Список основных обозначений

В. В. Стрижов
Вычислительный центр РАН

Матрицы обозначены заглавными буквами, векторы — полужирными прописными буквами, множества — каллиграфическими буквами.

$\mathbb{R}$ — множество действительных чисел

$\mathbb{N}$ — множество натуральных чисел

$\mathsf{E}(y)$ — математическое ожидание случайной величины

$\mathsf{D}(y)$ — дисперсия случайной величины

$\mathbf{X}$ — матрица плана, $\mathbf{X} = [x_j^i] \in \mathbb{R}^{m\times n}$, множество (объектов) элементов выборки $\mathbf{X} = [\mathbf{x}_1^\intercal, \dots, \mathbf{x}_m^\intercal]^\intercal$

$\mathbf{x}_i$ — i-й элемент выборки, $\mathbf{x}_i \in \mathbb{R}^n$

$\mathbf{x}$ — многомерная свободная переменная, $\mathbf{x} = [x_1, \dots, x_n]^\intercal \in \mathbb{R}^n$

$\mathbf{X}_\mathcal{A}$ — подмножество признаков, заданное индексным множеством $\mathcal{A}$

$\boldsymbol{\chi}_j$ — реализации j-й свободной переменной, признак, j-й столбец матрицы $\mathbf{X}$, $\boldsymbol{\chi}_j = [x_{1j}, \dots, x_{mj}]^\intercal \in \mathbb{R}^m$

y — зависимая переменная, случайная величина

$\mathbf{y}$ — зависимые переменные, многомерная случайная величина $\mathbf{y} = [y_1, \dots, y_m]^\intercal \in \mathbb{R}^m$

$\mathfrak{D}$ — выборка, множество пар $\{(\mathbf{x}_i, y_i) | i = 1, \dots, m\}$, также $\mathfrak{D} = (\mathbf{X}, \mathbf{y})$

$\mathcal{I}$ — множество индексов (объектов) элементов выборки; разбиение множества $\mathcal{I} = \mathcal{L} \sqcup \mathcal{T}$

$\mathcal{B}$ — множество индексов опорных объектов, $\mathcal{B} \subseteq \mathcal{I}$

$\mathcal{J}$ — множество индексов свободных переменных (признаков)

$\mathcal{A}$ — множество индексов активных признаков, $\mathcal{A} \subseteq \mathcal{J}$

m — число зависимых переменных, размерность пространства зависимых переменных, $m = |\mathcal{I}|$

n — число свободных переменных, размерность пространства свободной переменной, $n = |\mathcal{J}|$

f — регрессионная модель, $f = f(\mathbf{w}, \mathbf{x})$, по определению $f : (\mathbf{w}, \mathbf{x}) \mapsto \hat{y}$

$\mathbf{f}$ — регрессионная модель (вектор-функция), $\mathbf{f} = [f(\mathbf{w}, \mathbf{x}_1), \dots, f(\mathbf{w}, \mathbf{x}_m)]^\intercal$

$\mathbf{w}$ — вектор параметров $\mathbf{w} = [w_1, \dots, w_n]^\intercal$ модели

$\boldsymbol{\varepsilon}$ — многомерная случайная величина $\boldsymbol{\varepsilon} = [\varepsilon_1, \dots, \varepsilon_m]^\intercal$, вектор регрессионных остатков

σ_ε^2 — дисперсия элементов вектора регрессионных остатков, описываемых ковариационной матрицей $\sigma_\varepsilon^2\mathbf{I}$

$\mathbf{A}$ — обратная ковариационная матрица многомерной случайной величины $\mathbf{w}$

$\mathbf{B}$ — обратная ковариационная матрица многомерной случайной величины $\mathbf{y}$, вариант — $\boldsymbol{\varepsilon}$

$\mathbf{J}$ — матрица Якоби фукнции f с элементами $J_{ij} = \left[\frac{\partial f(\mathbf{w},\mathbf{x}_i)}{\partial w_j}\right], i \in \mathcal{I}, j \in \mathcal{J}$

∇S — градиент функции ошибки $S(\mathbf{w})$ в пространстве параметров $\mathcal{W} \ni \mathbf{w}$, $\nabla S(\mathbf{w}) = \left[\frac{\partial S(\mathbf{w})}{\partial w_j}\right], j \in \mathcal{J}$

$\mathbf{H}$ — матрица Гессе фукнции f с элементами $\mathbf{H} = \left[\frac{\partial^2 S(\mathbf{w})}{\partial w_j w_k}\right], j, k \in \mathcal{J}$, $\mathbf{H} = \nabla^2 S(\mathbf{w})$

g — порождающая функция, $g = g(\mathbf{w}, \cdot)$

$\mathcal{G}$ — множество порождающий функций, $\mathcal{G} = \{g\}$

$\mathcal{F}$ — множество индуктивно-порожденных регрессионных моделей, $\mathcal{F} = \{f\}$

