Dimensionality reduction for multivariate signal decoding problem

Roman Isachenko

Submitted to the Skolkovo Institute of Science and Technology on 01 June 2018

ABSTRACT:

This study is devoted to the problem of signal decoding for Brain Computer Interface (BCI) modelling. BCI helps disabled people to recover their mobility. The goal of the research is to build a model which predicts a limb position by brain signals. The challenge of the investigation is redundancy in data description. High correlation in measurements leads to correlation in input space. Additionally, the target space are correlated due to dependent consequent hand positions.

To overcome correlations in feature representation, feature selection are used. However, the majority of feature selection methods ignore the dependencies in the target space. We suggest a novel approach to feature selection in multivariate regression. To take into account the correlations in the target matrix, the proposed approach extend the ideas of the quadratic programming feature selection (QPFS) algorithm. The QPFS algorithm select non-correlated features, which are relevant to the targets. The proposed methods weigh the targets by their importances.

The computational experiment was carried out on the real public electrocorticogram (ECOG) dataset. The proposed algorithms show high performance compared to the known strategy. We analyze the robustness of the predictive model and the stability of the selected feature sets.

Skoltech Research Advisor:
Name: Maxim Fedorov
Degree: Professor
Title: Director, CREI CDISE

MIPT Research Advisor:
Name: Vadim Strijov
Degree:TO DO
Title: TO DO

Contents

1. INTRODUCTION

The research investigates the problem of signal decoding for Brain Computer Interface (BCI) [1]. BCI aims to develop systems that help people with a severe motor control disability to recover mobility. The minimally-invasive implant records cortical signals and the model decodes them on real time to predict the coordinates of an exoskeleton limbs [2, 3]. The subject placed inside the exoskeleton can drive it by imagining movements as if they were making the movement by themselves.

The challenge to build such model is redundancy in initial data description. The features are highly multicorrelated due to spatial nature of the data. The brain sensors are close to each other. It leads to the redundant measurements and instability of the final model. In addition, the redundant data description requires excess computations which lead to real-time delay. To overcome this problem dimensionality reduction [4,5] or feature selection [6,7] methods are used.

The dimensionality reduction algorithms find the optimal combinations of the initial features and use these combinations as the model features. For ECoG-based data the most suitable dimensionality reduction algorithm is partial least squares (PLS) [8–10]. The algorithm projects the features and the targets onto the joint latent space and maximizes the covariances between projected vectors. It allows to save information about initial input and target matrices and find their relations. The dimensionality of latent space is much less than the size of initial data description. It leads to stable linear model built at the small number of features. The overview of recent advances in PLS algorihms is given in [11, 12]. In this case we obtain the linear model with small latent dimension. However, the final model use the whole range of the initial features and it does not allow to detect useless features.

Feature selection is a special case of dimensionality reduction when the latent representation is choosen as a subset of initial data description. Here the model are built on the subset of the features. One of the approach to feature selection is to maximize feature relevances and minimize pairwise feature redundancy. This approach was recently proposed and investigated in [13, 14]. Quadratic programmic feature selection (QPFS) [15] uses this approach to construct the optimization problem. It was shown in [16] that QPFS algorithm outperforms many existing feature selection methods for the univariate regression problem. The QPFS algorithm introduces two functions: Sim and Rel. Sim estimates the redundancy between features, Rel contains relevances between each feature and the target vector. QPFS minimizes the function Sim and maximizes the function Rel simultaneously. The algorithm solves the following optimization problem

$$(1-\alpha)\cdot\underbrace{\mathbf{z}^{\mathsf{T}}\mathbf{Q}\mathbf{z}}_{\mathrm{Sim}(\mathbf{X})}-\alpha\cdot\underbrace{\mathbf{b}^{\mathsf{T}}\mathbf{z}}_{\mathrm{Rel}(\mathbf{X},\boldsymbol{\nu})}\to\min_{\substack{\mathbf{z}\geq\mathbf{0}_n\\ \mathbf{1}_n^{\mathsf{T}}\mathbf{z}=1}}. \tag{1}$$

The matrix $\mathbf{Q}\in\mathbb{R}^{n\times n}$ entries measure the pairwise similarities between features. The vector $\mathbf{b}\in\mathbb{R}^{n}$ expresses the similarities between each feature and the target vector. The normalized vector $\mathbf{z}$ shows the importance of each feature. The function (1) penalizes the dependent features by the function Sim and encourages features relevant to the target by the function Rel. The parameter α controls the trade-off between Sim and the Rel. To measure similarity the authors use the absolute value of sample correlation coefficient or sample mutual information coefficient between pairs of features for the function Sim, and between the features and the target vector for the function Rel.

The paper [MOTRENKO] proposes a multi-way version of QPFS algorithm for tensor ECoG-based data. It was shown that QPFS is appropriate feature selection method for brain signal decoding problem. We consider the multivariate problem, where the dependent variable is a vector. It refers to the prediction of limb position for not just one timestamp, but for some period of time. The subsequent hand positions are correlated. It leads to correlations in the model targets. In this situation feature selection algorithms do not take into account these dependencies. Hence, the selected feature subset is not optimal. We propose methods to take into account the dependencies in both input and output spaces. It allows to get the stable model with fewer variables. We refer to the original QPFS algortihm as our baseline for the computational experiment.

The experiment was carried out in the ECoG data from the NeuroTycho project [1]. We compared the proposed methods for multivariate feature selection with the baseline strategy and with PLS algporithm. The stability of the proposed methods were investigated. The proposed algorithms outperform the baseline algorithm and select the less number number of the features. [EXTEND].

[1] http://neurotycho.org/food-tracking-task

2. PROBLEM STATEMENT

2.1. Multivariate regression

The goal is to forecast a dependent variable $\mathbf{y} \in \mathbb{R}^r$ with r targets from an independent input object $\mathbf{x} \in \mathbb{R}^n$ with n features. We assume there is a linear dependence

$$\mathbf{y} = \mathbf{\Theta}\mathbf{x} + \boldsymbol{\varepsilon} \tag{2}$$

between the object $\mathbf{x}$ and the target variable $\mathbf{y}$, where $\mathbf{\Theta} \in \mathbb{R}^{r \times n}$ is a matrix of model parameters, $\boldsymbol{\varepsilon} \in \mathbb{R}^r$ is a residual vector. One has to find the matrix of the model parameters $\mathbf{\Theta}$ given a dataset $(\mathbf{X}, \mathbf{Y})$, where $\mathbf{X} \in \mathbb{R}^{m \times n}$ is a design matrix, $\mathbf{Y} \in \mathbb{R}^{m \times r}$ is a target matrix

$$\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^{\mathsf{T}} = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n]; \quad \mathbf{Y} = [\mathbf{y}_1, \dots, \mathbf{y}_m]^{\mathsf{T}} = [\boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_r]. \tag{3}$$

The columns $\boldsymbol{\chi}_j$ of the matrix $\mathbf{X}$ respond to object features.

The optimal parameters are determined by minimization of an error function. Define the quadratic loss function:

$$\mathcal{L}(\mathbf{\Theta}|\mathbf{X}, \mathbf{Y}) = \left\| \underset{m \times r}{\mathbf{Y}} - \underset{m \times n}{\mathbf{X}} \cdot \underset{r \times n}{\mathbf{\Theta}}^{\mathsf{T}} \right\|_2^2 \to \min_{\mathbf{\Theta}}. \tag{4}$$

The solution of (4) is given by

$$\mathbf{\Theta} = \mathbf{Y}^{\mathsf{T}}\mathbf{X}(\mathbf{X}^{\mathsf{T}}\mathbf{X})^{-1}. \tag{5}$$

The linear dependent columns of the matrix $\mathbf{X}$ leads to an instable solution for the optimization problem (4). If there is a vector $\boldsymbol{\alpha} \neq \mathbf{0}_n$ such that $\mathbf{X}\boldsymbol{\alpha} = \mathbf{0}_m$, then adding the vector $\boldsymbol{\alpha}$ to any column of the matrix $\mathbf{\Theta}$ does not change the value of the loss function $\mathcal{L}(\mathbf{\Theta}|\mathbf{X}, \mathbf{Y})$. In this case the matrix $\mathbf{X}^{\mathsf{T}}\mathbf{X}$ is not invertible. To avoid the strong linear dependence, dimensionality reduction and feature selection techniques are used.

2.2. Dimensionality reduction

To eliminate the linear dependence and reduce the dimensionality of the input space, the principal components analysis (PCA) is widely used algorithm. The main disadvantage of the PCA method is its insensitivity to the interrelation between the features and the targets. The partial least squares algorithm projects the design matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$

to the latent space with low dimensionality ($l < r < n$). The PLS algorithm finds the latent space matrices $\mathbf{T}, \mathbf{U} \in \mathbb{R}^{m \times l}$ that best describe the original matrices $\mathbf{X}$ and $\mathbf{Y}$.

The design matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$ are projected into the latent space in the following way:

$$\underset{m\times n}{\mathbf{X}} = \underset{m\times l}{\mathbf{T}} \cdot \underset{l\times n}{\mathbf{P}^{\mathsf{T}}} + \underset{m\times n}{\mathbf{F}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{t}_k} \cdot \underset{1\times n}{\mathbf{p}_k^{\mathsf{T}}} + \underset{m\times n}{\mathbf{F}}, \tag{6}$$

$$\underset{m\times r}{\mathbf{Y}} = \underset{m\times l}{\mathbf{U}} \cdot \underset{l\times r}{\mathbf{Q}^{\mathsf{T}}} + \underset{m\times r}{\mathbf{E}} = \sum_{k=1}^{l} \underset{m\times 1}{\mathbf{u}_k} \cdot \underset{1\times r}{\mathbf{q}_k^{\mathsf{T}}} + \underset{m\times r}{\mathbf{E}}, \tag{7}$$

where $\mathbf{T}$ and $\mathbf{U}$ are scores matrices in the latent space; $\mathbf{P}$ and $\mathbf{Q}$ are loading matrices; $\mathbf{E}$, $\mathbf{F}$ are residual matrices. PLS maximizes the linear relation between columns of matrices $\mathbf{T}$ and $\mathbf{U}$

$$\mathbf{U} \approx \mathbf{TB}, \quad \mathbf{B} = \operatorname{diag}(\beta_k), \quad \beta_k = \mathbf{u}_k^{\mathsf{T}}\mathbf{t}_k/(\mathbf{t}_k^{\mathsf{T}}\mathbf{t}_k). \tag{8}$$

We use the PLS algorithm as the dimensionality reduction algorithm in this research. The theoretical explanation of the PLS algorithm are given in section 3.

