# NON SMOOTH OPTIMIZATION METHODS IN THE PROBLEMS OF CONSTRUCTING A LINEAR CLASSIFIER

**Yurii I. Zhuravlev, Yuryi Laptin, Alexander Vinogradov, Nikolay Zhurbenko, Aleksey Likhovid**

***Abstract**: We consider the technique using nonsmooth optimization methods for pattern recognition problems. The results of numerical experiments of comparison of the proposed approach with support vector machines are presented.*

***Keywords**: cluster, decision rule, discriminant function, linear and nonlinear programming, nonsmooth optimization*

***ACM Classification Keywords**: G.1.6 Optimization - Gradient methods, I.5 Pattern Recognition; I.5.2 Design Methodology - Classifier design and evaluation*

***Acknowledgement**: This work was done in the framework of Joint project of the National Academy of Sciences of Ukraine and the Russian Foundation for Fundamental Research № 10-01-90419.*

## Introduction

Mathematical models of the problems of constructing linear and nonlinear discriminant functions (classifiers) have been considered in many papers (see, e.g., [1, 2]). An approach proposed in [3–6] has certain advantages over the method of support vector machines (SVM). Mathematical model proposed in these papers can be conveniently represented in the form of convex optimization problems. The technique using efficient nonsmooth optimization methods [7] to solve these problems will be considered below. We present the results of computational experiments for specific test problems of large dimension.

The first section presents new results [8-11], allowing efficiently generate the equivalent unconstrained optimization problem for problems with constraints. In the second section we describe the mathematical models of constructing classifiers. In the third section characteristics of test problems and results of computational experiments are given.

## 1. A brief description of methods for solving nonsmooth convex constrained optimization problems

The scheme for solving optimization problems with constraints consists in construction of an equivalent unconstrained optimization problem, and in its solution by efficient subgradient algorithms (for example, r-algorithm by N.Z.Shor [7]).

To construct an equivalent unconstrained optimization problem we use the following approaches:

- exact penalty functions method;

- conical extensions of functions [10, 11].

Exact penalty functions are considered in a large number of publications, for example in [7,12,13]. The practical usage of penalty functions and various generalizations are considered in these papers.

Convex programming problem with constraints has the form: find

$$f^* = \min\{f(x) : x \in C\} \quad (1)$$

where $C = \left\{x \in R^n : h_i(x) \le 0, i = 1,..., m\right\}$, $f, h_i : R^n \to R$ – convex functions.

Let $f$, $h_i$ take finite values at all $x$. We will consider the penalty function of the form

$$S(x,s) = f(x) + s \cdot h^+(x), \; s \in R, \; s \ge 0, \quad (2)$$

where $h(x) = \max\{h_i(x), \; i = 1,..., m\}$, $x^+ = \max\{0, \; x\}$. Consider the problem: find

$$S^*(s) = \min\left\{S(x,s) : x \in R^n\right\} \quad (3)$$

Penalty function $S(x,s)$ is exact for given values of penalty coefficients $s$, if the solutions of (1) and (3) coincide.

To select the values of the penalty coefficients it is usually necessary to solve the auxiliary dual problems, or this selection is put on the user, which leads either to an overestimation of the values used, or to the necessity of multiple solving the same problem for the satisfactory selection of the penalty coefficients. In [8, 9] an approach to build an automatic procedure for determining the values of penalty coefficients during the optimization process is proposed. Consider a brief description of this approach.

We assume that $C$ is a bounded closed set. Denote by $S'(x,s,p)$, $f'(x,p)$, $h'(x,p)$ derivatives of functions $S$, $f$, $h$ at a point $x \in R^n$ in the direction $p$ for the fixed value $s$, $p(x,y) = (y-x)/\|y-x\|, \; y \ne x$.

