Problem Statement

We consider the regression problem of predicting an target variable $y \in \mathbb{R}$ given an object $\mathbf{x}\mathbb{R}^n$. The goal is to build a model $f(\mathbf{x}|\mathbf{w}), \mathbf{w} \in \mathbb{R}^p$ which outcomes a prediction for each input object. Let assume that we are given the design matrix $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_m]^\intercal \in \mathbb{R}^{m \times n}$ and the target vector $\mathbf{y} = [y_1, \ldots, y_m]^\intercal \in \mathbb{R}^m$. Each i-th matrix $\mathbf{X}$ row represents an object which is associated with the i-th vector $\mathbf{y}$ element. The goal is to find the optimal weight vector $\mathbf{w}^*$.

The weights $\mathbf{w}$ are fitted by the minimization of an error function:

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^r} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f). \tag{1}$$

The most common choice for the error function $S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f)$ in the regression task is the squared error between real target values and predicted ones

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \|\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})\|_2^2 = \sum_{i=1}^{m} (y_i - f(\mathbf{x}_i|\mathbf{w}))^2.$$

The problem 1 could be solved by any neural network optimization method [].

The number of model weights p could be extremely huge. In this case the solution of the problem 1 leads to overfitting. To eliminate this problem we propose to select the subset $\mathcal{A} \subseteq \{1, \ldots, p\}$ of the active weights. The weights which are not active are supposed to be zero. To choose the subset $\mathcal{A}$ from all possible 2^p combinations let introduce a quality criteria $Q(\mathcal{A}|\mathbf{X}, \mathbf{y}, f)$. This function evaluates the quality of a particular active set $\mathcal{A}$. We desire to find the optimal subset $\mathcal{A}^*$ which minimizes the quality criteria

$$\mathcal{A}^* = \arg\min_{\mathcal{A} \subseteq \{1, \ldots, p\}} Q(\mathcal{A}|\mathbf{X}, \mathbf{y}, f). \tag{2}$$

If the solution $\mathcal{A}^*$ of the 2 is given the next step is to determine the optimal model weights $\mathbf{w}^*$ by solving a problem

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^p} S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f), \quad \text{subject to } w_j = 0 \text{ for } j \notin \mathcal{A}^*. \tag{3}$$

QPFS

To find the optimal subset $\mathcal{A}^*$ we suggest to use QPFS algorithm. The original algorithm selects features for the linear regression task, where the model $f(\mathbf{x}|\mathbf{w}) = \mathbf{x}^\intercal \mathbf{w}$ and the problem 1

$$S(\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \|\mathbf{y} - \mathbf{X}\mathbf{w}\|_2^2 \to \min_{\mathbf{w} \in \mathbb{R}^n}.$$

The goal of the QPFS is to select not correlated features which are relevant to target vector. To formalise this approach let introduce two functions: Sim and Rel. The former measures the redundancy between features, the latter contains relevances between each feature and target vector. We want to minimize the Sim function and maximize the Rel simultaneously.

The QPFS method offers the explicit way to construct the functions Sim and Rel. The method minimizes the following functional

$$\underbrace{\mathbf{z}^\mathsf{T}\mathbf{Q}\mathbf{z}}_{\text{Sim}} - \alpha \cdot \underbrace{\mathbf{b}^\mathsf{T}\mathbf{z}}_{\text{Rel}} \to \min_{\mathbf{z}\in[0,1]^n}. \tag{4}$$

This functional is an analogue of the described quality criteria 2. The first term is associated with the Sim function and the second with the Rel. 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 $\mathbf{y}$. The normalized vector $\mathbf{z}$ shows the importance of each feature. This functional penalizes the dependent features by the function Sim and encourages features relevant to the target by the function Rel. The parameter α allows to control the trade-off between the Sim and the Rel terms. To find the optimal feature subset the thresholding for $\mathbf{z}$ is applied:

$$j \in \mathcal{A} \Leftrightarrow z_j > \tau.$$

To measure similarity it was proposed to use the absolute value of sample Pearson correlation coefficient or sample mutual information between pairs of features for the Sim function and between features and target vector for the Rel function. The problem 4 is convex if the matrix $\mathbf{Q}$ is positive semidefinite. In general it is not always true. To satisfy this condition we shift the matrix $\mathbf{Q}$ spectrum and replace the matrix $\mathbf{Q}$ by $\mathbf{Q} - \lambda_{\min}\mathbf{I}$, where $\lambda_{\min}$ is a $\mathbf{Q}$ minimal eigenvalue.