2.3. Feature selection

Feature selection is a special case of dimensionality reduction, where the loading matrices $\mathbf{T}$ and $\mathbf{U}$ are the submtrices of the design matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$.

The feature selection goal is to find the boolean vector $\mathbf{a} = \{0,1\}^n$, which components indicate whether the feature are selected. To obtain the optimal vector $\mathbf{a}$ among all possible $2^n - 1$ options, introduce the feature selection error function $S(\mathbf{a}|\mathbf{X}, \mathbf{Y})$. We state the feature selection problem as follows

$$\mathbf{a} = \underset{\mathbf{a}' \in \{0,1\}^n}{\arg\min}\, S(\mathbf{a}'|\mathbf{X}, \mathbf{Y}). \tag{9}$$

The goal of feature selection is to construct the appropriate function $S(\mathbf{a}|\mathbf{X}, \mathbf{Y})$. The particular examples for the considered feature selection algorithms are given below and summarizes in the Table 1.

The problem 9 are hard to solve due to discrete binary domain $\{0,1\}^n$. We relax the problem 9 to the continuous domain $[0,1]^n$. The relaxed feature selection problem is

$$\mathbf{z} = \underset{\mathbf{z}' \in [0,1]^n}{\arg\min}\, S(\mathbf{z}'|\mathbf{X}, \mathbf{Y}). \tag{10}$$

Here vector $\mathbf{z}$ is a normalized feature scores. Firstly, solve the problem 10 to obtain the feature scores $\mathbf{z}$. Then the solution of 9 is recovered by thresholding:

$$\mathbf{a} = [a_j]_{j=1}^n, \quad a_j = \begin{cases} 1, & z_j > \tau; \\ 0, & \text{otherwise}. \end{cases} \tag{11}$$

Here the value τ is a hyperparameter.

Once the solution $\mathbf{a}$ of (9) is known, the problem (4) becomes

$$\mathcal{L}(\mathbf{\Theta}_\mathbf{a}|\mathbf{X}_\mathbf{a}, \mathbf{Y}) = \left\|\mathbf{Y} - \mathbf{X}_\mathbf{a}\mathbf{\Theta}_\mathbf{a}^\mathsf{T}\right\|_2^2 \to \min_{\mathbf{\Theta}_\mathbf{a}}, \tag{12}$$

where the subscript $\mathbf{a}$ indicates the submatrix with the columns for which components of $\mathbf{a}$ equal 1.

2.4. Quadratic Programming Feature Selection

Our base algorithm for feature selection is quadratic programming feature selection algorithm. The QPFS algorithm selects non-correlated features, which are relevant to the target vector $\boldsymbol{\nu}$ for the linear regression problem with $r = 1$

$$\|\boldsymbol{\nu} - \mathbf{X}\boldsymbol{\theta}\|_2^2 \to \min_{\boldsymbol{\theta} \in \mathbb{R}^n}. \tag{13}$$

The authors of the original QPFS paper suggested the way to select α for 1 and make $\text{Sim}(\mathbf{X})$ and $\text{Rel}(\mathbf{X}, \boldsymbol{\nu})$ impacts the same:

$$\alpha = \frac{\overline{\mathbf{Q}}}{\overline{\mathbf{Q}} + \overline{\mathbf{b}}}, \tag{14}$$

where $\overline{\mathbf{Q}}$, $\overline{\mathbf{b}}$ are the mean values of $\mathbf{Q}$ and $\mathbf{b}$ respectively. The QPFS parameters are defined as follows:

$$\mathbf{Q} = \left[\text{sim}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)\right]_{i,j=1}^n, \quad \mathbf{b} = [\text{sim}(\boldsymbol{\chi}_i, \boldsymbol{\nu})]_{i=1}^n. \tag{15}$$

Here the function $\text{sim}(\cdot,\cdot)$ is a similarity measure. The common ways to define this function are the absolute value of sample Pearson correlation coefficient

$$\text{sim}(\boldsymbol{\chi}, \boldsymbol{\nu}) = |\text{corr}(\boldsymbol{\chi}, \boldsymbol{\nu})| = \left| \frac{\sum_{i=1}^m (\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}})(\boldsymbol{\nu}_i - \overline{\boldsymbol{\nu}})}{\sqrt{\sum_{i=1}^m (\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}})^2 \sum_{i=1}^m (\boldsymbol{\nu}_i - \overline{\boldsymbol{\nu}})^2}} \right|, \tag{16}$$

or the sample mutual information coefficient

$$\mathrm{sim}(\boldsymbol{\chi}, \boldsymbol{\nu}) = I(\boldsymbol{\chi}, \boldsymbol{\nu}) = \int \int p(\boldsymbol{\chi}, \boldsymbol{\nu}) \log(\frac{p(\boldsymbol{\chi}, \boldsymbol{\nu})}{p(\boldsymbol{\chi})p(\boldsymbol{\nu})}) d\boldsymbol{\chi} d\boldsymbol{\nu}, \qquad (17)$$

We use the correlation coefficient (16) as a similarity measure $\mathrm{sim}(\cdot, \cdot)$. The other ways to define $\mathbf{Q}$ and $\mathbf{b}$ are considered in [16].

The problem (1) is convex if the matrix $\mathbf{Q}$ is positive semidefinite. In general it is not always true. To satisfy this condition, the matrix $\mathbf{Q}$ spectrum is shifted and the matrix $\mathbf{Q}$ is replaced by $\mathbf{Q} - \lambda_{\text{min}}\mathbf{I}$, where λ_{min} is a $\mathbf{Q}$ minimal eigenvalue. The original paper [15] suggests the way to solve the quadratic problem (1) efficiently. In [17] the sequential minimal optimization framework is proposed for solving (1).

3. PARTIAL LEAST SQUARES REGRESSION

The pseudocode of the PLS regression algorithm is given in the algorithm 1. In each of the l steps the algorithm iteratively calculates columns $\mathbf{t}_k$, $\mathbf{u}_k$, $\mathbf{p}_k$, $\mathbf{q}_k$ of the matrices $\mathbf{T}$, $\mathbf{U}$, $\mathbf{P}$, $\mathbf{Q}$, respectively. After the computation of the next set of vectors, the one-rank approximations are subtracted from the matrices $\mathbf{X}$, $\mathbf{Y}$. This step is called a matrix deflation. In the first step one has to normalize the columns of the original matrices (subtract the mean and divide by the standard deviation). During the test mode we need to normalize test data, compute the model prediction (2), and then perform the reverse normalization.

Algorithm 1 PLSR algorithm

```
Require: $\mathbf{X}, \mathbf{Y}, l$;
Ensure: $\mathbf{T}, \mathbf{P}, \mathbf{Q}$;
 1: normalize matrices $\mathbf{X}$ и $\mathbf{Y}$ by columns
 2: initialize $\mathbf{u}_0$ (the first column of $\mathbf{Y}$)
 3: $\mathbf{X}_1 = \mathbf{X}; \mathbf{Y}_1 = \mathbf{Y}$
 4: for $k = 1, \dots, l$ do
 5:    repeat
 6:       $\mathbf{w}_k := \mathbf{X}_k^{\mathsf{T}}\mathbf{u}_{k-1}/(\mathbf{u}_{k-1}^{\mathsf{T}}\mathbf{u}_{k-1}); \quad \mathbf{w}_k := \frac{\mathbf{w}_k}{\|\mathbf{w}_k\|}$
 7:       $\mathbf{t}_k := \mathbf{X}_k\mathbf{w}_k$
 8:       $\mathbf{c}_k := \mathbf{Y}_k^{\mathsf{T}}\mathbf{t}_k/(\mathbf{t}_k^{\mathsf{T}}\mathbf{t}_k); \quad \mathbf{c}_k := \frac{\mathbf{c}_k}{\|\mathbf{c}_k\|}$
 9:       $\mathbf{u}_k := \mathbf{Y}_k\mathbf{c}_k$
10:    until $\mathbf{t}_k$ stabilizes
11:    $\mathbf{p}_k := \mathbf{X}_k^{\mathsf{T}}\mathbf{t}_k/(\mathbf{t}_k^{\mathsf{T}}\mathbf{t}_k), \ \mathbf{q}_k := \mathbf{Y}_k^{\mathsf{T}}\mathbf{u}_k/(\mathbf{u}_k^{\mathsf{T}}\mathbf{u}_k)$
12:    $\mathbf{X}_{k+1} := \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^{\mathsf{T}}$
13:    $\mathbf{Y}_{k+1} := \mathbf{Y}_k - \mathbf{u}_k\mathbf{q}_k^{\mathsf{T}} \approx \mathbf{Y}_k - \mathbf{t}_k \cdot \left(\frac{\mathbf{Y}^{\mathsf{T}}\mathbf{t}_k}{\mathbf{t}_k^{\mathsf{T}}\mathbf{t}_k}\right)^{\mathsf{T}}$
```

The vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ from the inner loop of the algorithm 1 contain information about the design matrix $\mathbf{X}$ and the target matrix $\mathbf{Y}$, respectively. The blocks of steps (6)–(7) and (8)-(9) are analogues of the PCA algorithm for the matrices $\mathbf{X}$ and $\mathbf{Y}$ [18]. Sequential repetition of the blocks takes into account the interaction between the matrices $\mathbf{X}$ and $\mathbf{Y}$.

The theoretical explanation of the PLS algorithm follows from the statements.

Proposition 1. *The best description of the matrices $\mathbf{X}$ and $\mathbf{Y}$ taking into account their interrelation is achieved by maximization of the covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$.*

The statement follows from the equation

$$\operatorname{cov}(\mathbf{t}_k, \mathbf{u}_k) = \operatorname{corr}(\mathbf{t}_k, \mathbf{u}_k) \cdot \sqrt{\operatorname{var}(\mathbf{t}_k)} \cdot \sqrt{\operatorname{var}(\mathbf{u}_k)}. \tag{18}$$

Maximization of the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ variances corresponds to keeping information about original matrices, the correlation of these vectors corresponds to interrelation between $\mathbf{X}$ and $\mathbf{Y}$. ■

In the inner loop of the algorithm 1 the normalized weight vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ are calculated. These vectors construct the matrices $\mathbf{W}$ and $\mathbf{C}$, respectively.

Proposition 2. *The vector $\mathbf{w}_k$ and $\mathbf{c}_k$ are eigenvectors of the matrices $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$ and $\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k$, corresponding to the maximum eigenvalues.*

$$\mathbf{w}_k \propto \mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{t}_{k-1} \propto \mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_{k-1}, \tag{19}$$

$$\mathbf{c}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{t}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{u}_{k-1} \propto \mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{c}_{k-1}, \tag{20}$$

where the $\propto$ symbol means equality up to a multiplicative constant.

