

Past-future Mutual Information Estimation in Sparse Information Conditions

Yuval Shalev^a and Irad Ben-Gal^b

Laboratory for AI, Machine Learning, Business & Data Analytics, Department of Industrial Engineering,

Keywords: Past-future Mutual Information, Context Tree, Transfer Entropy, Time Series Analysis.

Abstract: We introduce the CT-PFMI, a context tree based algorithm that estimates the past-future mutual information (PFMI) between different time series. By applying a pruning phase of the context tree algorithm, uninformative past sequences are removed from PFMI estimation along with their false contributions. In situations where most of the past data is uninformative, the CT-PFMI shows better estimates to the true PFMI than other benchmark methods as demonstrated in a simulated study. By implementing CT-PFMI on real stock prices data we also demonstrate how the algorithm provides useful insights when analyzing the interactions between financial time series.


1 INTRODUCTION


Accurate estimation of the mutual information between the past of one time series and the future of another is an important task in time series analysis. For instance, the transfer entropy (Schreiber, 2000), that measures the conditional past-future mutual information (PFMI) between the past of one or more time series and an output time series that are conditioned on the past of the output time series, has been widely explored in the past two decades in various domains such as neural-science and economics (Bossmmaier et al., 2016). However, a difficulty arises when PFMI needs to be estimated from data observations. The number of possible sequences that potentially contributes to the mutual information increases exponentially with the number of time lags. When most realized past sequences are uninformative about the future, a condition we call sparse PFMI, large number of false contributors could lead to overestimation of PFMI, hence associating predictive power to uninformative sequences.

The methods that are used to estimate PFMI, usually in the context of transfer entropy estimation, are based on commonly used MI estimation methods ranging from naive binning (also called the Plug-in method) to bias and variance corrections such as the nearest neighbors method (Montalto et al., 2014;

Runge et al., 2012). When applied to time series, these methods resolve the time dimensionality problem by removing uninformative time lags entirely. Nevertheless, to the best of our knowledge, none of these methods apply estimation correction at a realization level, which has a greater potential for dimensionality reduction and can provide an insightful perspective on the nature of the underlying interactions.

We provide such a solution by estimating the PFMI using an expansion of the context tree (CT) algorithm which is called the input/output context tree (I/O CT) algorithm (Ben-Gal et al., 2005; Brice and Jiang, 2009). This algorithm parses the input time series into a tree of contexts (sequences), where in each node, the conditional probability of the output given the context is assigned. Next, only nodes with conditional probabilities that are significantly different from those of their parent nodes (often measured by the Kullback-Liebler divergence) are kept, and the others are pruned. This algorithm, as well as other algorithms from the Variable Order Markov Models family, were proposed to overcome overfitting in learning tasks such as classification and prediction (Ben-Gal et al., 2005; Begleiter et al., 2004; Shmilovici and Ben-Gal, 2012; Yang et al., 2014). Estimating the information between a time series' past and future was usually not one of the tasks these algorithms were used for. We show how to estimate PFMI between time series as the sum of the Kullback-Leibler divergence (Kullback and Leibler, 1951) be-

^a  <https://orcid.org/0000-0003-2125-9735>

^b  <https://orcid.org/0000-0003-2411-5518>

tween the root node and the leaves of I/O CT. The proposed procedure is implemented by a proposed context tree past-future mutual information algorithm (CT-PFMI): First, a full I/O CT is built. Second, the PFMI is calculated for descending values of the pruning constant c , a positive parameter which defines the number of pruned sequences (Ben-Gal et al., 2003). Third, by identifying the threshold at which redundant information is removed, a value of c is chosen to obtain an estimate for the "filtered" PFMI as well as most of the informative sequences.

In the results section it is shown that in simulated sparse PFMI condition, the CT-PFMI estimates the PFMI more accurately than benchmark methods. The proposed CT-PFMI is also implemented on real time series of stock prices returns, that due to market efficiency, follow the sparse PFMI condition (Shmilovici and Ben-Gal, 2012). The outcome of the CT-PFMI algorithm can also be exploited to gain important insights by performing a higher-resolution analysis of the PFMI contributors as demonstrated by real time series data.

To conclude, the first contribution of this paper is to demonstrate the extraction of PFMI from an I/O CT constructed from input and output time series. The second contribution is the introduction of a novel algorithm, called the CT-PFMI. This algorithm, is used for PFMI estimation, while offering a new method of identifying the value of the pruning constant that governs the compression rate. The third contribution is showing how the CT-PFMI algorithm can be used for in-depth analysis of interaction's insights in the data.