S — функция ошибки, $S = S(\mathbf{w})$, полный вариант $S = S(\mathbf{w}|\mathcal{D}, f)$ при заданной выборке $\mathcal{D}$ и фиксированной модели f

$[\cdot]$ — элементы матрицы или вектора, например: матрица $\mathbf{X} = [x_{ij}]$, вектор $\mathbf{y} = [y_1, \ldots, y_m]^\mathsf{T}$

$\|\cdot\|$ — евклидова норма вектора $\|\cdot\|_2$, если нижним индексом не указано иное

$\langle\cdot,\cdot\rangle$ — скалярное произведение двух векторов

Справочная информация. Взятие градиента или производной по элементам вектора:

$$\frac{\partial \mathbf{x}^\mathsf{T}\mathbf{A}\mathbf{x}}{\partial \mathbf{x}} = \mathbf{x}^\mathsf{T}\mathbf{A} + \mathbf{A}\mathbf{x};$$

поэлементно,

$$\frac{\partial}{\partial x_i}\sum_{j,k=1}^{m} a_j b_{ki} a_k = \sum_{j=1}^{n} x_j b_{ji} + \sum_{k=1}^{n} b_{ik} x_k.$$

02.10.2013

Comprehensive study of feature selection methods to solve multicollinearity problem according to evaluation criteria

A. M. Katrutsa[a,b,*], V. V. Strijov[a]

[a]*Moscow Institute of Physics and Technology, Institutskiy lane 9, Dolgoprudny city, 141700, Russian Federation*
[b]*Skolkovo Institute of Science and Technology, Nobel St., 3, Skolkovo, 143025, Russian Federation*

Abstract

This paper presents a comprehensive analysis of multicollinearity problem in data fitting. Data fitting is stated as a single-objective optimization problem where an objective function indicates the error of approximation the target vector with a some function of given features. The linear dependence between features means that the multicollinerity problem exists and leads to unstability and redundancy of the built model. These problems are addressed by introducing a feature selection method based on a quadratic programming approach. This approach takes into account the positions of the features and the target vector and select features according to relevance and similarity measures, which are defined by a user. Therefore, the built model is less redundant and more stable. To evaluate the quality of the proposed feature selection method and compare it with others we use different criteria to measure unstability and redundancy. In the experiments we compare proposed approach with other feature selection methods: LARS, Lasso, Ridge, Stepwise and Genetic algorithm. We show that the quadratic programming approach gives the best results according to considered criteria on the test and real data sets.

Keywords: data fitting, feature selection, multicollinearity, quadratic programming, evaluation criteria, test data sets

1. Introduction

This paper addresses the multicollinearity problem and proposes its comprehensive analysis. *Multicollinearity* is a strong correlation between features, which affect the target vector simultaneously. Due to multicollinearity the common methods of regression analysis like least squares build unstable models of excessive complexity. The formal definitions of model stability, complexity and redundancy are given in Section 5.

To treat multicollinearity problem feature selection methods are used. Most of previously proposed feature selection methods that solve multicollinearity problem are based on different heuristics [1, 2], greedy searches [3, 4] or regularization techniques [5, 6]. These approaches do not take into account the data set configuration and do not guarantee optimality of the obtained feature subset [7]. In constrast, we propose to use *quadratic programming approach* [8] to solve multicollinearity problem that corrects disadvantages mentioned above. This approach is based on two ideas: the first one is to represent features as some binary vector, and the second one is to define the feature subset quality criterion as quadratic form. The first term of the quadratic

*Corresponding author
Email address: aleksandr.katrutsa@phystech.edu (A. M. Katrutsa)

parameters are changing continuously.

2 Feature selection problem statement

Let $\mathfrak{D} = \{(\mathbf{X}, \mathbf{y})\}$ be the given data set, where the design matrix

$$\mathbf{X} = [\boldsymbol{\chi}_1, \ldots, \boldsymbol{\chi}_j, \ldots, \boldsymbol{\chi}_n],\ \mathbf{X} \in \mathbb{R}^{m \times n} \text{ and } j \in \mathcal{J} = \{1, \ldots, n\}.$$