The statement follows from the fact that the update rule for vectors $\mathbf{w}_k$, $\mathbf{c}_k$ coincides with the iteration of the power method for the maximum eigenvalue.

Let a matrix $\mathbf{A}$ be diagonalizable, $\mathbf{x}$ be some vector, then

$$\lim_{k\to\infty} \mathbf{A}^k\mathbf{x} = \lambda_{\max}(\mathbf{A}) \cdot \mathbf{v}_{\max}, \tag{21}$$

where $\mathbf{v}_{\max}$ is the eigenvector $\mathbf{A}$, corresponding to the maximum eigenvalue $\lambda_{\max}(\mathbf{A})$. ■

Proposition 3. *The update rule for the vectors in steps (6)–(9) of the algorithm 1 corresponds to the maximization of the covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$.*

The maximum covariance between the vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ is equal to the maximum eigenvalue of the matrix $\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k$:

$$\begin{aligned}\max_{\mathbf{t}_k,\mathbf{u}_k} \operatorname{cov}(\mathbf{t}_k, \mathbf{u}_k)^2 = \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{X}_k\mathbf{w}_k, \mathbf{Y}_k\mathbf{c}_k\right)^2 = \max_{\substack{\|\mathbf{w}_k\|=1\\ \|\mathbf{c}_k\|=1}} \operatorname{cov}\left(\mathbf{c}_k^\mathsf{T}\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right)^2 = \\ = \max_{\|\mathbf{w}_k\|=1} \operatorname{cov}\left\|\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k\right\|^2 = \max_{\|\mathbf{w}_k\|=1} \mathbf{w}_k^\mathsf{T}\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\mathbf{w}_k = \\ = \lambda_{\max}\left(\mathbf{X}_k^\mathsf{T}\mathbf{Y}_k\mathbf{Y}_k^\mathsf{T}\mathbf{X}_k\right),\end{aligned} \tag{22}$$

where $\lambda_{\max}(\mathbf{A})$ is the maximum eigenvalue of the matrix $\mathbf{A}$. Using the statement 2, we obtain the required result. ■

After the inner loop the following step (11) is to compute vectors $\mathbf{p}_k$, $\mathbf{q}_k$ by projection of the matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ columns to the vector $\mathbf{t}_k$. Before proceeding with the next iteration one has to deflate the matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ by the one-rank approximations $\mathbf{t}_k\mathbf{p}_k^{\mathsf{T}}$ and $\mathbf{t}_k\mathbf{q}_k^{\mathsf{T}}$

$$\mathbf{X}_{k+1} = \mathbf{X}_k - \mathbf{t}_k\mathbf{p}_k^{\mathsf{T}} = \mathbf{X} - \sum_k \mathbf{t}_k\mathbf{p}_k^{\mathsf{T}}, \tag{23}$$

$$\mathbf{Y}_{k+1} = \mathbf{Y}_k - \mathbf{t}_k\mathbf{q}_k^{\mathsf{T}} = \mathbf{Y} - \sum_k \mathbf{t}_k\mathbf{q}_k^{\mathsf{T}}. \tag{24}$$

Each next vector $\mathbf{t}_{k+1}$ turns out to be orthogonal to all vectors $\mathbf{t}_i$, $i = 1, \dots, k$.

Let assume that the dimension of the input, the target, and the latent spaces are equal to 2 ($n = r = l = 2$). Figure 1 shows the result of the PLS algorithm in this case. Blue and green dots represent the rows of the matrices $\mathbf{X}$ and $\mathbf{Y}$, respectively. The dots were generated from a normal distribution with zero expectation. Contours of the distribution covariance matrices are shown in red. Black contours are unit circles. Red arrows correspond to principal components for the set of points. Black arrows correspond to the vectors of the matrices $\mathbf{W}$ and $\mathbf{C}$ from the PLS algorithm. The vectors $\mathbf{t}_k$ and $\mathbf{u}_k$ are equal to the projected matrices $\mathbf{X}_k$ and $\mathbf{Y}_k$ to the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$, respectively, and are denoted by black pluses. Taking into account the interaction between the matrices $\mathbf{X}$ and $\mathbf{Y}$ the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ deviate from the principal components directions. The deviation of the vectors $\mathbf{w}_k$ is insignificant. In the first iteration, $\mathbf{c}_1$ is close to the principal component pc_1, but the vectors $\mathbf{c}_k$ in the next iterations could strongly correlate. The difference in the vectors $\mathbf{w}_k$ and $\mathbf{c}_k$ the behaviour is associated with the deflation process. In particular, we subtract from $\mathbf{Y}$ the one-rank approximation found in the space of the design matrix $\mathbf{X}$.

To obtain the model prediction and find the model parameters, multiply the both hand sides of the (6) by the matrix $\mathbf{W}$. Since the residual matrix $\mathbf{E}$ rows are orthogonal to the columns of the matrix $\mathbf{W}$, we have

$$\mathbf{X}\mathbf{W} = \mathbf{T}\mathbf{P}^{\mathsf{T}}\mathbf{W}. \tag{25}$$

The linear transformation between objects in the input and latent spaces has the form

$$\mathbf{T} = \mathbf{X}\mathbf{W}^*, \quad \text{where } \mathbf{W}^* = \mathbf{W}(\mathbf{P}^{\mathsf{T}}\mathbf{W})^{-1}. \tag{26}$$

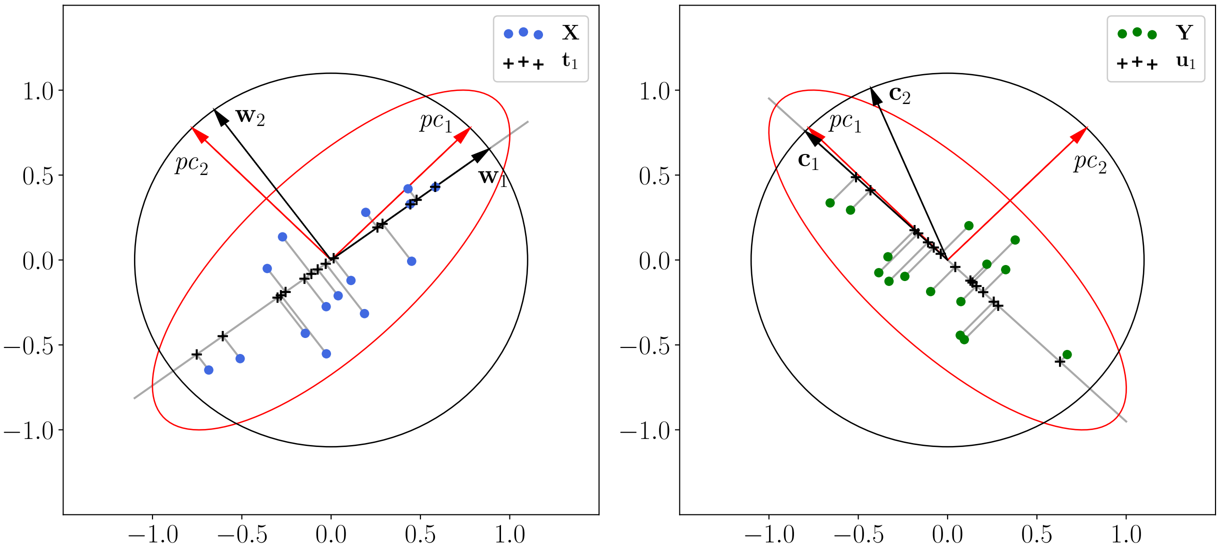


Figure 1: The result of the PLS algorithm for the case $n = r = l = 2$

.

The matrix of the model parameters 2 could be found from equations (7), (26)

$$\mathbf{Y} = \mathbf{U}\mathbf{Q}^\mathsf{T} + \mathbf{E} \approx \mathbf{T}\mathbf{B}\mathbf{Q}^\mathsf{T} + \mathbf{E} = \mathbf{X}\mathbf{W}^*\mathbf{B}\mathbf{Q}^\mathsf{T} + \mathbf{E} = \mathbf{X}\boldsymbol{\Theta} + \mathbf{E}. \tag{27}$$

Thus, the model parameters (2) are equal to

$$\boldsymbol{\Theta} = \mathbf{W}(\mathbf{P}^\mathsf{T}\mathbf{W})^{-1}\mathbf{B}\mathbf{Q}^\mathsf{T}. \tag{28}$$

The final model (27) is a linear model which are low-dimensional in the latent space. It reduces the data redundancy and improves the model stability.

4. MULTIVARIATE QPFS

We are aimed to propose the algorithms which suitable for feature selection in multivariate case. If the target space is multidimensional it prone to redundancy and correlations between the targets. In this section we consider the algorithms that take into account the probable dependencies in both input and target spaces.

4.1. Relevance aggregation (RelAgg).

First approach to apply the QPFS algorithm to the multivariate case ($r > 1$) is to aggregate feature relevances through all r components. The term $\text{Sim}(\mathbf{X})$ is still the same, the matrix $\mathbf{Q}$ is defined by (15). The vector $\mathbf{b}$ is aggregated across all targets and is defined by

$$\mathbf{b} = \left[\sum_{k=1}^{r} |\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_k)| \right]_{i=1}^{n}. \tag{29}$$

The drawback of this approach is that it does not use the dependencies in the columns of the matrix $\mathbf{Y}$. Observe the following example:

$$\mathbf{X} = [\boldsymbol{\chi}_1, \boldsymbol{\chi}_2, \boldsymbol{\chi}_3], \quad \mathbf{Y} = [\underbrace{\boldsymbol{\nu}_1, \boldsymbol{\nu}_1, \dots, \boldsymbol{\nu}_1}_{r-1}, \boldsymbol{\nu}_2], \tag{30}$$

We have three features and r targets, where first $r-1$ target are identical. The pairwise features similarities are given by the matrix $\mathbf{Q}$. The matrix $\mathbf{B}$ entries show pairwise features relevances to the targets. The vector $\mathbf{b}$ is obtained by summation of the matrix $\mathbf{B}$ over columns.

$$\mathbf{Q} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0.8 \\ 0 & 0.8 & 1 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0.4 & \dots & 0.4 & 0 \\ 0.5 & \dots & 0.5 & 0.8 \\ \underbrace{0.8 \quad \dots \quad 0.8}_{r-1} & & & 0.1 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} (r-1) \cdot 0.4 + 0 \\ (r-1) \cdot 0.5 + 0.8 \\ (r-1) \cdot 0.8 + 0.1. \end{bmatrix} \tag{31}$$

We would like to select only two features. For such configuration the best feature subset is $[\boldsymbol{\chi}_1, \boldsymbol{\chi}_2]$. The feature $\boldsymbol{\chi}_2$ predicts the second target $\boldsymbol{\nu}_2$ and the combination of features $\boldsymbol{\chi}_1, \boldsymbol{\chi}_2$ predicts the first component. The QPFS algorithm for $r = 2$ gives the solution $\mathbf{z} = [0.37, 0.61, 0.02]$. It coincides with our knowledge. However, if we add the collinear columns to the matrix $\mathbf{Y}$ and increase r to 5, the QPFS solution will be $\mathbf{z} = [0.40, 0.17, 0.43]$.