2 RELATED WORK

In the previous section we mentioned the works on transfer entropy (Schreiber, 2000; Bossomaier et al., 2016) as an important source for discussion on estimating the information flow between the past and the future of time series.

Researchers such as (Runge et al., 2012; Montalto et al., 2014) used standard methods of MI estimation, such as binning (Cover and Thomas, 2012) or nearest-neighbours (Kraskov et al., 2004), to estimate TE. According to those methods, when a specific time lag is found to be informative in some specific realizations, all its realizations, including the uninformative ones, are included in the estimation. In sequential data, where the number of different realizations is potentially large, this drawback can be crucial by adding many uninformative sequences to the estimation affecting both the TE accuracy as well as the extracted insights from the data.

To overcome this challenge, we utilize the CT algorithm, a member of the family of Variable Order Markov Models that were originally constructed for compression of a single time series, and found it to be well-suited to the prediction task of discrete time series (Weinberger et al., 1995; Begleiter et al., 2004; Shmilovici and Ben-Gal, 2012). Variable Order Markov Models and their usage have been extensively explored (Begleiter et al., 2004; Shmilovici and Ben-Gal, 2012; Yang et al., 2014; Slonim et al., 2003; LARGERON-LETÉNO, 2003; Society et al., 2014; Chim and Deng, 2007; Ben-Gal et al., 2003; Begleiter et al., 2013; Ben-Gal et al., 2005; Kusters and Ignatenko, 2015). Two works were found that incorporated Variable Order models and information or entropy (Schürmann and Grassberger, 1996; Slonim et al., 2003), yet none of them used these models for PFMI estimation.

Ben-Gal et al (Ben-Gal et al., 2005) and later Brice et al (Brice and Jiang, 2009) proposed an input/output formulation of the context tree algorithm (I/O CT), where the branches of the context tree belong to one time series and the leaves belong to another time series. In this way, the researchers could incorporate data from different time series for learning tasks, such as structure learning and anomaly detection within the CT framework.

Let us also note that the CT-PFMI algorithm is scalable using methods presented in (Satish et al., 2014; Kaniwa et al., 2017; Tiwari and Arya, 2018; Satish et al., 2014; Tiwari and Arya, 2018).

3 PRELIMINARIES AND MATHEMATICAL BACKGROUND

Henceforth, unless stated otherwise, random variables are represented by capital letters, while their realizations are denoted by lower-case letters; multi-dimensional variables and arrays are denoted by bold letters.

Mutual Information (Cover and Thomas, 2012): Given two discrete random variables \mathbf{X} and \mathbf{Y} , the Mutual Information between them is defined as

$$I(\mathbf{X};\mathbf{Y}) = \sum_{x \in \mathbf{X}} \sum_{y \in \mathbf{Y}} P(x,y) \log \frac{P(x,y)}{P(x)P(y)}. \quad (1)$$

$I(\mathbf{X};\mathbf{Y})$ is a positive symmetrical measure. The Kullback-Liebler divergence (D_{KL}) between arbitrary probability functions $Q(\cdot)$ and $P(\cdot)$ is given by

$$D_{KL}(Q(X, Y) \parallel P(X, Y)) = \sum_{x \in X} \sum_{y \in Y} Q(x, y) \log \frac{Q(x, y)}{P(x, y)}. \quad (2)$$

Following Eq.(2), the $I(X; Y)$ can be written as

$$I(X; Y) = \langle D_{KL}(P(Y|X) \parallel P(Y)) \rangle_{P(X)}, \quad (3)$$

where $\langle \cdot \rangle_{P(\cdot)}$ is the expectation with respect to the subscript distribution.