Model linearization

Let assume that we have the model $f(\mathbf{x}|\mathbf{w})$ and the weight vector $\mathbf{w}$ and we want to find the new weights by adding the updates $\Delta\mathbf{w}$ to the existing weights $\mathbf{w}$. Similar to the Levenberg-Marquardt algorithm, we use the linear approximation of the model

$$\mathbf{f}(\mathbf{X}|\mathbf{w} + \Delta\mathbf{w}) \approx \mathbf{f}(\mathbf{X}|\mathbf{w}) + \mathbf{J} \cdot \Delta\mathbf{w},$$

where $\mathbf{J} \in \mathbb{R}^{m\times p}$ is a Jacobian matrix

$$\mathbf{J} = \begin{pmatrix} \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial f(\mathbf{x}_1|\mathbf{w})}{\partial w_p} \\ \cdots & \cdots & \cdots \\ \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_1} & \cdots & \frac{\partial f(\mathbf{x}_m|\mathbf{w})}{\partial w_p} \end{pmatrix}. \tag{5}$$

In this case the problem 1 is a linear regression problem with the target vector $\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})$, matrix $\mathbf{J}$ and weights $\Delta\mathbf{w}$

$$S(\mathbf{w} + \Delta\mathbf{w}|\mathbf{X}, \mathbf{y}, f) = \|\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w} + \Delta\mathbf{w})\|_2^2 \approx \|(\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})) - \mathbf{J} \cdot \Delta\mathbf{w}\|_2^2. \tag{6}$$

To find the most significant set of weights for the current point $\mathbf{w}$ we can apply QPFS algorithm. Significance means the relevance of the weight update to the residual vector and pairwise weights updates independence through training data.

We use backpropagation procedure to update the network weights. The most of the neural network optimization methods use the gradient of the model to update network weights. It allows to get the Jacobian matrix $\mathbf{J}$ for free from optimization process.

We want to investigate the dependence of the QPFS solution for the problem 6. Let assume that we have some weight vector $\mathbf{w}^0$ which lies near the optimal weight vector $\mathbf{w}^*$. We consider the line segment

$$\mathbf{w}_\beta = \beta\mathbf{w}^* + (1-\beta)\mathbf{w}^0;\ \beta \in [0,1].$$

Linear model

In the case of linear model $f(\mathbf{x}|\mathbf{w}) = \mathbf{x}^\intercal\mathbf{w}$, the Jacobian matrix $\mathbf{J}$ equals the design matrix $\mathbf{X}$. The problem 6 has the form

$$\|(\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})) - \mathbf{J}\cdot\Delta\mathbf{w}\|_2^2 = \|(\mathbf{y} - \mathbf{X}\mathbf{w}) - \mathbf{X}\cdot\Delta\mathbf{w}\|_2^2.$$

The matrix $\mathbf{Q}$ entries are the similarities between features, the vector $\mathbf{b}$ entries are the feature relevances to the residual vector $\mathbf{y} - \mathbf{X}\mathbf{w}$. Therefore, the $\mathbf{Q}$ matrix is constant with respect to the model weights $\mathbf{w}$. The vector $\mathbf{b}$ entries decreases as the β coefficient is getting smaller. The residual vector $\mathbf{y} - \mathbf{X}\mathbf{w}^*$ is orthogonal to the matrix $\mathbf{X}$ columns for the optimal weights $\mathbf{w}^*$, since the correlation and the mutual information coefficients equals zero for the orthogonal vectors.

Two layer neural network

We consider the feed-forward two layer neural network as the nonlinear model. In this case the model $f(\mathbf{x}|\mathbf{w})$ is given by

$$f(\mathbf{x}|\mathbf{w}) = \sigma(\mathbf{x}^\intercal\mathbf{W}_1)\mathbf{W}_2.$$

Here $\mathbf{W}_1 \in \mathbb{R}^{n\times h}$ the weight matrix which connects the input features with h hidden units, $\sigma(\cdot)$ is a nonlinearity function which applied element-wise, and $\mathbf{W}_2 \in \mathbb{R}^{h\times 1}$ the weight matrix which connects the hidden units with output. The model weight vector $\mathbf{w}$ is a concatenation of vectorized matrices $\mathbf{W}_1$, $\mathbf{W}_2$.