Here we lose the relevant feature $\boldsymbol{\chi}_2$ and select the redundant feature $\boldsymbol{\chi}_3$.

4.2. Symmetric importances (SymImp).

To take into account the dependencies in the columns of the matrix $\mathbf{Y}$ we extend the QPFS function (1) to the multivariate case. We add the term $\text{Sim}(\mathbf{Y})$ and modify the term $\text{Rel}(\mathbf{X}, \mathbf{Y})$:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y}_{\text{Rel}(\mathbf{X},\mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^{\mathsf{T}} \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \rightarrow \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n,\, \mathbf{1}_n^{\mathsf{T}} \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r,\, \mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y = 1}}. \tag{32}$$

Determine the entries of matrices $\mathbf{Q}_x \in \mathbb{R}^{n \times n}$, $\mathbf{Q}_y \in \mathbb{R}^{r \times r}$, $\mathbf{B} \in \mathbb{R}^{n \times r}$ in the following way

$$\mathbf{Q}_x = \left[\left| \text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) \right| \right]_{i,j=1}^{n}, \quad \mathbf{Q}_y = \left[\left| \text{corr}(\boldsymbol{\nu}_i, \boldsymbol{\nu}_j) \right| \right]_{i,j=1}^{r}, \quad \mathbf{B} = \left[\left| \text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_j) \right| \right]_{\substack{i=1,\dots,n \\ j=1,\dots,r}}. \tag{33}$$

The vector $\mathbf{z}_x$ shows the features importances, while $\mathbf{z}_y$ is a vector with the targets importances. The correlated targets will be penalized by $\text{Sim}(\mathbf{Y})$ and have the lower importances.

The coefficients α_1, α_2, and α_3 control the influence of each term on the function (57) and satisfy the conditions:

$$\alpha_1 + \alpha_2 + \alpha_3 = 1, \quad \alpha_i \geq 0,\, i = 1, 2, 3. \tag{34}$$

Proposition 4. *The balance between the terms Sim*$(\mathbf{X})$*, Rel*$(\mathbf{X}, \mathbf{Y})$*, and Rel*$(\mathbf{X}, \mathbf{Y})$ *for the problem* (57) *is achieved by the following coefficients:*

$$\alpha_1 = \frac{\overline{\mathbf{Q}}_y \overline{\mathbf{B}}}{\overline{\mathbf{Q}}_y \overline{\mathbf{B}} + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}; \tag{35}$$

$$\alpha_2 = \frac{\overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y}{\overline{\mathbf{Q}}_y \overline{\mathbf{B}} + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}; \tag{36}$$

$$\alpha_3 = \frac{\overline{\mathbf{Q}}_x \overline{\mathbf{B}}}{\overline{\mathbf{Q}}_y \overline{\mathbf{B}} + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}. \tag{37}$$

Here $\overline{\mathbf{Q}}_x$*,* $\overline{\mathbf{B}}$*,* $\overline{\mathbf{Q}}_y$ *are mean values of* $\mathbf{Q}_x$*,* $\mathbf{B}$*, and* $\mathbf{Q}_y$*, respectively.*

Proof. The desired values of α_1, α_2, and α_3 are given by solution of the following equations

$$\alpha_1 + \alpha_2 + \alpha_3 = 1; \tag{38}$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}} = \alpha_3 \overline{\mathbf{Q}}_y. \tag{39}$$

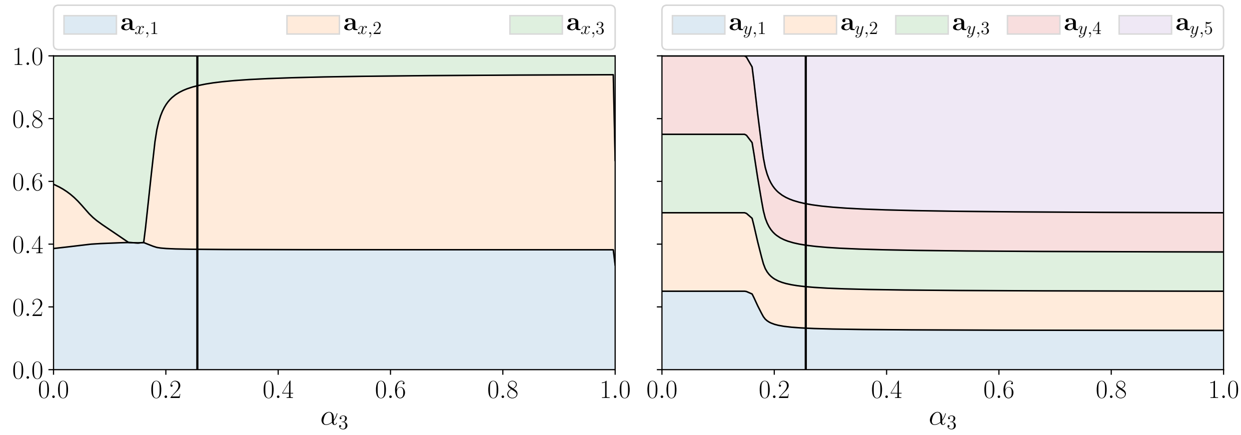


Figure 2: Feature importances $\mathbf{z}_x$ and $\mathbf{z}_y$ with respect to the α_3 coefficient

Here, the mean values $\overline{\mathbf{Q}}_x$, $\overline{\mathbf{B}}$, $\overline{\mathbf{Q}}_y$ of the corresponding matrices $\mathbf{Q}_x$, $\mathbf{B}$, and $\mathbf{Q}_y$ are the mean values of the terms $\text{Sim}(\mathbf{X})$, $\text{Rel}(\mathbf{X}, \mathbf{Y})$, and $\text{Rel}(\mathbf{X}, \mathbf{Y})$. $\square$

To investigate the impact of the term $\text{Sim}(\mathbf{Y})$ on the function (57), we balance the terms $\text{Sim}(\mathbf{X})$ and $\text{Rel}(\mathbf{X}, \mathbf{Y})$ by fixing the proportion between α_1 and α_2:

$$\alpha_1 = \frac{(1-\alpha_3)\overline{\mathbf{B}}}{\overline{\mathbf{Q}}_x + \overline{\mathbf{B}}}; \quad \alpha_2 = \frac{(1-\alpha_3)\overline{\mathbf{Q}}_x}{\overline{\mathbf{Q}}_x + \overline{\mathbf{B}}}; \quad \alpha_3 \in [0, 1]. \tag{40}$$

We apply the proposed algorithm to the discussed example (31). The given matrix $\mathbf{Q}$ corresponds to the matrix $\mathbf{Q}_x$. We additionally define the matrix $\mathbf{Q}_y$ by setting $\text{corr}(\boldsymbol{\nu}_1, \boldsymbol{\nu}_2) = 0.2$ and all others entries to one. Figure 2 shows the importances of features $\mathbf{z}_x$ and targets $\mathbf{z}_y$ with respect to α_3 coefficient. If α_3 is small, the impact of all targets are almost identical and the feature $\boldsymbol{\chi}_3$ dominates the feature $\boldsymbol{\chi}_2$. When α_3 becomes larger than 0.2, the importance $(\mathbf{z}_y)_5$ of the target ϕ_5 grows up along with the importance of the feature $\boldsymbol{\chi}_2$.

4.3. Minimax QPFS (MinMax and MaxMin).

The function (57) is symmetric with respect to $\mathbf{z}_x$ and $\mathbf{z}_y$. It penalizes the features that are correlated and are not relevant to the targets. At the same time it penalizes the targets that are correlated and are not sufficiently explained by the features. It leads to small importances for the targets which are difficult to predict by the features and large importances for the targets which are strongly correlated with the features. It contradicts with the intuition. Our goal is to predict all targets, especially which are difficult to explain, by selected relevant and non-correlated the features. We express this into two related problems.

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} \to \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n,\\ \mathbf{1}_n^{\mathsf{T}}\mathbf{z}_x=1}}; \tag{41}$$

$$\alpha_3 \cdot \underbrace{\mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})} + \alpha_2 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} \to \min_{\substack{\mathbf{z}_y \geq \mathbf{0}_r,\\ \mathbf{1}_r^{\mathsf{T}}\mathbf{z}_y=1}}. \tag{42}$$

The difference is in the term Rel. In feature space the non-relevant components should have smaller scores. Meanwhile, the targets that are not relevant to the features should have larger scores. The problems (41) and (42) are merged into the joint min-max or max-min formulation

$$\min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n\\ \mathbf{1}_n^{\mathsf{T}}\mathbf{z}_x=1}} \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r\\ \mathbf{1}_r^{\mathsf{T}}\mathbf{z}_y=1}} f(\mathbf{z}_x, \mathbf{z}_y), \quad \left(\text{or} \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r\\ \mathbf{1}_r^{\mathsf{T}}\mathbf{z}_y=1}} \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n\\ \mathbf{1}_n^{\mathsf{T}}\mathbf{z}_x=1}} f(\mathbf{z}_x, \mathbf{z}_y) \right), \tag{43}$$

where

$$f(\mathbf{z}_x, \mathbf{z}_y) = \alpha_1 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x}_{\mathrm{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y}_{\mathrm{Rel}(\mathbf{X},\mathbf{Y})} - \alpha_3 \cdot \underbrace{\mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y}_{\mathrm{Sim}(\mathbf{Y})}. \tag{44}$$

Theorem 1. *For positive definite matrices $\mathbf{Q}_x$ and $\mathbf{Q}_y$ the max-min and min-max problems* (43) *have the same optimal value.*

Proof. Denote

$$\mathbb{C}^n = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_n, \, \mathbf{1}_n^{\mathsf{T}}\mathbf{z} = 1\}, \quad \mathbb{C}^r = \{\mathbf{z} : \mathbf{z} \geq \mathbf{0}_r, \, \mathbf{1}_r^{\mathsf{T}}\mathbf{z} = 1\}. \tag{45}$$

