DIMACS Technical Report 2004-08
April 2004

SIMULTANEOUS FEATURE SELECTION AND MARGIN MAXIMIZATION USING SADDLE POINT APPROACH

by

Yuri Goncharov
Artezio LLC.
Moscow, Russia
ygoncharov@artezio.ru

Ilya Muchnik
DIMACS,
Rutgers University,
New Brunswik, NJ
muchnik@dimacs.rutgers.edu

Leonid Shvartser
Friman 12,
Lod, Israel 71216
lshvartser@hotmail.com

DIMACS is a collaborative project of Rutgers University, Princeton University, AT&T Labs--Research, Bell Labs, NEC Laboratories America and Telcordia Technologies, as well as affiliate members Avaya Labs, HP Labs, IBM Research and Microsoft Research. DIMACS was founded as an NSF Science and Technology Center.

Abstract

A new SVM wrapper method, which simultaneously maximizes margin and minimizes feature space is introduced. For these purposes we modify the standard criterion by adding to the basic objective function a third term, which directly penalizes a chosen set of variables. The new criterion divided the set of all variables into three subsets: deleted, selected and weighted features. We are showing that the question can be formulated as a particular min-max problem for convex-concave functions, which in turn can be solved by saddle point polynomial algorithms. We analyzed a set of such algorithms and realized one, which is taking to account specificity of our problem. The algorithm is examined on a classification Benchmark and its ability to improve the recognition results is shown. We also show that the developed method can be easily transferred to the Support Vector Regression case.

1. Introduction

This work was motivated by feature selection problem in learning classification via SVM methods [45]. According SVM approach an optimal hyper-plane is related to maximum margin between sets of samples from complementary classes in the training data, which is bounded the critical factor of the learning generalization called VC-dimension (or weak margin if the sets are not separable [1, 2]). From the prospective of the feature selection problem it is a very interesting question how to find a subset of features which space gives the largest margin between classes on the training data.[1] We introduce a new type of SVMs – saddle point SVM (SP-SVM) and we are showing that the question can be formulated as a particular min-max problem for convex-concave functions, which in turn can be solved by saddle point algorithms.[2] This specification of the problem gives algorithms, which are polynomial. In literature for feature selection methods based on optimization of quality criteria called as wrapped methods [23-29]. Taking into account that the margin can be considered as a criterion of a classifier quality we will call the algorithms "exact wrapped algorithms". It is looking both for a coordinate subspace and a set of classifiers with the maximal margin. One can use it in a combination with a cross-validation procedure, for instance, to estimate on the same testing data two classifiers: constructed in original space (getting by a regular SVM method) and based on the considered optimal space. An example of such experiments will be done in section 4.

The paper is organized into 6 sections and 2 Appendices. In the second section we describe our basic results: exact wrapped methods of feature selection for classification problem by maximizing margin between classes. In the third section we described our saddle point algorithm in which we take into account specifics of our problems. In the fourth section we describe and analyze the obtained experimental results with this algorithm. In the section 5 we show that the developed saddle point approach can be

[1] The first paper in which the feature selection problem is integrated with SVM method was [32]. This work explores idea that weights of a linear SVM classifier can be interpreted as significance estimates of the corresponding variables. As it was demonstrated by experiments the procedure proposed in [32] gives a significant reduction of the original space without reducing the classification accuracy. However, it doesn't relate directly to the margin maximization task.

[2] In [31] J. Bi has investigated multi criteria approach to the feature selection problem in the framework of the VC-dimension minimization. According to this approach the solution is represented as a Pareto-optimal set of classifiers (MOPSVM). Unfortunately, the proposed method doesn't guarantee to achieve the solution. Our criterion is different. We follow the standard SVM idea to design a single criterion, which maximized margin, but we modify the standard criterion by adding to the basic objective function a new term, which directly penalizes a chosen set of variables. One could find below that the new criterion divided the set of all variables into three subsets: deleted, selected and weighted features. For a convenient comparison of these two methods we give a detail summary of the method proposed by J. Bi in the Appendix 4.

easily extend on SVM methods for regression modeling.[3] Detail analysis of this opportunity is our plan on a future research.

As far as this paper investigates the properties of a new introduced saddle point criterion we decided to add a section 6, which shows that the convex-concave theory does not transfers automatically to the cases of non-linear kernels. In spite of it the following method for non-linear kernels can be used. It is very often for SVM practitioners to map every object to the space of vectors of a kernel function between the chosen object and all objects in the training data including the considered object itself; and to build a linear classifier in the new space. Our SP-SVM method can be also used in this new space, which can improve the classification results.

In the Appendix 1 we present a survey about saddle points algorithms to give readers a convenient way to analyze the proposed methods autonomously. This Appendix together with section 3 shows that unless proven polynomial properties of the saddle point algorithms their implementation is an independent interesting task. In our experiments (section 4) we used small cardinality samples. The reason of it was a strong dependency of the algorithm's speed of the cardinality of training set. The first improvement we have to do in our future work is to develop new algorithms which speed would not be so depended from the training set size.

Appendix 2 dedicated to MOPSVM feature selection algorithm [31] of Bi.

We hope that this paper which brings to "SVM methods community" the idea of using saddle point algorithms for learning processes will give a new opportunity to improve a power of the methods in general. We suppose that SP-SVM will be useful in a wider field of data analysis problems as already mentioned regression modeling as well as new ones for instance: feature interactions, PCA-like and clustering problems.

2. Exact Wrapped Method for Feature Selection in Learning Classification

In this section we introduce new criteria to receive a classifier with maximum margin by searching subspaces of a given space. The methods to find saddle points related to the mentioned optimal classifiers we call as exact wrapped methods. The feature selection problem considered under SVM methods is investigated in [30-34]. In [32, 33] was proposed an efficient greedy-like procedure, which worked as a standard wrapped algorithm [23].

We introduce a three terms criterion as a modification of SVM setting and define a problem as a problem of searching for a set of variables that gives optimum to the described criterion.

This problem became discrete-continuous and thus is very hard to solve. We load our problem into a continuous one, which is searching for a transformation of space of

[3] A feature selection method for SVM regression from very different point of view proposed in [34]. This method exploits the fact that linear SVM with l_1 - norm regularization inherently performs feature selection as a side effect of minimizing capacity of the SVM model.

variables in such a way that feature selection and margin maximization will be done together.

The new problem is also not good for its effective solution because it is not convex. We change it to a problem of successive minimization, which has the same global optimal solutions.

The last problem is a problem of non-smoothed convex optimization, which we show using the dual form of this problem. Such problem can already been solved with polynomial algorithms (see Appendix 1).

We introduce an additional criterion for our problem such that gives more stability to a found solution. This criterion is formulated as a saddle point of a convex-concave function on a close convex compact.

In subsection 2.2 we analyze properties of the proposed formalism especially investigating the geometry of variables and parameters.

2.1. SP-SVM Setting.

Let $\{\boldsymbol{x}_i, y_i\}$, $i = 1,2,...,N$ is a training set of points $\boldsymbol{x}_i \in \mathbf{R}^M$ with labels $y_i \in \{-1, 1\}$. Assign set of variables as Ω, $Q \subseteq \Omega$ is an arbitrary subset of variables. We formulate the feature selection problem, where the margin criterion is represented by the formula:

$$\begin{gathered}\frac{1}{2}\left\|\mathbf{w}_Q\right\|^2 + C\sum_{i=1}^{N}\delta_Q^i + A|Q| \to \min_{Q\subseteq\Omega, w_Q, b_Q, \delta_Q^i} \\ y_i\left(\mathbf{w}_Q^T\mathbf{x}_i^Q + b_Q\right) \geq +1 - \delta_Q^i, \quad y_i = +1, \\ y_i\left(\mathbf{w}_Q^T\mathbf{x}_i^Q + b_Q\right) \leq -1 + \delta_Q^i, \quad y_i = -1, \\ \delta_Q^i \geq 0, i = 1,..., N.\end{gathered} \tag{1}$$

The term $A|Q|$ with the positive constant A is introduced in order to reduce the cardinality of the extreme subspace that we look for.

It is easy to see that this "discrete-continuous" problem is very hard to be solved. That's why let us extend this problem for its continuous analog:

To find a vector of scaling factors (scaling factor for each variable), such that from one hand maximize margin and from the other hand choose a small set of variables (feature selection). Let us formulate it in a following way:

$$\begin{aligned}
&\frac{1}{2}\sum_{l=1}^{M} w_l^2 + C\sum_{i=1}^{N}\delta^i + A\sum_{l=1}^{M} z_l^2 \to \min_{w,\delta,b,z} \\
&y_i\left(\sum_{l=1}^{M} w_l \mathbf{x}_i^l z_l + b\right) \geq +1-\delta^i, \qquad y_i = +1, \\
&y_i\left(\sum_{l=1}^{M} w_l \mathbf{x}_i^l z_l + b\right) \leq -1+\delta^i, \qquad y_i = \text{-}1, \\
&\delta^i \geq 0, \qquad i = 1,\ldots,N, \\
&0 \leq z_l \leq 1, \qquad l = 1,\ldots,M.
\end{aligned} \tag{2}$$

This setting means that every variable l in the desired space multiplied by z_l.[4]

This problem is non-convex, because of the matrices of their constraints are not positive and not negative semidefinite. Then problem (2) seems very hard to solve, but it is easy to show that it can be substituted by another easier problem which solution is coincident to this one.[5]

Let us formulate the new problem:

$$F^p(\mathbf{z}) = \left\{ \begin{aligned}
&\frac{1}{2}\sum_{l=1}^{M} w_l^2 + C\sum_{i=1}^{N}\delta^i + A\sum_{l=1}^{M} z_l^2 \to \min_{w,\delta} \\
&y_i\left(\sum_{l=1}^{M} w_l \mathbf{x}_i^l z_l + b\right) \geq +1-\delta^i, \qquad y_i = +1, \\
&y_i\left(\sum_{l=1}^{M} w_l \mathbf{x}_i^l z_l + b\right) \leq -1+\delta^i, \qquad y_i = \text{-}1, \\
&\delta^i \geq 0, \qquad i = 1,\ldots,N
\end{aligned} \right\} \to \min_{0\leq z_l\leq 1, l=1,\ldots,M} \tag{3}$$

Every subproblem in figure brackets of (3) is convex. To show it introduce the following pair of auxiliary problems:

---

[4] Term $A\sum_{i=1}^{n} z_i^2$ can be changed with term $A\sum_{i=1}^{n} z_i$, but we'll see below that usage of the first one is more convenient for dual formulations.

[5] *Indeed the global solutions of* (2) *are coincident to the global solutions of* (3).

Proof. Let $(w_{(23)}, \delta_{(23)}, z_{(23)})$ is a solution of problem (3) with optimal objective value $F_{(23)}$. $(w_{(23)}, \delta_{(23)}, z_{(23)})$ is a feasible solution for (2) then $F_{(22)} \leq F_{(23)}$. Let $(w_{(22)}, \delta_{(22)}, z_{(22)})$ is a solution of (2) with objective $F_{(22)}$. $F^p(z_{(22)}) \leq F_{(22)}$ because $(\delta_{(22)}, z_{(22)})$ is a feasible point for a problem in square brackets of (3). $F_{(23)} \leq F^p(z_{(22)})$ as a solution of (3). We have $F_{(23)} \leq F^p(z_{(22)}) \leq F_{(22)} \leq F_{(23)}$, which proves that $F_{(22)} = F_{(23)}$ and $(w_{(23)}, \delta_{(23)}, z_{(23)})$ is a solution of (2) and $(w_{(22)}, \delta_{(22)}, z_{(22)})$ is a solution of (3). ♦

$$F_1^d(\mathbf{z}) = \left\{ \begin{array}{c} \sum_{j=1}^{N} \lambda_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (y_j y_k \sum_{l=1}^{M} z_l^2 x_j^l x_k^l) \lambda_j \lambda_k + A \sum_{l=1}^{M} z_l^2 \to \max\limits_{\boldsymbol{\lambda}} \\ \sum_{j=1}^{N} \lambda_j y_j = 0, j = 1,..., N, \\ 0 \le \lambda_j \le C, j = 1,..., N, \end{array} \right\} \to \min\limits_{0 \le z_l \le 1, l=1,...,M.} \quad (4)$$

$$F_2^d(\mathbf{z}) = \left\{ \begin{array}{c} \sum_{j=1}^{N} \lambda_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l) \lambda_j \lambda_k + A \sum_{l=1}^{M} z_l \to \max\limits_{\boldsymbol{\lambda}} \\ \sum_{j=1}^{N} \lambda_j y_j = 0, j = 1,..., N, \\ 0 \le \lambda_j \le C, j = 1,..., N, \end{array} \right\} \to \min\limits_{0 \le z_l \le 1, l=1,...,M.} \quad (5)$$

Proposition 1. Solutions of (3) are coincident to solutions of (4). Solutions of (2) are square roots of the solutions of (4): $\mathbf{z}^*_{(5)} = \sqrt{\mathbf{z}^*_{(4)}}$.

Proof. The first part of the proposition proved by a transformation $\hat{x}_i^l = z_l x_i^l, i = 1,..., N, l = 1,..., M$ and taking to account that for each chosen **z** the problem in figure brackets of (4) without last constant term is dual to the problem in figure brackets of (3) also without last constant term. The second part of the proposition proved by a transformation $\hat{\mathbf{z}}_l = \sqrt{\mathbf{z}_l}, l = 1,..., M$. ♦

Remark 1. The set of constraints in (4), (5) is constant and does not depend of **z**. Then the functions $F_1^d(\mathbf{z})$, $F_2^d(\mathbf{z})$ are convex according to Theorem 5.5 from book [6]. It seems more attractive to work with $F_2^d(\mathbf{z})$ because variable **z** included in it in a linear form. The equivalence of (4) and (5) shows that introduction of term $A\sum_{i=1}^{n} z_i^2$ more convenient for work than $A\sum_{i=1}^{n} z_i$, because it gives an opportunity to solve a linear by **z** problem (5).

Now we obtain a problem of minimization of non-smooth convex function on a convex set $0 \le z_l \le 1, l = 1,..., M$. The algorithms for such optimization analyzed in the next Section 3 and Appendix 1.