The first order optimization condition tells that in the optimal point the first derivatives of the error function are equal to

$$\left.\frac{\partial S(\mathbf{w}|\mathbf{X},\mathbf{y},f)}{\partial\mathbf{w}}\right|_{\mathbf{w}=\mathbf{w}^*} = \left.(\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w}))^\intercal\mathbf{J}\right|_{\mathbf{w}=\mathbf{w}^*} = 0.$$

This implies the orthogonality between residual vector $\mathbf{y} - \mathbf{f}(\mathbf{X}|\mathbf{w})$ and the columns of the Jacobian matrix $\mathbf{J}$. It means that in the optimum point $\mathbf{w}^*$ the linear term $\mathbf{b}$ will be around zero.

Experiment

In the experiment we used the Boston House Pricing dataset (objects: 506, features: 13).

First of all we investigate the influence of α coefficient to the QPFS function 4. Figure 1 shows the features scores $\mathbf{z} \in \mathbb{R}^{13}$ for correlation and mutual information coefficients with respect to different α. If α is large enough the Rel term is prevailed.

All features is active in this case. The algorithm does not penalize features and selects features relevant to the target. As α goes down the Sim term starts to penalize feature interactions and the active set of features shrinks. It worth to note that correlations mutual information works in different ways.

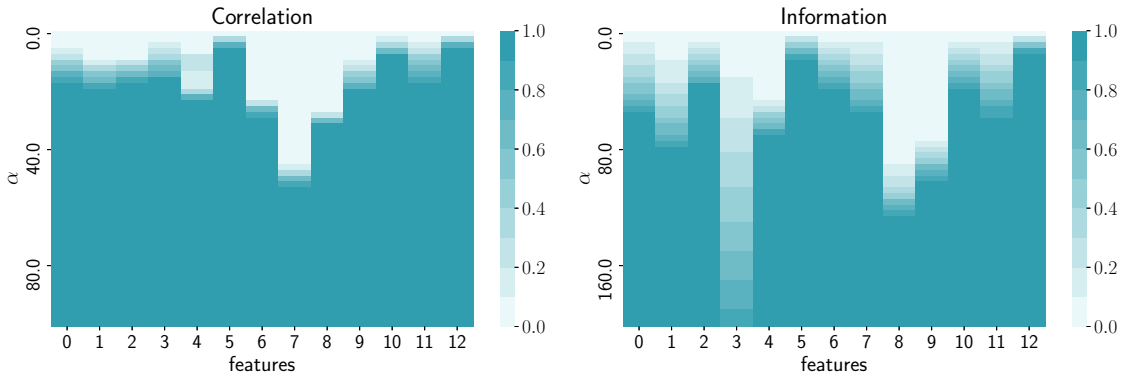

Figure 1: QPFS scores with respect to α coefficient

The authors of the original QPFS paper suggested the way to select α and make Sim and Rel terms impact are equal

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

where $\overline{\mathbf{Q}}$, $\overline{\mathbf{b}}$ are the mean values of $\mathbf{Q}$ and $\mathbf{b}$ correspondingly.

We fitted the linear model for this data. The error landscape is shown in the figure 2 for two randomly selected weights. We add the random noise to the optimum weights $\mathbf{w}^*$ to get the point $\mathbf{w}^0$. The behaviour of the Rel term vector $\mathbf{b}$ on the line segment between $\mathbf{w}^0$ and $\mathbf{w}^*$ is illustrated in the figure 3.

The landscape for the neural network model is more complex. We use only 2 hidden units to get not excessively complex model. The optimal weight vector $\mathbf{w}^*$ is obtained by backpropagation optimization procedure. Figure 4 shows the error function on the grid of two neural network weights from $\mathbf{W}_1$. We use the same strategy to investigate how the linear term vector $\mathbf{b}$ is changing moving from $\mathbf{w}^0$ to $\mathbf{w}^*$. The result are shown in the figure 5.