The vector $\boldsymbol{\chi}_j$ is called the j-th feature and the vector $\mathbf{y} = [y_1, \ldots, y_m]^\intercal \in \mathbb{Y} \subset \mathbb{R}^m$ is called the target vector. Assume that the target vector $\mathbf{y}$ and design matrix $\mathbf{X}$ are related through the following equation:

$$\mathbf{y} = \mathbf{f}(\mathbf{w}, \mathbf{X}) + \boldsymbol{\varepsilon}, \tag{1}$$

where $\mathbf{f}$ maps the cartesian product of the feasible parameter space and the space of the $m \times n$ matrices to the target vector domain, and $\boldsymbol{\varepsilon}$ is the residual vector. The data fit problem is to estimate the parameter vector $\mathbf{w}^*$,

$$\mathbf{w}^* = \arg\min_{\mathbf{w} \in \mathbb{R}^n} S(\mathbf{w} | \mathfrak{D}_{\mathcal{L}}, \mathcal{A}, \mathbf{f}), \tag{2}$$

where S is the error function. The set $\mathfrak{D}_{\mathcal{L}} \subset \mathfrak{D}$ is a training set and the set $\mathcal{A} \subseteq \mathcal{J}$ is the *active index set* used in computing the error function S. In the stresstest procedure we use the quadratic error function

$$S = \|\mathbf{y} - \mathbf{f}(\mathbf{w}, \mathbf{X})\|_2^2 \tag{3}$$

and the linear regression function $\mathbf{f}(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w}$. The introduced stresstest procedure could be applied to the generalised linear model selection algorithms, where the model is $\mathbf{f} = \boldsymbol{\mu}^{-1}(\mathbf{X}\mathbf{w})$ and $\boldsymbol{\mu}$ is a link function.

Definition 2.1 Let $\mathcal{A}^*$ denote *the optimum index set*, the solution of the problem

$$\mathcal{A}^* = \arg\min_{\mathcal{A} \subseteq \mathcal{J}} S_{\mathfrak{m}}(\mathcal{A} | \mathbf{w}^*, \mathfrak{D}_{\mathcal{C}}, \mathbf{f}), \tag{4}$$

where $\mathfrak{D}_{\mathcal{C}} \subset \mathfrak{D}$ is the test set, $\mathbf{w}^*$ is the solution of the problem (2) and $S_{\mathfrak{m}}$ is an error function corresponding to a feature selection method $\mathfrak{m}$ (5).

The feature selection problem (4) is to find the optimum index set $\mathcal{A}^*$. It must exclude indices of noisy and multicollinear features. It is expected that if one uses features indexed by the set $\mathcal{A}^*$ then it brings more stable solution of the problem (2), in comparison to the case of $\mathcal{A} \equiv \mathcal{J}$.

In the computational experiment we consider the feature selection methods from the set $\mathfrak{M} = \{\text{Lasso, LARS, Stepwise, ElasticNet, Ridge}\}$.

Definition 2.2 A feature selection method $\mathfrak{m} \in \mathfrak{M}$ is a map from the complete index set $\mathcal{J}$ to active index set $\mathcal{A} \subseteq \mathcal{J}$:

$$\mathfrak{m} : \mathcal{J} \to \mathcal{A}. \tag{5}$$

According to this definition we consider the terms feature selection problem and the model selection problem to be synonyms.

Definition 2.3 Let a model be a pair $(\mathbf{f}, \mathcal{A})$, where $\mathcal{A} \subseteq \mathcal{J}$ is an index set. The model selection problem is to find the optimum pair $(\mathbf{f}^*, \mathcal{A}^*)$ which minimizes the error function S (3).

Definition 2.4 Call *the model complexity* C the cardinality of the active index set $\mathcal{A}$, number of the selected features:

$$C = |\mathcal{A}|.$$

Definition 2.5 Define *the model stability* R be logarithm of the condition number κ of the matrix $\mathbf{X}^\intercal\mathbf{X}$:

$$R = \ln \kappa = \ln \frac{\lambda_{\max}}{\lambda_{\min}},$$

where $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and the minimum non-zero eigenvalue of the matrix $\mathbf{X}^\intercal\mathbf{X}$. The features with indices from the corresponding active set $\mathcal{A}$ are used in computing the condition number κ.

3 Multicollinearity analysis in feature selection

In this section we give definitions of multicollinear features, correlated features and features correlated with the target vector. In the following subsections we list and study the multicollinearity criteria.

Assume that the features $\boldsymbol{\chi}_j$ and the target vector $\mathbf{y}$ are normalized:

$$\|\mathbf{y}\|_2 = 1 \text{ and } \|\boldsymbol{\chi}_j\|_2 = 1,\ j \in \mathcal{J}. \tag{6}$$

Consider active index subset $\mathcal{A} \subseteq \mathcal{J}$.

Definition 3.1 The features with indices from the set $\mathcal{A}$ are called *multicollinear* if there exist the index j, the coefficients a_k, the index $k \in \mathcal{A} \setminus j$ and sufficiently small positive number $\delta > 0$ such that

$$\left\| \boldsymbol{\chi}_j - \sum_{k \in \mathcal{A} \setminus j} a_k \boldsymbol{\chi}_k \right\|_2^2 < \delta. \tag{7}$$

The smaller δ the higher *degree of multicollinearity*.