The sets $\mathbb{C}^n$ and $\mathbb{C}^r$ are compact and convex. The function $f : \mathbb{C}^n \times \mathbb{C}^r \to \mathbb{R}$ is a continuous function. If $\mathbf{Q}_x$ and $\mathbf{Q}_y$ are positive definite matrices, the function f is convex-concave, i.e. $f(\cdot, \mathbf{z}_y) : \mathbb{C}^n \to \mathbb{R}$ is convex for fixed $\mathbf{z}_y$, and $f(\mathbf{z}_x, \cdot) : \mathbb{C}^r \to \mathbb{R}$ is concave for fixed $\mathbf{z}_x$. In this case Neumann's minimax theorem states

$$\min_{\mathbf{z}_x \in \mathbb{C}^n} \max_{\mathbf{z}_y \in \mathbb{C}^r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\mathbf{z}_y \in \mathbb{C}^r} \min_{\mathbf{z}_x \in \mathbb{C}^n} f(\mathbf{z}_x, \mathbf{z}_y). \tag{46}$$

□

To solve the min-max problem (43), fix some $\mathbf{z}_x \in \mathbb{C}^n$. For fixed vector $\mathbf{z}_x$ we solve the problem

$$\max_{\mathbf{z}_y \in \mathbb{C}_r} f(\mathbf{z}_x, \mathbf{z}_y) = \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r\\ \mathbf{1}_r^{\mathsf{T}}\mathbf{z}_y=1}} \left[\alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y\right]. \tag{47}$$

The Lagrangian for this problem is

$$L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^{\mathsf{T}} \mathbf{Q}_y \mathbf{z}_y + \lambda \cdot (\mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y - 1) + \boldsymbol{\mu}^{\mathsf{T}} \mathbf{z}_y. \tag{48}$$

Here the Lagrange multipliers $\boldsymbol{\mu}$, corresponding to the inequality constraints $\mathbf{z}_y \geq \mathbf{0}_r$, are restricted to be non-negative. The dual problem is

$$\min_{\lambda,\, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \min_{\lambda,\, \boldsymbol{\mu} \geq \mathbf{0}_r} \left[\max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) \right]. \tag{49}$$

The strong duality holds for (47). Therefore, the optimal value for (47) equals the optimal value for (49). It allows to solve the problem

$$\min_{\mathbf{z}_x \in \mathbb{C}^n,\, \lambda,\, \boldsymbol{\mu} \geq \mathbf{0}_r} g(\mathbf{z}_y, \lambda, \boldsymbol{\mu}) \tag{50}$$

instead of (43).

Setting the gradient of the Langrangian $\nabla_{\mathbf{z}_y} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})$ to zero, we obtain an optimal value $\mathbf{z}_y$:

$$\mathbf{z}_y = \frac{1}{2\alpha_3} \mathbf{Q}_y^{-1} \left(-\alpha_2 \cdot \mathbf{B}^{\mathsf{T}} \mathbf{z}_x + \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu} \right). \tag{51}$$

The dual function is equal to

$$\begin{aligned} g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \max_{\mathbf{z}_y \in \mathbb{R}^r} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = \mathbf{z}_x^{\mathsf{T}} \left(-\frac{\alpha_2^2}{4\alpha_3} \cdot \mathbf{B} \mathbf{Q}_y^{-1} \mathbf{B}^{\mathsf{T}} - \alpha_1 \cdot \mathbf{Q}_x \right) \mathbf{z}_x \\ - \frac{1}{4\alpha_3} \lambda^2 \cdot \mathbf{1}_r^{\mathsf{T}} \mathbf{Q}_y^{-1} \mathbf{1}_r - \frac{1}{4\alpha_3} \cdot \boldsymbol{\mu}^{\mathsf{T}} \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \lambda \cdot \mathbf{1}_r^{\mathsf{T}} \mathbf{Q}_y^{-1} \mathbf{B}^{\mathsf{T}} \mathbf{z}_x \\ - \frac{1}{2\alpha_3} \lambda \cdot \mathbf{1}_r^{\mathsf{T}} \mathbf{Q}_y^{-1} \boldsymbol{\mu} + \frac{\alpha_2}{2\alpha_3} \cdot \boldsymbol{\mu}^{\mathsf{T}} \mathbf{Q}_y^{-1} \mathbf{B}^{\mathsf{T}} \mathbf{z}_x + \lambda. \end{aligned} \tag{52}$$

It brings to the quadratic problem (50) with $n + r + 1$ variables.

4.4. Minimax Relevances (MaxRel).

The problem (50) is not convex. If we shift the spectrum for the matrix of quadratic form (52), the optimality is lost. To overcome this problem, we drop the term $\text{Sim}(\mathbf{Y})$.

$$\min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n \\ \mathbf{1}_n^{\mathsf{T}} \mathbf{z}_x = 1}} \max_{\substack{\mathbf{z}_y \geq \mathbf{0}_r \\ \mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y = 1}} \left[(1 - \alpha) \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x - \alpha \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y \right]. \tag{53}$$

The Lagrangian for the problem (53) with the fixed vector $\mathbf{z}_x$ is

$$L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu}) = (1-\alpha) \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x - \alpha \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y + \lambda \cdot (\mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y - 1) + \boldsymbol{\mu}^{\mathsf{T}} \mathbf{z}_y. \tag{54}$$

Setting the gradient of the Langrangian $\nabla_{\mathbf{z}_y} L(\mathbf{z}_x, \mathbf{z}_y, \lambda, \boldsymbol{\mu})$ to zero, we obtain:

$$\alpha \cdot \mathbf{B}^{\mathsf{T}} \mathbf{z}_x = \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu}. \tag{55}$$

The dual function is equal to

$$g(\mathbf{z}_x, \lambda, \boldsymbol{\mu}) = \begin{cases} (1-\alpha) \cdot \mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x - \lambda, & \alpha \cdot \mathbf{B}^{\mathsf{T}} \mathbf{z}_x = \lambda \cdot \mathbf{1}_r + \boldsymbol{\mu}; \\ +\infty, & \text{otherwise}. \end{cases} \tag{56}$$

In this case the feature scores are the solution of (50).

4.5. Asymmetric Importance (AsymImp)

Another way to overcome the problem of SymImp strategy is to add penalty for targets, which are well-explained by the features. We add the term $\mathbf{b}^{\mathsf{T}} \mathbf{z}_y$ to the term $\text{Rel}(\mathbf{X}, \mathbf{Y})$:

$$\alpha_1 \cdot \underbrace{\mathbf{z}_x^{\mathsf{T}} \mathbf{Q}_x \mathbf{z}_x}_{\text{Sim}(\mathbf{X})} - \alpha_2 \cdot \underbrace{\left(\mathbf{z}_x^{\mathsf{T}} \mathbf{B} \mathbf{z}_y - \mathbf{b}^{\mathsf{T}} \mathbf{z}_y\right)}_{\text{Rel}(\mathbf{X}, \mathbf{Y})} + \alpha_3 \cdot \underbrace{\mathbf{z}_y^{\mathsf{T}} \mathbf{Q}_y \mathbf{z}_y}_{\text{Sim}(\mathbf{Y})} \to \min_{\substack{\mathbf{z}_x \geq \mathbf{0}_n, \, \mathbf{1}_n^{\mathsf{T}} \mathbf{z}_x = 1 \\ \mathbf{z}_y \geq \mathbf{0}_r, \, \mathbf{1}_r^{\mathsf{T}} \mathbf{z}_y = 1}}. \tag{57}$$

We define the vector $\mathbf{b}$ as follows

$$b_j = \max_{i=1,\dots n} [\mathbf{B}]_{i,j}. \tag{58}$$

It brings to the positive coefficient for the importancies $\mathbf{z}_y$ in $\text{Rel}(\mathbf{X}, \mathbf{Y})$.

Proposition 5. *The balance between the terms Sim*$(\mathbf{X})$*, Rel*$(\mathbf{X}, \mathbf{Y})$*, and Rel*$(\mathbf{X}, \mathbf{Y})$ *for the problem* (57) *is achieved by the following coefficients:*

$$\alpha_1 = \frac{\overline{\mathbf{Q}}_y \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right)}{\overline{\mathbf{Q}}_y \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right) + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}; \tag{59}$$

$$\alpha_2 = \frac{\overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y}{\overline{\mathbf{Q}}_y \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right) + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}; \tag{60}$$

$$\alpha_3 = \frac{\overline{\mathbf{Q}}_x \overline{\mathbf{B}}}{\overline{\mathbf{Q}}_y \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right) + \overline{\mathbf{Q}}_x \overline{\mathbf{Q}}_y + \overline{\mathbf{Q}}_x \overline{\mathbf{B}}}. \tag{61}$$

Algorithm	Strategy	Error function $S(\mathbf{a}\|\mathbf{X}, \mathbf{Y})$
RelAgg	$\min\bigl[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X}, \mathbf{Y})\bigr]$	$\min\limits_{\mathbf{z}_x}\bigl[(1-\alpha)\cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{1}_r\bigr]$
SymImp	$\min\bigl[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X}, \mathbf{Y})$ $+ \mathrm{Sim}(\mathbf{Y})\bigr]$	$\min\limits_{\mathbf{z}_x,\, \mathbf{z}_y}\bigl[\alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y + \alpha_3 \cdot \mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y\bigr]$
MinMax	$\min\bigl[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X}, \mathbf{Y})\bigr]$ $\max\bigl[\mathrm{Rel}(\mathbf{X}, \mathbf{Y}) + \mathrm{Sim}(\mathbf{Y})\bigr]$	$\min\limits_{\mathbf{z}_x}\max\limits_{\mathbf{z}_y}\bigl[\alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y - \alpha_3 \cdot \mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y\bigr]$
AsymImp	$\max\bigl[\mathrm{Rel}(\mathbf{X}, \mathbf{Y}) + \mathrm{Sim}(\mathbf{Y})\bigr]$ $\min\bigl[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X}, \mathbf{Y})\bigr]$	$\min\limits_{\mathbf{z}_x, \mathbf{z}_y}\bigl[\alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha_2 \cdot \bigl(\mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y - \mathbf{b}^{\mathsf{T}}\mathbf{z}_y\bigr) + \alpha_3 \cdot \mathbf{z}_y^{\mathsf{T}}\mathbf{Q}_y\mathbf{z}_y\bigr]$
MaxRel	$\min\bigl[\mathrm{Sim}(\mathbf{X}) - \mathrm{Rel}(\mathbf{X}, \mathbf{Y})\bigr]$ $\max\bigl[\mathrm{Rel}(\mathbf{X}, \mathbf{Y})\bigr]$	$\min\limits_{\mathbf{z}_x}\max\limits_{\mathbf{z}_y}\bigl[(1-\alpha)\cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{B}\mathbf{z}_y\bigr]$

