Problem Statement

Let $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_m]^T = [\boldsymbol{\chi}_1, \dots, \boldsymbol{\chi}_n] \in \mathbb{R}^{m \times n}$ be a design matrix, where $\mathbf{x}_i \in \mathbb{R}^n$ is the i-th object, $\boldsymbol{\chi}_j \in \mathbb{R}^m$ is the j-th feature. Denote by $\mathcal{J} = \{1, \dots, n\}$ the feature index set, and let $\mathcal{A} \subseteq \mathcal{J}$ be a feature index subset. Let $\mathbf{y} \in \{1, \dots, K\}^n$ be the target vector. Suppose the function $\mathbf{h}$ approximates the probabilities of class label taking on each of the K possible values given object $\mathbf{x}$ feature index subset $\mathcal{A}$ and model parameters $\mathbf{W} = \left[\mathbf{w}^{(1)}, \dots, \mathbf{w}^{(K)}\right] \in \mathbb{R}^{n \times K}$

$$\mathbf{h}(\mathbf{x}, \mathcal{A}, \mathbf{W}) = \begin{bmatrix} p(y = 1|\mathbf{x}, \mathcal{A}, \mathbf{W}) \\ \dots \\ p(y = K|\mathbf{x}, \mathcal{A}, \mathbf{W}) \end{bmatrix}.$$

Let $\mathbf{a} \in \mathbb{B}^n = \{0{,}1\}^n$ be an indicator vector such that $a_j = 1$ if and only if $j \in \mathcal{A}$. The vector $\mathbf{a}^*$ and the index set $\mathcal{A}^*$ are related by

$$a_j^* = 1 \Leftrightarrow j \in \mathcal{A}^*, \; j \in \mathcal{J}. \tag{1}$$

The data fitting problem is to find parameters $\mathbf{W}^*$ such that

$$\mathbf{W}^* = \mathop{\arg\min}_{\mathbf{W} \in \mathbb{R}^{n \times K}} S\left(\mathbf{W}, \mathbf{a}|\mathbf{X}, \mathbf{y}, \mathbf{h}\right), \tag{2}$$

where S is the error function, which validates the quality of the parameters $\mathbf{W}$ and the corresponding feature index subset $\mathcal{A}$ given a design matrix $\mathbf{X}$, a target vector $\mathbf{y}$ and a hypothesis function $\mathbf{h}$.

The features are assumed to be noisy, irrelevant or multicollinear, which leads to additional error in estimating the optimum model parameters $\mathbf{W}^*$ and increases the instability of them. Feature selection methods can be used to remove certain features from the design matrix $\mathbf{X}$. The feature selection procedure reduces the dimensionality of problem (2) and improves the stability of the optimum parameters $\mathbf{W}^*$. The feature selection problem is

$$\mathbf{a}^* = \mathop{\arg\min}_{\mathbf{a} \in \mathbb{B}^n} Q(\mathbf{a}|\mathbf{X}, \mathbf{y}), \tag{3}$$

where $Q : \mathbb{B}^n \to \mathbb{R}$ is a quality criterion that determines the quality of a selected feature index subset $\mathcal{A} \subseteq \mathcal{J}$.

Problem (3) does not necessarily require estimation of the optimum parameters $\mathbf{W}^*$ for data fitting procedure. It uses the relationships between the features $\boldsymbol{\chi}_j, j \in \mathcal{J}$ and the target vector $\mathbf{y}$. Similarly, the solution of the problem (2) is optimal for the given feature subset $\mathcal{A}$.

One could combine the data fitting problem with feature selection procedure

$$\mathbf{W}^*, \mathbf{a}^* = \mathop{\arg\min}_{\substack{\mathbf{W} \in \mathbb{R}^{n \times K} \\ \mathbf{a} \in \mathbb{B}^n}} S\left(\mathbf{W}, \mathbf{a}|\mathbf{X}, \mathbf{y}, \mathbf{h}\right). \tag{4}$$

The problem (4) is mixed integer optimization problem which includes continuous variables $\mathbf{W}$ and binary variables $\mathbf{a}$.