Let $\tilde{x}$ be a solution of (3), and convergent sequences $x^k \in R^n$, $y^k \in C$, $k = 0,1, ...$ are given, $x^k \to \tilde{x}$ when $k \to \infty$. The sequence $x^k$ is generated during the solution of problem (3) by algorithm for unconstrained optimization, the point $y^k$ is determined by an auxiliary rule for the current point $x^k$. Such rules can be defined in various ways, for example, we may assign $y^k = y^0$, where $y^0$ is an initial feasible point such that $h(y^0) < 0$, or may choose $y^k$ among the feasible points generated in previous iterations.

Let $x^k \notin C$. We denote by $\pi_C(x^k, y^k)$ the intersection point of the segment $\left[x^k, y^k\right]$ with the boundary of the set $C$, $\overline{x}^k = \pi_C(x^k, y^k)$.

Theorem **1** [9]. Let $\varepsilon > 0$, $s > 0$ such that for each $x^k$, $x^k \notin C$, $k = 0,1, ...$ *the following constraint* is satisfied

$$S'(\overline{x}^k, s, p(\overline{x}^k, x^k)) \ge \varepsilon \quad (4)$$

Then $\tilde{x}$ *is* a solution of *(1), i.e.* $S(x,s)$ *is* the exact penalty function.

Thus, for using this approach to determine the value of penalty coefficient $s$ it is necessary to check the condition (4) at each step of the optimization algorithm, which requires the solution of one-dimensional problem of finding the intersection point $\overline{x}^k = \pi_C(x^k, y^k)$ of the segment $\left[ x^k, y^k \right]$ with the boundary of the set $C$. This search procedure can be implemented effectively.

In the case when inequality (4) at some iteration of the algorithm is violated, we will increase the penalty $s$, so that inequality (4) is satisfied. This increase must be not less than $B$, where $B > 0$ is a given parameter. It is easy to see that if there exists such finite $\overline{s}$ that for $s > \overline{s}$ the inequality (4) holds on all iterations of the algorithm, then the amount of such penalty increases is finite throughout the optimization process.

Theorem **2** [9]. Given a sequence $x^k \in R^n$, $k = 0,1, \ldots$ converging to a solution $\tilde{x}$ of problem *(3),* $y^k = y^0$, $k = 1, 2, \ldots,$ $h(y^0) < 0$. Then there exists $\overline{s} < \infty$ such that the conditions of Theorem *1* are *satisfied* for $s > \overline{s}$.

In [9] a special rule of the choice of $y^k$ was considered. It was showed that $\overline{s} = \sum_{i=1}^{m} u_i^*$ where $u_i^*$, $i = 1, \ldots, m$ are optimal values of dual variables.

Theorem 2 allows us to construct the automatic determination of penalty coefficients during the optimization process in the case when the starting point $y^0$, $h(y^0) < 0$, is known. If this point is unknown, then the solving process of the problem is divided into two phases - the first phase is to find the point $y^0$, $h(y^0) < 0$, on the second phase the original problem is solved.

Conical extensions of functions [10, 11] is another approach to generate an equivalent unconstrained optimization problem. The objective function of this problem coincides with the objective function of the original problem on the feasible set. Outside of the feasible set the formed function is defined by the behavior of the objective function of the original problem on the boundary of the feasible set. The original objective function can not be defined outside the feasible set. As above consider a brief description of this approach.

As before, the problem (1) is considered. It is assumed that $C$ is a closed bounded set, a feasible point $x^0 \in C$ such that $h_i(x^0) < 0$, $i = 1, \ldots, m$ and a number $E$, $E < f(x^0)$ are given.

For $x \notin C$ we denote $\pi_C(x^0, x)$ the intersection point of the segment $[x^0, x]$ with boundary of the set $C$. Let

$$R_C(x^0, x) = \frac{\left\| x - x^0 \right\|}{\left\| \pi_C(x^0, x) - x^0 \right\|}, \tag{5}$$

$$\chi^E(x) = E + (f(\pi_C(x^0, x)) - E) \cdot R_C(x^0, x), \tag{6}$$

$$\psi^E(x) = \begin{cases} f(x), & \text{if } x \in C \\ \chi^E(x), & \text{if } x \notin C \end{cases}. \tag{7}$$

It is easy to see that $\psi^E(x)$ is a continuous function. Consider the problem of finding

$$\psi^{*E} = \inf\left\{\psi^E(x) : x \in R^n\right\}. \quad (8)$$

Lemma **1** [11]. Let $E < f^*$ *, then* $\psi^{*E} = f^*$ .

