Reconstruction of hand trajectory with video

Eduard Vladimirov[1*], Vadim Strijov[1†] and Roman Isachenko[1†]

[1]Moscow Institute of Physics and Technology, Inststitutskii per. 9, Dolgoprudny, 141700, Moscow Region, Russia.

*Corresponding author(s). E-mail(s): vladimirov.ea@phystech.edu;
Contributing authors: strijov@phystech.edu; roman.isachenko@phystech.edu;
†These authors contributed equally to this work.

Abstract

In this paper, we consider the problem of forecasting a time series with a complex structure. The complex structure means the presence of non-linear dependencies and a varying period. We must to find causal relationships between time series. In order to do this, we reduce the dimension of trajectory spaces. The paper introduces a new way for the consistent dimensional reduction of time series. The proposed method combines the partial least squares method and convergent cross mapping method. To demonstrate the results of the work we solve the problem of hand trajectory reconstruction with video.

Keywords: Pose estimation, Time series, Phase trajectory, Trajectory subspace, Convergent cross mapping, Partial least squares

1 Introduction

In this paper, we solve the problem of forecasting a time series based on other time series. One of the challenges is to detect relationships between time series and exclude unrelated time series from the predictive model. Solving this problem improves model's quality.

In this paper, we apply the convergent cross mapping method (CCM) or the Sugihara method [**?**, **?**], which is effective for time series generated by a

dynamic system. It is based on a comparison of the nearest neighbors in the trajectory space of the time series $\mathbf{y}$ obtained from the time series $\mathbf{x}$.

When constructing a predictive model, we use a trajectory matrix (or shift matrix) that describes the phase space of a time series. For example, in the method of singular spectral analysis (SSA) [**?**, **?**, **?**], the time series prediction is based on the spectral decomposition of the covariance matrix obtained from the trajectory matrix. In CCM, shift matrices are used for checking the presence of a Lipschitz mapping between trajectory spaces.

However, the dimension of the trajectory space may be extremely high, which leads to instability of the predictive model. In this case, it is necessary to reduce the dimension of the trajectory space by constructing a projection of the phase trajectory into some subspace. There is no specific way for CCM to select a subspace in which the phase trajectory is approximated. In the paper [**?**] this problem is solved using spherical regression. According to this method, information about the desired subspace is extracted from the set of empirical directions $\{\mathbf{x}_i - \mathbf{x}_j \mid i < j\}$, where $\mathbf{x}_i$ — elements of the trajectory space in spherical coordinates. In the work [**?**] automatic selection of a pair of principal components is used. The main idea is to compare the spectral densities of the principal components. A simple iteration over the principal components [**?**] is also used.

The partial least squares method (PLS) [**?**, **?**] selects the most significant features and builds new ones as their linear combination. This allows us to obtain a simple, accurate and stable predictive model. Along with PLS, the canonical correlation analysis (CCA) [**?**] method is used. It is similar to PLS except that the former method maximizes the covariance between projections, and the latter — correlation. The disadvantage of these models is their low accuracy in estimating nonlinear dependencies between data. The nonlinear extensions of PLS [**?**] and CCA [**?**] have been developed. This article uses the PLS-Autoencoder model [**?**], which converts the source data using autoencoders.

The theoretical part of the paper shows how to apply the Sugihara method to reduce the dimension of the trajectory space and how to combine the ideas of the PLS and CCM methods. To achieve the latter goal, a new metric for the consistency of latent projections has been introduced.

The algorithm of sequential locally weighted global linear map (SMAP) [**?**] is used as a model for predicting time series from a set of time series.

The experiment is carried on a set of manually collected data. It is a collection of key points obtained from a video of a person's movement, as well as accelerometer and gyroscope readings taken from a person's hand. In the experiment, a time series forecast is constructed using the detected related components of the time series.

2 Problem statement

Let the values of the multidimensional time series

$$\mathbf{S}_y(t) = [S_y^1(t), \ldots, S_y^r(t)]^\mathsf{T}$$

be available at time points $t = 1, 2, \ldots, n$ We assume that a set of auxiliary time series $S_x^1(t), \ldots, S_x^m(t)$ affects the values of $\mathbf{S}_y(t)$.

It is necessary to predict the values of the original time series $\mathbf{S}_y(t)$ at time points $n+1, \ldots, n+p$. We assume that the values of the auxiliary time series are available in the time period for which the prediction of the time series $\mathbf{S}_y(t)$ is carried out.