This study explores the softmax probability function

$$p(y = k|\mathbf{x}, \mathcal{A}, \mathbf{W}) = \frac{\exp\left(\mathbf{x}_{\mathcal{A}}^{\mathsf{T}}\mathbf{w}_{\mathcal{A}}^{(k)}\right)}{\sum_{j=1}^{K}\exp\left(\mathbf{x}_{\mathcal{A}}^{\mathsf{T}}\mathbf{w}_{\mathcal{A}}^{(j)}\right)},$$

where $\mathbf{x}_{\mathcal{A}}, \mathbf{w}_{\mathcal{A}}^{(j)}$ is the reduced object and parameter vector consisting of elements with indices in $\mathcal{A}$, i.e. non-zero $\mathbf{a}$ elements.

The considered error function is

$$S(\mathbf{W}, \mathcal{A}|\mathbf{X}, \mathbf{y}, \mathbf{h}) = -\sum_{i=1}^{m}\sum_{k=1}^{K}\mathbb{I}\{y_i = k\}\log p(y = k|\mathbf{x}, \mathcal{A}, \mathbf{W}), \tag{5}$$

where the function $\mathbb{I}(A)$ equals to 1 if A is true or 0 otherwise.

The error function (5) coincides with the logistic regression error function in the case of binary classification $K = 2$.

1 Quadratic Optimization Approach to the Multicollinearity Problem

In (Katrutsa stress), it shown that none of the feature selection methods considered (LARS, Lasso, Ridge, Stepwise and Genetic algorithm) solve problem (2) and give a model that is simultaneously stable, accurate and nonredundant. In contrast, in (Katrutsa QP) it was proposed quadratic programming approach to solving the multicollinearity problem for regression problem.

The main idea of the proposed approach is to minimize the number of similar features and maximize the number of relevant features. To formalize this idea it was represented the criterion Q from problem (3) as a quadratic function

$$Q(\mathbf{a}) = \mathbf{a}^{\mathsf{T}}\mathbf{Q}\mathbf{a} - \mathbf{b}^{\mathsf{T}}\mathbf{a}, \tag{6}$$

where $\mathbf{Q} \in \mathbb{R}^{n\times n}$ is a matrix of pairwise feature similarities, and $\mathbf{b} \in \mathbb{R}^{n}$ is a vector of the relevances of features to the target vector.

To compute the matrix $\mathbf{Q}$ and the vector $\mathbf{b}$, it was introduced the functions Sim and Rel:

$$\begin{aligned}&\text{Sim: } \mathcal{J} \times \mathcal{J} \to [0, 1],\\ &\text{Rel: } \mathcal{J} \to [0, 1].\end{aligned} \tag{7}$$

These functions are problem-dependent, defined by the user before performing feature selection, and indicate how to measure feature similarity (Sim) and relevance to the target vector (Rel).

The matrix $\mathbf{Q}$ is computed using Sim:

$$\mathbf{Q} = [q_{ij}] = \text{Sim}\left(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j\right), \tag{8}$$

and the vector $\mathbf{b}$ is computed using Rel:

$$\mathbf{b} = [b_i] = \text{Rel}\left(\boldsymbol{\chi}_i\right). \tag{9}$$

The optimum feature index set $\mathcal{A}^*$ is defined by solution of the optimization problem

$$\mathbf{a}^* = \arg\min_{\mathbf{a}\in\mathbb{B}^n} \mathbf{a}^\mathsf{T}\mathbf{Q}\mathbf{a} - \mathbf{b}^\mathsf{T}\mathbf{a}, \tag{10}$$

One of the example of defining the Sim and Rel function proposed in (Katrutsa QP) is the following

$$q_{ij} = \text{Sim}(\boldsymbol{\chi}_i, \boldsymbol{\chi}_j) = |\hat{\rho}_{ij}|, \tag{11}$$

$$b_i = \text{Rel}\left(\boldsymbol{\chi}_i|\mathbf{v}\right) = |\hat{\rho}_{iy}|, \tag{12}$$

where $\hat{\rho}_{ij}$ are sample correlation coefficients between features $\boldsymbol{\chi}_i$ and $\boldsymbol{\chi}_j$, $\hat{\rho}_{iy}$ are the sample correlation coefficient between feature $\boldsymbol{\chi}_i$ and target vector $\mathbf{y}$. The sample correlation coefficient is defined as

$$\hat{\rho}_{ij} = \frac{(\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}_i})^\mathsf{T}(\boldsymbol{\chi}_j - \overline{\boldsymbol{\chi}_j})}{\|\boldsymbol{\chi}_i - \overline{\boldsymbol{\chi}_i}\|_2 \|\boldsymbol{\chi}_j - \overline{\boldsymbol{\chi}_j}\|_2}, \qquad \overline{\boldsymbol{\chi}_i} = [\overline{\chi}_i, \dots, \overline{\chi}_i]^\mathsf{T}, \qquad \overline{\boldsymbol{\chi}_j} = [\overline{\chi}_j, \dots, \overline{\chi}_j]^\mathsf{T} \tag{13}$$

where $\overline{\chi}_i$ and $\overline{\chi}_j$ are the means of $\boldsymbol{\chi}_i$ and $\boldsymbol{\chi}_j$ respectively.