**Theorem 3** [11]. Let $C$ is a closed bounded set, $C \subset \operatorname{int} \operatorname{dom} f$ . Then there exists a finite number $E^*$ such that $\psi^E(x)$ is a convex function for all $E \le E^*$.

We denote $g_f(x)$ , $g_h(x)$ subgradients of functions $f$ , $h$ at the point $x$ .

**Theorem 4** [11]. Let $\overline{x} = \pi_C(x^0, x)$ . *Then the vector*

$$g = g_f(\overline{x}) + \frac{E - f(\overline{x}) - \left\langle g_f(\overline{x}), x^0 - \overline{x}\right\rangle}{\left\langle g_h(\overline{x}), x^0 - \overline{x}\right\rangle} g_h(\overline{x}) \quad (9)$$

Is a subgradient of function $\chi^E(x)$ at the point $x$ (subgradient of function $\psi^E(x)$ if $x \notin C$ ).

Thus, if $f^*$ , $E^*$ are known, and conditions of the Lemma 1 and Theorem 3 are satisfied, then any algorithm for minimizing convex functions can be used for solving the problem (8). The solution of the problem (8) is a solution of the original problem (1).

Consider the case where the values of $f^*$ and $E^*$ are unknown. Denote as $f'(x, p)$ the derivative of function $f$ at the point $x$ in direction $p$ , $p(x^0, x) = (x - x^0)\Big/\left\|x - x^0\right\|$ . Suppose that we use a convergent algorithm $A$ for unconstrained minimization of convex functions, at each iteration of which the value of objective function and its subgradient are computed.

**Theorem 5** [11]. Let numbers $E$ and $\delta > 0$ are given, algorithm $A$ is used to solve problem (8), and the following condition is satisfied at each iteration $k$ of the algorithm: if $x^k \notin C$ then

$$E < f(\overline{x}^k) - \delta, \quad (10)$$

$$E < f(\overline{x}^k) - f'(\overline{x}^k, p(\overline{x}^k)) \cdot \left\|\overline{x}^k - x^0\right\| \quad (11)$$

*where* $\overline{x}^k = \pi_C(x^0 x^k)$, $x^k$ *is* the current point at the iteration $k$ . Then the sequence of points generated by the algorithm $A$ converges to the solution *of problem (1).*

If at some iteration $k$ inequalities (10), (11) are violated, then it's necessary to change the value $E$ iteratively: $E = \Delta - B$ , where $\Delta = \min\left\{f(\overline{x}^k), \quad f(\overline{x}^k) + f'(\overline{x}^k, -p(\overline{x}^k)) \cdot \left\|\overline{x}^k - x^0\right\|\right\}$ , $B > 0$ is a given parameter.

In view of finiteness $f^*$ and $E^*$ there will be just finite number of changes of the value $E$ , after which the algorithm converges to the optimal solution of problem (1).

Thus, the approach under consideration allows to construct an equivalent unconstrained optimization problem, and to solve the original problem using unconstrained optimization algorithm.

## 2. Description of the mathematical models for constructing classifiers

The problems of constructing linear classifiers are considered in [3-5].

Given a set of linear functions $f_i(x,W^i)=\left\langle w^i,x\right\rangle+w_0^i,\ i=1,...,\ m$, where $x\in R^n$ is an attribute vector, $W^i=(w_0^i,w^i)\in R^{n+1}$ is a parameter vector, $i=1,...,m$.