In order to calculate the future values of a time series, we must determine a functional dependence illustrating the relationship between the past values of $\mathbf{S}_y(t)$ and the future ones, as well as taking into account the influence of auxiliary time series $S_x^1(t), \ldots, S_x^m(t)$.

Definition 1 The prediction model with external factors is a function:

$$\mathbf{S}_y(t) = \mathbf{F}(\mathbf{w}, \mathbf{S}_y(t-1), \ldots, S_x^1(t), \ldots, S_x^m(t), \ldots) + \boldsymbol{\epsilon}_t.$$

We need to create a model for which the mean square deviation of the true value from the predicted value tends to the minimum for a given p:

$$\widehat{E} = \frac{1}{p} \sum_{i=n+1}^{n+p} \boldsymbol{\epsilon}_{i2}^{\,2} \to \min_{\mathbf{w}}.$$

The specificity of this problem is that the size m of the time series set is quite large and that among the time series $S_x^1(t), \ldots, S_x^m(t)$ there are many highly correlated ones. Therefore, using the entire set to predict the time series $\mathbf{S}_y(t)$ leads to poor forecast quality.

One way to solve this problem is to select a fixed number of time series that have the greatest impact on the target variable using the CCM method. For a pair of time series

$$\left(S_x^i(t), S_y^j(t)\right) \quad i = 1, \ldots, m \quad j = 1, \ldots r$$

it determines the impact measure of the time series $S_x^i(t)$ on the target variable $S_y^j(t)$. Next, select the time series from the set with the maximum impact measure.

2.1 CCM method

Let's define the trajectory matrix of the time series $\mathbf{x} = [x_1, \ldots, x_n]$ as follows:

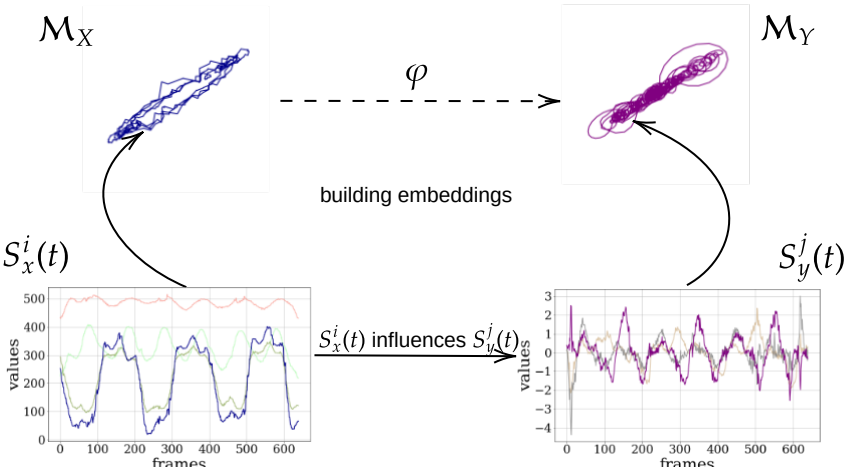


Fig. 1: Application of the CCM method to select the most significant components of a time series

$$\mathbf{H_x} = \begin{bmatrix} x_1 & x_2 & \dots & x_\tau \\ x_2 & x_3 & \dots & x_{\tau+1} \\ \vdots & \vdots & \ddots & \vdots \\ x_N & x_{N+1} & \dots & x_n \end{bmatrix},$$

where N is the number of delays, $\tau = n - N + 1$.

Denote the i-th column of the matrix $\mathbf{H_x}$ by $\mathbf{x}_i$. The matrix $\mathbf{H_x}$ takes the form:

$$\mathbf{H_x} = [\mathbf{x}_1, \dots, \mathbf{x}_\tau], \qquad \mathbf{x}_i = [x_i, x_{i+1}, \dots, x_{i+N-1}]^\mathsf{T}$$

Note that all vectors $\mathbf{x}_t$ belong to the $N-$ dimensional trajectory space $\mathbb{H}_\mathbf{x} \subseteq \mathbb{R}^N$ of the time series $\mathbf{x}$ and form the phase trajectory $\mathbf{x}(t) \in \mathbb{R}^N$.