We propose to extend this approach to the classification problems. The definition of the Q matrix is the same as in (11). In order to define relevancies to the target vector we build the model with only one feature $\boldsymbol{\chi}_i$ and the target vector $\mathbf{y}$. Let denote by $\hat{\mathbf{y}}_i$ the predictions of ith such model

$$\hat{\mathbf{y}}_i = \text{sign}\left(v_i^{(0)} + v_i^{(1)}\boldsymbol{\chi}_i\right), \quad i = 1, \dots, n.$$

We propose to use logistic regression to fit parameters $\{(v_i^{(0)}, v_i^{(1)})\}_{i=1}^n$. Then the elements of $\mathbf{b} = [b_i]$ are defined as the absolute values of the sample correlation coefficient between the feature $\boldsymbol{\chi}_i$ and the target vector $\mathbf{y}$:

$$b_i = \text{Rel}\left(\boldsymbol{\chi}_i\right) = |\hat{\rho}_{\hat{y}_i y}|. \tag{14}$$

1.1 Convex representation of the feature selection problem

The quadratic programming approach to the multicollinearity problem leads to problem (10), which is NP-hard because of the Boolean domain. Therefore, this problem is approximated with a convex optimization problem to solve it efficiently.

Assume that Sim gives a positive semidefinite matrix $\mathbf{Q}$. Then the quadratic form (6) is a convex function. To represent problem (10) in convex form, we have to replace the non-convex set $\mathbb{B}^n$ with a convex set. A natural way to achieve this is to use the convex hull of $\mathbb{B}^n$:

$$\text{Conv}(\mathbb{B}^n) = [0, 1]^n.$$

Problem (10) is now approximated by the following *convex optimization problem*:

$$\begin{aligned} \mathbf{z}^* &= \arg\min_{\mathbf{z}\in[0,1]^n} \mathbf{z}^\mathsf{T}\mathbf{Q}\mathbf{z} - \mathbf{b}^\mathsf{T}\mathbf{z} \\ &\text{s.t. } \|\mathbf{z}\|_1 \leqslant 1. \end{aligned} \tag{15}$$

This constraint is added to show that $\mathbf{z}^*$ can be treated as a vector of non-normalized probabilities for every feature to be selected in the active set $\mathcal{A}^*$.

To return from a continuous vector $\mathbf{z}^*$ to a Boolean vector $\mathbf{a}^*$ and consequently to an active set $\mathcal{A}^*$ (see equation (1)), we use the *significance threshold* τ.

The value τ is a significance threshold if $z_j^* > \tau$ if and only if $a_j^* = 1$ and $j \in \mathcal{A}^*$.

Tuning the value of τ is problem-dependent and is based on the appropriate error rate, the number of features selected and the values of the evaluation criteria. To obtain the most appropriate significance threshold for a specific problem, we need to set a range of values for τ.

Mixed Integer Optimization Software

We use python-embedded CVXPY free software package for solving mixed integer optimization problem.

The considered types of problems which have open source solvers:

- LP — Linear Programming;
- SOCP — Second-Order Cone Programming;
- SDP — Semidefinite Programming;
- EXP — Problems with Exponential cone constraints;
- MIP — Mixed Integer Programming

Table 1 shows available solvers and types of the problems which they can handle.

Table 1: Solvers vs Problems

	LP	SOCP	SDP	EXP	MIP
CBC	X				X
GLPK	X				
GLPK-MI	X				X
Elemental	X	X			
ECOS	X	X		X	
ECOS-BB	X	X		X	X
GUROBI	X	X			X
MOSEK	X	X	X		
CVXOPT	X	X	X	X	
SCS	X	X	X	X	

In this paper GUROBI solver is used.

Computational experiment

This section provides experiments on the real datasets to show the perfomance of the proposed approach. In the computational experiment we compare the following methods: Quadratic Programming approach, Mixed Integer Programming, Lasso, Ridge, Elastic Net.