Definition 3.2 Call the features indexed i, j be *correlated* if there exists sufficiently small positive number $\delta_{ij} > 0$ such that:

$$\|\boldsymbol{\chi}_i - \boldsymbol{\chi}_j\|_2^2 < \delta_{ij}. \tag{8}$$

From this definition it follows that $\delta_{ij} = \delta_{ji}$. In the special case $a_k = 0$ $k \neq j$ and $a_k = 1$ $k = j$ the inequalities (8) and (7) are identically.

Definition 3.3 A feature $\boldsymbol{\chi}_j$ is called *correlated with the target vector* $\mathbf{y}$ if there exists sufficiently small positive number $\delta_{yj} > 0$ such that

$$\|\mathbf{y} - \boldsymbol{\chi}_j\|_2^2 < \delta_{yj}.$$

Further used the following notations RSS (Residual Sum of Squares) and TSS (Total Sum of Squares):

$$\text{RSS} = S(\mathfrak{D}_{\mathcal{L}}, \mathbf{w}^*) = \|\boldsymbol{\varepsilon}\|_2^2 \quad \text{and} \quad \text{TSS} = \sum_{i=1}^{m}(y_i - \overline{y})^2, \text{ where } \overline{y} = \frac{1}{m}\sum_{i=1}^{m} y_i. \tag{9}$$

3.1 Variance inflation factor

The variance inflation factor VIF_j is used as a multicollinearity indicator [17]. The VIF_j is defined for j-th feature and shows a linear dependence between j-th feature and the other features.

To compute VIF_j estimate the parameter vector $\mathbf{w}^*$ according to the problem (1) assuming $\mathbf{y} = \boldsymbol{\chi}_j$ and extracting j-th feature from the index set $\mathcal{J} = \mathcal{J} \setminus j$. The functions RSS and TSS are computed similar to (9). The VIF_j is computed with the following equation:

$$\text{VIF}_j = \frac{1}{1 - R_j^2},$$

where $R_j^2 = 1 - \frac{\text{RSS}}{\text{TSS}}$ is the coefficient of determination.

According to [17] any $\text{VIF}_j \gtrsim 5$ indicates that the associated elements of the vector $\mathbf{w}^*$ are poorly estimated because of multicollinearity. Denote by VIF the maximum value of VIF_j for all $j \in \mathcal{J}$:

$$\text{VIF} = \max_{j \in \mathcal{J}} \text{VIF}_j.$$

However, VIF_j can be infinitely large for some features. In this case it is impossible to determine which features must be removed from the active set. This is major disadvantage of the variance inflation factor.

Another multicollinearity indicator is the condition number κ of the matrix $\mathbf{X}^\intercal\mathbf{X}$. The condition number is defined as:

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}},$$

where the $\lambda_{\max}$ and $\lambda_{\min}$ are the maximum and minimum non-zero eigenvalues of the matrix $\mathbf{X}^\intercal\mathbf{X}$.

The condition number shows how much does the matrix $\mathbf{X}^\intercal\mathbf{X}$ close to the singular matrix. The larger κ the more ill-conditioned matrix $\mathbf{X}^\intercal\mathbf{X}$.

3.2 The Belsley criterion

To detect and remove indices of the multicollinear features from the active index set we state the direct optimization problem using the Belsley criterion. We propose the new criterion

Generation of simple structured IR functions by genetic algorithm without stagnation

Kulunchakov A. S.[a], Strijov V. V.[b]

[a] *Moscow Institute of Physics and Technology*
[b] *Computing Centre of the Russian Academy of Sciences*

Abstract

This paper investigates an approach to construct new ranking models for Information Retrieval. The IR ranking model depends on the document description. It includes the term frequency and document frequency. The model ranks documents upon a user request. The quality of the model is defined by the difference between the documents, which experts assess as relative to the request, and the ranked ones. To boost the model quality a modified genetic algorithm was developed. It generates models as superpositions of primitive functions and selects the best according to the quality criterion. The main impact of the research if the new technique to avoid stagnation and to control structural complexity of the consequently generated models. To solve problems of stagnation and complexity, a new criterion of model selection was introduced. It uses structural metric and penalty functions, which are defined in space of generated superpositions. To show that the newly discovered models outperform the other state-of-the-art IR scoring models the authors perform a computational experiment on TREC datasets. It shows that the resulted algorithm is significantly faster than the exhaustive one. It constructs better ranking models according to the MAP criterion. The obtained models are much simpler than the models, which were constructed with alternative approaches. The proposed technique is significant for developing the information retrieval systems based on expert assessments of the query-document relevance.