To detect the relationship between the time series $\mathbf{x}$ and $\mathbf{y}$, take the element $\mathbf{x}_0$ from the trajectory space $\mathbb{H}_\mathbf{x}$ and find k nearest neighbors in the same space. Let's denote their time indices (from near to far) by $t_1, \dots, t_k$.

Since both time series are defined on the same timeline, then we can uniquely obtain the value of the time series $\mathbf{y}$ at time point $t_0 \in \{1, \dots, n\}$ by the value of the time series $\mathbf{x}$ and vice versa. Let's introduce the mapping from $\mathbb{H}_\mathbf{x}$ to $\mathbb{H}_\mathbf{y}$ as follows:

$$\phi : \mathbf{x}_0 \mapsto \widehat{\mathbf{y}_0} = \sum_{i=1}^{k} w_i \mathbf{y}_{t_i}, \qquad w_i = \frac{u_i}{\sum\limits_{j=1}^{k} u_j}, \qquad u_i = \exp\bigl(-\mathbf{x}_0 - \mathbf{x}_{t_i}\bigr).$$

Definition 2 The time series $\mathbf{x}$ and $\mathbf{y}$ are called **linked** if the mapping ϕ is Lipschitz:

$$\rho_{\mathbb{H}_\mathbf{y}}(\phi(\mathbf{x}_i), \phi(\mathbf{x}_j)) \leq C\rho_{\mathbb{H}_\mathbf{x}}(\mathbf{x}_i, \mathbf{x}_j) \qquad \mathbf{x}_i, \mathbf{x}_j \in \mathbb{H}_\mathbf{x}.$$

We introduce a metric proximity function of vectors in the vicinity of $U_k(\mathbf{x}_{t_0})$ and $U_k(\mathbf{y}_{t_0})$ to check for connectivity:

$$L(\mathbf{x}, \mathbf{y}) = \frac{R(U_k(\mathbf{x}_{t_0}))}{R(U_k(\mathbf{y}_{t_0}))}, \qquad R(U_k(\mathbf{x}_{t_0})) = \frac{1}{k}\sum_{i=1}^{k} \rho_{\mathbb{H}_\mathbf{x}}(\mathbf{x}_{t_0}, \mathbf{x}_{t_j}). \tag{1}$$

If $L(x, y)$ is greater than the specified threshold $C(n)$, then the time series $\mathbf{y}$ depends on the time series $\mathbf{x}$.

2.2 PLS method

Another way to solve above stated problem is to reduce the dimension in a consistent manner. The partial least squares method restores the relationship between the datasets $\mathbf{X}$ and $\mathbf{Y}$. The object matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$ are projected onto the latent space $\mathbb{R}^l$ of smaller dimension as follows:

$$\underset{n\times d}{\mathbf{X}} = \underset{n\times K}{\mathbf{T}} \cdot \underset{K\times d}{\mathbf{P}^\mathsf{T}} + \underset{n\times d}{\mathbf{E}}$$

$$\underset{n\times s}{\mathbf{Y}} = \underset{n\times K}{\mathbf{U}} \cdot \underset{K\times s}{\mathbf{Q}^\mathsf{T}} + \underset{n\times s}{\mathbf{F}},$$

where $\mathbf{T}$ and $\mathbf{U}$ — matrices describing objects and targets in the latent space, $\mathbf{P}$ and $\mathbf{Q}$ — transition matrices from the latent space to the original, $\mathbf{E}$ and $\mathbf{F}$ — remainder matrices.

The source data transformation function has the form:

$$f(\mathbf{X}) = \mathbf{X}\mathbf{W}_\mathbf{x} \qquad g(\mathbf{Y}) = \mathbf{Y}\mathbf{W}_\mathbf{y},$$

where the weight matrices $\mathbf{W}_\mathbf{x} \in \mathbb{R}^{d\times K}$ and $\mathbf{W}_\mathbf{y} \in \mathbb{R}^{s\times K}$ are found by maximizing the sample covariance:

$$(\mathbf{W}_\mathbf{x}, \mathbf{W}_\mathbf{y}) = \underset{\mathbf{W}_\mathbf{x}, \mathbf{W}_\mathbf{y}}{\operatorname{argmax}} \operatorname{Cov}(\mathbf{X}\mathbf{W}_\mathbf{x}, \mathbf{Y}\mathbf{W}_\mathbf{y})$$