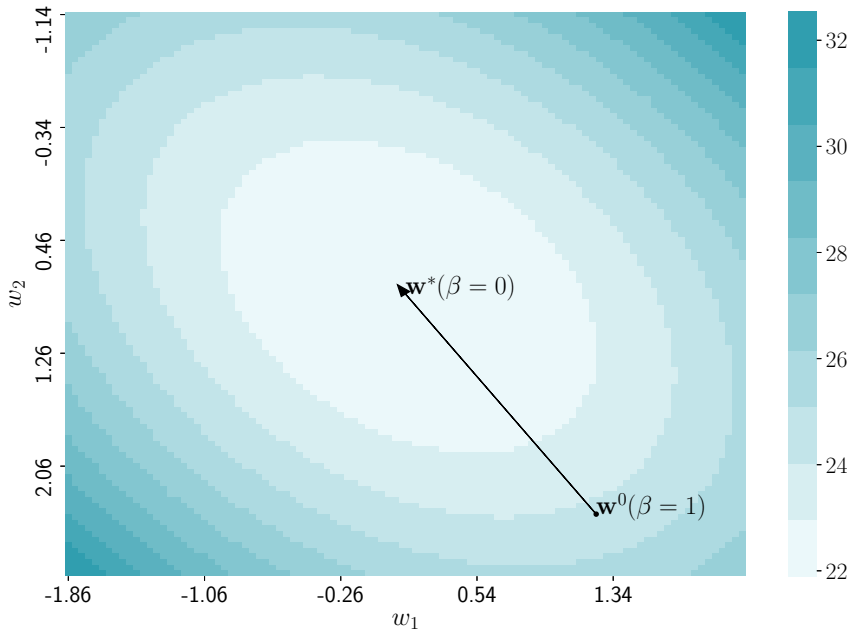

Figure 2: Error function landscape near optimal weight point for linear model.

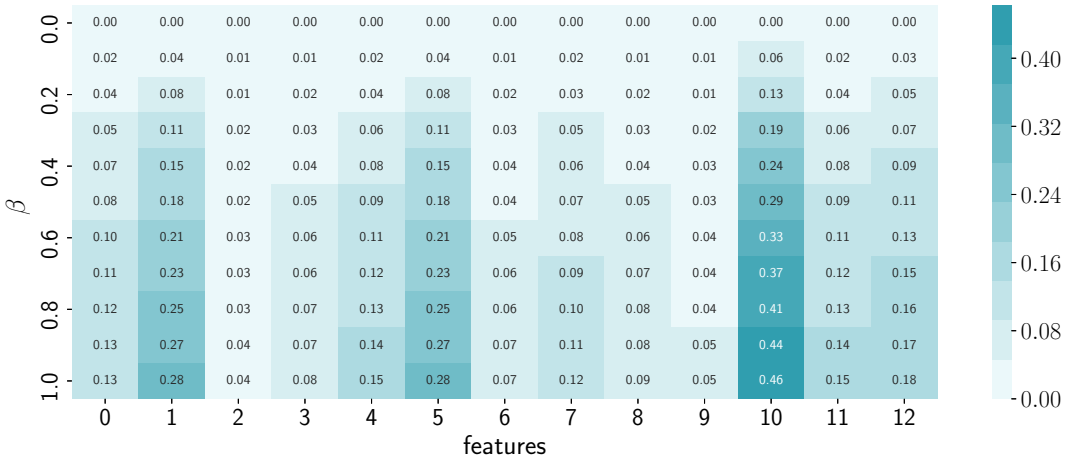

Figure 3: Relevance scores for linear model with respect to β coefficient.