The Past-future Mutual Information: To explain PFMI, we use the notation in (Bialek et al., 2001; Still, 2014) whom introduce a similar measure to PFMI called the predictive information (PI), that is the mutual information between two random vectors, one representing the past τ_p time lags, $\bar{\mathbf{X}}_{\tau_p}$ and another representing time series values from the future τ_f time lags, $\bar{\mathbf{Y}}_{\tau_f}$. Following Eq.(3), the PI can be defined by using D_{KL}

$$PI(\bar{\mathbf{X}}_{\tau_p}; \bar{\mathbf{Y}}_{\tau_f}) = \langle D_{KL}(P(\bar{\mathbf{X}}_{\tau_f} | \bar{\mathbf{X}}_{\tau_p}) \parallel P(\bar{\mathbf{Y}}_{\tau_f})) \rangle_{P(\bar{\mathbf{X}}_{\tau_p})}. \quad (4)$$

and,

$$PI(\bar{\mathbf{X}}_{\tau_p}; \bar{\mathbf{Y}}_{\tau_{f=1}}) = PFMI(\bar{\mathbf{X}}_{\tau_p}; \bar{\mathbf{Y}}). \quad (5)$$

Context Tree (CT) Algorithm (Weinberger et al., 1995; Ben-Gal et al., 2003): Given a sequence of length N , \mathbf{x}^N , generated from a tree source X , the CT algorithm finds a finite set \mathcal{S} of size $|\mathcal{S}|$ of contexts $\mathcal{S}(x^N)$. \mathcal{S} satisfies the requirement that the conditional probability to obtain a symbol given the whole sequence preceding that symbol is close enough to the

Table 1: Optimal contexts of the I/O CT of Deutsche Bank (input) to HSBC (output) as obtained with the CT-PFMI algorithm and the pruning constant tuning algorithms (see Section 5). The returns are discretized to "1" for positive return, "0" for zero return and "-1" for negative return with respect to the previous minute.

Optimal Context	Context Probability	Conditional probability
root	-	(0.42, 0.16, 0.42)
("-1")	0.369	(0.45, 0.15, 0.40)
("0")	0.111	(0.40, 0.20, 0.40)
("1")	0.370	(0.40, 0.15, 0.45)
("-1", "0")	0.057	(0.43, 0.20, 0.37)
("1", "0")	0.058	(0.37, 0.20, 0.43)
("0", "0")	0.011	(0.37, 0.27, 0.36)
("0", "0", "1")	0.011	(0.36, 0.25, 0.39)
("0", "0", "0", "-1")	0.003	(0.35, 0.30, 0.35)
("0", "0", "0", "1")	0.003	(0.33, 0.30, 0.37)
("0", "0", "0", "0")	0.002	(0.32, 0.33, 0.35)
("0", "0", "0", "0", "0")	0.005	(0.06, 0.87, 0.07)

conditional probability of obtaining the symbol given a context, i.e.,

$$P(x|\mathbf{x}^N) \cong P(x|\mathcal{S}(\mathbf{x}^N)). \quad (6)$$

Given Eq.(6), when $|\mathcal{S}|$ sequences are informative, the number of conditional probability parameters that are required to describe \mathbf{x}^N equals $|\mathcal{S}|(d-1)$, where d is the alphabet size of X .

To obtain \mathcal{S} , the learning algorithm constructs a context tree where each node holds a set of ordered counters that represent the distribution of symbols that follow that context, which is defined by the path to that node (Ben-Gal et al., 2003). At the next step, a pruning procedure is performed to leave only those contexts in \mathcal{S} (called optimal contexts (Ben-Gal et al., 2003)) - with corresponding nodes in the tree that represent the conditional distribution of the output variable conditioned on the context which is different from the distributions of the output variable conditioned only on part of the context (represented by the path from the tree root to the parent node). Table 1 shows all the optimal contexts and their corresponding conditional probabilities in a I/O context tree obtained in stock returns data that will be discussed in the result section. Fig. 1 shows in a context tree formation some of the optimal contexts obtained in this table.

Descriptions of the main principles of the CT Algorithm, including how to obtain \mathcal{S} and a numerical example appear in (Ben-Gal et al., 2003).

The I/O CT (Ben-Gal et al., 2005; Brice and Jiang, 2009) algorithm is a generalization of the CT algorithm where the tree's contexts are from the input sequence and the leaves represent counters of the output sequence, in contrast to Eq.(6), where the input and the output are from the same time series

$$P(y|\mathbf{x}^N) \cong P(y|\mathcal{S}(\mathbf{x}^N)). \quad (7)$$

4 THE CONTEXT TREE PAST-FUTURE MUTUAL INFORMATION ALGORITHM

Let $\{\bar{\mathbf{x}}; \tilde{\bar{\mathbf{x}}}\} \in \bar{\mathbf{x}}_{\tau_p}$ represent the informative and uninformative contexts respectively from the input time series, $\bar{\mathbf{y}}$ represents the symbols from the output time series and $PFMI(\bar{\mathbf{x}}_{\tau_p}; \bar{\mathbf{y}})$ represent the estimated PFMI. We define the uninformative sequences as those with conditioning probability with respect to the output that do not result in a conditional distribution of the output time series, which is significantly different from unconditional marginal distribution of