The PLS algorithm works for previously column-normalized matrices $\mathbf{X}$ and $\mathbf{Y}$ and the number of components K as follows. Let's set $\mathbf{X}_1 = \mathbf{X}$, $\mathbf{Y}_1 = \mathbf{Y}$. Next, for each $k \in [1, K]$:

1. calculate $\mathbf{a}_k \in \mathbb{R}^d$ and $\mathbf{b}_k \in \mathbb{R}^s$, the first left and right singular vectors of the matrix $\mathbf{X}_k^T\mathbf{Y}_k$; from the definition it follows that $(\mathbf{a}_k, \mathbf{b}_k) = \underset{\mathbf{a},\mathbf{b}}{\operatorname{argmax}}\operatorname{Cov}(\mathbf{X}_k\mathbf{a}, \mathbf{Y}_k\mathbf{b})$.
2. project the matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ onto singular vectors: $\mathbf{t}_k = \mathbf{X}_k\mathbf{a}_k$, $\mathbf{u}_k = \mathbf{Y}_k\mathbf{b}_k$. Best rank-one approximation

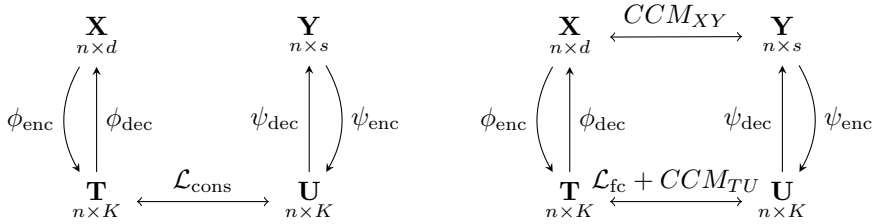


Fig. 2: Left: PLS-Autoencoder, right: PLS-CCM

3. regress the matrix $\mathbf{X}_k$ by the vector $\mathbf{t}_k$, that is, find the vector $\mathbf{p}_k$ such that the matrix $\mathbf{t}_k\mathbf{p}_k^\mathsf{T}$ is the best rank-one approximation of the matrix $\mathbf{X}_k$ by the Frobenius norm; do the same with the matrix $\mathbf{Y}_k$ and the vector $\mathbf{u}_k$ and get the vector $\mathbf{q}_k$.
4. subtract from the matrix $\mathbf{X}_k$ its rank-one approximation from the previous step, denote this matrix $\mathbf{X}_{k+1}$; similarly, get the matrix $\mathbf{Y}_{k+1}$.

You can get an explicit form of the matrices $\mathbf{W_x}$ and $\mathbf{W_y}$ from the PLS algorithm. Note that:

$$\mathbf{X} \cdot \mathbf{A}(\mathbf{P}^\mathsf{T}\mathbf{A})^{-1} = (\mathbf{T}\mathbf{P}^\mathsf{T}\mathbf{A} + \mathbf{E}\mathbf{A})(\mathbf{P}^\mathsf{T}\mathbf{A})^{-1} \approx \mathbf{T},$$

where the matrices $\mathbf{A}$, $\mathbf{P}$, $\mathbf{T}$ are formed from the columns $\mathbf{a}_k$, $\mathbf{p}_k$, $\mathbf{t}_k$ respectively. Similarly, $\mathbf{Y} \cdot \mathbf{B}(\mathbf{Q}^\mathsf{T}\mathbf{B})^{-1} \approx \mathbf{U}$, where the matrices $\mathbf{B}$, $\mathbf{Q}$, $\mathbf{U}$ are formed from the columns $\mathbf{b}_k$, $\mathbf{q}_k$, $\mathbf{u}_k$, respectively. Thus:

$$\mathbf{W_x} = \mathbf{A}(\mathbf{P}^\mathsf{T}\mathbf{A})^{-1}, \qquad \mathbf{W_y} = \mathbf{B}(\mathbf{Q}^\mathsf{T}\mathbf{B})^{-1}.$$

2.3 PLS-Autoencoder and PLS-CCM methods

The main disadvantage of the classical PLS method is the low quality when working with data that have complex nonlinear dependencies. For this reason, extensions of the linear PLS method, that transform input data using smooth nonlinear functions, have been developed .

One of such extensions is the PLS-Autoencoder method. Neural networks act as parametric functions that translate the source data into the latent space and vice versa. Multilayer perceptrons are used in this work.