The criterion from (5) can be considered as a function of two vector variables (**z**, **λ**):

$$L(\mathbf{z}, \boldsymbol{\lambda}) = \sum_{j=1}^{N} \lambda_j - \frac{1}{2} \sum_{j=1}^{N} \sum_{k=1}^{N} (y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l) \lambda_j \lambda_k + A \sum_{l=1}^{M} z_l , \quad (6)$$

which is investigated over two convex compact sets:

$$Z = \{\mathbf{z} \mid 0 \le z_l \le 1,\ l = 1,...,M\}, \tag{7}$$

$$\Lambda = \{\boldsymbol{\lambda} \mid 0 \le \lambda_j \le C,\ \sum_{j=1}^{N} \lambda_j y_j = 0, j = 1,...,N\}. \tag{8}$$

Consider a saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ of (6):

$$L(\mathbf{z}^*, \boldsymbol{\lambda}) \le L(\mathbf{z}^*, \boldsymbol{\lambda}^*) \le L(\mathbf{z}, \boldsymbol{\lambda}^*), \forall \mathbf{z} \in Z, \forall \boldsymbol{\lambda} \in \Lambda. \tag{9}$$

Let us solve a defined problem of simultaneous feature selection and margin maximization as problem (9). It may be seen from Appendix 1 that a solution of (9) is also a solution of (5). Solution of (9) is a stable solution of (5) in sense of Nash equilibrium [42]. It may be interpreted in an intuitive way as a game of two players: one of them chooses $\mathbf{z}$ from Z, the second one chooses $\boldsymbol{\lambda}$ from Λ. Their compromise solution is $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$.

2.2. Saddle Point Properties.

In this section we will analyze the properties of the introduced formalization from the following points of view: what are the conditions for strict elimination of variable, what are the conditions for proven necessarily remaining of a variable, what are the conditions for "fuzzy answer" – a variable is only weighted, not eliminated, not remained. Moreover we analyze a geometrical meaning of introduced above penalty parameter A.

The introduced saddle point properties described by the following

Theorem 1.

1) There exists a saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ of problem (9):

$$\begin{cases} (\mathbf{z}^*, \boldsymbol{\lambda}^*) = \arg\min_{\mathbf{z}} \max_{\lambda} L(\mathbf{z}, \boldsymbol{\lambda}); \\ (\mathbf{z}^*, \boldsymbol{\lambda}^*) = \arg\max_{\lambda} \min_{\mathbf{z}} L(\mathbf{z}, \boldsymbol{\lambda}); \\ \sum_{j=1}^{N} \lambda_j y_j = 0, j = 1,...,N, \\ 0 \le \lambda_j \le C, j = 1,...,N, \\ 0 \le z_j \le 1, j = 1,...,M; \end{cases} \tag{10}$$

2) If there are no such l, $l=1,...,M$ that $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* = 2A$ then the saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ of problem (10) is a solution of problem (11), with $z_l=1$ if $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* > 2A$ and $z_l=0$ if $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* < 2A$;

3) A component z_l of a saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ of problem (10) can be different from 0 or 1 only if $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* = 2A$.

4) Let $(\mathbf{z}_{(11)}^*, \boldsymbol{\lambda}_{(11)}^*)$ is a solution of problem (11):

$$\begin{cases} \min\limits_{\mathbf{z}} \max\limits_{\lambda} L(\mathbf{z},\boldsymbol{\lambda}) = \sum\limits_{j=1}^{N}\lambda_j - \frac{1}{2}\sum\limits_{j=1}^{N}\sum\limits_{k=1}^{N}(y_j y_k \sum\limits_{l=1}^{M} z_l x_j^l x_k^l)\lambda_j\lambda_k + A\sum\limits_{l=1}^{M} z_l \\ \sum\limits_{j=1}^{N}\lambda_j y_j = 0, j = 1,...,N, \\ 0 \le \lambda_j \le C, j = 1,...,N, \\ z_j \in \{0,1\}, j = 1,...,M, \end{cases} \quad (11)$$

and $(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^*)$ is a saddle point of (8) with every $0<z_j<1$ arbitrary changed to 0 or 1. The solution of (11) is bounded with

$$L(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^*) \le L(\mathbf{z}_{(11)}^*, \boldsymbol{\lambda}_{(11)}^*) \le \max_{\lambda} L(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}). \quad (12)$$

Proof.

1) It is known [6] (see Appendix 1) that a function, which is concave by part of its arguments and convex by the complemented part, has a saddle point on a close convex set. The function $L(\mathbf{z}, \boldsymbol{\lambda})$ is concave by $\boldsymbol{\lambda}$ (multiplication of each component of $\boldsymbol{x}$ to a positive value, does not disturbs Gramm property) and linear by $\mathbf{z}$. Moreover, the frame of definition for problem (9) is convex and close.

2) The criterion $\sum_{j=1}^{N}\lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l)\lambda_j\lambda_k + A\sum_{l=1}^{M} z_l$ can be rewritten as $\sum_{j=1}^{N}\lambda_j - \sum_{l=1}^{M} z_l(\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j\lambda_k - A)$. $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ is a saddle point of $L(\mathbf{z},\boldsymbol{\lambda})$, which means that $L(\mathbf{z}^*, \boldsymbol{\lambda}^*) = \max\limits_{\lambda}\min\limits_{\mathbf{z}} L(\mathbf{z},\boldsymbol{\lambda})$ on a defined frame of definition.

Function $L(\mathbf{z},\boldsymbol{\lambda})$ is monotonic by $\mathbf{z}$ for any fixed $\boldsymbol{\lambda}$ that's why its solution will be obtained on the ends of intervals [0,1]: z_l=1 for positive coefficients

$\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j \lambda_k - A$, z_l=0 for negative coefficients. If there are no zero coefficients ($\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j \lambda_k - A$) in a solution then the statement is proved.

3) This statement is an accomplishment of reasoning of 2).

4) The left part of inequality (6) is a consequence of the fact that it is a solution of the same problem on a widened set. The right part can be obtained as

$$L(\mathbf{z}^*_{(4)}, \boldsymbol{\lambda}^*_{(4)}) = \min_{z} \max_{\boldsymbol{\lambda}} L(\mathbf{z}, \boldsymbol{\lambda}) \le \max_{\boldsymbol{\lambda}} L(\mathbf{z}^*_{\{0,1\}}, \boldsymbol{\lambda}).$$

♦

Theorem 1 says that sometimes the solution of (1) will be obtained as a solution of (3), which is polynomial. In general case the algorithm divides the set of variables into 3 parts: "definitely non-exist", "definitely exist" and "weighted". For someone likes that there will not be "weighted" variables we consider several conditions and strategies how to determine $\mathbf{z}$ for the cases when the equality $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* = 2A$ is hold.

1) If nevertheless there will be a small set of such axes then one can make an exhaustive search of their combinations and calculate a confidence level (6) for each of them. The decision of rejection of a set of axes can be obtained if this set gives a long confidence interval.
2) If, in opposite, this "singular" set is large for exhaustive search then only the hypotheses of one-by-one inclusion for every of such axes can be examined.
3) If an axe l has a large weight z_l and a tight confidence interval then it will be remained, otherwise it will be rejected [36]. Another variant is to remain these continuous coefficients z_l, which amount will be small, as is: we have no confidence to remove l-th axe and no confidence to remain it. Let us take it with its weight z_l. This variant stays algorithm as an effective, non-exhaustive one.

Let us discuss a geometrical meaning of parameter A. It can be easily obtained that $\mathbf{w} = \sum_{i=1}^{N} y_i \lambda_i \mathbf{x}_i$, and $\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^* \lambda_k^* = w_l^2$. Thus it is obvious that $2A$ is a threshold for measure of weakness of l-th slope of a separated hyperplane: a coordinate with a weak slope doesn't influence the separation process.

Note that in the case $A=0$ all the space correct to the axes with $w_l=0$ will be chosen.

A saddle point topic has to be discussed in more deep way. According to the definition (3) we can solve only the *min max* problem, which can have solutions that are not saddle points (see Appendix 1). The saddle point was introduced in this work for technical purposes: it supports a discrete solution that we need. Nevertheless on the other hand it supports the Nash equilibrium, which we hope gives more stable classification rules. To explain it consider two independent problems:

$$\min_{\mathbf{z}} \max_{\lambda} L(\mathbf{z}, \boldsymbol{\lambda})$$

and

$$\max_{\lambda} \min_{\mathbf{z}} L(\mathbf{z}, \boldsymbol{\lambda}).$$

The first one is interpreted as following: *to find the best margin classification learning for every combination of variables and after it to choose a combination of variables supports the best margin from the bests.*

The second one is interpreted as following: *to find the best margin combination of variables for every classification learning and after it to choose a classification learning supports the best margin from the bests.*

Now we have two independent criteria for the feature selection problem, which are suitable to intuition. Saddle point algorithm solves both of them simultaneously and it has to be more stable than separate solution of one of them.

3. Saddle Point Algorithm

In the previous section we formulated a problem of learning classification, which simultaneously maximizes margin and minimizes feature space as a saddle point problem. In this section we will describe an algorithm for saddle point search. In order to make this new topic in learning field more clearly for reader this section organized so that the algorithm is described twice

- theoretically using the necessary notions from convex analysis;
- in the form of pseudocode.

We analyzed several saddle-point search algorithms to solve problem (9). The results of our analysis presented in Appendix 1.

At first, algorithm described in [20] was programmed. Our choice based on the fact that in [20] was described computation experience of applying the saddle-point search algorithm for problems where function $L(\mathbf{z}, \boldsymbol{\lambda})$ is linear by variable $\mathbf{z}$ and quadratic by $\boldsymbol{\lambda}$. Computations show poor convergence of points generating by the algorithm. The same behavior of the algorithm remains in the case when good approximation to saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ is provided as starting point.

The second algorithm realized was separate calculation of $\mathbf{z}$-component and $\boldsymbol{\lambda}$-component of saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ (see subsection A1.1. and formulas (A6), (A7)). We also used constructions of subgradient (Definition A3, Appendix 1) and projection ((A15), Appendix 1).

Lets consider functions $\psi(\mathbf{z})$ and $\varphi(\boldsymbol{\lambda})$ defined below:

$$\psi(\mathbf{z}) = \max_{\boldsymbol{\lambda}} L(\mathbf{z}, \boldsymbol{\lambda}) = \sum_{j=1}^{N} \lambda_j - \sum_{j=1}^{N} \sum_{k=1}^{N} \left(y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l \right) \lambda_j \lambda_k + A \sum_{l=1}^{M} z_l , \qquad (13)$$

$$\sum_{j=1}^{N} \lambda_j y_j = 0 \,,\; 0 \le \lambda_j \le C, j=1,...,N.$$

$$\varphi(\boldsymbol{\lambda}) = \min_{\mathbf{z}} L(\mathbf{z},\boldsymbol{\lambda}) = \sum_{j=1}^{N} \lambda_j - \sum_{j=1}^{N}\sum_{k=1}^{N}\left(y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l \right)\lambda_j \lambda_k + A\sum_{l=1}^{M} z_l \,, \tag{14}$$

$$0 \le z_l \le 1,\; l=1,...,M.$$

Theorem 5.5 from book [6] says:

Let I -arbitrary set, $f_i(x)$ - convex function for $\forall i \in I$ then $f(x) = \sup\{ f_i(x) \mid i \in I \}$ is a convex function .

This theorem can be rewritten in the form:

Let I -arbitrary set, $f_i(x)$ - concave function for $\forall i \in I$ then $f(x) = \inf\{ f_i(x) \mid i \in I \}$ is a concave function .

In our case $L(\mathbf{z},\boldsymbol{\lambda})$ is linear and hence convex by $\mathbf{z}$ for fixed $\boldsymbol{\lambda}$. So taking $I = \left\{ \boldsymbol{\lambda} \mid \sum_{j=1}^{N} \lambda_j y_j = 0, C \ge \lambda_j \ge 0, j = 1,..., N \right\}$ we see that $\psi(\mathbf{z})$ is convex function.

On the other hand $L(\mathbf{z},\boldsymbol{\lambda})$ is concave by $\boldsymbol{\lambda}$ for fixed $\mathbf{z}$. Taking $I = \{ \mathbf{z} \mid 0 \le z_l \le 1, l = 1,..., M \}$ we see that $\varphi(\lambda)$ is concave function. Any solution of the problem:

$$\min_{\mathbf{z}} \{ \psi(\mathbf{z}) \mid 0 \le z_l \le 1, l = 1,..., M \} \tag{15}$$

gives $\mathbf{z}$-component of saddle point ($\mathbf{z}^*$, $\boldsymbol{\lambda}^*$).

Any solution of the problem:

$$\max_{\boldsymbol{\lambda}} \left\{ \varphi(\boldsymbol{\lambda}) \mid \sum_{j=1}^{N} \lambda_j y_j = 0, \lambda_j \ge 0, j = 1,..., N \right\} \tag{16}$$

gives $\boldsymbol{\lambda}$-component of saddle point ($\mathbf{z}^*$, $\boldsymbol{\lambda}^*$).

We realized algorithm of non-differentiable convex (concave) optimization for solving problems (16) and (15).

The algorithm uses the following scheme for generating sequence of points converging to solution of (16) and (15)[6]:

[6] Each step of iterative procedure consists of two sub steps: step by subgradient and projection to the set of constraints. This is the reason for usage of indexes t, $t+\frac{1}{2}$, $t+1$ in the definition of the procedure [20].

$$\mathbf{z}^{t+\frac{1}{2}} := \mathbf{z}^t - \alpha_t^z \partial\psi(\mathbf{z}^t),\ \ \mathbf{z}^{t+1} := \pi_Z\left(\mathbf{z}^{t+\frac{1}{2}}\right), t = 0,1,... \tag{17}$$

$$\boldsymbol{\lambda}^{t+\frac{1}{2}} := \boldsymbol{\lambda}^t + \alpha_t^\lambda \partial\varphi(\boldsymbol{\lambda}^t),\ \ \boldsymbol{\lambda}^{t+1} := \pi_\Lambda\left(\boldsymbol{\lambda}^{t+\frac{1}{2}}\right), t = 0,1,... \tag{18}$$

where t- is a iteration number, $\partial\psi(\mathbf{z})$ is subgradient of function $\psi(\mathbf{z})$ in point $\mathbf{z}^t$,$\pi_Z(\mathbf{z})$ is projection of point $\mathbf{z}\in\boldsymbol{R}^M$ on set $Z=\{\ 0 \le z_l \le 1,\ l=1,...,M\ \}$, $\partial\varphi(\boldsymbol{\lambda})$ is subgradient of function $(-\varphi(\lambda))$ (supergradient) in point $\boldsymbol{\lambda}^t$, $\pi_\Lambda(\boldsymbol{\lambda})$ is projection of point $\boldsymbol{\lambda}\in\boldsymbol{R}^N$ on set $\Lambda = \left\{\sum_{j=1}^{N} \lambda_j y_j = 0, 0 \le \lambda_j \le C, j = 1,...,N\right\}$, α_t^z and α_t^λ are step-sizes.