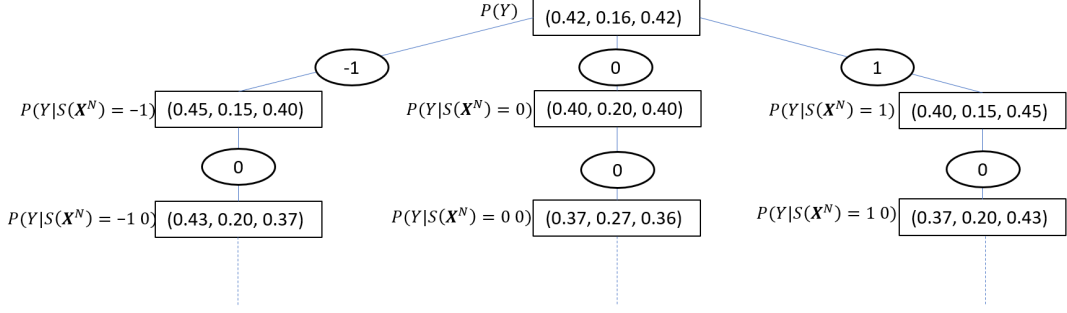


Figure 1: The I/O CT representation of some of the optimal contexts in Table 1 as obtained from HSBC (input) to Deutsche Bank (output) stock prices time series. Each edge represents a single context realizations. Consecutive edges represent contexts (sequences) in reverse order. The nodes represent the conditional probabilities of the output time series given the input context between the root to that node of the tree. The root (at the top of the tree) contains the marginal distribution of the output time series.

the output. Formally, $\{\hat{\mathbf{x}} : D_{KL}(P(\bar{\mathbf{y}}|\hat{\mathbf{x}}) \| P(\bar{\mathbf{y}})) = 0\}$. Due to the finite size of the data, often the empirical measurement leads to $D_{KL}(\hat{P}(\bar{\mathbf{y}}|\hat{\mathbf{x}}) \| \hat{P}(\bar{\mathbf{y}})) > 0$, so positive bias can occur. In the sparse PFMI condition, where $\frac{|\hat{\mathbf{x}}|}{|\mathbf{x}|} \ll 1$, removing these contexts can significantly decrease \widehat{PFMI} estimation error and enhance better understanding of the "source of information" (Tishby et al., 2000).

To achieve this goal, we apply some of the principles implemented in (Slonim et al., 2003), to introduce a novel method for \widehat{PFMI} estimation using the I/O CT. Let \mathbf{X}^N and \mathbf{Y}^N be the input and the output time series of length N respectively. As discussed in Section 3, the root node of the I/O CT represents the marginal (unconditioned) distribution of \mathbf{Y}^N (the symbols' frequency in \mathbf{Y}^N). The estimated PFMI between the input and the output time series is the sum of the D_{KL} between the root node and the conditional probabilities given the contexts in \mathcal{S} , weighted by the probabilities of these contexts, following Eqs.(4) and (5) is

$$\widehat{PFMI}_c = \langle D_{KL}(\hat{P}(\bar{\mathbf{y}}|\mathcal{S}_c(\hat{\mathbf{x}})) \| \hat{P}(\bar{\mathbf{y}})) \rangle_{\hat{P}(\mathcal{S}_c(\hat{\mathbf{x}}))}, \quad (8)$$

where \widehat{PFMI}_c is the empirical PFMI obtained from the I/O CT algorithm with a pruning constant c and $\mathcal{S}_c(\hat{\mathbf{x}})$ is its corresponding optimal contexts set. To continue with the running example of stocks returns data, we use Table 1 that represents the obtained context tree. using Eq.(8), \widehat{PFMI} with $c = 1$ can be calculated as follows

$$\begin{aligned} \widehat{PFMI}_1 = & 0.369 \cdot D_{KL}(0.45, 0.15, 0.40) \| (0.42, 0.16, 0.42) + \\ & 0.111 \cdot D_{KL}(0.40, 0.20, 0.40) \| (0.42, 0.16, 0.42) + \\ & \dots + \\ & 0.005 \cdot D_{KL}(0.06, 0.87, 0.07) \| (0.42, 0.16, 0.42) = \\ & 0.016 \text{ bits.} \end{aligned} \quad (9)$$