Keywords: information retrieval, evolutionary stagnation, ranking function, genetic programming, overfitting

Email addresses: kulu-andrej@yandex.com (Kulunchakov A. S.), strijov@gmail.com (Strijov V. V.)

to have as high quality as the stored superpositions. This superposition highly probably will be eliminated. Therefore the population will pass to the next iteration without changes. The genetic algorithm stops actual generation.

To outperform the ranking functions found in [2], one needs to extend the set of superpositions considered there. To perform it, a modified genetic algorithm is proposed. First, it detects evolutionary stagnation and replaces the worst stored superpositions with random ones. This detection is implemented with a structural metric on superpositions. Regularizers solve the problem of overfitting. They penalize the excessive structural complexity of superpositions. The paper analyzes various pairs regularizer-metric and chooses the pair providing a selection of better ranking superpositions. All strengths and weakness of compared approaches are summarized in Table 1.

The paper [2] uses TREC collections to test ranking functions. To make the comparison of approaches consistent, the present paper also use these collections. The collection TREC-7 (trec.nist.gov) is used as the train dataset to evaluate quality of generated superpositions. The collections TREC-5, TREC-6, TREC-8 are used as test datasets to test selected superpositions.

2. Problem statement

There given a collection C consisting of documents $\{d_i\}_{i=1}^{|C|}$ and queries $Q = \{q_j\}_{j=1}^{|Q|}$. For each query $q \in Q$ some documents C_q from C are ranked by experts. These ranks g are binary

$$g : Q \times C_q \to \mathbb{Y} = \{0, 1\},$$

where 1 corresponds to relevant documents and 0 to irrelevant.

To approximate g, superpositions of grammar elements are generated. The grammar $\mathfrak{G}$ is a set $\{g_1, \ldots, g_m, x_w^d, y_w\}$, where each g_i stands for an mathematical function and x_w^d, y_w stand for variables. These variables are tf-idf features of *document-query* pair (d, q). Feature x_w^d is a frequency of the word $w \in q$ in d, feature y_w is a frequency of w in C:

$$x_w^d = t_d^w \log\left(1 + \frac{l_a}{l_d}\right), \quad y_w = \frac{N_w}{|C|}, \tag{1}$$

where N_w is the number of documents from C containing w, t_d^w is the frequency of w in d, l_d is the number of words in d (the size of a document d), l_a is an average size of documents in C. Each superposition f of grammar elements is stored as a directed labeled tree T_f with vertices labeled by elements from $\mathfrak{G}$. The set of these superpositions is defined as $\mathfrak{F}$.

The value of f on a pair (d, q) is defined as a sum of its values on (d, w), where w is a word from q:

$$f(d, q) = \sum_{w \in q} f(x_w^d, y_w).$$

The superposition f ranks the documents for each q. The quality of f is the mean average precision [1]

$$\mathrm{MAP}(f, C, Q) = \frac{1}{|Q|} \sum_{q=1}^{Q} \mathrm{AveP}(f, q),$$

where

$$\mathrm{AveP}(f, q) = \frac{\sum_{k=1}^{|C_q|} \big(\mathrm{Prec}(k) \times g(k)\big)}{\sum_{k=1}^{|C_q|} \mathrm{Rel}(k)}, \quad \mathrm{Prec}(k) = \frac{\sum_{s=1}^{k} g(s)}{k},$$

where $g(k) \in \{0, 1\}$ is a relevance of the k-th document from C.

This paper aims at finding the superposition f, which maximizes the following quality function

$$f^* = \operatorname*{argmax}_{f \in \mathfrak{F}} \mathcal{S}(f, C, Q), \quad \mathcal{S}(f, C, Q) = \mathrm{MAP}(f, C, Q) - \mathrm{R}(f), \tag{2}$$

where R is a regularizer controlling the structural complexity of f.

The exhaustive algorithm in [2] generates random ranking superpositions consisting at most of 8 elements of the grammar $\mathfrak{G}$. Let $\mathfrak{F}_0$ be the set of the best superpositions selected in [2]. The solution f^* is compared with the superpositions from $\mathfrak{F}_0$ with respect to to MAP.

3. Generation of superpositions

IR ranking functions are superpositions of expert-given primitive functions. These superpositions are generated by the genetic algorithm. It uses an expertly given grammar $\mathfrak{G}$ and constructs superpositions of its elements. On each iteration it keeps stores a population of the best selected superpositions. To update them and pass to the next iteration, it generates new superpositions with use of the stored ones. Since the superpositions are represented as trees, the algorithm applies crossover $c(f, h)$ and mutation $m(f)$ operations to the stored trees

$$c(f, h) : \mathfrak{F} \times \mathfrak{F} \to \mathfrak{F}, \quad m(f) : \mathfrak{F} \to \mathfrak{F},$$

Definition 1. *Crossover operation $c(f, h) : \mathfrak{F} \times \mathfrak{F} \to \mathfrak{F}$ produces a new superpositions from given f and h. This operation represents f and h as trees, uniformly selected a subtree for each of them and swaps these subtrees.*