We realized several well-known methods of setting values for the step-sizes. The best result gives a method that uses bounds on the optimal value of optimized function [5]. Lets describe computation of step-size α_t^z.

If we know value $\psi(\mathbf{z}^*)$, where $\mathbf{z}^*$ is solution of (15), then we can set step value to

$$\alpha_t^z = \frac{\psi(\mathbf{z}^t) - \psi(\mathbf{z}^*)}{\left\| \partial\psi(\mathbf{z}^t) \right\|^2}. \tag{19}$$

Such choice of step-size gives convergence of sequence (17) to $\mathbf{z}^*$, see , for example theorem 7.2 from book [5].

The value of $\psi(\mathbf{z}^*)$ is unknown, we just looking for it. As it is shown in paragraphs 5.3 and 7.2 of [5] a low estimate of $\psi(\mathbf{z}^*)$ can be used for practical computations.

Obviously $\psi(\mathbf{z}) \ge \varphi(\boldsymbol{\lambda})$ for all $\mathbf{z}\in Z$, $\boldsymbol{\lambda}\in\Lambda$. Assign φ^{rec} as maximal record value of $\varphi(\boldsymbol{\lambda})$ achieved during optimization process to iteration t. φ^{rec} is a low estimate of $\psi(\mathbf{z}^*)$ ($\psi(\mathbf{z}^*) \ge \varphi^{rec}$) and we have step-size formula:

$$\alpha_t^z = \frac{\psi(\mathbf{z}^t) - \varphi^{rec}}{\left\| \partial\psi(\mathbf{z}^t) \right\|^2} \tag{20}$$

In the same way we get formula for value of step-size by $\boldsymbol{\lambda}$:

$$\alpha_t^{\lambda} = \frac{\psi^{rec} - \varphi(\boldsymbol{\lambda}^t)}{\left\| \partial\varphi(\boldsymbol{\lambda}^t) \right\|^2}. \tag{21}$$

To calculate $\partial\psi(\mathbf{z})$ - subgradient[7] of $\psi(\mathbf{z})$ in point $\mathbf{z}$ we should get $\boldsymbol{\lambda}^{\max}$ - solution of the problem:

$$\arg\max_{\boldsymbol{\lambda}} L(\mathbf{z},\boldsymbol{\lambda}) = \sum_{j=1}^{N}\lambda_j - \sum_{j=1}^{N}\sum_{k=1}^{N}\left(y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l \right)\lambda_j\lambda_k + A\sum_{l=1}^{M} z_l \tag{22}$$
$$\sum_{j=1}^{N}\lambda_j y_j = 0, 0 \le \lambda_j \le C,\ j = 1,...,N$$

and calculate derivative of function $L(\mathbf{z}, \boldsymbol{\lambda}^{\max})$ by $\mathbf{z}$. The formula for l-component of subgradient $\partial\psi(\mathbf{z})$ will be[8]:

$$\frac{\partial\psi(\mathbf{z})}{\partial z_l} = -\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j^{\max}\lambda_k^{\max} + A, \tag{23}$$

where $\boldsymbol{\lambda}^{\max}$ is a solution of ***dual SVM problem***:

$$\begin{aligned} &\sum_{j=1}^{N}\lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(y_j y_k \sum_{l=1}^{M}\hat{x}_j^l\hat{x}_k^l)\lambda_j\lambda_k \to \max_{\boldsymbol{\lambda}} \\ &\sum_{j=1}^{N}\lambda_j y_j = 0,\ j = 1,...,N, \\ &0 \le \lambda_j \le C,\ j = 1,...,N, \end{aligned} \tag{24}$$

and[9] $\hat{\mathbf{x}}^l = \sqrt{z_l}\,\mathbf{x}^l$, $l=1,...,M$.

---

[7] See Definition A3 in Appendix 1. Using geometrical language **subgradient d** is opposite to the first n components of normal vector (**-d**, 1) of **supporting hyperplane** of a set $S=\{(\mathbf{x}, y) \mid y \ge f(\mathbf{x})\}$ in point $(\mathbf{x}, f(\mathbf{x}))$ (a hyperplane that contains S in one of its closed halfspaces and intersects the closure of S with at least one point).

[8] Functions $\psi(\mathbf{z})$ and $\varphi(\boldsymbol{\lambda})$ are non-differentiable. That's why $\frac{\partial\psi(\mathbf{z})}{\partial z_l}$ and $\frac{\partial\varphi(\boldsymbol{\lambda})}{\partial\lambda_j}$ are only the notations for the corresponding components of subgradient vectors.

[9] To find $\boldsymbol{\lambda}^{\max}$ we use SVM Light program [39, 40]. We transformed SVM Light to DLL after the permission of its author Thorsten Joachims. Input data to SVM Light was preprocessed by multiplying each l-column of data matrix with $\sqrt{z_l}$.

To calculate $\partial\varphi(\boldsymbol{\lambda})$ - supergradient of $\varphi(\boldsymbol{\lambda})$ in point $\boldsymbol{\lambda}$ we should get $\mathbf{z}^{\min}$ - solution of the ***linear programming*** problem:

$$\arg\min_{\mathbf{z}} L(\mathbf{z},\boldsymbol{\lambda}) = \sum_{j=1}^{N}\lambda_j - \sum_{j=1}^{N}\sum_{k=1}^{N}\left(y_j y_k \sum_{l=1}^{M} z_l x_j^l x_k^l \right)\lambda_j\lambda_k + A\sum_{l=1}^{M} z_l \qquad (25)$$
$$0 \le \mathbf{z}_l \le 1,\ l=1,\ldots,M.$$

and calculate derivative of function $L(\mathbf{z}^{\min}, \boldsymbol{\lambda})$ by $\boldsymbol{\lambda}$. The formula for j-component of supergradient $\partial\varphi(\boldsymbol{\lambda})$ will be:

$$\frac{\partial\varphi(\boldsymbol{\lambda})}{\partial\lambda_j} = 1 - \sum_{k=1}^{N}\left(y_j y_k \sum_{l=1}^{M} z_l^{\min} x_j^l x_k^l \right)\lambda_k \,. \qquad (26)$$

To find $\mathbf{z}^{\min}$ we calculate coefficients of linear by $\mathbf{z}$ function $L(\mathbf{z},\boldsymbol{\lambda})$:

$$p_l = -\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N} y_j y_k x_j^l x_k^l \lambda_j\lambda_k + A \qquad (27)$$

and calculate

$$z_l^{\min} = \begin{cases} 1, & p_l < 0, \\ 0, & p_l \ge 0. \end{cases} \qquad (28)$$

Projection $\mathbf{z}^{\text{pr}} = \pi_Z(\hat{\mathbf{z}})$, $Z=\{\ 0 \le z_l \le 1,\ l=1,\ldots,M\ \}$, $\hat{\mathbf{z}} \in \mathbf{R}^M$ calculated by simple formula:

$$z_l^{pr} = \begin{cases} 0, & \hat{z}_l < 0, \\ z_l, & 0 \le \hat{z}_l \le 1, l = 1,\ldots,M, \\ 1, & \hat{z}_l > 1. \end{cases} \qquad (29)$$

Projection $\boldsymbol{\lambda}^{pr} = \pi_\Lambda(\hat{\boldsymbol{\lambda}})$, $\Lambda = \left\{ \sum_{j=1}^{N}\lambda_j y_j = 0, 0 \le \lambda_j \le C, j = 1,\ldots,N \right\}$, $\hat{\boldsymbol{\lambda}} \in \mathbf{R}^N$ is given by solution of ***quadratic programming*** problem:

$$\min_{\boldsymbol{\lambda}} \left\{ \sum_{j=1}^{N}\left(\lambda_j - \hat{\lambda}_j\right)^2 \mid \sum_{j=1}^{N}\lambda_j y_j = 0, 0 \le \lambda_j \le C, j = 1,\ldots,N \right\} \qquad (30)$$

Now we give a pseudocode description of the algorithm:

Initialization.

Choose $\mathbf{z}^0 \in Z$, $\boldsymbol{\lambda}^0 \in \Lambda$.

($Z=\{0 \le z_l \le 1,\ l=1,...,M\}$, $\Lambda = \left\{ \sum_{j=1}^{N} \lambda_j y_j = 0, 0 \le \lambda_j \le C, j = 1,..., N \right\}$).

Set solution accuracy $\varepsilon>0$

Set $t=0$.

Init record values $\psi^{rec} = \psi(\mathbf{z}^0)$, $\varphi^{rec} = \varphi(\boldsymbol{\lambda}^0)$, $\mathbf{z}^{rec} = \mathbf{z}^0$, $\boldsymbol{\lambda}^{rec} = \boldsymbol{\lambda}^0$, where ψ, φ calculated using formulas (13), (14).

Stopping test.

if $\dfrac{\psi^{rec} - \varphi^{rec}}{\left(1 + \psi^{rec}\right)} \le \varepsilon$

then $\mathbf{z}^* = \mathbf{z}^{rec}$, $\boldsymbol{\lambda}^* = \boldsymbol{\lambda}^{rec}$. Stop.

Step by z.

Calculate $\partial\psi(\mathbf{z}^t)$ using procedure (22), (23) and $\alpha_t^z = \dfrac{\psi\left(\mathbf{z}^t\right) - \varphi^{rec}}{\left\| \partial\psi\left(\mathbf{z}^t\right) \right\|^2}$ using (20).

Calculate $\mathbf{z}^{t+\frac{1}{2}} = \mathbf{z}^t - \alpha_t^z \partial\psi\left(\mathbf{z}^t\right)$ (see (17)).

Project $\mathbf{z}^{t+1} = \pi_Z\left(\mathbf{z}^{t+\frac{1}{2}}\right)$ using (29).

Step by $\boldsymbol{\lambda}$.

Calculate $\varphi(\boldsymbol{\lambda}^t)$ using procedure (25) - (28) and $\alpha_t^\lambda = \dfrac{\psi^{rec} - \varphi\left(\boldsymbol{\lambda}^t\right)}{\left\| \partial\varphi\left(\boldsymbol{\lambda}^t\right) \right\|^2}$ using (21).

Calculate $\boldsymbol{\lambda}^{t+\frac{1}{2}} = \boldsymbol{\lambda}^t + \alpha_t^\lambda \partial\psi\left(\boldsymbol{\lambda}^t\right)$ (see (18)).

Project $\boldsymbol{\lambda}^{t+1} = \pi_Z\left(\boldsymbol{\lambda}^{t+\frac{1}{2}}\right)$ by solving a quadratic programming problem (30).

Update records

If $\psi(\mathbf{z}^{t+1}) < \psi^{rec}$, then $\psi(\mathbf{z}^{t+1}) = \psi^{rec}$, $\mathbf{z}^{rec} = \mathbf{z}^{t+1}$.

If $\varphi(\boldsymbol{\lambda}^{t+1}) > \varphi^{rec}$ then $\varphi^{rec} = \varphi(\boldsymbol{\lambda}^{t+1})$, $\boldsymbol{\lambda}^{rec} = \boldsymbol{\lambda}^{t+1}$.

$t=t+1$

go to Stopping test

Remark 2. *Additional stopping criteria can be used in optimization process by* $\mathbf{z}$ *or* $\boldsymbol{\lambda}$. *The following theorem 7.3 from* [5] *gives such criterion:*

Let f(x)-convex function on R^n, *Q convex set,* $Q \subset R^n$, $x^* \in Q$ *then* x^**-global minimum of f(x) on Q* $\Leftrightarrow$. *subgradient* $\partial f(\mathbf{x}^*), \forall \mathbf{x} \in Q \ \left(\partial f(\mathbf{x}^*), \mathbf{x} - \mathbf{x}^*\right) \geq 0.$

4. Experimental Results

4.1. Experimental Framework

The proposed algorithm is still time consuming. It is working rather fast on data arrays with ~30 observations (calculation time <1sec), but it takes to it more than hour to process real data arrays with more than 1000 observations.

In order to study the properties of proposed SP-SVM mechanism the following experimental framework was realized. Let *M* is overall number of objects in a Benchmark set.

1. Any given matrix was used for generation of submatrices of objects of length *L*, which is taken randomly with uniform distribution from interval 15-30. Thus each matrix contained *L* objects and all *N* variables. Proportion of positive and negative labels was saved as in the original big matrix.
2. Parameter *C* was chosen according to [41]: taken SVM light, chosen default $C=(average||x||)^{-1}$, extracted support vectors and calculated default C only with support vectors.
3. Parameter *A* chosen randomly with uniform distribution from a predefined interval. The boundaries for *A* were chosen experimentally in such a way that the solutions would be non-trivial (small $A \sim 0$ remain all the variables, large *A* – delete a lot, may be all the variables). The

boundaries were chosen as a function of the hyperplane coefficients **w** for a case A=0.

4. Saddle point training algorithm found a $(\mathbf{z}^*, \boldsymbol{\lambda}^*)$ approximation to saddle point.
5. Three resulted sets of variables are stored:
   a. "Deleted" – those, which obtained weight 0 (assigned as not important) and were deleted by the Saddle Point algorithm;
   b. "Assigned" – those, which obtained weight 1 (assigned as highly important) by the Saddle Point algorithm;
   c. "Weighted" – those, which obtained weight $0 \le A \le 1$ (see section 2.2).
6. All objects the complementary (M-L) objects were used as examined objects.
7. 3 recognition quality evaluators for SVM were calculated according to SVM light terminology:
   a. Accuracy = (overall number of true classified objects)/(overall number of objects for examination).
   b. Precision = (number of true classified objects in positive (+1) class)/(overall number of objects in positive (+1) class).
   c. Recall =(number of true classified objects in positive (+1) class)/(number of true classified objects in both classes).
8. The same set of 3 evaluators was calculated for the SP-SVM recognizer.

4.2. Benchmark

We made experiments with different data sets. Inasmuch as all the results and all the open problems were rather common for all the sets we decided that in this first report we illustrate the method implementation on only one Benchmark: Vowel dataset and to prepare a separated report only about the experiments on the different Benchmark sets.

The Vowel dataset contains 992 points, corresponding to 11 English vowel sounds, represented by ***10 features*** and a label. The features are derived from analysis of sample windowed segments of the speech signal and are real-valued.