So far, the extraction of \widehat{PFMI} from CT with a given c value has been described. A tuning method for finding the value of c that results in a good separation between informative and uninformative contexts is now proposed by utilizing the statistics gained by the first stage in the CT algorithm. Consider the vector \mathbf{c} of indexed pruning constant values c_i . The empirical second derivative of \widehat{PFMI}_{c_i} with respect to $|\mathcal{S}_{c_i}|$ can be obtained by

$$\frac{\partial^2 \widehat{PFMI}_{c_i}}{\partial |\mathcal{S}_{c_i}|^2} = \frac{\widehat{PFMI}_{c_{i+1}} + \widehat{PFMI}_{c_{i-1}} - 2\widehat{PFMI}_{c_i}}{(|\mathcal{S}_{c_{i+1}}| - |\mathcal{S}_{c_{i-1}}|)^2}. \quad (10)$$

When the absolute value of Eq.(10) reaches a greater value than a threshold ϵ , the correspondent pruning constant is chosen. The second derivative is used to enable the detection of changes from higher than a linear order (e.g, a curved shaped changes) in the \widehat{PFMI} . Linear decrease is expected to happen when uninformative contexts are removed. The reason for this behaviour lies in the pruning threshold of the CT algorithm. This threshold equals to the probability of a context times a parent-child D_{KL} measure. In the uninformative case, incrementally increasing the pruning constant will result in the pruning of all the leaves in the same tree level in a reverse order. Hence, in each incremental increase in the pruning constant c ,

the same size of \widehat{PFMI} is subtracted. When one of the contexts contains a significant amount of information, its pruning will result in a higher order change in the empirical PFMI.

\widehat{PFMI} extraction and the tuning of the pruning constant c constitute the CT-PFMI algorithm (see Algorithm 1). First, the estimated PFMI is extracted iteratively from the I/O CT for decreasing values of c . When the second derivative condition is satisfied, the algorithm stops and returns the values of c and the PFMI of the last iteration. Note that the full I/O CT is constructed only once in the first iteration, so the complexity of this algorithm is dominated by this construction with complexity of $O(N \log N)$ (Ben-Gal et al., 2003).

Considering the \widehat{PFMI} randomness, we need to reject the null hypothesis that $\widehat{PFMI} = 0$, especially in sparse PFMI condition. Here, we adopt the approach of (Vicente et al., 2011) by setting the stopping threshold ε to be higher than the 95 percentile value of \widehat{PFMI} obtained by repeatedly reshuffling the time series and measuring the resulting \widehat{PFMI} .

Algorithm 1: Context Tree Past-Future Mutual Information.

```

1: Input:  $\mathbf{x}^N, \mathbf{y}^N, \mathbf{c}, \varepsilon$ 
2: Implement on  $\mathbf{x}^N, \mathbf{y}^N$  the first stage of the I/O CT algorithm to obtain a full I/O context tree
3: for  $i$  in 1 to  $|\mathbf{c}|-1$  do
4:   Implement the following stages of the I/O CT algorithm
5:   with  $\mathbf{c}_{i-1}, \mathbf{c}_i, \mathbf{c}_{i+1}$ , and obtain  $\mathcal{S}_{c_{i-1}}, \mathcal{S}_{c_i}, \mathcal{S}_{c_{i+1}}$ 
6:   Calculate  $\widehat{PFMI}_{c_{i-1}}, \widehat{PFMI}_{c_i}, \widehat{PFMI}_{c_{i+1}}$ 
7:   if  $|\mathcal{S}_{c_{i-1}}| = |\mathcal{S}_{c_{i+1}}|$  then
8:      $dv2 \leftarrow 0$ 
9:   else
10:     $dv2 \leftarrow \left| \frac{\partial^2 \widehat{PFMI}_{c_i}}{\partial |\mathcal{S}_{c_i}|^2} \right|$ 
11:   end if
12:   if  $dv2 > \varepsilon$  then
13:     return  $c_i$ 
14:   end if
15: end for
16: return 0

```

5 EMPIRICAL RESULTS

This section shows the results of a simulation setup with a known ground truth, which is used to measure the performance of the CT-PFMI algorithm compared to benchmark methods in sparse PFMI environment. Later, a real financial time series is used as an example for the CT-PFMI algorithm usage for PFMI estimation and a high-resolution data analysis.