The loss function of this model has the form:

$$\mathcal{L} = \lambda_1 \cdot \mathcal{L}^X_{\text{recov}}(\mathbf{X}, \hat{\mathbf{X}}) + \lambda_2 \cdot \mathcal{L}^Y_{\text{recov}}(\mathbf{Y}, \hat{\mathbf{Y}}) + \lambda_3 \cdot \mathcal{L}_{\text{cons}}(\mathbf{T}, \mathbf{U}), \qquad \lambda_1, \lambda_2, \lambda_3 > 0,$$

$$\mathcal{L}^X_{\text{recov}}(\mathbf{X}, \hat{\mathbf{X}}) = \mathbf{X} - \hat{\mathbf{X}}_2^2\,, \text{ where } \hat{\mathbf{X}} = \phi_{\text{dec}}(\phi_{\text{enc}}(\mathbf{X})),$$

$$\mathcal{L}^Y_{\text{recov}}(\mathbf{Y}, \hat{\mathbf{Y}}) = \mathbf{Y} - \hat{\mathbf{Y}}_2^2\,, \text{ where } \hat{\mathbf{Y}} = \psi_{\text{dec}}(\psi_{\text{enc}}(\mathbf{Y})),$$

$$\mathcal{L}_{\text{cons}}(\mathbf{T}, \mathbf{U}) = \frac{1}{1 + \left(\frac{1}{n}\,\text{tr}\left(\mathbf{U}^\mathsf{T}_{\text{centered}}\mathbf{T}_{\text{centered}}\right)\right)^2}$$

where $\mathcal{L}_{\text{recov}}$ is responsible for how accurately the original data is restored from their projections into the latent space, and $\mathcal{L}_{\text{cons}}$ is responsible for the connectivity of low-dimensional latent representations.

It is worth emphasizing that $\mathcal{L}_{\text{cons}}$ maximizes the sum square of the corresponding features covariance, which are columns of the matrices $\mathbf{T}$ and $\mathbf{U}$. Thus, this method does not take into account the consistency of objects in the latent space, that is, rows of matrices $\mathbf{T}$ and $\mathbf{U}$.

The new PLS-CCM method takes into account object consistency using metric functions from the CCM method. It is an extension of PLS-Autoencoder, only a new loss function is added:

$$\mathcal{L}_{\text{oc}}(\mathbf{X}, \mathbf{Y}, \mathbf{U}, \mathbf{T}) = (CCM_{XY} - CCM_{UT})^2,$$

where CCM_{XY} — the value characterizing the quality of approximation $\mathbf{y}_n$ using $\mathbf{x}_n$ constructed in the trajectory space consisting of the first $n-1$ objects, and CCM_{UT} — the same value, obtained from the matrices $\mathbf{U}$ and $\mathbf{T}$.

3 Computational experiment

The aim of the experiment is to compare different methods of consistent dimensionality reduction of spaces. These methods are used to predict the trajectory of the hand movement according to the corresponding video sequence. An important experiment part is the study of the results of the time series prediction model applied to the elements of phase trajectories space and to the elements of the trajectory subspace of a smaller dimension.

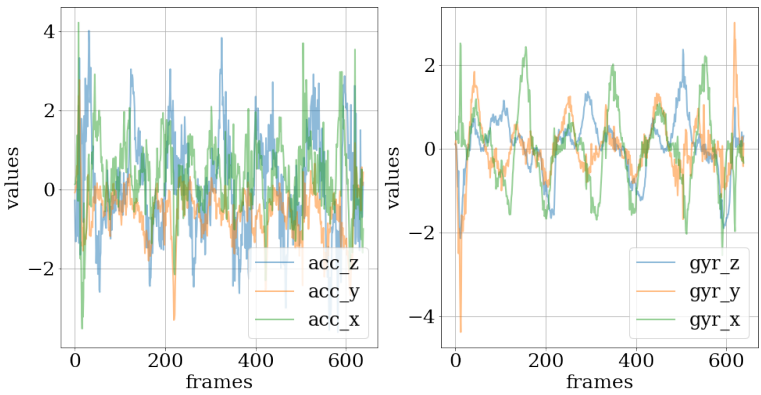
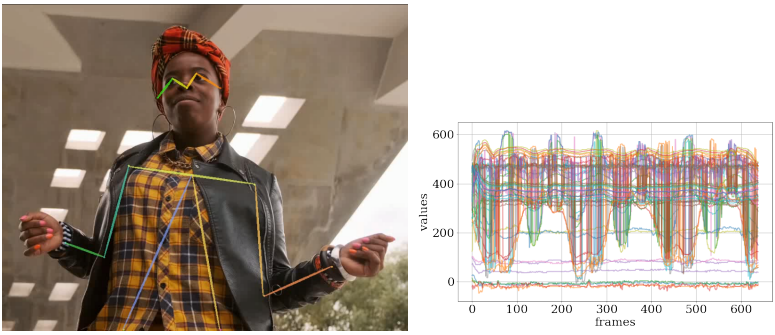