Let introduce the notations $W=(W^1,...,W^m)$, $W\in R^L,\ L=m(n+1)$. When $m>2$ we consider the linear classification algorithms (linear classifiers) in the form:

$$a(x,W)=\arg\max_i\left\{f_i(x,W^i):\ i=1,...,\ m\right\},\ x\in R^n,\ W\in R^L. \tag{12}$$

When $m=2$ ,then the linear classifiers are defined by linear functions $f(x,W)=(w,x)+w_0,\ W=(w,w_0)\in R^{n+1}$, and are presented in the form

$$a(x,W)=\begin{cases}1,\ \text{if}\ f(x,W)\ >0,\\ 2,\ \text{if}\ f(x,W)\le 0,\end{cases} \tag{13}$$

We consider a given finite family of disjoint sets (training set) of points: $\Omega_i=\left\{x^t:x^t\in R^n,\ t\in T_i\right\}$, $i=1,...,\ m$ , $T=\bigcup_{i=1}^{m}T_i$ .

It is said that the classifier $a(x,W)$ correctly separates the points of $\Omega_i,\ i=1,...,\ m$, if $a(x,W)=i$ for all $x\in\Omega_i,\ i=1,...,\ m$. We define function $i(t)$ returning index of the set which contains the point $x^t\in\Omega_{i(t)}$, $t\in T$.

If $m>2$, then the value

$$g^t(W)=\min\left\{f_i(x^t,W^i)-f_j(x^t,W^j):\ j\in\{1,...,\ m\}\setminus i,\ i=i(t)\right\}=$$

$$=\min\left\{\left\langle w^i-w^j,x^t\right\rangle+w_0^i-w_0^j:\ j\in\{1,...,\ m\}\setminus i,\ i=i(t)\right\} \tag{14}$$

is called a gap of classifier $a(x,W)$ at the point $x^t$, $t\in T$.

In the case of $m=2$ a classifier gap at the point $x^t$ is the value

$$g^t(W)=\begin{cases}\ f(x^t,W),\text{if}\ t\in T_1,\\ -f(x^t,W),\ \text{if}\ t\in T_2.\end{cases} \tag{15}$$

The value $g(W)=\min\left\{g^t(W):t\in T\right\}$ is called a gap of classifier $a(x,W)$ on the family of sets $\Omega_i,\ i=1,...,\ m$ . The classifier $a(x,W)$ correctly separates the points of the sets $\Omega_i,\ i=1,...,\ m$ , if $g(W)>0$. The sets $\Omega_i,\ i=1,...,\ m$ are called linearly separable, if there exist a linear classifier that correctly separates the points of these sets.

If the sets $\Omega_i$, $i=1,...,\ m$ are linearly separable, then the problem of constructing an optimal classifier (determination of the parameters $W$) has the following form: find

$$g^* = \max_W \left\{ g(W) : \eta(W) \le 1,\ W \in R^L \right\}. \tag{16}$$

Here $\eta(W)$ is the norm of vector $W$, $\eta(W) = \sqrt{\sum_{i=1}^{m}\sum_{j=1}^{n}(w_j^i)^2}$.

This problem can be written in the equivalent forms

$$\eta^* = \min_V \left\{ \eta(V) : g(V) \ge 1,\ V \in R^L \right\}, \tag{17}$$

$$\eta^* = \min_V \left\{ \eta(V) : g^t(V) \ge 1, t \in T, V \in R^L \right\}. \tag{18}$$

The equivalence is understood in the sense that if $W^*$ is the optimal solution of problem (16) then equalities $V^* = W^* \Big/ g^*$, $\eta^* = 1 \Big/ g^*$ hold for the optimal solution $V^*$ of problem (17) or (18).

A continuous relaxation of the problem of minimizing the empirical risk is proposed in [4-5] to build a classifier in the case of linearly inseparable sets. This relaxation has the form: find

$$q^* = \min \sum_{t \in T} d^t(W) \tag{19}$$

under constraints

$$\eta(W)^2 \le 1, \tag{20}$$

$$\sum_{t \in T_i} d^t(W) \le |T_i| - 1,\ i = 1,...,\ m \tag{21}$$

$$d^t(W) \le 1,\ t \in T, \tag{22}$$

where $d^t(W) = \max\left(0,\ \frac{1}{B}\left(\bar{\delta} - g^t(W)\right)\right)$, $\bar{\delta} > 0$ is a parameter of the reliability required for separating points of the training set, $B$ is a parameter (a sufficiently large positive number).