The value of f on a pair (d, q) is defined as a sum of its values on (d, w), where w is a word from q:

$$f(d, q) = \sum_{w \in q} f(x_w^d, y_w).$$

The superposition f ranks the documents for each q. The quality of f is the mean average precision [1]

$$\mathrm{MAP}(f, C, Q) = \frac{1}{|Q|} \sum_{q=1}^{Q} \mathrm{AveP}(f, q),$$

where

$$\mathrm{AveP}(f, q) = \frac{\sum_{k=1}^{|C_q|} \big(\mathrm{Prec}(k) \times g(k)\big)}{\sum_{k=1}^{|C_q|} \mathrm{Rel}(k)}, \quad \mathrm{Prec}(k) = \frac{\sum_{s=1}^{k} g(s)}{k},$$

where $g(k) \in \{0, 1\}$ is a relevance of the k-th document from C.

This paper aims at finding the superposition f, which maximizes the following quality function

$$f^* = \operatorname*{argmax}_{f \in \mathfrak{F}} \mathcal{S}(f, C, Q), \quad \mathcal{S}(f, C, Q) = \mathrm{MAP}(f, C, Q) - \mathrm{R}(f), \tag{2}$$

where R is a regularizer controlling the structural complexity of f.

The exhaustive algorithm in [2] generates random ranking superpositions consisting at most of 8 elements of the grammar $\mathfrak{G}$. Let $\mathfrak{F}_0$ be the set of the best superpositions selected in [2]. The solution f^* is compared with the superpositions from $\mathfrak{F}_0$ with respect to to MAP.

3. Generation of superpositions

IR ranking functions are superpositions of expert-given primitive functions. These superpositions are generated by the genetic algorithm. It uses an expertly given grammar $\mathfrak{G}$ and constructs superpositions of its elements. On each iteration it keeps stores a population of the best selected superpositions. To update them and pass to the next iteration, it generates new superpositions with use of the stored ones. Since the superpositions are represented as trees, the algorithm applies crossover $c(f, h)$ and mutation $m(f)$ operations to the stored trees

$$c(f, h) : \mathfrak{F} \times \mathfrak{F} \to \mathfrak{F}, \quad m(f) : \mathfrak{F} \to \mathfrak{F},$$

Definition 1. *Crossover operation $c(f, h) : \mathfrak{F} \times \mathfrak{F} \to \mathfrak{F}$ produces a new superpositions from given f and h. This operation represents f and h as trees, uniformly selected a subtree for each of them and swaps these subtrees.*

Motrenko A., Strijov V., Weber G.-W. Bayesian sample size estimation for logistic regression // Journal of Computational and Applied Mathematics, 2014, 255 : 743-752.

Sample Size Bayesian Estimation for Logistic Regression[☆]

Anastasiya Motrenko[a], Vadim Strijov[b], Gerhard-Wilhelm Weber[c]

[a]*Moscow Institute of Physics and Technology, Moscow, Russia*
[b]*Computing Center of the Russian Academy of Sciences, Moscow, Russia*
[c]*Institute of Applied Mathematics, Middle East Technical University, Ankara, Turkey*

Abstract

The problem of sample size estimation is important in the medical applications, especially in the cases of expensive measurements of immune biomarkers. The papers describes the problem of logistic regression analysis including model feature selection and includes the sample size determination algorithms, namely methods of univariate statistics, logistics regression, cross-validation and Bayesian inference. The authors, treating the regression model parameters as a multivariate variable, propose to estimate the sample size using the distance between parameter distribution functions on cross-validated data sets.

Keywords: logistic regression, sample size, feature selection, Bayesian inference, Kullback-Leibler divergence

1. Introduction

The paper is devoted to the logistic regression analysis [1], applied to classification problems in biomedicine. A group of patients is investigated as a sample set; each patient is described with a set of features, named as biomarkers and is classified into two classes. Since the patient measurement is expensive the problem is to reduce number of measured features in order to increase sample size.

The responsive variable is assumed to follow a Bernoulli distribution. Also, parameters of the regression function are evaluated [2, 3].

With given set of features, the model is excessively complex. The problem is to select a set of features of a smaller size, that will classify patients effectively. In logistic regression, features are usually selected by stepwise regression [4, 5]. In the computational experiment, exhaustive search is implemented. This makes the experts sure that all possible combinations of the features were considered. The authors use the area under ROC curve [6] as the optimum criterion in the feature selection procedure.