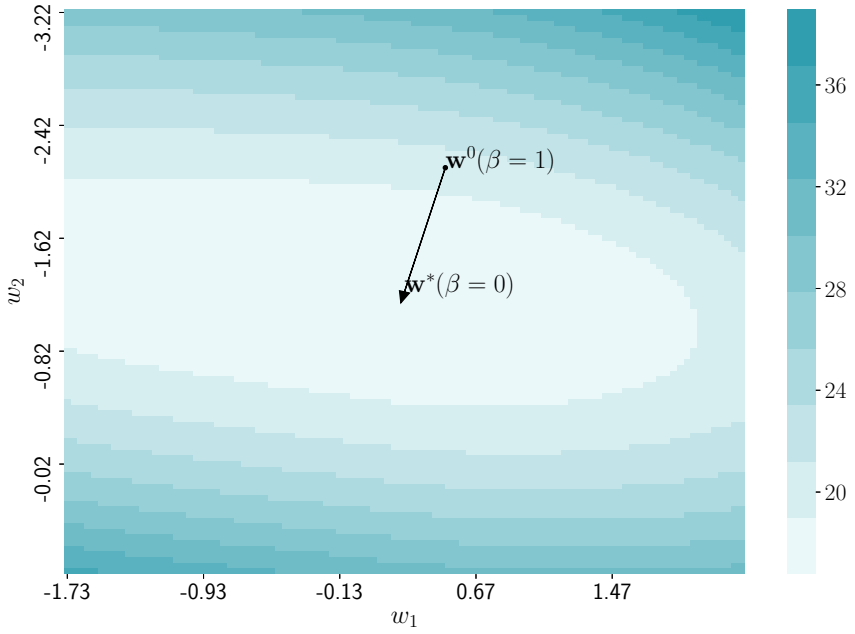

Figure 4: Error function landscape near optimal weight point for neural network.

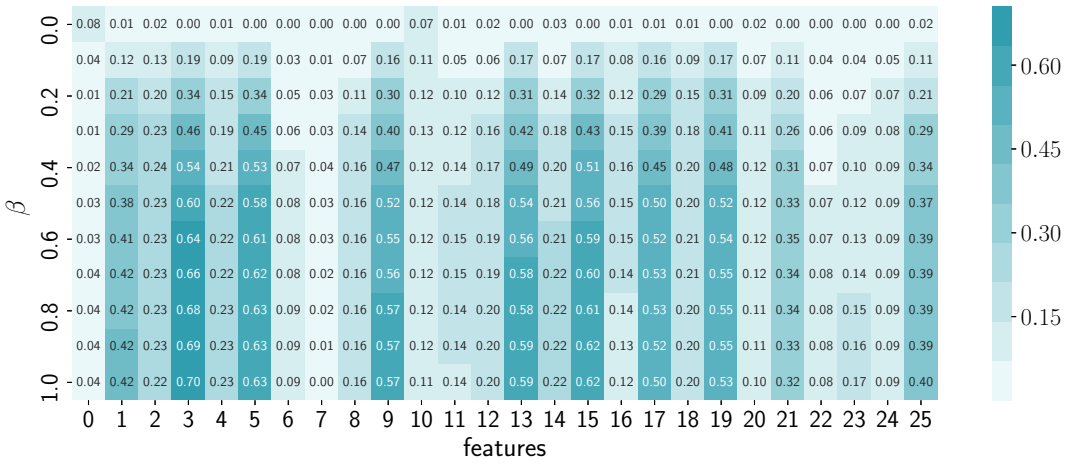

Figure 5: Relevance scores for neural network with respect to β coefficient.

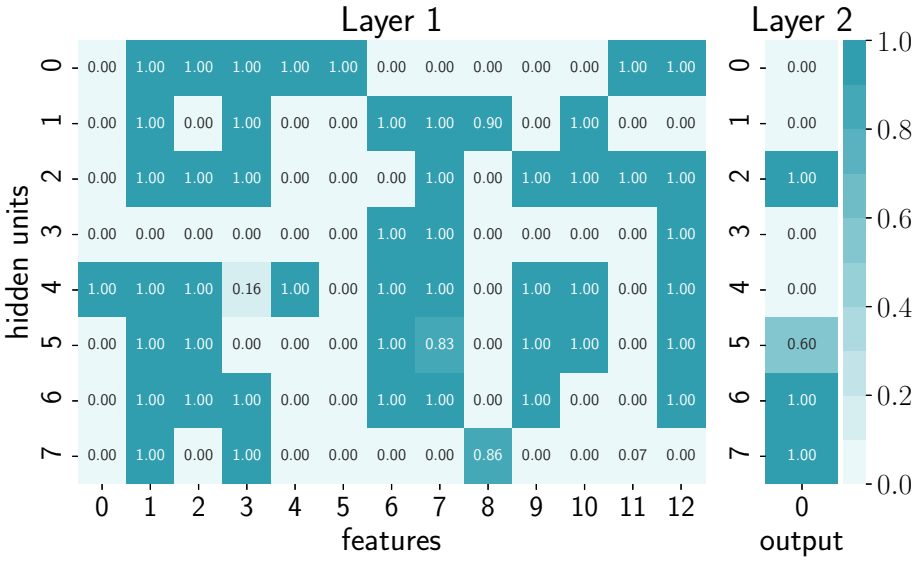

Figure 6: Neural network weight scores maps for mutual information similarity coefficient