5.1 PFMI Estimation in Sparse PFMI Conditions, a Simulated Study

In this example, \widehat{PFMI} is measured between an input time series with alphabet size starting from 20 to 90 symbols and the output binary time series. The time series length is 5000 discrete time steps. The sparse PFMI condition is achieved by randomly choosing two of the alphabet symbols to be informative with the following conditional probability:

$$\begin{aligned}
P(\bar{y} = 1 | \bar{x}_1) &= 0.95 \\
P(\bar{y} = 0 | \bar{x}_1) &= 0.05 \\
P(\bar{y} = 1 | \bar{x}_2) &= 0.05 \\
P(\bar{y} = 0 | \bar{x}_2) &= 0.95.
\end{aligned}$$

One hundred simulation runs were performed per each alphabet size. When the size of the alphabet increases, the sparse PFMI condition becomes more significant. The CT-PFMI performances were compared to the commonly used plug-in (Cover and Thomas, 2012) method and the K-NN method (Kraskov et al., 2004) which is used in many recent studies on TE (Runge et al., 2012; Vicente et al., 2011; Montalto et al., 2014). The PFMI estimation error of CT-PFMI and the benchmark methods relatively to the true theoretical PFMI is shown in Fig.2, as a function of the dictionary (alphabet) size. Three values of K in the K-NN method were used, testing different bias-variance trade-offs. Fig.2 demonstrates the robustness of CT-PFMI estimations to increasing size of uninformative sequences, showing relatively small increase in estimation error while the benchmark methods that show significant increase with the plug-in method that is the most sensitive to increasing alphabet size. K-NN method with $k = 10$ shows the best results for this method. The fact that CT-PFMI can remove uninformative sequences, and not only assign to them a small contribution, supports this robustness.

5.2 The CT-PFMI Algorithm - Example of Real Stock Prices Data

Stock market time series analysis is an example of a real-world application of the CT-PFMI algorithm. In this case, the sparse PFMI condition is a reasonable assumption because of market efficiency (Shmilovici and Ben-Gal, 2012). That is, in an efficient market only few historical pattern or contexts exist that can be used for predictions, while most of these patterns are insignificant (Shmilovici and Ben-Gal, 2012). The dataset comprises minute-by-minute time series of stock prices of eight large banks in the U.S. for the

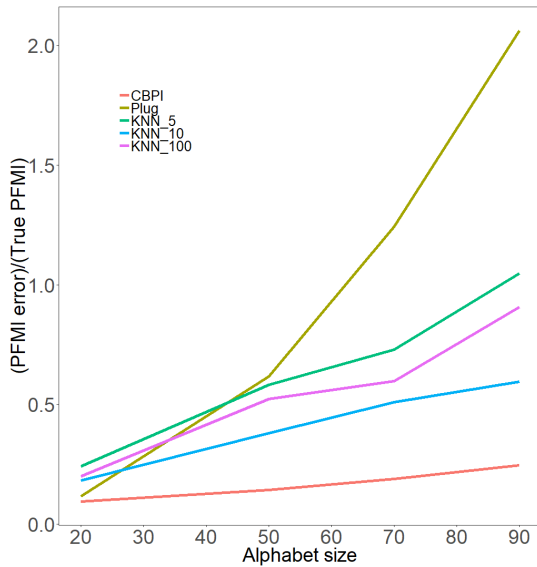


Figure 2: Average PFMI estimation error of the CT-PFMI algorithm and the benchmark methods with respect to the true PFMI theoretical value in different values of alphabet size. The K-NN with different number of neighbors (k) was calculated using the Parmigene R package (Sales and Romualdi, 2011).

period of 1.2008-1.2010 that because of the banking crisis within these years, has a potential of nonzero \widehat{PFMI} in between banks (Dimpfl and Peter, 2014). The length of the time series was 197,000, hence, a distributed I/O CT algorithm was implemented.

Stock prices were discretized to $+1$, 0 and -1 for positive, zero and negative changes, respectively, relatively to the price of the previous minute. For each bank, the PFMI was obtained by implementing the algorithm of Section 4 for various values of $1/c$ (see Fig.3). All curves exhibit a similar behavior of a phase where uninformative sequences are removed followed by a steep drop in PFMI after crossing a certain pruning constant threshold that corresponded to pruning of sequences from \mathcal{S} . The Pruning constant obtained from the CT-PFMI algorithm ranged between 0.13 to 1.33, depending on the input/output pair. These values corresponds to filtering 96 percent of sequences.