Table 1: Overview of proposed multivariate QPFS algorithms

Proof. The desired values of α_1, α_2, and α_3 are given by solution of the following equations

$$\alpha_1 + \alpha_2 + \alpha_3 = 1; \tag{62}$$

$$\alpha_1 \overline{\mathbf{Q}}_x = \alpha_2 \overline{\mathbf{B}}; \tag{63}$$

$$\alpha_2 \left(\overline{\mathbf{b}} - \overline{\mathbf{B}}\right) = \alpha_3 \overline{\mathbf{Q}}_y. \tag{64}$$

□

Proposition 6. *For the case* $r = 1$ *the proposed functions* (57)*,* (43)*, and* (53) *coincide with the original QPFS algorithm* (1)*.*

Proof. If r is equal to 1, then $\mathbf{Q}_y = q_y$ is a scalar, $\mathbf{z}_y = 1$, $\mathbf{B} = \mathbf{b}$. It reduces the problems (57), (43), and (53) to

$$\alpha_1 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{Q}_x\mathbf{z}_x - \alpha_2 \cdot \mathbf{z}_x^{\mathsf{T}}\mathbf{b} \to \min_{\mathbf{z}_x \geq \mathbf{0}_n,\, \mathbf{1}_n^{\mathsf{T}}\mathbf{z}_x = 1}. \tag{65}$$

Setting $\alpha = \frac{\alpha_2}{\alpha_1 + \alpha_2}$ brings to the original QPFS problem (1). □

To summarize all proposed strategies for multivariate feature selection, Table 1 shows the core ideas and error functions for each method.

5. EXPERIMENT

We carried out computational experiment with ECoG data from the NeuroTycho project. The input data consists of brain voltage signals recorded from 32 channels. The goal is to predict 3D hand position in the next moments given the input signal. The example of input signals and the 3D wrist coordinates are shown in Figure 3. The initial voltage signals are transformed to the spatial-temporal representation using wavelet transformation with Morlet mother wavelet. The procedure of extracting feature representation from the raw data are described in details in [19, 20]. We unfold the data and feature description at each time moment has dimension equals to 32 (channels) × 27 (frequencies) = 864. Each object is the representation of local history time segment with duration $\Delta t = 1s$. The time step between objects is $\delta t = 0.05s$. The final matrices are $\mathbf{X} \in \mathbb{R}^{18900 \times 864}$ and $\mathbf{Y} \in \mathbb{R}^{18900 \times 3k}$, where k is a number of timestamps that we predict. We split our data into train and test parts with the ratio 0.67.

5.1. Metrics

To evaluate the selected feature subset we introduce criteria that estimates the quality of feature selection procedure. We measure multicorrelation by mean value of miltiple correlation coefficient as follows

$$R^2 = \frac{1}{r}\text{tr}\left(\mathbf{C}^{\mathsf{T}}\mathbf{R}^{-1}\mathbf{C}\right); \quad \text{where } \mathbf{C} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\nu}_j)]_{\substack{i=1,\dots,n,\\ j=1,\dots,r}} \mathbf{R} = [\text{corr}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j)]_{i,j=1}^{n}. \tag{66}$$

This coefficient lies between 0 and 1. The bigger R^2 means the better feature subset we have.

The model stability is given by the logarithm of the ratio between minimal eigenvalue $\lambda_{\min}$ and maximum eigenvalue $\lambda_{\max}$ of the matrix $\mathbf{X}^{\mathsf{T}}\mathbf{X}$:

$$\text{Stability} = \ln \frac{\lambda_{\min}}{\lambda_{\max}}. \tag{67}$$

This criteria should be minimal.

The scaled Root Mean Squared Error (sRMSE) shows the quality of the model prediction. We estimate sRMSE on train and test data.

$$\text{sRMSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}}) = \sqrt{\frac{\text{MSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}})}{\text{MSE}(\mathbf{Y}, \overline{\mathbf{Y}})}} = \frac{\|\mathbf{Y} - \widehat{\mathbf{Y}}_{\mathbf{a}}\|_2}{\|\mathbf{Y} - \overline{\mathbf{Y}}\|_2}. \tag{68}$$

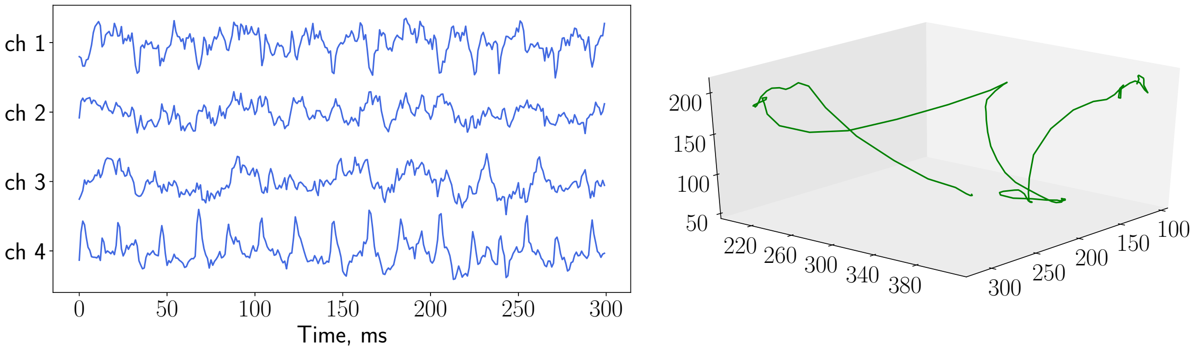


Figure 3: Brain signals and the corresponding hand position

Here $\widehat{\mathbf{Y}}_{\mathbf{a}} = \mathbf{X}_{\mathbf{a}}\mathbf{\Theta}_{\mathbf{a}}^{\mathsf{T}}$ is a model prediction and $\overline{\mathbf{Y}}$ is a constant prediction obtained by averaging the targets across all objects.

Akaike Information Criteria (AIC) is a trade-off between prediction quality and the size of selected subset $\|\mathbf{a}\|_0$:

$$\text{AIC} = m \ln \left(\frac{\text{MSE}(\mathbf{Y}, \widehat{\mathbf{Y}}_{\mathbf{a}})}{m} \right) + 2\|\mathbf{a}\|_0, \tag{69}$$

where $\|\mathbf{a}\|_0 = \#\{j : a_j \neq 0\} = \sum_{j=1}^{n} a_j$.

5.2. Results

To show the redundancy in the data representation we solve the QPFS problem for our data. Figures 4 and 5 show the result, where we use the Relevance Aggregation strategy and $k = 1$. QPFS scores $\mathbf{z}_x$ decrease sharply. It allows to use the elbow rule to choose the threshold value τ. In our experiments we set $\tau = 10^{-4}$. Only about one hundred features have scores significantly greater than zero. If add features to this amount the test error stops to decrease.

Figure 6 shows the dependencies in the matrices $\mathbf{X}$ and $\mathbf{Y}$. Frequencies in the matrix $\mathbf{X}$ are highly correlated. The frequencies are choosen in logarithmic scale, the closer the frequencies are the higher the correlations. In the target matrix $\mathbf{Y}$ the correlations between axes are not significant in comparison with the correlations between consequent moments and the correlations decay with time.

We apply the QPFS algorithm with SymImp strategy for different values of α_3 coefficient according to formulas (40). The dependence between target scores $\mathbf{z}_y$ with respect to α_3 for different values of k is shown in Figure 7. If we predict wrist coordinates only for one timestamp $k = 1$, targets scores are almost the same. It tells about the independence

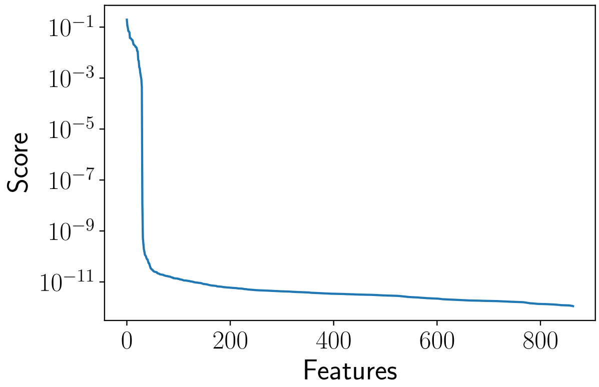


Figure 4: Sorted feature importances for the QPFS algorithm

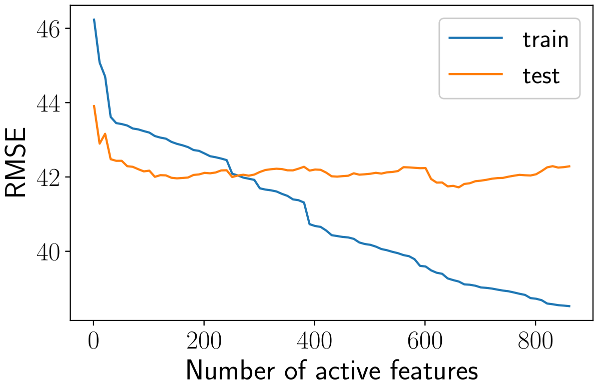


Figure 5: RMSE w.r.t. number of selected features (features are ranked by QPFS algorithm)

