_0'(\lambda) = 2\cdot g_1(\lambda-1) + g_2(\lambda-1) > 0.$$

*Thus for i = 0, the assertion of **Theorem 2** is proven. According to duality, the assertion is true when i = 2.*

Finally, the whole assertion of **Theorem 2** is proven. □