The problem of classification is associated with minimum sample size determination. In the paper, the following methods are discussed:

[☆]This project was supported by the Russian Foundation for Basic Research, grant 12-07-31095.
Email address: strijov@ccas.ru (Vadim Strijov)

1. Method of confidence intervals: a method of univariate statistics.
2. Method of sample size evaluation in logistic regression [7, 8]: unlike the previous one, this method considers the distribution of the responsive variable according to the logistic regression model.
3. Cross-validation: a method which evaluates sample size by observing potential overfitting [9, 10].
4. Comparing different subsets of the same sample by computing Kullback-Leibler [11] divergence between probability density functions of model parameters, evaluated at these subsets.

The data, used while conducting computational experiment can be found here [12].

2. Classification problem

Consider the sample set $D = \{(\mathbf{x}_i, y_i) : i = 1, \ldots, m\}$, of m objects (patients). Each patient is described by n features (biomarkers), $\mathbf{x}_i \in \mathbb{R}^n$ and belongs to one of two classes: $y_i \in \{0, 1\}$. The logistic regression problem assumes that the vector of responsive variables $\mathbf{y} = [y_1, \ldots, y_m]^T$ is a vector of Bernoulli random variables, $y_i \sim \mathcal{B}(\theta_i)$ with the probability density function

$$p(\mathbf{y}|\mathbf{w}) = \prod_{i=1}^{m} \theta_i^{y_i}(1-\theta_i)^{1-y_i}. \tag{1}$$

We use the maximim likelihood method, write the error function for (1) as

$$E(\mathbf{w}) = -\ln p(\mathbf{y}|\mathbf{w}) = -\sum_{i=1}^{m} y_i \ln \theta_i + (1-y_i)\ln(1-\theta_i). \tag{2}$$

find vector of parameters $\hat{\mathbf{w}}$ of regression function, one has to solve the following optimization problem:

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}\in\mathbb{R}^n} E(\mathbf{w}). \tag{3}$$

Let us define the probability of a case as

$$f(\mathbf{x}_i^T\mathbf{w}) = \frac{1}{1+\exp(-\mathbf{x}_i^T\mathbf{w})} = \theta_i. \tag{4}$$

To solve the problem (3), using

$$\frac{df(\xi)}{d\xi} = f(1-f),$$

we compute gradient of the error function $E(\mathbf{w})$:

$$\nabla E(\mathbf{w}) = -\sum_{i=1}^{m}\big(y_i(1-\theta_i) - (1-y_i)\theta_i\big)\mathbf{x}_i = \sum_{i=1}^{m}(\theta_i - y_i)\mathbf{x}_i = \mathbf{X}^T(\boldsymbol{\theta} - \mathbf{y}),$$

in which $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_m]^T$ and the matrix $\mathbf{X} = \left[\mathbf{x}_1^T, \ldots, \mathbf{x}_m^T\right]^T$ represents features sets.

Parameters are evaluated by Newton-Raphson method. Denote by $\mathbf{\Sigma}$ a diagonal matrix with diagonal elements $\Sigma_{ii} = \theta_i(1-\theta_i)$ $(i = 1, \ldots, m)$. Set the initial value $\mathbf{w} = [w_1, \ldots, w_n]^T$ of $\hat{\mathbf{w}}$

$$w_j = \sum_{i=1}^{m} y_i(1-y_i) \quad (j = 1, \ldots, n),$$

Then the $(k+1)$-th iteration of evaluation of $\hat{\mathbf{w}}$ is

$$\begin{gathered} \mathbf{w}_{k+1} = \mathbf{w}_k - (\mathbf{X}^T\mathbf{\Sigma}\mathbf{X})^{-1}\mathbf{X}^T(\boldsymbol{\theta} - \mathbf{y}) = \\ (\mathbf{X}^T\mathbf{\Sigma}\mathbf{X})^{-1}\mathbf{X}^T\mathbf{\Sigma}(\mathbf{X}\mathbf{w}_k - \mathbf{\Sigma}^{-1}(\boldsymbol{\theta} - \mathbf{y})). \end{gathered} \tag{5}$$

The process is repeated until the Euclidean distance $\| \mathbf{w}_{k+1} - \mathbf{w}_k \|$ is sufficiently small. Thus, the classification algorithm is defined as:

$$a(\mathbf{x}, c_0) = \operatorname{sign}\big(f(\mathbf{x}, \mathbf{w}) - c_0\big), \tag{6}$$

where c_0 is a cut-off value of regression function (4), defined by (7).

Quality of classification. Let us use an additional to (1) quality functional AUC, or the area under the ROC-curve. Introduce $\mathrm{TPR}(\xi)$, which stands for true positive rate

$$\mathrm{TPR}(\xi) = \frac{1}{m}\sum_{i=1}^{m}[a(\mathbf{x}_i, \xi) = 1][y_i = 1]$$

and $\mathrm{FPR}(\xi)$means the false positive rate

$$\mathrm{FPR}(\xi) = \frac{1}{m}\sum_{i=1}^{m}[a(\mathbf{x}_i, \xi) = 1][y_i = 0].$$

Here, the following denotation is used:

$$[y = 1] = \begin{cases} 1, & y = 1; \\ 0, & y \neq 1. \end{cases}$$

Thus, the bigger AUC value is, the better is the classifier.