Dataset is available from website of Hastie, Tibshirani, Friedman (HTF) book: http://www-stat-class.Stanford.EDU/~tibs/ElemStatLearn/
(in Data section). It is also in UCI (format is somewhat different than on HTF website) ftp://ftp.ics.uci.edu/pub/machine-learning-databases/undocumented/connectionist-bench/vowel.

We extracted only the first two classes from this dataset.

4.3. Results

The data was centralized and normalized by columns:

$$\hat{\mathbf{x}}_i = \frac{(\mathbf{x}_i - \bar{x}_i)}{\|\mathbf{x}_i - \bar{x}_i\|}, \quad i=1,\ldots,M,$$

where $\bar{x}_i = \frac{1}{N_{train}} \sum_{j=1}^{N_{train}} x_i^j$, N_{train} is number of objects only in training set.

Calculating $\|\mathbf{x}_i - \bar{\mathbf{x}}_i\|$ as norm over only the training objects.

There were generated 20 sub matrices. 31 random uniform selections of parameter A from interval[10] P=[0.17* $\min_{i=1,\ldots,M} w_i^0$, 0.6* $\max_{i=1,\ldots,M} w_i^0$] were done for each sub matrix.

The results presented in Tables 1, 2.

In Table 1 presented results of comparison of maximal Accuracy of saddle point in the examined interval P with SVM results for a same data submatrix. Maximum of Accuracy is taken over all the 31 choices of parameter A from the interval P. These results show possibilities to decrease the feature space and to increase recognition quality simultaneously.

Let us remark that these results prove only the existence of an optimal value of parameter A, which gives maximal Accuracy on the recognition stage. The choice of this optimal value is a plan of our future investigations. This is the reason that we tried to choose this parameter in a very simple way: to chose such A' that gives an average value of the optimized saddle point criterion. Once more, average is taken over all the 31 choices of parameter A from the interval P. The results of this set of experiments presented in Table 2.

One can see from the Table 1 that the Maximal Accuracy of saddle point is in average 5-7% better than SVM Accuracy, in some cases this difference exceeds 15%. At the same time approximately half of the variables are eliminated (the average number of eliminated variables over all the 20 submatrixes is 4.6 from 10). Average SVM Accuracy is 70.64; average saddle point Accuracy is 78.35. The Accuracy obtained with saddle point algorithm is better than the SVM Accuracy in 18 cases from 20. In case 8 saddle point and SVM Accuracies are equal. The only case of saddle point accuracy greater than SVM accuracy is case number 18. Let us remark that the SVM Accuracy in this case has a large value for the examined material and saddle point Accuracy is approximately the same. One can see that even if 70% of variables are eliminated (cases 3 and 11) the saddle point Accuracy is greater than the SVM Accuracy.

Obviously the results of the second set of experiments are worse. We add a column 3 to the Table 2. This column shows that in approximately 39% of all the 20*31=620 experiments the value of saddle point Accuracy is not worse than the value of SVM Accuracy. Average SVM Accuracy is 70.64; average saddle point Accuracy is 70.0. For the choice of result corresponding to average of saddle point values on interval P only in 8 cases from 20 the saddle point Accuracy is better than the SVM Accuracy. In cases 3,5,20 the saddle point Accuracy is much worse than the SVM Accuracy. In other 8 cases when saddle point Accuracy is worse than the SVM Accuracy it is still not much more badly. At the same time the number of deleted variables is rather big. In average 5.6

[10] $\mathbf{w}^0$ is a normal of hyperplane for A=0.

variables from 10 were eliminated. Once more cases 7 and 11 remove 70% of variables and the Accuracy remains very close to the SVM Accuracy, only a little worse.

This analysis of Table 2 shows that even in case of such simple choice of parameter A the algorithm does feature selection without big losses in Accuracy. Analysis of Table 1 shows that to find a method of searching for an optimal A is a very important task.

Number of Submatrix	Number of Objects in Submatrix	Accuracy SVM	Max Accuracy Saddle Point	Precision in Max Accuracy SP (SVM Precision)	Recall in Max Accuracy SP (SVM Recall)	Number of deleted variables in Max Accuracy SP (z_i=0)	Number of definitely assigned variables in Max Accuracy SP (z_i=1)
1	18	75.3	80.24	98.76 (95)	61.53 (63.1)	4	2
2	30	75.33	85.33	93.33 (76)	54.68 (50.44)	5	2
3	20	52.5	68.125	66.66 (25)	51.37 (25)	7	1
4	24	68.58	80.76	88.46 (70.51)	54.76 (51.4)	6	1
5	15	50.3	72.72	62.79 (4.65)	45 (4.81)	6	1
6	29	74.17	79.47	89.61 (84.41)	57.5 (58.03)	5	1
7	21	80.5	83.64	98.71 (93.58)	57.89 (57.03)	5	1
8	24	67.3	67.3	51.28 (51.28)	38.09 (38.09)	1	7
9	26	68.83	72.07	100 (100)	65.76 (68.86)	6	3
10	21	64.77	77.98	72.83 (65.43)	47.58 (51.45)	6	1
11	28	75.65	80.92	86.3 (84.93)	51.21 (53.91)	7	1
12	24	76.28	80.12	81.48 (67.9)	74.07 (52.8)	0	7
13	15	68.94	88.81	83.54 (100)	46.15 (71.17)	6	2
14	19	71.875	83.75	83.95 (64.19)	50.746 (45.21)	5	3
15	20	76.31	86.18	84.81 (74.68)	51.145 (50.86)	6	2
16	28	72.04	77.63	82.92 (89.02)	54.4 (62.93)	6	2
17	19	74.54	80.6	86.74 (74.69)	54.13 (50.4)	3	2
18	15	84.21	82.89	78.48 (83.54)	49.2 (51.56)	0	8
19	24	76.92	79.48	82.05 (80.76)	51.61 (52.5)	6	1
20	21	58.49	59.1	55.12 (84.6)	45.74 (70.96)	2	6

Table 1. Vowel Data Processing.
Maximal Accuracy.

Number of Submatrix	Number of Objects in Submatrix	% of times when (*Saddle Point Accuracy* ≥ *SVM Accuracy*) in experimental interval	Accuracy SVM	Accuracy Saddle Point corresponding to average value of criterion in experimental interval	Precision in Saddle Point corresponding to average value of criterion in experimental interval (SVM Precision)	Recall in Saddle Point corresponding to average value of criterion in experimental interval (SVM Recall)	Number of deleted variables in Saddle Point corresponding to average value of criterion in experimental interval (z_i=0)	Number of definitely assigned variables in Saddle Point corresponding to average value of criterion in experimental interval (z_i=1)
1	18	100	75.3	77.16	100 (95)	64.8 (63.1)	3	4
2	30	61	75.33	85.33	93.33 (76)	52.0 (50.44)	5	2
3	20	23	52.5	46.87	0 (25)	0 (25)	6	1
4	24	94	68.58	74.35	71.79 (70.51)	48.27 (51.4)	5	1
5	15	55	50.3	47.87	0 (4.65)	0 (4.81)	5	1
6	29	42	74.17	78.14	94.8 (84.41)	61.86 (58.03)	5	2
7	21	31	80.5	74.84	84.61 (93.58)	55.46 (57.03)	7	1
8	24	10	67.3	62.82	61.53 (51.28)	48.97 (38.09)	5	2
9	26	13	68.83	61.68	100 (100)	76.84 (68.86)	8	1
10	21	23	64.77	61.0	34.56 (65.43)	28.86 (51.45)	6	1
11	28	77	75.65	72.36	87.67 (84.93)	58.18 (53.91)	7	1
12	24	13	76.28	75.0	66.66 (67.9)	46.15 (52.8)	6	2
13	15	20	68.94	88.19	82.27 (100)	45.77 (71.17)	6	2
14	19	55	71.875	69.37	54.32 (64.19)	39.6 (45.21)	5	3
15	20	29	76.31	82.23	81.01 (74.68)	51.2 (50.86)	6	2
16	28	35	72.04	74.53	84.14 (89.02)	57.5 (62.93)	5	3
17	19	55	74.54	70.9	67.46 (74.69)	47.86 (50.4)	3	3
18	15	0	84.21	69.07	55.69 (83.54)	41.9 (51.56)	6	1
19	24	45	76.92	78.84	79.48 (80.76)	50.4 (52.5)	6	1
20	21	6	58.49	49.68	61.53 (84.6)	60.75 (70.96)	7	3

Table 2. Vowel Data Processing.
Average value of optimized criterion in experimental interval.

5. Exact Wrapped Method for Feature Selection in Regression

The SVM linear regression model is based on the same ideas [1]. Suppose that we have a training sample $\{\boldsymbol{x}_i, y_i\}$, $\boldsymbol{x}_i \in \mathbf{R}^M$, $y_i \in \mathbf{R}$, $i=1,...,N$. Margin maximization (VC-dimension minimization) now reflects in attempt to approximate as more as possible experimental points with y-axe ε-tube:

$$\begin{cases} \dfrac{1}{2}\|\mathbf{w}\|^2 \to \min\limits_{w,b} \\ y_i - \left(\mathbf{w}^T \mathbf{x}_i - b\right) \le \varepsilon, \\ y_i - \left(\mathbf{w}^T \mathbf{x}_i - b\right) \ge -\varepsilon, \\ i = 1,..., N. \end{cases} \tag{31}$$

The same as in the classification case: to approximate all the points with ε precision can be infeasible task, and a structural risk minimization is needed:

$$\begin{cases} \dfrac{1}{2}\|\mathbf{w}\|^2 + C\sum\limits_{i=1}^{N}\left(\delta_i + \delta_i^*\right) \to \min\limits_{w,b,\delta_i,\delta_i^*} \\ y_i - \left(\mathbf{w}^T \mathbf{x}_i - b\right) \le \varepsilon + \delta_i, \\ y_i - \left(\mathbf{w}^T \mathbf{x}_i - b\right) \ge -\varepsilon - \delta_i^* \\ \delta_i, \delta_i^* \ge 0,\ i = 1,..., N, \end{cases} \tag{32}$$

which means that the points into the ε-tube do not penalized and the points outside the ε-tube penalized linearly. This problem has the following dual:

$$\begin{cases} \sum\limits_{j=1}^{N} y_j\left(\lambda_j - \lambda_j^*\right) - \dfrac{1}{2}\sum\limits_{j=1}^{N}\sum\limits_{k=1}^{N} \mathbf{x}_j^T \mathbf{x}_k \left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right) - \varepsilon\sum\limits_{j=1}^{N}\left(\lambda_j + \lambda_j^*\right) \to \max\limits_{\lambda,\lambda^*} \\ \sum\limits_{j=1}^{N}\left(\lambda_j - \lambda_j^*\right) = 0, \\ 0 \le \lambda_j, \lambda_j^* \le C,\ j = 1,..., N \end{cases} \tag{33}$$

The same way we introduce a feature selection problem

$$F(Q)=\left\{\begin{array}{c}\min\limits_{w_Q,b_Q,\delta_Q^i,\delta_Q^*}\dfrac{1}{2}\left\|\mathbf{w}_Q\right\|^2+C\sum\limits_{i=1}^{N}\left(\delta_Q^i+\delta_Q^{i\,*}\right)+A|Q| \\ y_i-\left(\mathbf{w}_Q^T\mathbf{x}_i^Q-b_Q\right)\le\varepsilon+\delta_Q^i, \\ y_i-\left(\mathbf{w}_Q^T\mathbf{x}_i^Q-b_Q\right)\ge-\varepsilon-\delta_Q^{i\,*}, \\ \delta_Q^i,\delta_Q^{i\,*}\ge 0, i=1,...,N,\end{array}\right\}\to\min_Q \tag{34}$$

and nest this problem into a continuous one

$$\left\{\begin{array}{l}\min\limits_{\mathbf{z}}\max\limits_{\lambda,\lambda^*}L(\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\lambda}^*)=\sum\limits_{j=1}^{N}y_j\left(\lambda_j-\lambda_j^{\,*}\right)- \\ -\dfrac{1}{2}\sum\limits_{j=1}^{N}\sum\limits_{k=1}^{N}\left(\lambda_j-\lambda_j^{\,*}\right)\left(\lambda_k-\lambda_k^{\,*}\right)\sum\limits_{l=1}^{M}z_l x_j^l x_k^l- \\ -\varepsilon\sum\limits_{j=1}^{N}\left(\lambda_j+\lambda_j^{\,*}\right)+A\sum\limits_{l=1}^{M}z_l \\ \quad\sum\limits_{j=1}^{N}(\lambda_j-\lambda_j^{\,*})=0, j=1,...,N, \\ \quad 0\le\lambda_j,\lambda_j^{\,*}\le C, j=1,...,N, \\ \quad 0\le z_j\le 1, j=1,...,M.\end{array}\right. \tag{35}$$

Thus we ones more obtain a convex-concave function on a close convex set we transfer here straightforward the results of the previous section.

1) There exists a saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\lambda}^{**})$ of problem (35):

$$\left\{\begin{array}{l}(\mathbf{z}^*,\boldsymbol{\lambda}^*,\boldsymbol{\lambda}^{**})=\arg\min\limits_{\mathbf{z}}\max\limits_{\lambda,\lambda^*}L(\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\lambda}^*); \\ (\mathbf{z}^*,\boldsymbol{\lambda}^*,\boldsymbol{\lambda}^{**})=\arg\max\limits_{\lambda,\lambda^*}\min\limits_{\mathbf{z}}L(\mathbf{z},\boldsymbol{\lambda},\boldsymbol{\lambda}^*); \\ \quad\sum\limits_{j=1}^{N}(\lambda_j-\lambda_j^{\,*})=0, j=1,...,N, \\ \quad 0\le\lambda_j,\lambda_j^{\,*}\le C, j=1,...,N,, \\ \quad 0\le z_j\le 1, j=1,...,M;\end{array}\right. \tag{36}$$

2) If there are no such *l, l=1,...,M* that $\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right)x_j^l x_k^l = 2A$ then the saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\lambda}^{**})$ of problem (31) is a solution of problem (35), with $z_l=1$ if $\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right)x_j^l x_k^l > 2A$ and $z_l=0$ if $\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right)x_j^l x_k^l < 2A$;

3) A component z_l of a saddle point $(\mathbf{z}^*, \boldsymbol{\lambda}^*, \boldsymbol{\lambda}^{**})$ of problem (36) can be different from 0 or 1 only if $\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right)x_j^l x_k^l = 2A.$

4) Let $(\mathbf{z}_{(34)}^*, \boldsymbol{\lambda}_{(34)}^*, \boldsymbol{\lambda}_{(34)}^{**})$ is a solution of problem (34), $(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^{**})$ is a saddle point of (31) with every $0<z_j<1$ arbitrary changed to 0 or 1. The solution of (34) is bounded with

$$l(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^*, \boldsymbol{\lambda}_{\{0,1\}}^{**}) \le L(\mathbf{z}_{(34)}^*, \boldsymbol{\lambda}_{(34)}^*) \le \max_{\lambda,\lambda^*} L(\mathbf{z}_{\{0,1\}}^*, \boldsymbol{\lambda}, \boldsymbol{\lambda}^*). \qquad (37)$$

5) $\mathbf{w} = \sum_{j=1}^{N}\left(\lambda_j - \lambda_j^*\right)\mathbf{x}_j$, $\sum_{j=1}^{N}\sum_{k=1}^{N}\left(\lambda_j - \lambda_j^*\right)\left(\lambda_k - \lambda_k^*\right)x_j^l x_k^l = w_l^2$ and A is a parameter reflects a *l*-th hyperplane slope significance.