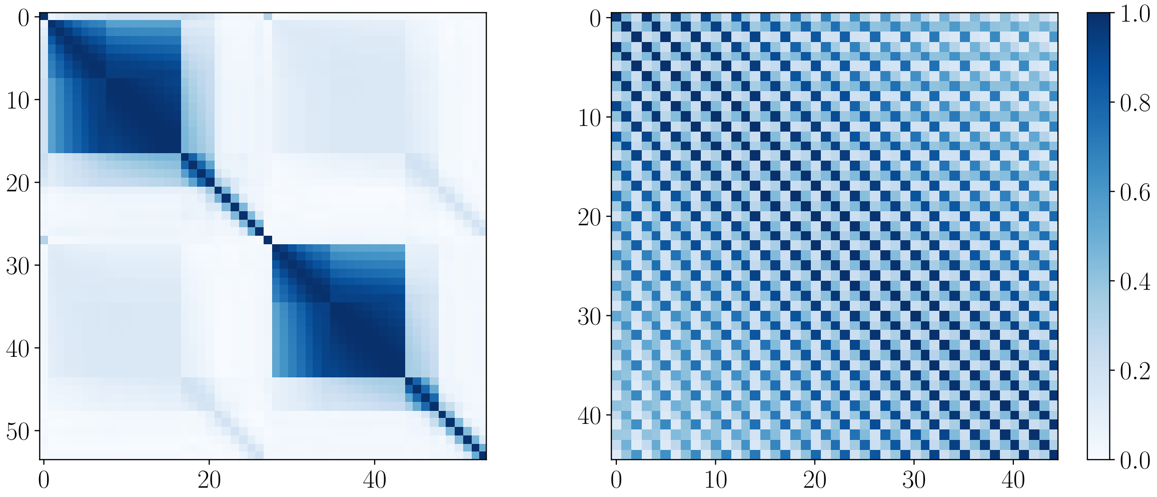
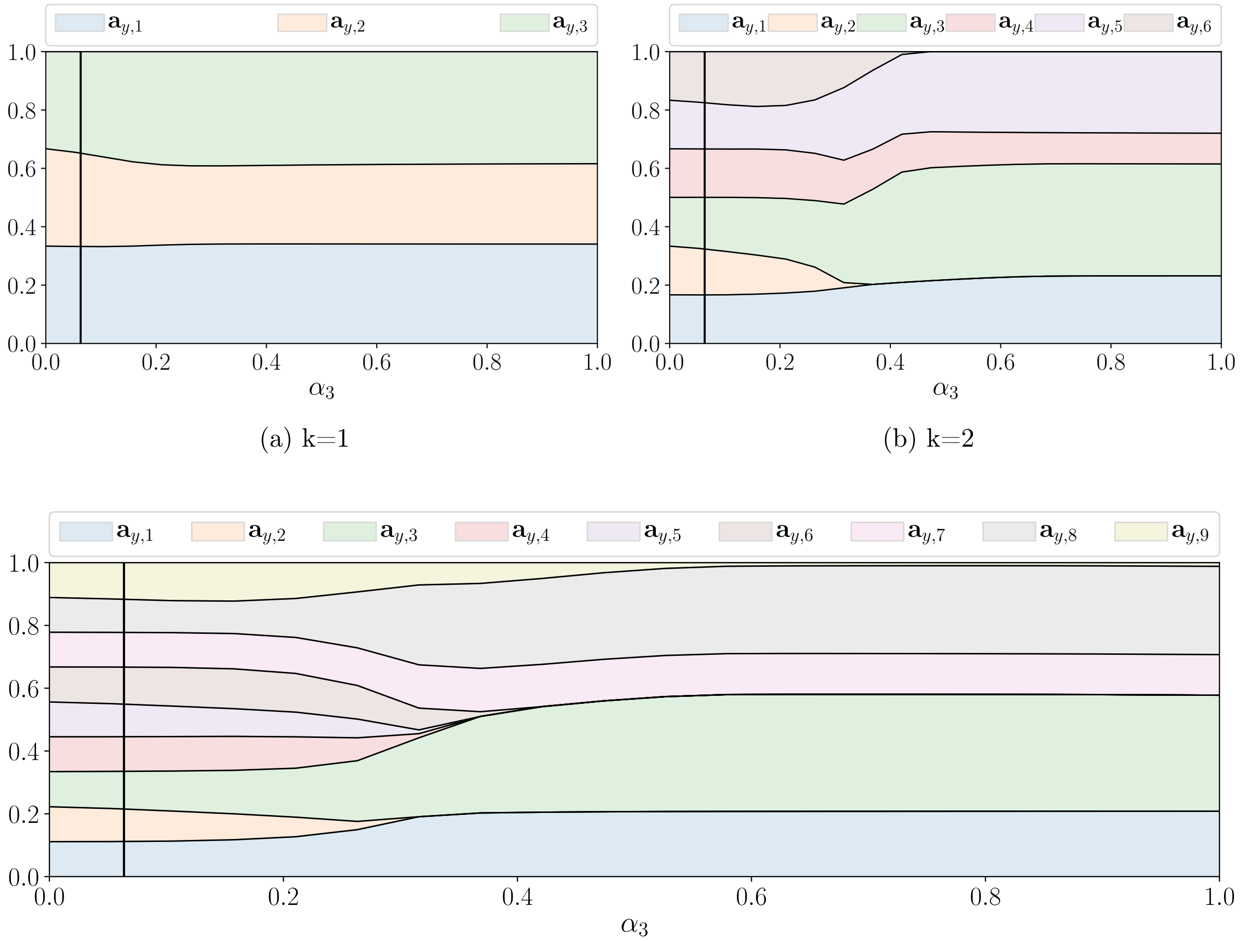


Figure 6: Correlation matrices for $\mathbf{X}$ and $\mathbf{Y}$

between x, y, and z coordinates. For $k = 2$ and $k = 3$ the scores of some targets become zero when α_3 increases. The vertical lines correspond to the optimal value of coefficient α_3 obtained by (37). The importances $\mathbf{a}_y$ for this value of α_3 are similar. It means that the algorithm does not distinguish the targets for $k = 1, 2, 3$.

We compare the proposed strategies of multivariate QPFS that are given in Table 1 for the ECoG dataset. Firstly, we apply all methods to get feature importances. Then we fit linear model with increasing number of features. For each method the features are sorted by the obtained importances. We show how the described metrics are changed with the increasing feature set size. Figure 8 illustrates the results for prediction of $k = 30$ timestamps. Here the feature importances threshold τ are shown by colored ticks. These thresholds are larger for the proposed methods with comparison to the baseline RelAgg strategy. The SymImp strategy has the largest threshold, it does not allow to get the small feature subset for this algorithm. However, this strategy shows the best performance in terms of sRMSE on



(a) k=1

(b) k=2

(c) k=3

Figure 7: Target importances $\mathbf{z}_y$ with respect to α_3 for QPFS with Symmetric Importance

test data. The second value of performance is given by AsymImp strategy, followed by MaxRel. All proposed algorithms give the less test error compared to the RelAgg strategy. The Stability criteria is5 also increased for the proposed algorithms. Here we consider the AsymImp strategy is the best in terms of prediction quality and the size of selected feature subset.

To compare the structure of the selected feature subsets and investigate the stability of the selection procedure, we use bootstrap approach. First, the bootstrap data are generated. Then for each data we solve the feature selection problem. The obtained feature importances are compared. We calculate the average pairwise Spearman correlation coefficient and the ℓ_2 distance to obtain the measure of the algorithms stability. Table 2 shows the average error, the size of the subset and the described statistics for each method. The error was calculated by fitting the model on the 50 features with the largest importances. The MaxRel strategy shows the worst stability. AsymImp gives the least error on the test data. The size of selected feature subsets are overestimated using the equal threshold $\tau = 10^{-4}$. The value

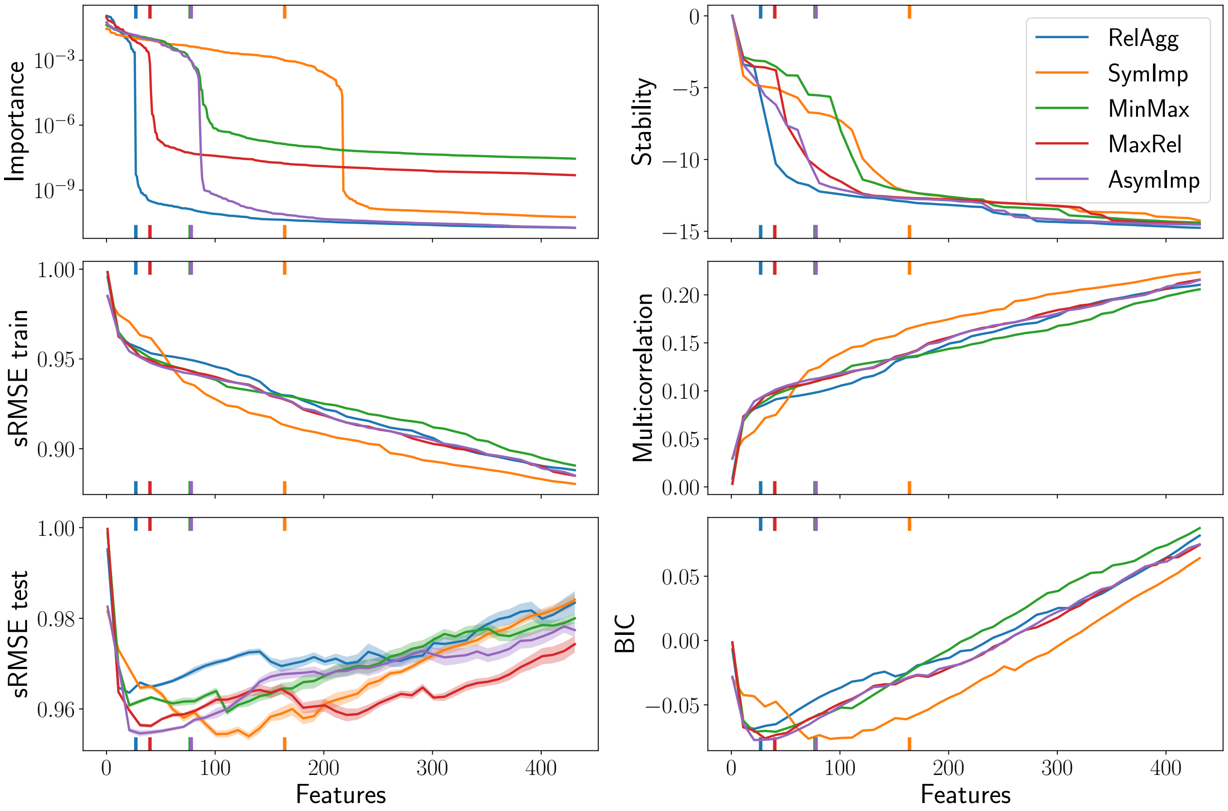


Figure 8: Feature selection algorithms evaluation for ECoG data, prediction of $k = 30$ timestamps

Table 2: The stability of the selected feature subset

	sRMSE	$\|\mathbf{a}\|_0$	Spearman ρ	ℓ_2 dist
RelAgg	0.965 ± 0.002	26.8 ± 3.8	0.915 ± 0.016	0.145 ± 0.018
SymImp	0.961 ± 0.001	224.4 ± 9.0	0.910 ± 0.017	0.025 ± 0.002
MinMax	0.961 ± 0.002	101.0 ± 2.1	0.932 ± 0.009	0.059 ± 0.004
MaxRel	0.958 ± 0.003	41.2 ± 5.2	0.862 ± 0.027	0.178 ± 0.010
AsymImp	0.955 ± 0.001	85.8 ± 10.2	0.926 ± 0.011	0.078 ± 0.007

of τ should be cross-validated to get the optimal threshold and the feature subset size.