Using the descriptive power of the CT-PFMI algorithm, hierarchical analysis can be obtained. For example, in the higher level, a geographic orientation can be identified when looking at Fig.3. The estimated PFMI between the European banks HSBC and DB is higher than the estimated PFMI between these banks and the American banks.

Moving to lower hierarchies of the interactions, the conditional probabilities of the output sequences given the contexts in \mathcal{S} differ from the marginal distri-

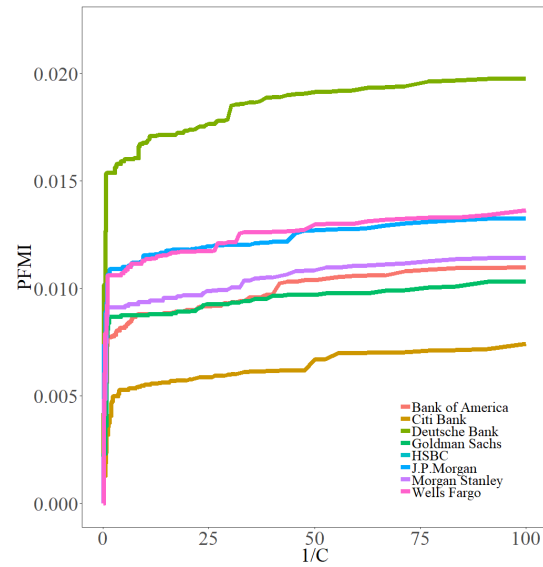


Figure 3: Estimated PFMI of large banks' stock prices in the Wall Street stock exchange (input) with respect to the stock prices of HSBC bank (output), calculated as a function of the inverse of the pruning constant c . Shuffled input time series showed maximum PFMI values of $\approx 5 \cdot 10^{-5}$.

bution of the output in the probabilities of each symbol, but the symmetry between -1 and $+1$ is relatively preserved. For example, see the contexts obtained with the I/O CT of DB to HSBC in Table 1. Hence, for trading purposes, additional information is needed.

Another conclusion can be drawn from the contexts' length. The average memory of the process is 1.5 symbols, as calculated by multiplication of all contexts' lengths by their respective probabilities (see Table 1). This observation implies that most of the information within $\tau_p = 2$.

6 CONCLUSIONS

We showed how the Input/Output context tree algorithm can be utilized to measure the past-future mutual information between time series. Using that, we demonstrated how the pruning constant parameter of the I/O CT algorithm can be calibrated in a way that separates informative versus uninformative sequences. This approach constitutes the CT-PFMI algorithm for PFMI estimation. We used sparse past-future predictive information (sparse PFMI) simulated data with a known theoretical PFMI values to benchmark the CT-PFMI algorithm against other common PFMI estimation methods. This comparison shows the advantages of the CT-PFMI algorithm over the benchmark methods under sparse PFMI condi-

tions. The CT-PFMI algorithm was also implemented on real stock prices data to show the sparse PFMI effect between pairs of real-world time series. It was also demonstrated how the CT-PFMI algorithm can be used for in-depth analyses of interactions between time series.

ACKNOWLEDGEMENTS

This research was funded by the Koret foundation grant for Smart Cities and Digital Living 2030.