6. Nonlinear Kernels.

6.1. Negative Attempt of Generalization

It may seem that the results of this work can be easily transferred to nonlinear kernels. Let us show that this way immediately took us to no convex problems. Let us discuss the following problem:

$$F(z) = \sum_{j=1}^{N}\lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(y_j y_k \prod_{l=1}^{M} e^{-\frac{z_l (x_j^l - x_k^l)^2}{2\sigma^2}})\lambda_j\lambda_k, \quad 0 \le z \le 1.$$

For one-dimensional case it reduced to:

$$F(z) = \sum_{j=1}^{N}\lambda_j - \frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(y_j y_k e^{-\frac{z(x_j - x_k)^2}{2\sigma^2}})\lambda_j\lambda_k, \quad 0 \le z \le 1.$$

The necessary and sufficient condition for convexity of 2 times differentiable function is $F''(z) \geq 0$ ($F''(z) \leq 0$ for concavity). The second derivative of $F(z)$ is:

$$F''(z) = -\frac{1}{2}\sum_{j=1}^{N}\sum_{k=1}^{N}(y_j y_k (\frac{(x_j - x_k)^2}{2\sigma^2})^2 e^{-\frac{z(x_j-x_k)^2}{2\sigma^2}})\lambda_j \lambda_k$$

This function is non-negative if matrix $a_{ij}=(\frac{(x_j - x_k)^2}{2\sigma^2})^2 e^{-\frac{z(x_j-x_k)^2}{2\sigma^2}}$ is negative semidefinite and vice versa.

Let us assume an example of three points on a line with distance 2σ between the successive ones. Let y_j=+1 on the extreme points and y_j=-1 on the medium point. Let us take only two successive points. We obtain the following matrix a_{ij}:

$$\begin{Vmatrix} 0 & 4e^{-2z} \\ 4e^{-2z} & 0 \end{Vmatrix}.$$

Choose $\boldsymbol{\lambda}$=**1**, and z=0. We obtain that $F''(z)$=-(1/2)*(-16)=8 > 0.

Now take all the 3 points. We obtain the following matrix a_{ij}:

$$\begin{Vmatrix} 0 & 4e^{-2z} & 64e^{-8z} \\ 4e^{-2z} & 0 & 4e^{-2z} \\ 64e^{-8z} & 4e^{-2z} & 0 \end{Vmatrix}.$$

Once more choose $\boldsymbol{\lambda}$=**1**, and z=0. We obtain that $F''(z)=$(-1/2)*(128-16)=-56<0.
We see that a sign of $F''(z)$ is changed and depends on a sample distribution in space.

6.2. Indirect Kernel Method

Indirect Kernel method is very popular in the present time. It consists of two steps:

1) To chose a Kernel function $K(x,y)$ and to fill out the matrix $\mathbf{K}=\|K(x_i,x_j)\|$, $i,j=1,..,N$, where (x_i,x_j) is a pair of objects.
2) To use matrix **K** "object-object" as a matrix "object-feature" and to run linear SVM algorithm on this data matrix.

This method sometimes gives better results than the straightforward application of nonlinear kernels.

In our case, we can apply SP-SVM to this scheme. Objects and features here have the same sense, therefore if i-th object became a support vector, but it has weight z_i=0,

then one can suppose that it may be not taken to account. It means that SP-SVM can decrease drastically number of support vectors in indirect kernel method.

This assumption was not investigated in the present work; its study is in one of our future plans.

Appendix 1. Saddle Point Algorithms

This Appendix is dedicated to description and brief analysis of the saddle point theory, methods and algorithms. We didn't find such survey in literature.

As it was mentioned in the Introduction of the report we consider the saddle point extremes as very interesting structures, which match many ideas of simultaneous search in multiple criteria problems. We have emphasized several of the problems on the end of the Introduction. In the section 5 we have showed that presented in the report simultaneous analysis to build a classifier and to find "the best subspace" in which the classifier has to work can be directly extended to the regression model construction together with finding the best subspace for the model. We are sure that in every such different case one has to develop different type of the saddle point algorithm. For this reason it is important to have a survey in which one could a foundation of these algorithms, which will allow her or him to do the development. This was the basic reason for us to add to the report this survey. It was also an additional reason: (a) we hope to find collaborators who will find an interest to develop new learning methods based on the saddle points structures and who will decide to do a study to develop efficient saddle points procedures. For such readers it is convenient to have a complete picture about the new technology.[11]

The Appendix organized as following.

Section A1.1 dedicated to the main theoretical principles of saddle point of convex-concave function on a convex compact [3-8]. The saddle point existence theorems are formulated. The principle of separate search for the saddle point components is assigned here. This principle is the basis of our algorithm, which is described in section 4. The end of this section illustrates impossibility of gradient search for saddle points. A simple example from [8] shows that the trajectory of search can enter a loop, which never converges to a saddle point. This fact motivated a lot of theoretical investigations and various approaches to saddle point search.

Section A1.2.1 contains notions that are necessary for understanding of non-differentiable convex optimization theory and algorithms [3-8]. Some of them, such as duality gap estimator (A5) and subgradient (Definition A3) are important for our scheme described in Section 4.

Sections A1.2.2 – A1.2.4 describe methods of saddle point search for a smooth function on a strict convex set [3]. This section is written to introduce additional notions and principles, which will be useful in further analysis. Here we introduce such notions as iterative sequence, iterative step, and projection operator. All these notions used by us in Section 4.

Sections A1.2.5 – A1.2.6 dedicated to theoretical analysis of controlled continuous gradient processes [7-9, 43, 44]. These processes were introduced to theoretical understanding of saddle point convergence problems and to construct schemes for their solving. It played very important role in our understanding of algorithm convergence problems in our own scheme.

Section A1.2.7 contains an example of algorithm using a principle of parallel search for the saddle point components [13]. This principle was implemented in our scheme as we mentioned above.

Section A1.2.8 describes so-called "perturbation method" for saddle point search [17-19] with non-cooperative game interpretation.

Method of levels described in sections A1.3.1 – A1.3.2 is important as a proof of polinomiality of saddle point search [14-15].

---

[11] As we have mentioned several times our polynomial algorithm described in the section 4., even it is enough, to analyze well some practical problems it has a strong limitation. It can work only with training data of very small size (not more than 100-200 objects). At the same time we are sure that it could be modified to be able to work with "real for pattern recognition case" training data of size of many thousand objects. We hope that the survey gives a base to improve the algorithm to work on a large data.

The conditional ε-subgradient method [20] described in section A1.3.3 was chose by us as a first attempt to apply it to our problem. This method had been seemed us interesting from one hand because it was applied by its authors and there are some works contain results of its practical work. From the other hand its principle of searching only for one component of saddle point and simultaneous calculation of the second one seemed our important. Unfortunately we obtained very poor convergence results for this scheme and decided that it is not applicable for our problems.

Section A1.3.4 describes an interior-point scheme in the saddle point search. We describe a polynomial method for a very special class of saddle point problems [21]: convex-concave function with one of its components taken from the space of symmetric positive semidefinite matrices.

A1.1. Introduction

This section dedicated to the main theoretical principles of saddle point of convex-concave functions.

Let function $L(x,y)\in\boldsymbol{R}$, $(x,y)\in X\times Y$, X and Y some sets.

Definition A1. *Point $(x^*,y^*)\in X\times Y$ satisfying condition*

$$L(x^*,y)\le L(x^*,y^*)\le L(x^*,y),\ \ \forall x\in X,\ \forall y\in Y \tag{A1}$$

is a saddle point of function L(x,y) on X×Y

Equivalent condition for existence of saddle-point is the following equation:

$$\min_{x\in X}\max_{y\in Y} L(x,y)=\max_{y\in Y}\min_{x\in X} L(x,y) \tag{A2}$$

given all *min* and *max* values exist.

Let's introduce functions:

$$\varphi(y)=\min_{x\in X} L(x,y) \tag{A3}$$

$$\psi(x)=\max_{y\in Y} L(x,y) \tag{A4}$$

Proposition A1 [3]. *If $X\subset\boldsymbol{R}^N$, $Y\subset\boldsymbol{R}^M$ compact sets and L(x,y) a continuous function on X×Y then existence of saddle-point of function L(x,y) on X×Y is equivalent to fulfillment of (A2).*

Remark A1. *For saddle-point (x^*,y^*) we have*

$$\min_{x\in X}\max_{y\in Y} L(x,y)=\max_{y\in Y}\min_{x\in X} L(x,y)=L(x^*,y^*)$$

and

$$\max_{y\in Y}\varphi(y)=\varphi(y^*)=L(x^*,y^*)=\psi(x^*)=\min_{x\in X}\psi(x)\,.$$

So we can estimate $L(x^*,y^*)$ as following:

$$\varphi(y) \le L(x^*, y^*) \le \psi(x), \forall x \in X, \forall y \in Y. \quad \text{(A5)}$$

Definition A2. *Function L(x,y),x∈X,y∈Y is called convex-concave function on X×Y if L(x,$\bar{y}$) is convex by x∈X for any fixed $\bar{y}$∈Y and L($\bar{x}$,y) is concave by y for any fixed $\bar{x}$.*

Proposition A2. *Let X∈$\mathbf{R}^n$, Y∈$\mathbf{R}^m$ are closed, bounded, convex sets, convex-concave function L(x,y),x∈X,y∈Y is continuous on Ω=X×Y. Then function L(x,y) has saddle-point (x^*,y^*).*

See for example [6].

From now we assume that $X \in \mathbf{R}^n$, $Y \in \mathbf{R}^m$ are compact convex sets, function $L(x,y)$, $x \in X, y \in Y$ is convex-concave and continuous on $\Omega = X \times Y$.

Let's introduce sets:

$$X^* = \operatorname{argmin}\{\psi(x) : x \in X\} \quad \text{(A6)}$$

$$Y^* = \operatorname{argmax}\{\varphi(y) : y \in Y\} \quad \text{(A7)}$$

Any point $(x,y) \in X^* \times Y^*$ is a saddle-point of $L(x,y)$.

It's well known that $\psi(x)$ is convex and $\varphi(y)$ is concave. So we have the straight-forward method to calculate saddle-point of $L(x,y)$:

- Solve convex problem $\min_{x \in X} \psi(x)$ and get x-component of saddle-point (x^*, y^*).
- Solve concave problem $\max_{y \in Y} \varphi(y)$ and get y-component of saddle-point (x^*, y^*).

Notice that if we need to find only *x*-component or *y*- component of saddle-point then only one of the optimization problem must be solved.

And we will see later in conditional ε-subgradient method that we don't need to solve both problems and while search for *x*-component *y*-component can be constructed via some average scheme.

Let's look at a example from [8]:

$L(x,y)=x \cdot y, \quad X=[-1,1], \quad Y=[-1,1]$ (A8)

(0,0) - is a saddle-point of $L(x,y)$: $0 \cdot y \le 0 \cdot 0 \le x \cdot 0$.

This example shows that in general it is insufficient to solve only one of the problems $\min_{x \in X} \psi(x)$ and $\max_{y \in Y} \varphi(y)$ to find saddle point. We can try to apply standard gradient method: descend by *x* and ascend by *y*. This gradient method can be written in the form:

$$\frac{dx}{dt} = -ay, \quad \frac{dy}{dt} = ax, \quad a>0, \quad x(t_0)=x^0, \quad y(t_0)=y^0.$$

Then $xdx+ydy=0$ and $x^2+y^2=r^2$. We see that trajectory of gradient method does not converge to saddle-point.

This example demonstrates that standard optimization methods do not directly extend to saddle-point search problem.

A1.2. Algorithms for Saddle-Points Search

A1.2.1. Theoretical Background and Notions

This section contains notions that are necessary for understanding of non-differentiable convex optimization theory and algorithms.

By (x,y) we denote scalar product of vectors x and y, $L_X(x,y)$ and $L_Y(x,y)$ are partial derivatives of function $L(x,y)$.

Strict convexity of a set X means that $\forall x_1 \in X, \forall x_2 \in X, x_1 \neq x_2, 0 < \lambda < 1$ point $\lambda x_1 + (1-\lambda)x_2$ is an interior point of X.

Strong convex function f with constant $l > 0$ is a function that satisfies the following inequality:

$$f(\lambda x + (1-\lambda)y) \leq \lambda f(x) + (1-\lambda)f(y) - l\lambda(1-\lambda)\frac{\| x-y \|^2}{2}, \quad 0 \leq \lambda \leq 1.$$

For differentiable function f strong convexity is equivalent to the following inequality:

$$f(x+y) \geq f(x) + (\nabla f(x), y) + \frac{l\| y \|^2}{2}, \quad \forall x, y \in \mathbf{R}^n.$$

Some algorithms described below require functions satisfy Lipschitz constraints. The following lemma from [3] states that convex function is a Lipschitz function on any convex bounded set.

Lemma 1. *Let f(x) is a convex function, $G \subset R^n$, G is a convex bounded set. Then there exists $C < \infty$ such that* $|f(x) - f(y)| \leq C\| x-y \|, \quad \forall x, y \in G$.