Consider the problem: find

$$\eta^* = \min\left\{ \frac{1}{2}\eta(V)^2 + C\sum_{t \in T} \xi^t : g^t(V) \ge 1 - \xi^t, \xi^t \ge 0, t \in T, V \in R^{n+1} \right\}. \tag{23}$$

This problem is solved by the method of support vector machines (SVM) for the case $m = 2$. The SVM method is used to build an optimal classifier for linearly separable classes, and for linearly nonseparable classes.

Note that for linearly separable classes the problems (18) and (23) have the same solutions if coefficient $C$ is sufficiently large. This follows from the theorem on non-smooth penalties [7,12].

In [5] it was shown that in the case of linearly nonseparable classes the problem (23) can be obtained from (19) - (22) by Lagrangian relaxation with a special selection of the values of the Lagrange multipliers.

A choice of coefficient $C$ is a significant problem when the problem (23) is used in the case of linearly nonseparable classes. It should be noted that this problem does not arise when problem (19) - (22) is used.

Consider the characteristics of problems (16) and (19)-(22) which are useful when the approaches described in the previous section are used:

The point $W^0 = 0$ is an interior point of the feasible set.

1. The optimal values of these problems are always greater than or equal to zero.
2. Implementation of one-dimensional search for a point on the boundary of the feasible set is simple: let $\eta^k = \eta(W^k)^2$ is a squared norm of the points $W^k$, $\eta^k > 1$, then the point $\bar{W} = \frac{W^k}{\sqrt{\eta^k}}$ is the required point on the boundary of the feasible set.
3. Functions $g^t(W)$ have the property – $g^t(\alpha W) = \alpha g^t(W)$.

## 3. Software implementation and results of computational experiments

Software implementation for the following approaches to the problems under consideration was developed:

- for problems (16) and (19)-(22) - a method of exact penalty functions with automatic adjustment of the penalty factor, the method of convex conical extensions;
- for problems (18) and (23) - a method of exact penalty functions without the automatic adjustment of the penalty coefficient.

Unconstrained optimization problems, to which the original problems with the constraints are reduced, were solved using $r$-algorithm by N.Z.Shor [7].

The problems of constructing linear classifiers for two classes were generated randomly for computational experiments. The parameters of problems varied in the range:

- the dimension $n$ of attribute space $R^n$ – from 5 to 100;
- the number of points in the training set – from 40 to 100 000.

The points in the training set for each class were generated on the basis of a uniform distribution within the unit cube. These cubes are shifted relative to each other along the first coordinate, so that the distance between them is equal to unity. For each problem $P_0$, constructed in such way, a family of problems $P_i$, $i = 1,..., 10$ was generated by reducing the distance between the classes (cubes). The distance between classes of the problem $P_i$ is equal to $2^{-i}$. All problems from the generated families are linearly separable.

To construct linearly nonseparable problems (sets) a membership to a class of some points of training set is changed.

According to the results of computational experiments we can do the following conclusions:

– the method of exact penalty functions with automatic adjustment of the penalty coefficient and the method of convex conical extensions showed approximately the same efficiency for the problem (1 1), all problems from the generated families were solved successfully (the accuracy of the objective function $\sim 10^{-6}$), the iteration

number of the $r$-algorithm changed from $\sim 100$ for the dimension $n = 5$ to $\sim 1500$ for the dimension $n = 100$;

– the choice of coefficient $C$ is essential when using the model SVM (problem (23)), in the computational experiments we used the value $C = 1000$, and the problems $P_i$, $i \le 5$ from the generated families were solved successfully (separating hyperplanes were found), but the problems $P_i$, $i \ge 7$ were not solved (separating hyperplanes were not found).

The developed software tools were compared with existing software (LIBSVM – http://www.csie.ntu.edu.tw/~cjlin/libsvm/). In the table the elapsed time for solving problems of constructing a linear classifier in the space of dimension $n = 100$, depending on the number of points in the training set, is shown. The standard settings for LIBSVM were used.