Wer fit the PLS regression model for the data to compare the dimensionality reduction and feature selection. Figure 9 shows the example of the model prediction. Three solid lines show 3D coordinates of the hand position and the dashed lines are the coordinates predictions.

Figure 10 demonstrates the scaled RMSE on train and test data with respect to the dimensionality of the latent space l. The test error achieves minimum value at hte point $l = 11$. PLS regression is more flexible approach compared to the linear model built on the subset of features. It leads to the less error, but the model are not sparse.

Figure **??** compares 3 models: linear regression and PLS regression built on 100 features given by qpfs and PLS regression with all features. We do not include linear regression with

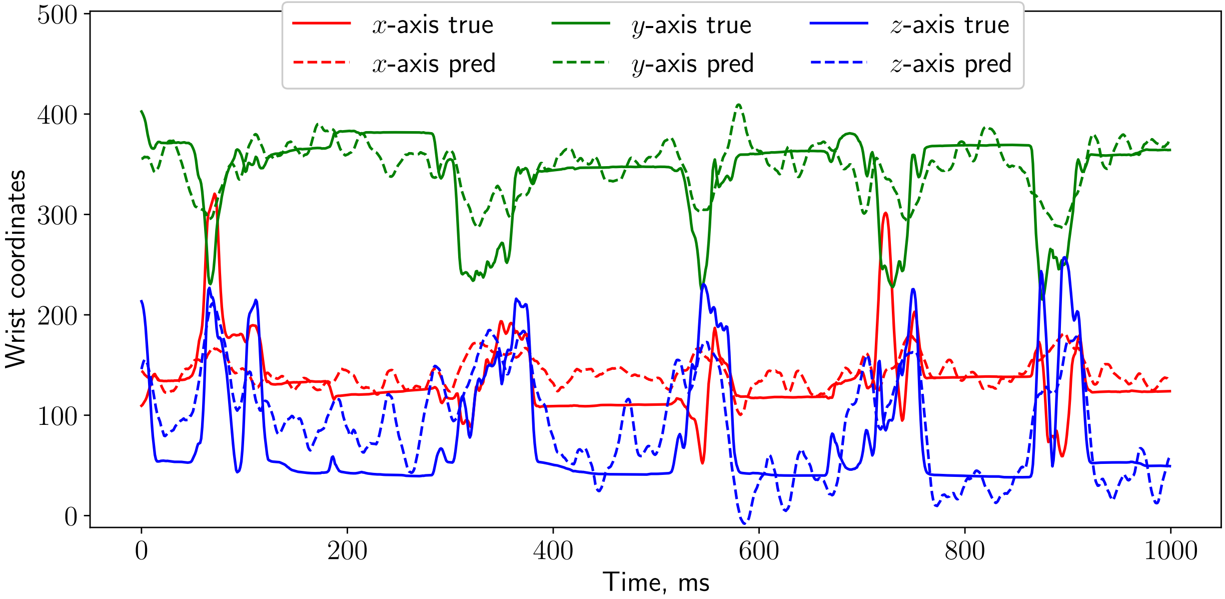


Figure 9: Example of 3D hand position prediction given by PLS algorithm

all features because its results are close to the constant prediction. We use the AsymImp strategy for QPFS in this experiment. The number of PLS latent dimension is $l = 15$. Here PLS regression are significantly better than linear regression with QPFS features. It means that the latter model is not flexible enough. The best result is given by combination of PLS regression model with QPFS features. This model is sparse since it uses only 100 QPFS features. The ability of PLS model to find the optimal latent data representation allows to improve model performance.

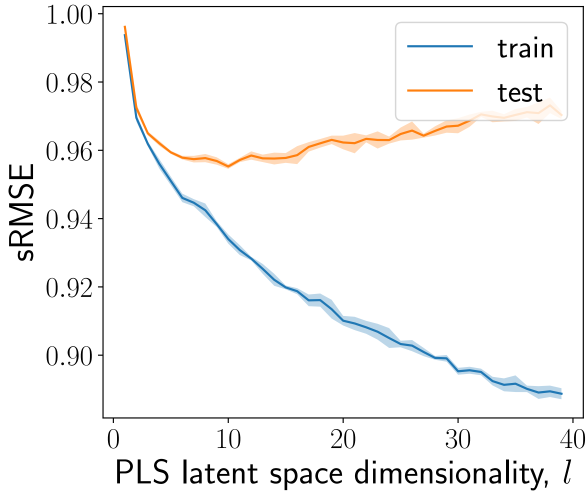


Figure 10: Test scaled RMSE for PLS regression model

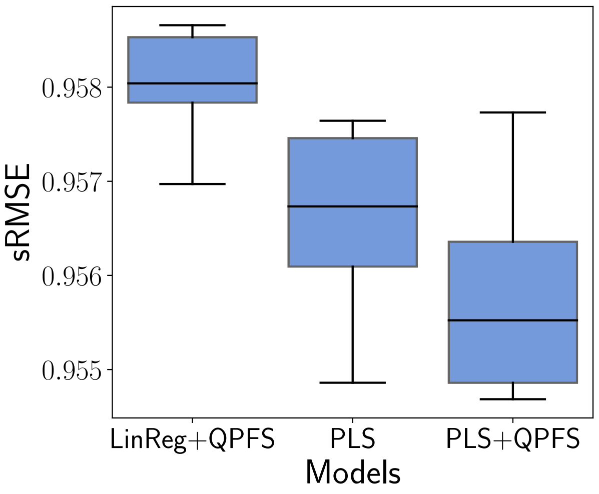


Figure 11: sRMSE box plots for different models

6. CONCLUSION

The study investigates the problem of signal decoding in relation to modelling Brain Computer Interface. To build a stable edequate model, it was proposed to reduce dimensionality of the problem using the dependencies in both input and target spaces. The partial least squares algorithm is considered as linear model for dimensionality reduction. The quadratic programming approach is investigated as feature selection algorithm. The multivariate extensions for the QPFS algorithms are proposed. The algorithms solve feature selection in a single quadratic programming optimization problem. The resulting feature subset includes non-correlated features which are relevant to the most difficult targets.

The computational experiments were carried out on the ECoG data. The resulting model predicts the limb position of an exoskeleton by brain signals. The proposed algorithms outperforms the baseline algorithm and reduce the problem dimension significantly.

References

[1] Thomas Costecalde, Tetiana Aksenova, Napoleon Torres-Martinez, Andriy Eliseyev, Corinne Mestais, Cecile Moro, and Alim Louis Benabid. A long-term bci study with ecog recordings in freely moving rats. *Neuromodulation: Technology at the Neural Interface*, 21(2):149–159, 2018.

[2] Corinne S Mestais, Guillaume Charvet, Fabien Sauter-Starace, Michael Foerster, David Ratel, and Alim Louis Benabid. Wimagine: Wireless 64-channel ecog recording implant for long term clinical applications. *IEEE transactions on neural systems and rehabilitation engineering*, 23(1):10–21, 2015.

[3] Andrey Eliseyev, Corinne Mestais, Guillaume Charvet, Fabien Sauter, Neil Abroug, Nana Arizumi, Serpil Cokgungor, Thomas Costecalde, Michael Foerster, Louis Korczowski, et al. Clinatec® bci platform based on the ecog-recording implant wimagine® and the innovative signal-processing: preclinical results. In *Engineering in Medicine and Biology Society (EMBC), 2014 36th Annual International Conference of the IEEE*, pages 1222–1225. IEEE, 2014.

[4] Hyonho Chun and Sündüz Keleş. Sparse partial least squares regression for simultaneous dimension reduction and variable selection. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(1):3–25, 2010.

[5] Tahir Mehmood, Kristian Hovde Liland, Lars Snipen, and Solve Sæbø. A review of variable selection methods in partial least squares regression. *Chemometrics and Intelligent Laboratory Systems*, 118:62–69, 2012.

[6] A. M. Katrutsa and V. V. Strijov. Stress test procedure for feature selection algorithms. *Chemometrics and Intelligent Laboratory Systems*, 142:172–183, 2015.

[7] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.

[8] Andrey Eliseyev and Tatiana Aksenova. Stable and artifact-resistant decoding of 3d hand trajectories from ecog signals using the generalized additive model. *Journal of neural engineering*, 11(6):066005, 2014.

[9] Sarah Engel, Tetiana Aksenova, and Andrey Eliseyev. Kernel-based npls for continuous trajectory decoding from ecog data for bci applications. In *International Conference on Latent Variable Analysis and Signal Separation*, pages 417–426. Springer, 2017.

[10] Andrey Eliseyev, Cecile Moro, Jean Faber, Alexander Wyss, Napoleon Torres, Corinne Mestais, Alim Louis Benabid, and Tetiana Aksenova. L1-penalized n-way pls for subset of electrodes selection in bci experiments. *Journal of neural engineering*, 9(4):045010, 2012.

[11] Roman Rosipal and Nicole Kramer. Overview and Recent Advances in Partial Least Squares. *C. Saunders et al. (Eds.): SLSFS 2005, LNCS 3940*, pages 34–51, 2006.

[12] Roman Rosipal. Nonlinear partial least squares an overview. In *Chemoinformatics and advanced machine learning perspectives: complex computational methods and collaborative techniques*, pages 169–189. IGI Global, 2011.

[13] Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

[14] Makoto Yamada, Wittawat Jitkrittum, Leonid Sigal, Eric P Xing, and Masashi Sugiyama. High-dimensional feature selection by feature-wise kernelized lasso. *Neural computation*, 26(1):185–207, 2014.

[15] Irene Rodriguez-Lujan, Ramon Huerta, Charles Elkan, and Carlos Santa Cruz. Quadratic programming feature selection. *Journal of Machine Learning Research*, 11(Apr):1491–1516, 2010.

[16] Alexandr Katrutsa and Vadim Strijov. Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria. *Expert Systems with Applications*, 76:1–11, 2017.

[17] Yamuna Prasad, KK Biswas, and Parag Singla. Scaling-up quadratic programming feature selection. In *AAAI (Late-Breaking Developments)*, 2013.

[18] Paul Geladi and Bruce R Kowalski. Partial least-squares regression: a tutorial. *Analytica chimica acta*, 185:1–17, 1986.

[19] Zenas C Chao, Yasuo Nagasaka, and Naotaka Fujii. Long-term asynchronous decoding of arm motion using electrocorticographic signals in monkey. *Frontiers in neuroengineering*, 3:3, 2010.

[20] Andrey Eliseyev and Tetiana Aksenova. Penalized multi-way partial least squares for smooth trajectory decoding from electrocorticographic (ecog) recording. *PloS one*, 11(5):e0154878, 2016.