Various methods for saddle-point computations were analyzed in [12].
The main device for analysis is the following function:

$$\Phi(z,w) = L(u,y) - L(x,v), \quad z=[x,y], w=[u,v]. \tag{A9}$$

Let $z^* = [x^*, y^*] \in \Omega = X \times Y$ then

$$\min_{w \in \Omega} \Phi(z^*, w) = \Phi(z^*, w) \Leftrightarrow z^* \text{ is a saddle point of } L(x,y). \tag{A10}$$

Let $L(x,y) \in C^1$ and

$$h(z) = \Phi_w(z,z) = [L_x(x,y), -L_y(x,y)] \tag{A11}$$

If $z^* \in \Omega$ - saddle-point then

$$\psi(z^*) = \min_{w \in \Omega}(h(z^*), w - z^*) = 0 \tag{A12}$$

otherwise we have descent direction of $\Phi(z^*,w)$ in the point (z^*, z^*) and remain in Ω. But this contradict with (A10).

Condition (A12) is also sufficient for z^* to be a saddle-point.

Analysis of non-smoothed functions based on the subgradient notion:

Definition A3. *Let f(x) - convex function on R^n. Vector $a \in R^n$ is called subgradient of function f(x) in point x if the following inequality satisfied:*

$$f(x+y) \geq f(x)+(a,y), \quad \forall y \in R^n.$$

and denoted by ∂f(x).

See for example [5].

A1.2.2. Conditional Gradient Method

This section describe method of saddle point search for a smooth function on a strict convex set [3].

According to (A12) the following function introduced:

$$\psi(z) = \min_{w \in \Omega}(h(z), w - z) = 0 \tag{A13}$$

and point $\theta(z) \in \Omega$ that gives minimum value in (A13) : $\psi(z)=(h(z),\theta(z)-z)$.

So in conditional gradient method the sequence $\{z_k\}$ built such that $\psi(z_k) \to 0$ and limit of $\{z_k\}$ is saddle-point z^*:

$$\begin{gathered} z_0 \in \Omega, \\ z_{k+1} = z_k^{\alpha_k}, \quad z_k^{\alpha} = z_k + \alpha\zeta_k, \quad \zeta_k = \theta(z_k) - z_k, \\ \alpha_k = arg \min_{\alpha \in [0,1]} \left| \psi(z_k^{\alpha}) \right|. \end{gathered} \tag{A14}$$

Theorem A1[3]. *If $L(x,y) \in C^1(\Omega)$, Ω - strict convex compact, $h(z) \neq 0$ on Ω then sequence (A14) has following properties:*

1. $\psi(z_k) \to 0$ *and limit of $\{z_k\}$ is saddle-point z^*.*
2. *If $\psi'(z)$ is Lipschitz function on Ω then* $\psi(z_k) = O(\frac{1}{k})$.
3. *If $\psi'(z)$ is Lipschitz function on Ω, $L(x,y) \in C^2(\Omega)$ and strong convex-concave: $(L_{xx}(z)u, u) \geq m||u||^2$, $(-L_{yy}(z)v, v) \geq m||v||^2$, $m > 0$ then*

$$\left| \psi(z_k) \right| < \left| \psi(z_0) \right| q^k, q < 1 \text{ and } \left\| z_k - z^* \right\|^2 \leq \frac{2}{m} \left| \psi(z_0) \right| q^k.$$

We see from theorem that conditional gradient method requires the strict convexity of Ω to guarantee convergence and cannot been applied to polyhedral sets. For example, $X=[0;1]^n$ - is not strict convex set. On other hand if our saddle-point search problem satisfy conditions of theorem then we must have "quick" procedure to solve problem (A13).

Let $\pi_{\Omega.}$ - projection operator on Ω:

$$\pi_{\Omega.}(z) \in \Omega, \left\| \pi_{\Omega.}(z) - z \right\| = \min_{w \in \Omega} \left\| w - z \right\|. \tag{A15}$$

The most part of the algorithms described below use projection operators, which demands effective procedures for their calculation. For example: projection on set $l \leq x \leq b$ is trivial task and projection on set $Ax \leq b$ requires a quadratic programming solution.

A1.2.3. Gradient Projection Method

This section describes saddle point search algorithm that use projection operator that also used in our saddle point search algorithm from section 4.

Gradient projection method uses the following result:

$$\pi_{\Omega.}\left(z^* - \alpha h\left(z^*\right)\right) = z^*, \alpha > 0 \Leftrightarrow z^* \text{ is a saddle point of } L(x,y). \quad \text{(A16)}$$

The following iterative sequence used to find a saddle-point:

$$z_{k+1} = \pi_{\Omega}\left(z_k - \alpha z^* h\left(z_k\right)\right), k=0,1,2,\ldots \quad \text{(A17)}$$

where $z_0 \in \Omega$ and α is sufficiently small.

Theorem A2[3]. *If the following is true:*

1. $L(x,y) \in C^1(\Omega)$,
2. $(h(z)-h(w), z-w) \geq m(r)||z-w||^2$, $z,w \in \Omega_r$ *and* $||h(z)-h(w)|| \leq M(r)||z-w||, z,w \in \Omega_r$, *where* $\Omega_r=(z \in \Omega : ||z_0-z|| \leq r)$, $m(r) \geq 0$ *is a non-increasing function,* $M(r)$ *is a non-decreasing function,*
3. $\exists r_0 > 0$ *with* $r_0 m_0(r_0) > ||h(z^0)||$,
4. $0 < \alpha < \sup\limits_{r>0} \dfrac{2}{rM^2(r)}\left(rm(r) - \left\|h(z_0)\right\|\right)$,

then sequence (A17) converges to a saddle point z^* *with linear speed:*

$$|| z_k - z || \leq Cq^k, \; q < 1.$$

A1.2.4 Extensions of Brown-Robinson Method for Matrix Games

This section describes one of the first saddle-point search algorithm that was developed and was reported to have bad performance on practice.

Let

$$\alpha_k \in (0,1], \quad \alpha_k \to 0, \quad \sum_{i=1}^{\infty} \alpha_k = \infty$$

and k-approximation $[x_k,y_k] \in \Omega$ found. Find u_k - minimum of $L(x, y_k)$ on X, find v_k - maximum of $L(x_k, y)$ on Y.

Calculate next approximation:

$$x_{k+1} = x_k + \alpha_k(u_k - x_k), \; y_{k+1} = y_k + \alpha_k(v_k - y_k).$$

Sequence $\{[x_k,y_k]\}$ converges to saddle-point. But it's well known that speed of convergence of Brown-Robinson method and its generalization is very low. For example Brown-Robinson method applied to matrix game has following approximation error for game-value on k-iteration:

$error \leq max\left\|a_{ij}\right\| 2^n k^{-\frac{1}{n-2}}$, (a_{ij}) - game matrix, n-dimension of (x,y).

We can also mark the absence of results of comparative tests for various methods for saddle-point problems.

In last decade considerable progress was made in understanding of saddle-point problems and new algorithms were proposed:

- Controlled continuous gradient processes [7-9],
- Bundle and levels methods [14-16],
- Perturbation methods [17-19],
- Subgradient methods [20], [36] .

A1.2.5. Controlled Saddle Differential Gradient Processes

This method gives continuous interpretation of saddle-point search process.[12] It is more "theoretical" than practical algorithm at first sight. Yet iterative variants of algorithm can be written. Method says that feedback control must be used in saddle-point search process.

We'll see that some kind of feedback is present in other algorithms of saddle-point problem: perturbation, "probe" step, etc. Following notations from [8] we have saddle-point problem:

$$L(x^*, p) \le L(x^*, p^*) \le L(x, p^*), \forall x \in Q \subset R^n, \forall p \in P \subset R^n, \tag{A18}$$

where $L(x,p)$ differentiable convex by x and concave by p and P, Q are closed and convex sets.

Necessary conditions for (x^*,p^*) to be a saddle-point is:

$$\begin{aligned} x^* &= \pi_Q\left(x^* - \alpha \nabla L_x\left(x^*, p^*\right)\right), \\ p^* &= \pi_P\left(p^* + \alpha \nabla L_p\left(x^*, p^*\right)\right), \end{aligned} \tag{A19}$$

where π_Q-projection operator on Q, π_P-projection operator on P,$\nabla L_x(x,p)$, $\nabla L_p(x,p)$ - gradients of $L(x,p)$ by x and p.

So (x^*,p^*) can be considered a fixed point of operator given by right side of (A19).

$$\begin{aligned} x^* &= \arg\min \{L(z,p^*) : z \in Q\} \\ p^* &= \operatorname{argmin}\{-L(x^*,y) : y \in P\} \end{aligned} \tag{A20}$$

Introduce the same function as in (A9):

$$\Phi(v,w) = L(z,p) - L(x,y), w = [z,y], v = [x,p]. \tag{A21}$$

Problem

$$v^* \in \operatorname{argmin}\{\Phi(v^*,w) : w \in \Omega = P \times Q\} \tag{A22}$$

is equivalent to problem (A20). Necessary and sufficient conditions for (A22) is:

$$v^* = \pi_\Omega(v^* - a\nabla\Phi_w(v^*,v^*)) \ . \tag{A23}$$

A1.2.6. Method

[12] For more information about this method see its author's (A. Antipin) personal Web page: http://www.ccas.ru/antipin/.index.html.

Discrepancy or difference between left and right side of (A23) (equal to zero only in point v^*) gives transform $\Omega \to R^{n+m}$.

Direct image of transform can be considered as vector field with stationary point v^*.

Lets state the problem to build trajectory such that linear combination of velocity vector and acceleration vector was equal to direction of vector field. If velocity and acceleration vectors are equal to zero in some point of vector field then trajectory remain in this point. In our case linear combination of velocity vector and acceleration vector is equal to zero in saddle point.

This problem described by system of differential equations:

$$\mu(t)\frac{d^2v}{d^2t}+\frac{dv}{dt}+v=\pi_{\Omega}\left(v-a(t)\nabla\Phi_w(v,v)\right),\ v(t_0)=v^0,\dot{v}(t_0)=\dot{v}^0\ . \qquad \text{(A24)}$$

System of first order with $\mu(t)=0$ was investigated in [9]. Continuous systems of first order also were investigated in [10].

Equation (A24) is composition of the system:

$$\begin{aligned}&\mu(t)\frac{d^2x}{d^2t}+\frac{dx}{dt}+x=\pi_Q\left(x-a(t)\nabla L_x(x,p)\right),\\ &\mu(t)\frac{d^2p}{d^2t}+\frac{dp}{dt}+p=\pi_P\left(p-a(t)\nabla L_p(x,p)\right),\\ &x(t_0)=x^0,\dot{x}(t_0)=\dot{x}^0,\ p(t_0)=p^0,\dot{p}(t_0)=\dot{p}^0\ .\end{aligned} \qquad \text{(A25)}$$

Right side of (A24) or (A25) is non–expansive operator and for $t \in t_0$ and $v^0 \in$ " there is unique trajectory of (A24) that nearly aways does not convergence to v^*. To make trajectory of (A24) to convergence to v^* Antipin proposes to use feedback control. In general case feedback control have the form:

$$u=u(v,\dot{v},\ddot{v}),\ \ where\ \dot{v}=\frac{dv}{dt},\ddot{v}=\frac{d^2v}{dt^2}.$$

In equilibrium points the feedback is equal to zero: $u=u(v,\dot{v},\ddot{v})=0$, where $\dot{v}(\infty)=\dot{v}^*,\ddot{v}(\infty)=\ddot{v}^*$. The choice of proper feedback must provide convergence of trajectory to equilibrium. To gain this aim lets introduce additive control to system (A24):

$$\mu(t)\frac{d^2v}{d^2t}+\frac{dv}{dt}+v=\pi_{\Omega}\left(v-a(t)\nabla\Phi_w(v+u,v)\right),\ v(t_0)=v^0,\dot{v}(t_0)=\dot{v}^0\ . \qquad \text{(A26)}$$

and state the control problem: from some class of controls $u=u(v,\dot{v},\ddot{v})$ choose control that provide convergence of trajectory of (A24) to equilibrium state v^* beginning from arbitrary initial state v^0.

General principles of dynamic control by various forms of feedback were investigated in [11].

Gradient processes controlled by derivative.

Lets control has the following form

$$u=\mu(t)\ddot{v}+\dot{v}. \qquad \text{(A27)}$$

The system (A26) closed by control (A27) takes the form:

$$\mu(t)\frac{d^2v}{d^2t}+\frac{dv}{dt}+v=\pi_{\Omega}\left(v-a(t)\nabla\Phi_w\left(v+\dot{v}+\mu(t)\ddot{v},v\right)\right), v(t_0)=v^0, \dot{v}(t_0)=\dot{v}^0 . \qquad \text{(A28)}$$

This differential system may cause some problem while numerical integration.

For example, iterative version of (A28) has unsolved for v^{n+1} form:

$v^{n+1}=f(v^{n+1},v^n,v^{n-1})$, where f is nonlinear function.

Theorem A3 [8]. *If*

1. *Problem (A18) has nonempty solution set.*
2. *Function Φ(v,w) defined as (A21) satisfies Lipschitz conditions:*

$$\left\|\Phi(v,w+h)-\Phi(v,w)-(\nabla\Phi_w(v,w),h)\right\|\le\frac{1}{2}\|\Phi\|\|h\|^2, \forall w, w+h, w\in\Omega .$$

3. Ω - convex, closed set.
4. *a(t) ∈C[0;∞],* $\mu(t)\in C^2[0;\infty]$,

   a(t)≤a$_0$, $\dot{\mu}(t)\le 0$, $\ddot{\mu}(t)\ge 0$, $t\ge 0$

   $\lim_{t\to 0}\mu(t)=\mu_\infty$, $1-a_0\|\Phi\|-\mu_\infty>0$,

   then

   $\forall v^0, \dot{v}^0$ *trajectory of process (A34) converges to a saddle-point:*

   $v(t)=(x(t),p(t))\to v^*=(x^*,p^*)\in\Omega$, *for* $t\to\infty$.

Remark A2. There are exist functions a(t) and μ(t) that satisfy conditions of theorem:.

$a(t)=a_0\left(1+\frac{1}{t+1}\right)$, $\mu(t)=\mu_\infty+\frac{1}{t+1}$,

a_0 and μ_∞ such that $1-a_0|\Phi|-\mu_\infty>0$.

Remark A3. The first condition of theorem is satisfied when *L(x,p)* is continuous on Ω .