Table.

| Number of points | The solution time, in seconds | |
|---|---|---|
| | LIBSVM | Automatic adjustment of the penalty coefficient |
| 5000 | 9.421 | 20.8 |
| 10000 | 24.234 | 24.3 |
| 25000 | 83.468 | 43 |
| 40000 | 186.484 | 51,1 |
| 50000 | 266.203 | 84,8 |

Thus, the methods of nonsmooth optimization provide greater opportunities in the construction of linear classifiers in comparison with traditional approaches. At the same time the performance of the developed programs is comparable with existing software.

## References

1. Vladimir Vapnik. Estimation of Dependences Based on Empirical Data. – Springer Verlag, 2006, 2nd edition.
2. Thorsten Joachims. Learning to Classify Text Using Support Vector Machines: Methods, Theory, and Algorithms. Kluwer, 2002
3. Laptin Yu., Likhovid A. P., Vinogradov A.P. Approaches to Construction of Linear Classifiers in the Case of Many Classes // Pattern Recognition and Image Analysis, Vol. 20, No. 2, 2010, p. 137-145.
4. Zhuravlev Yu., Laptin Yu., A.Vinogradov Minimization of empirical risk in linear classifier problem // New Trends in Classification and Data Mining, ITHEA, Sofia, Bulgaria, 2010. – Pages 9-15
5. Журавлев Ю.И., Лаптин Ю.П., Виноградов А.П. Минимизация эмпирического риска и задачи построения линейных классификаторов // Кибернетика и системный анализ. 2011, № 4.- С. 155 – 164.
6. Журавлев Ю.И., Лаптин Ю.П., Виноградов А.П. Построение нелинейных классификаторов в случае многих классов // Applicable Information vodels. ITHEA, Sofia, 2011. – P. 7 – 13
7. Shor N. Z. Nondifferentiable Optimization and Polynomial Problems. – Amsterdam / Dordrecht / London: Kluwer Academic Publishers, 1998. – 381 p.
8. Лаптин Ю.П. Некоторые вопросы использования негладких штрафных функций // Теорія оптимальних рішень. 2011, № 10. с. 127 – 135.

9. Лаптин Ю.П. Некоторые вопросы определения коэффициентов негладких штрафов // Теорія оптимальних рішень. 2012, № 11. (В печати).
10. Лаптин Ю.П. Один подход к решению нелинейных задач оптимизации с ограничениями // Кибернетика и системный анализ. 2009, № 3. С. 182 – 187.
11. Лаптин Ю.П., Лиховид А.П. Использование выпуклых продолжений функций для решения нелинейных задач оптимизации // Управляющие машины и системы. 2010, № 6. – С. 25–31.
12. Пшеничный Б.Н. Метод линеаризации. – М.: Наука. – 1983. – 136 с.
13. Evtushenko Y.G., Rubinov A.M., and Zhadan V.G. General Lagrange-type functions in constrained global optimization. Optimization Methods and Software, 2001, vol. 16, Part I: pp.179-217, Part II: pp. 231-256.

## Information about authors

***Yurii I. Zhuravlev*** *– Academician of the RAS, Deputy Director, Dorodnicyn Computing Centre of the RAS, Vavilova 40, 119333 Moscow, Russian Federation; e-mail: zhuravlev@ccas.ru*

***Yuryi Laptin*** *– Senior Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: laptin_yu_p@mail.ru*

***Alexander Vinogradov*** *– Senior Researcher, Dorodnicyn Computing Centre of the RAS, Vavilova 40, 119333 Moscow, Russian Federation; e-mail: vngrccas@mail.ru*

***Nikolay Zhurbenko*** *– Senior Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: zhurbnick@yandex.ru*

***Aleksey Likhovid*** *– Researcher, V.M.Glushkov Institute of Cybernetics of the NASU, Prospekt Academika Glushkova, 40, 03650 Kyiv, Ukraine; e-mail: o.lykhovyd@gmail.com*