REFERENCES

- Begleiter, R., El-Yaniv, R., and Yona, G. (2004). On prediction using variable order markov models. *Journal of Artificial Intelligence Research*, 22:385–421.
- Begleiter, R., Elovici, Y., Hollander, Y., Mendelson, O., Rokach, L., and Saltzman, R. (2013). A fast and scalable method for threat detection in large-scale dns logs. In *Big Data, 2013 IEEE International Conference on*, pages 738–741. IEEE.
- Ben-Gal, I., Morag, G., and Shmilovici, A. (2003). Context-based statistical process control: A monitoring procedure for state-dependent processes. *Technometrics*, 45(4):293–311.
- Ben-Gal, I., Shani, A., Gohr, A., Grau, J., Arviv, S., Shmilovici, A., Posch, S., and Grosse, I. (2005). Identification of transcription factor binding sites with variable-order bayesian networks. *Bioinformatics*, 21(11):2657–2666.
- Bialek, W., Nemenman, I., and Tishby, N. (2001). Predictability, complexity, and learning. *Neural computation*, 13(11):2409–2463.
- Bossomaier, T., Barnett, L., Harré, M., and Lizier, J. T. (2016). *An introduction to transfer entropy*. Springer.
- Brice, P. and Jiang, W. (2009). A context tree method for multistage fault detection and isolation with applications to commercial video broadcasting systems. *IIE Transactions*, 41(9):776–789.
- Chim, H. and Deng, X. (2007). A new suffix tree similarity measure for document clustering. In *Proceedings of the 16th international conference on World Wide Web*, pages 121–130. ACM.
- Cover, T. M. and Thomas, J. A. (2012). *Elements of information theory*. John Wiley & Sons.
- Dimpfl, T. and Peter, F. J. (2014). The impact of the financial crisis on transatlantic information flows: An intraday analysis. *Journal of International Financial Markets, Institutions and Money*, 31:1–13.
- Kaniwa, F., Kuthadi, V. M., Dinakenyane, O., and Schroeder, H. (2017). Alphabet-dependent parallel algorithm for suffix tree construction for pattern searching. *arXiv preprint arXiv:1704.05660*.
- Kraskov, A., Stögbauer, H., and Grassberger, P. (2004). Estimating mutual information. *Physical review E*, 69(6):066138.
- Kullback, S. and Leibler, R. A. (1951). On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.
- Kusters, C. and Ignatenko, T. (2015). Dna sequence modeling based on context trees. In *Proc. 5th Jt. WIC/IEEE Symp. Inf. Theory Signal Process. Benelux*, pages 96–103.
- Largerion-Leténo, C. (2003). Prediction suffix trees for supervised classification of sequences. *Pattern Recognition Letters*, 24(16):3153–3164.
- Montalto, A., Faes, L., and Marinazzo, D. (2014). Mute: a matlab toolbox to compare established and novel estimators of the multivariate transfer entropy. *PloS one*, 9(10):e109462.
- Runge, J., Heitzig, J., Petoukhov, V., and Kurths, J. (2012). Escaping the curse of dimensionality in estimating multivariate transfer entropy. *Physical review letters*, 108(25):258701.
- Sales, G. and Romualdi, C. (2011). parmigene—a parallel r package for mutual information estimation and gene network reconstruction. *Bioinformatics*, 27(13):1876–1877.
- Satish, U. C., Kondikoppa, P., Park, S.-J., Patil, M., and Shah, R. (2014). Mapreduce based parallel suffix tree construction for human genome. In *Parallel and Distributed Systems (ICPADS), 2014 20th IEEE International Conference on*, pages 664–670. IEEE.
- Schreiber, T. (2000). Measuring information transfer. *Physical review letters*, 85(2):461.
- Schürmann, T. and Grassberger, P. (1996). Entropy estimation of symbol sequences. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, 6(3):414–427.
- Shmilovici, A. and Ben-Gal, I. (2012). Predicting stock returns using a variable order markov tree model. *Studies in Nonlinear Dynamics & Econometrics*, 16(5).
- Slonim, N., Bejerano, G., Fine, S., and Tishby, N. (2003). Discriminative feature selection via multiclass variable memory markov model. *EURASIP Journal on Applied Signal Processing*, 2003:93–102.
- Society, T. X., Wang, S., Jiang, Q., and Huang, J. Z. (2014). A novel variable-order markov model for clustering categorical sequences. *IEEE Transactions on Knowledge and Data Engineering*, 26(10):2339–2353.
- Still, S. (2014). Information bottleneck approach to predictive inference. *Entropy*, 16(2):968–989.
- Tishby, N., Pereira, F. C., and Bialek, W. (2000). The information bottleneck method. *arXiv preprint physics/0004057*.
- Tiwari, V. S. and Arya, A. (2018). Distributed context tree weighting (ctw) for route prediction. *Open Geospatial Data, Software and Standards*, 3(1):10.
- Vicente, R., Wibral, M., Lindner, M., and Pipa, G. (2011). Transfer entropy—a model-free measure of effective connectivity for the neurosciences. *Journal of computational neuroscience*, 30(1):45–67.
- Weinberger, M. J., Rissanen, J. J., and Feder, M. (1995). A universal finite memory source. *IEEE Transactions on Information Theory*, 41(3):643–652.
- Yang, J., Xu, J., Xu, M., Zheng, N., and Chen, Y. (2014). Predicting next location using a variable order markov model. In *Proceedings of the 5th ACM SIGSPATIAL International Workshop on GeoStreaming*, pages 37–42. ACM.