Gradient processes controlled by discrepancy

Gradient processes controlled by derivative has implicit form by derivative. To get explicit differential systems the control by discrepancy introduced:

$$u=\pi_{\Omega}(v-a\nabla\Phi_w(v,v))-v \qquad \text{(A29)}$$

The system (A24) closed by control (A29) takes explicit differential system:

$$\mu(t)\frac{d^2v}{d^2t}+\frac{dv}{dt}+v=\pi_{\Omega}(v-a(t)\nabla\Phi_w(\bar{v},v)), v(t_0)=v^0, \dot{v}(t_0)=\dot{v}^0 , \qquad \text{(A30)}$$

where

$$\bar{v}=\pi_{\Omega}(v\text{-}a\nabla\Phi_{w}(v,v)) \quad . \tag{A31}$$

Theorem A4.[8]. *If*

1. *Problem (A18) has nonempty solution set.*
2. *Function Φ(v,w) defined as (A21) satisfies Lipschitz conditions:*

$$\left\|\Phi(v,w+h)-\Phi(v,w)-(\nabla\Phi_{w}(v,w),h)\right\|\le\frac{1}{2}\|\Phi\|\,\|h\|^{2},\ \forall w,w+h,w\in\Omega.$$

$$\left\|\nabla\Phi_{w}(v+h,w)-\nabla\Phi_{w}(v,w)\right\|\le\|\nabla\Phi\|\,\|h\|,\ \forall v,v+h,w\in\Omega.$$

3. *Ω - convex, closed set.*
4. *$a(t)\in C[0;\infty]$, $\mu(t)\in C^{2}[0;\infty]$,*

$a(t)\le a_0$, $\dot{\mu}(t)\le 0$, $\ddot{\mu}(t)\ge 0$, $t\ge 0$

$$\lim_{t\to 0}\mu(t)=\mu_{\infty},\ \ 1-2\mu_{\infty}-\dot{\mu}(t)>0,\ \ \ 1-a_0\|\Phi\|-2a_0^{2}\|\nabla\Phi\|^{2}>0,$$

then

$\forall v^0,\dot{v}^0$ trajectory of process (A34) converges to a saddle-point:

$$v(t)=(x(t),p(t))\to v^{*}=(x^{*},p^{*})\in\Omega,\ \ \textit{for } t\to\infty.$$

<u>*Gradient processes with mixed control*</u>

The mixed control is control of variable *x* by derivative and control of variable *p* by discrepancy:

$$u_1=\pi_{P}(p+a(t)\nabla L_{P}(x,p))-p \tag{A32}$$

$$u_2=\mu(t)\ddot{x}+\dot{x}$$

The system closed by control (A32) takes the form:

$$\mu(t)\frac{d^{2}x}{d^{2}t}+\frac{dx}{dt}+x=\pi_{Q}(x-a(t)\nabla L_{x}(x,\bar{p})), \tag{A33}$$

$$\mu(t)\frac{d^{2}p}{d^{2}t}+\frac{dp}{dt}+p=\pi_{P}(p-a(t)\nabla L_{p}(x+\dot{x}+\mu(t)\ddot{x},p)), \tag{A34}$$

$$x(t_0)=x^{0},\dot{x}(t_0)=\dot{x}^{0},p(t_0)=p^{0},\dot{p}(t_0)=\dot{p}^{0}. \tag{A35}$$

where

$$\bar{p}=\pi_{P}(p+a(t)\nabla L_{P}(x,p)). \tag{A36}$$

Theorem A5 [8]**.** *If*

1. *Problem (A18) has nonempty solution set.*
2. *Convex-concave function L(x,p) satisfies Lipschitz conditions:*

$$L(x+h,p)-L(x,p)-(\nabla L_{x}(x,p),h)\le\frac{1}{2}L_{1}\|h\|^{2},\ \forall x,x+h\in Q,p\in P,$$

$$L(x,p+h)-L(x,p)-(\nabla L_p(x,p),h)\geq -\frac{1}{2}L_2\|h\|^2,\ \forall p,p+h\in P,x\in Q,$$

$$\|\nabla L_p(x+h,p)-\nabla L_p(x,p)\|\leq\|\nabla L\|\,\|h\|.$$

3. *Q,P - convex, closed sets.*
4. *a(t)* $\in C[0;\infty]$, $\mu(t)\in C^2[0;\infty]$,

a(t)$\leq a_0$, $\quad a_0\leq min\left(\frac{1}{\sqrt{L_2}},\frac{\sqrt{L_1+8\|\nabla L\|^2}-L_1}{4\|\nabla L\|^2}\right)$, $\dot{\mu}(t)\leq 0$, $\ddot{\mu}(t)\geq 0$, $t\geq 0$

$$\lim_{t\to 0}\mu(t)=\mu_\infty,\ 1-2\mu_\infty>0,\ \ 1-a_0L_1-2a_0^2\|\nabla L\|^2-\mu_\infty>0,$$

then

$\forall x^0, p^0, \dot{x}^0, \dot{p}^0$ *trajectory of process (A34) converges to a saddle-point:*

$(x(t),p(t))\to(x^*,p^*)\in Q\times P$, *for* $t\to\infty$.

A1.2.7. Extragradient Method

Extragradient method [13] alters gradient method (that does not work, as we saw from example (A8)) in a following way.

The main idea of algorithm is to make "predict" step in gradient direction of L from current point. Calculate gradient in this "probe" point and use this gradient for step from current point.

Let

1. $X\subset R^n, Y\subset R^m$ closed, convex sets.
2. $L(x,y)$-convex-concave, differentiable function and partial derivatives $L_X(x,y)$, $L_Y(x,y)$ satisfy Lipschitz conditions:

$$\|L_x(x,y)-L_x(x',y')\|\leq C\left(\|x-x'\|^2+\|y-y'\|^2\right)^{\frac{1}{2}},$$

$$\|L_y(x,y)-L_y(x',y')\|\leq C\left(\|x-x'\|^2+\|y-y'\|^2\right)^{\frac{1}{2}}.$$

3. Set of saddle-points $X^*\times Y^*$ is not empty.

Algorithm

Prediction. *Calculate "probe" point*

$$\bar{x}^k=\pi_X\left(x^k-\alpha L_x\left(x^k,y^k\right)\right)$$
$$\bar{y}^k=\pi_Y\left(y^k+\alpha L_y\left(x^k,y^k\right)\right)$$

Step. *Calculate next point*

$$x^{k+1}=\pi_X\left(x^k-\alpha L_x\left(\bar{x}^k,\bar{y}^k\right)\right),$$

$$y^{k+1} = \pi_Y\left(y^k + \alpha L_y\left(\bar{x}^k, \bar{y}^k\right)\right).$$

Theorem A6[13]**.** *If $0<\alpha<1/C$ then sequence $\{(x^k, y^k)\}$ generated by algorithm converges to (x^*, y^*)-saddle-point of $L(x,y)$ on $X\times Y$.*

Remark A4. *Extragradient method applied to Lagrangian of linear programming problem converges with linear speed.*

A1.2.8. Perturbation Method for Saddle-Point Computation

This section describes method for saddle point search [17-19] with non-cooperative game interpretation.

Let $L(x,y)$ finite convex-concave function on $X\times Y$ and $X\cdot R^n$, $Y\cdot R^m$ are closed, convex sets. Authors of the method has great justification in possibility to parallelize their method for saddle-point search of linear-programming Langrangian function (see [17]). In that case they can split entire saddle-point search algorithm on separate optimization processes for each variable and constrain of linear programming problem. Authors of the method described in [17] introduce primal and dual regularizations of function $L(x,y)$ in the following way:

Let us define a non–cooperative game with two players: *P* and *D***.** The objective of *P* is to minimize in the variables $x\in X$ the regularized primal function:

$$P(x,y) = \max_{\mu\in Y}\left(L(x,\mu) - \frac{\rho}{2}\|\mu - y\|^2\right). \quad \text{(A37)}$$

The objective of D is to maximize with respect to the variables $y\in Y$ the regularized dual function:

$$D(x,y) = \min_{\xi\in X}\left(L(\xi,y) - \frac{\rho}{2}\|\xi - x\|^2\right) \quad \text{(A38)}$$

where $\rho>0$ is some parameter.

A Nash equilibrium of the game defined as a point $(x^*,y^*)\in X\times Y$ with:

$$x^* \in \operatorname{argmin}\{P(x,y^*) : x\in X\}, \quad \text{(A39)}$$

$$y^* \in \operatorname{argmax}\{P(x^*,y) : y\in Y\}, \quad \text{(A40)}$$

Define the proximal mappings $\mu(x,y)$ as the solution of (A37) and $\xi(x,y)$ as the solution of the (A38).

Introduce the error functions:

$$E(x,y) = \max_{g\in\partial_x L(\xi,x)}\left(g, x-\xi\right) - \min_{h\in\partial_y L(x,\mu)}\left(h, y-\mu\right),$$

$$\Delta(x,y) = \|\xi - x\|^2 + \|\mu - y\|^2,$$

where $(\partial_x L(x,y), \partial_y L(x,\mu))$ - $\xi=\xi(x,y),\mu=\mu(x,y)$.

Proposition A3. *For all $x\in X$ and $y\in Y$, $\rho\Delta(x,y)\le E(x,y)\le L(x,\mu(x;y))-L(\xi(x,y),y)$.*

Theorem A7*. The following three statements are equivalent:*

1. *(x^*,y^*) is a Nash equilibrium of the game (A37)–(A38);*
2. *$E(x^*,y^*)=0$*
3. *(x^*,y^*) is a saddle-point of $L(x,y)$ on $X\times Y$.*

A1.3. Algorithms for Finding a Saddle-Point

According to theorem we can find Nash equilibrium of the game (A37)–(A38) to find saddle-point of L. Cone of feasible directions at $x\in X$ is denoted by $K_X(x)$ and cone of feasible directions at $y\in Y$ is denoted by $K_Y(y)$.

Initialization. Choose $x^0\in X$, $y^0\in Y$, $\gamma\in(0;2)$. Set $k=0$.

Prediction. Calculate $\mu^k=\mu(x^k,y^k)$ and $\xi^k=\xi(x^k,y^k)$.

Stopping test. If $E_k=E(x^k,y^k)=0$, then stop.

Direction finding. Find subgradients $L_x(x^k,\mu^k)$, $L_y(\xi^k,y^k)$ and calculate

$$d_x^k=\pi_{C_x^k}\left(-L_x(x^k,\mu^k)\right)$$

$$d_y^k=\pi_{C_y^k}\left(L_y(\xi^k,y^k)\right)$$

where C_x^k and C_y^k are closed convex cones and $K_X(x^k)\subset C_x^k$, $K_Y(y^k)\subset C_y^k$.

Stepsize calculation. Calculate

$$\tau_k=\frac{\gamma E_k}{\left\|d^k\right\|^2},$$

where $d^k=\left(d_x^k,d_y^k\right)$.

Step. Calculate next point

$$x^{k+1}=\pi_X\left(x^k-\tau d_x^k\right),$$
$$y^{k+1}=\pi_Y\left(y^k-\tau d_y^k\right),$$
$k=k+1$,

go to Prediction.

Theorem A8. *Let set of saddle-points $X^*\times Y^*$ is not empty. Then algorithm generates a sequence $\left\{\left(x^k,y^k\right)\right\}_{k=0}^{\infty}$ converges to a saddle-point of L on $X\times Y$.*

A1.3.1. Method of Levels for Saddle-Points Search.

This method belongs to a class of bundle methods [16]. These methods use information that was collected on all steps of optimization process unlike many other methods that use information from one or two previous steps only.

Lets describe the saddle-point search method for function defined on direct product of polytopes. Method is tailored for saddle-point search problem level method of minimizing convex function. Lets shortly describe it and show how it lead us to saddle-point search algorithm.

Method has one parameter $\lambda \in (0,1)$ and denoted by Lev_λ.

Given the problem:

$$f(x) \to min, \quad x \in G. \tag{A41}$$

where $G \in R^n$-polytope with diameter D, f-convex and Lipschitz function on G.

Method generates sequence of points x_j, which accumulate information about function f and allows to build approximation of function f with increasing accuracy.

Introduce piecewise-linear functions:

$$f_i(x) = \max_{1 \le j \le i} \left[f(x_j) + (x - x_j)^T \partial f(x_j) \right] \tag{A42}$$

where $x_j \in G$ -generated by algorithm points, $\partial f(x_j)$ - subgradient in x_j.

On step number i method uses all information accumulated on previous steps.

We have following properties:

$f_i(x) \le f(x)$ - low bound property,

$f_{i+1}(x) \ge f_i(x)$ - nondecreasing of low bounds property,

$f_i(x) = f(x)$, $1 \le j \le i$.

Note that problem $\min_{x \in G_i} f_i(x)$ is a linear programming problem:

$$\min t$$

$$f(x_1) + (x - x_1)^T \partial f(x_1) \le t,$$

$$f(x_2) + (x - x_2)^T \partial f(x_2) \le t,$$

$$f(x_i) + (x - x_i)^T \partial f(x_i) \le t, \; x \in G.$$

Denote minimal value of this problem $\underline{f}_i$. And again we have:

$\underline{f}_i \le f^*$, where f^* is optimal value of (A41),

$\underline{f}_{i+1} \ge \underline{f}_i$.

Record value achieved is:

$$\bar{f}_i = \min_{1 \le j \le i} f(x_j),$$

From $\underline{f}_i \le f^* \le \bar{f}_i$ we have non-increasing error estimates:

$$\Delta_i = f_i - \bar{f}_i$$

$$\Delta_{i+1} \le \Delta_i .$$

Then the next point x_{i+1} is calculated as projection of x_i on "level" set:

$$x_{i+1} = argmin\left[\|x - x_i\|^2, x \in G, f_i(x) \le l_i \right],$$

where l_i- is value of "level":

$l_i = f_i + \lambda\Delta_i$, $\lambda \in (0,1)$ - is a parameter of algorithm.

Algorithm stops when $\Delta_i \le \varepsilon$, where $\varepsilon>0$ is solution accuracy for problem (A41).

Record points $\bar{x}_i$ such that $f(\bar{x}_i) = \bar{f}_i$ is approximate solution for problem (A41).

Method of levels for saddle-point search is method of minimizing of function:

$$v(z) = \psi(x) - \varphi(y) \to min, \ z = (x,y) \in G = X \times Y, \qquad \text{(A43)}$$

where $\varphi(y)$and $\psi(x)$ defined as in (A3)-(A4)

$$\varphi(y)=\min_{x \in X} L(x,y)$$
$$\psi(x)=\max_{y \in Y} L(x,y)$$

and X and Y are polytopes.

Optimal value of problem (A43) is zero and reached on saddle-point $\nu(x^*,y^*)=0$.

Accuracy of solution of (A43) in point $(x,y) \in G$ is:

$$v(x,y) = \psi(x) - \varphi(y) = \left[\psi(x) - \min_{x \in X} \psi(x) \right] + \left[\max_{y \in Y} \varphi(y) - \varphi(y) \right].$$

If $\nu(x^*,y^*) \le \varepsilon$ then (x^*,y^*) called ε-solution of saddle-point problem.