Fig. 3: Accelerometer and gyroscope data obtained by hand movement

The data is a set of videos on which various hand movements (cyclic and chaotic) are performed. The data also includes accelerometer and gyroscope

readings with a frequency of 100 Hertz, fixed on one of the hands. These devices form a 6-dimensional time series: The accelerometer and gyroscope show changes in values along the X, Y, and Z axes. Next, using the alphapose framework [**?**, **?**, **?**], the coordinates of the limbs, namely 68 key points, are extracted from the video sequence. As a result, we obtain a multidimensional time series of dimension 136, each component of which shows a change in one of the coordinates of some key point. Then highly correlated components are excluded from the resulting time series. After that, the resulting multidimensional time series are reduced to one time scale by removing elements of a longer time series.



(a) The result of the alphapose framework (b) Keypoints data obtained from the video

Fig. 4: Video processing scheme

4 Error analysis

Table 1: Comparison of the error (MSE) of the predictive model applied in the trajectory space and in its subspace obtained by CCM

	acc_z	acc_y	acc_x	gyr_z	gyr_y	gyr_x
space	1.053 ± 2.223	0.401 ± 0.833	0.483 ± 0.825	0.084 ± 0.537	0.090 ± 0.094	0.063 ± 0.295
subspace	0.315 ± 0.461	0.043 ± 0.051	0.150 ± 0.177	0.001 ± 0.001	0.015 ± 0.031	0.001 ± 0.003

To begin with, let's compare the predictions quality of the forecasting model applied in the trajectory space and its subspace. The table 1 shows the mean-square error of the predictions of the accelerometer and gyroscope values for each of the axes and their standard deviations. It shows that the predictive model applied in the trajectory subspace gives more accurate predictions, since

most of the features of the original space are uninformative and many of them are highly correlated.

Next, we will consider various methods of dimension reduction of the trajectory space. 5 methods were compared: CCM (K video features are selected that have the greatest impact on one of the accelerometer or gyroscope readings), PLS, CCA, PLS-AE, PLS-CCM. These methods are applied on two data sets: corresponding to cyclic and arbitrary hand movements. In the PLS-AE and PLS-CCM methods, a multilayer perceptron with the LeakyReLU activation function is taken as encoding and recovery functions,

Table 2: The standard deviation between the true readings of the devices and the predictions obtained with one of the dimensionality reduction methods

	Target feature \ Method	CCM	PLS	CCA	PLS-AE	PLS-CCM
cyclic	acc_z	0.400	0.067	0.146	**0.065**	0.097
	acc_y	**0.011**	0.045	0.219	0.067	0.066
	acc_x	0.056	0.073	0.092	**0.045**	0.054
	gyr_z	**0.001**	0.034	0.105	0.031	0.018
	gyr_y	**0.002**	0.023	0.010	0.024	0.070
	gyr_x	0.027	0.045	0.196	0.011	**0.009**
chaotic	acc_z	1.015	**0.256**	0.405	0.357	0.325
	acc_y	0.547	0.075	**0.036**	0.155	0.156
	acc_x	0.568	0.382	0.628	0.364	**0.324**
	gyr_z	0.099	0.066	**0.021**	0.259	0.127
	gyr_y	0.263	0.032	**0.028**	0.103	0.172
	gyr_x	0.074	**0.039**	0.055	0.129	0.298

5 Conclusion

The paper proposes a method for generalizing the PLS and CCA methods using the Sugihara method by constructing embeddings and choosing a metric for evaluating the quality of approximation. A computational experiment was carried out on the data of devices and video series. It was found that the using data from the video improves the forecasting quality. It is shown that the predictive model is less stable when it is applied in the trajectory space.

In the future, it is planned to apply the method not to two-dimensional data that correspond to regular measurements of a certain value, but to sporadic time series. This means that the input data will be multi-index matrices.