Defining c_0 value. Every point $[\mathrm{FPR}(c_0), \mathrm{TPR}(c_0)]$ of the ROC-curve corresponds to some $c_0 \in [0, 1]$ value. As shown in figure 1, the most distant from segment [(0,0);(1,1)] point of the ROC-curve corresponds to the c_0 value used in (6):

$$\hat{c}_0 = \arg\max_{\xi\in[0,1]} \| \big(\mathrm{TPR}(\xi), \mathrm{FPR}(\xi)\big) - (\xi, \xi) \| = \arg\max_{\xi\in[0,1]} \sqrt{(TPR(\xi) - \xi)^2 - (FPR(\xi) - \xi)^2}. \tag{7}$$

Defining $\hat{c}_0$ includes computing AUC value and, therefore, computation of (6) and iterative estimation of parameters $\mathbf{w}$ according to (5).

Feature generation for classification and forecasting problems

N. P. Ivkin
Moscow Institute of Physics and Technology
ivkinnikita@gmail.com

Abstract

We propose a problem statement for analysis of complex objects such as video sequences with contents, e-mail letters with attached files, source codes of programs. The proposed problem statement helps to organize work on a project, to simplify code development and to reduce labor costs.

Feature generation problem statement

Let $\mathfrak{S}$ be a set of measurements such that

$$\mathfrak{S} = \{\mathfrak{s}_1, ..., \mathfrak{s}_m\}.$$

The element $\mathfrak{s}_i$ of the set $\mathfrak{S}$ can be a time series a video sequence or a scoring application. Let $\mathbf{y} = \{y_1, ..., y_m\}$ be a set of class labels, or target variables.

Together with the set $\mathfrak{S}$ a set $V = V(\mathfrak{S})$ is given. The set $V = V(\mathfrak{S})$ is called a vocabulary and contains knowledge about the set of measurements. The vocabulary can be obtained as the result of measurement structure analysis and used for model generation.

By $G = \{g_1, ..., g_n\}$ denote an expert-given set of primitive functions such that each function g_j maps an object $\mathfrak{s}_i$ to an element (i, j) of the design matrix $\mathbf{X}$:

$$g_j : \quad (\mathbf{b}_j, \mathfrak{s}_i, V) \mapsto x_{ij} \in \mathbb{R}^1,$$

where $\mathbf{b}_j$ is the set of parameters of the primitive function g_j. By f denote the regression model f together with the set of parameters $\mathbf{w}$. To find the optimal parameters $\hat{\mathbf{w}}$ we minimize a loss function $S(\mathbf{w}|f, \mathbf{X}, \mathbf{y})$ such that

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w} \in \mathbb{R}^n} S(\mathbf{w}|f, \mathbf{X}, \mathbf{y}).$$

Examples

In this section we investigate classification and forecasting problem statements as the examples of feature generation problem.

Linear regression. According to the regression problem statement the target variable y belongs to the set of real numbers, $y \in \mathbb{R}$. The model f maps each row of the matrix $\mathbf{X}$ to the set $\mathbb{R}$ such that

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \mathbf{X}\mathbf{w},$$

where $\mathbf{f} = [f(\mathbf{w}, \mathbf{x}_1), ..., f(\mathbf{w}, \mathbf{x}_m)]^T$. As an example of the loss function S, the sum-squared error can be considered:

$$S(\mathbf{w}|f, \mathbf{X}, \mathbf{y}) = \|\mathbf{f}(\mathbf{w}, \mathbf{X}) - \mathbf{y}\|_2^2.$$

Classification. According to the two-class classification problem the target variable y belongs to the set of class labels, $y \in \{0, 1\}$. Consider a logistic regression problem as an example of classification problem. The model f maps each row of the matrix $\mathbf{X}$ to the segment $[0, 1]$ such that

$$\mathbf{f}(\mathbf{w}, \mathbf{X}) = \frac{\mathbf{1}}{\mathbf{1} + \exp\left(-\mathbf{X}\mathbf{w}\right)},$$

where optimal parameters $\hat{\mathbf{w}}$ minimize a loss function

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} S(\mathbf{w}|f, \mathbf{X}, \mathbf{y}),$$

where

$$S(\mathbf{w}|f, \mathbf{X}, \mathbf{y}) = -\ln\left(\sum_{i=1}^{m} y_i \log f(\mathbf{x}_i, \mathbf{w}) + (1 - y_i) \log\big(1 - f(\mathbf{x}_i, \mathbf{w})\big)\right).$$