Lets we found points $z_j=(x_j,y_j) \in G$, $1 \le j \le i$ and calculated $L(z_j)$ and partial subgradients $\partial L_x(z_j)$ and partial supergradients $\partial L_y(z_j)$

Introduce functions:

$$\psi_i(x) = \max_{1 \le j \le j} \left[L(z_j) + (x - x_j)^T \partial L_x(z_j) \right], \qquad \text{(A44)}$$

$$\varphi_i(y) = \max_{1 \le j \le j} \left[L(z_j) + (y - y_j)^T \partial L_y(z_j) \right]. \qquad \text{(A45)}$$

We have:

$$\psi_i(x) \le \psi(x)$$

$$\varphi_i(y) \ge \varphi(y)$$

Function $\nu_i(x,y) = \psi_i(x) - \varphi_i(y)$ is lower bound for $\nu(x,y)$, but we have not equality $\nu_i(x_i,y_i)=\nu(x_i,y_i)$ as it was in case of convex function minimization. So we'll have point (x_i,y_i) and call it "search point" and "approximation" point" (x_i^*,y_i^*) that will be build from previous search points.

A1.3.2. Algorithm Lev_λ^s for Saddle-Point Search

Initialization. Choose $x_1 \in X$, $y_1 \in Y$, $\varepsilon>0$, $\lambda \in (0,1)$. Set $i=1$.

Iteration

Calculate $L(z_j)$, $\partial L_x(z_j)$, $\partial L_y(z_j)$.
Build models ψ_i, φ_i, ν_i

Solve linear programming problems:

$\min_{x \in X} \psi_i(x)$ and find dual variables $\lambda_j \ge 0$.
$\max_{y \in Y} \varphi_i(y)$ and find dual variables $\mu_j \ge 0$.

Calculate optimal value - Δ_i of ν_i

Calculation of approximation to saddle-point :

$z_i^* = (x_i^*, y_i^*)$

$$x_i^* = \sum_{j=1}^{i} \mu_j x_j$$

$$y_i^* = \sum_{j=1}^{i} \lambda_j y_j$$

Stopping test:

If $\Delta_i \le \varepsilon$ then z_i^* is ε-solution

Calculate next search point:

Solve quadratic programming problem:

$$z_{i+1} = (x_{i+1}, y_{i+1}) = \arg\min\left(\|z - z_i\|, z \in G, \nu_i(z) \le -(1-\lambda)\Delta_i \right),$$

$i=i+1$
goto Iteration.

Stop criterion follows from theorem

Theorem A9.

1. For method Lev_λ^S for all i have $\nu(z_i^*)\le\Delta_i$.

2.$\forall\varepsilon>0$ method finds ε-solution (i.e. $\nu(z_i^*)\le\varepsilon$) on iteration K with estimate: $K \le \left[c(\lambda)\frac{C^2D^2}{\varepsilon^2} \right]$, $c(\lambda) = \frac{2}{(1-\lambda)^2\lambda(2-\lambda)}$, where D-diameter of G, C-Lipschitz constant of function L.

This result proves that saddle point computation is polynomial: diameter D ~ $\sqrt[3]{N}$, where N is dimension of a space of variables.

A1.3.3. Conditional ε–subgradient Method

During computations in optimization algorithms we frequently solve subproblems inexactly (for example, in interior points algorithms). So we would like to have method that takes in account inexact solution of subproblems.

Method described in [20] allows us to build *x*-component of saddle-point $(x^*, y^*)\in X^*\times Y^*$ by means of a conditional ε–subgradient algorithm, while the *y*-component is constructed by means of a weighted average of the subproblem solutions generated within the subgradient method.

The following problem solved to find *x*-component:

$$\min_{x\in X}\psi(x) \tag{A46}$$

where:

$$\psi(x)=\max_{y\in Y}L(x,y) \tag{A47}$$

Lets denote the set of solutions to the problem (A47) by $Y(x)$. An ε–optimal solution, $\tilde{y}$, to the problem (A47) is defined by the following inequality

$$L(x,\tilde{y})\ge\psi(x)-\varepsilon \tag{A48}$$
$$\tilde{y}\in Y,\ \varepsilon\ge 0.$$

Let's introduce some notations and facts, see [4],[20]:

Let $S\subseteq R^n$ nonempty, closed, and convex set.
The normal cone to S is

$$N_S(x) := \begin{cases} \{ z\in R^n : z^T(y-x)\le 0, \forall y\in S \}, & x\in S, \\ \varnothing, & x\notin S. \end{cases}$$

Indicator function on S is

$$I_S(x) := \begin{cases} 0, & x\in S, \\ +\infty, & x\notin S. \end{cases}$$

Subdifferential operator ∂I_X of I_X is equal to N_X:

$$\partial I_X(x) = N_X(x), \forall x \in X .$$

$\gamma_\varepsilon(x)$ is called ε-subgradient of f at x (that is $\gamma_\varepsilon(x)$ is an element of the ε-subdifferential $\partial_\varepsilon f(x)$ of f at x) for $\varepsilon \geq 0$ if and only if

$$f(z) \geq f(x) + \gamma_\varepsilon(x)^T (z - x) - \varepsilon, \forall z \in \mathbf{R}^n \quad \text{(A49)}$$

With ε=0 we get definition of a subgradient(element of the subdifferential).
The following introduce the notion of conditional ε-subgradient, see [20].
For $x \in X$ and $\gamma_\varepsilon(x) \in \partial_\varepsilon f(x)$

$$\gamma_\varepsilon^X(x) := \gamma_\varepsilon(x) + \nu(x), \quad \text{(A50)}$$
$$\nu(x) \in N_X(x). \quad \text{(A51)}$$

$\gamma_\varepsilon^X(x)$ is called as conditional ε-subgradient of f at x.

Equivalent definition produced by substituition $z \in R^n$ with $z \in X$ in (A49):

$$f(z) \geq f(x) + \gamma_\varepsilon^X(x)^T (z - x) - \varepsilon, \forall z \in X .$$

To calculate conditional ε-subgradient according to (50), (51) the following facts will be useful:

Proposition A4 *Let $x \in X$ and $\tilde{y}$ is ε–optimal solution to the problem (A47). Then any subgradient $\tilde{\gamma}(x)$ of $L(\cdot, \tilde{y})$ at x is an ε-subgradient of $\psi(x)$ at x.*

Proposition A5 *Let* $X = \left\{ x \in \mathbf{R}^n \mid (a_i, x) \leq b_i, i = 1,...,m \right\}$.

Then

$$N_X(x) := \begin{cases} \left\{ v \in R^n : v = \sum_{i=1}^m a_i w_i; w_i((a_i, x) - b_i) = 0; w_i \geq 0, i = 1,...,m \right\}, & x \in X, \\ \varnothing, & x \notin X. \end{cases}$$

We will use the following divergent series step length rule in conditional ε-subgradient algorithm:

$$\alpha_t > 0, \forall t, \lim_{t \to \infty} \alpha_t = 0, \sum_{t=0}^{\infty} \alpha_t = \infty, \sum_{t=0}^{\infty} \alpha_t^2 < \infty \quad \text{(A52)}$$

We can use the following step length rule for example:

$$\alpha_t = \frac{1}{a+bt},\ a \geq 0,\ b > 0.$$

At first we present method to solve problem (A46):

$$x^{t+\frac{1}{2}} := x^t - \alpha_t \partial \gamma_\varepsilon^X(x^t),\quad x^{t+1} := \pi_X\left(x^{t+\frac{1}{2}}\right), t = 0,1,... \tag{A53}$$

Where π_X-projection operator on X.

Let X^* denote solution set for problem (A46).

Theorem A10 [20]. *Let $\{x^t\}$ is generated by method (A52)-(A53) applied to (A46). If $\mathbf{R}_+ \ni \{\varepsilon\} \to 0$, the sequence $\{\gamma_\varepsilon^X(x^t)\}$ is bounded, and if $\sum_{t=0}^{\infty} a_t \varepsilon_t < \infty$ then $\{x^t\}$ converges to an element of X^*.*

Remark A5. *If X is bounded then sequence $\{\gamma_\varepsilon^X(x^t)\}$ is bounded.*

Now lets show how the y-component of saddle-point computed within ε-subgradient method (A46-A47):

Let

$$A_t := \sum_{s=0}^{t-1} \alpha_s \tag{A54}$$

$$\hat{y}^t := A_t^{-1} \sum_{s=0}^{t-1} \alpha_s y_{\varepsilon_s}^s \tag{A55}$$

In other words A_t is the accumulated step length up to iteration t and $\hat{y}^t$ the weighted average of the inexact solutions of (A52-A53).

In [20] was shown that any accumulation point $\hat{y}^\infty$ of the sequence $\{\hat{y}^t\}$ together with the solution x^∞ obtained from subgradient scheme forms a saddle-point of L.

Let's denote distance from the point x to its projection on S:

$$dist(x,S) = \min_{y \in S} \|y - x\|.$$

Theorem A11. $L(x^\infty, \hat{y}^\infty) \geq L(x^\infty, y), \forall y \in Y.\ \{dist(\hat{y}^t, Y(x^\infty))\} \to 0.$

So for $(x^\infty, \hat{y}^\infty)$ we have left-most inequlity from definition of saddle-point.

Theorem A12. $L(x, \hat{y}^\infty) \geq L(x^\infty, \hat{y}^\infty), \forall x \in X.$

So we have right-most inequality for $(x^\infty, \hat{y}^\infty)$ from definition of saddle-point.

And finally we have

Theorem A13. $(x^\infty, \hat{y}^\infty)$ *solves (A46),* $dist\{(x^t, \hat{y}^t), \{x^\infty\} \times Y^*\} \to 0.$

In [20] is not described stop criterion for conditional ε-subgradient method and this could be considered as a flaw of the method. We can propose to use stop criterion that arises from the estimate the duality gap:

If $\psi(x^t) - \varphi(\hat{y}^t) < \Delta$ *then STOP.*

This requires solving the problem:

$$\varphi(\hat{y}^t)=\min_{x\in X} L(x,\hat{y}^t) \quad .$$

To reduce the computation work we can propose to evaluate the stop criterion value not on each step of method. In [16] described other variants of conditional ε-subgradient algorithm for minimization of convex function. But convergence of $\hat{y}^t$ to *y*-component of saddle-point for these variants is not proved.

The similar results about constructing *y*-component of saddle-point exist for "exact" subgradient method; see for example [37]. In [37] also suggested stop criterion of the saddle-point algorithm arises in problem of "topology optimization of sheets in contact".

A1.3.4. Interior-Point Method for a Special Class of Saddle-Point Problems

In [21] described polynomial method for solving special class of saddle-point problem. Method use results of Nemirovski on self-concordant convex-concave functions [22].

Let S^n denotes the space of $n\times n$ symmetric matrices and S^n_+ denotes the space of $n\times n$ symmetric positive semidefinite matrices.

Lets consider the following problem:

$$\max_{(c,Q)} \min_{x} \left(c^T x + \frac{1}{2} x^T Q x \right) \tag{A56}$$

$$Ax \geq b, \tag{A57}$$

$$c^L \leq c \leq c^U, \quad Q^L \leq Q \leq Q^U. \tag{A58}$$

where:

$$A \in \mathbf{R}^{m+n},$$
$$b \in \mathbf{R}^m,$$
$$c, c^L, c^U \in R^n, Q^L, Q^U \in S^n, Q \in S^n_+ \quad .$$

Denote

$$X := \{x \mid Ax \geq b\},$$
$$Y := \{(c,Q) \mid c^L \leq c \leq c^U, Q^L \leq Q \leq Q^U, Q^L, Q^U \in S^n, Q \in S^n_+\}$$
$$y := (c,Q)$$
$$L(x,y) := c^T x + \frac{1}{2} x^T Q x$$

We can rewrite (A56) as:

$$\max_{y\in Y} \min_{x\in X} L(x,y) \tag{A59}$$

Function *L* is quadratic by *x* and linear by *y*.
X and *Y* are convex, closed sets and *Y* is bounded.

Assume that X is bounded. This assumption meets in most real life applications. This can be achieved by adding box constrains with "Big M":-$M \le x_i \le M$. Solve (A59) by finding the saddle-point of L:

$$L(x^*,y) \le L(x^*,y^*) \le L(x^*,y), \quad \forall x \in X, \forall y \in Y \tag{A60}$$

We have convex-concave function $L(x,y)$ defined on $X \times Y$ and saddle-point (x^*,y^*) of $L(x,y)$ with equality:

$$\max_{y \in Y} \min_{x \in X} L(x,y) = \min_{x \in X} \max_{y \in Y} L(x,y) = L(x^*,y^*) \tag{A61}$$

As usual we introduce functions (see (A3) -(A4))

$$\varphi(y) = \min_{x \in X} L(x,y)$$

$$\psi(x) = \max_{y \in Y} L(x,y)$$

The value of duality gap will serve a measure of proximity to a saddle-point:

$$v(x,y) = \psi(x) - \varphi(y) = \left[\psi(x) - \min_{x \in X} \psi(x)\right] + \left[\max_{y \in Y} \varphi(y) - \varphi(y)\right]. \tag{A62}$$

Let X^0 and Y^0 denote the interiors of the sets X and Y, which we assume to be nonempty. Consider the following barriers functions for the sets X and Y:

$$F(x) = -\sum_{i=1}^{m} log\left(\left|Ax-b\right|_i\right), \forall x \in X^0 ,$$

$$G(y) = -\sum_{j=1}^{n} log\left(c_j^U - c_j\right) - \sum_{j=1}^{n} log\left(c_j - c_j^L\right) - \sum_{1 \le i \le j \le n} log\left(Q_{ij}^U - c_j\right) -$$

$$- \sum_{1 \le i \le j \le n} log\left(Q_{ij} - Q_{ij}^L\right) - log\ det(Q),$$

$$\forall y \in Y^0.$$

$F(x)$ is a self-concordant barrier for X with parameter m and $G(y)$ is a self-concordant barrier for Y with parameter n^2+4n.

For $t \ge 0$ consider the following saddle-barrier function:

$$L_t(x,y) := tL(x,y) + F(x) - G(y). \tag{A63}$$

$L_t(x,y)$ -is a strictly convex-concave function.

Lemma A1. *For each $t\geq 0$ there exists a unique saddle-point (x_t,y_t) of the function $L_t(x,y)$ in $X^0\times Y^0$.*

The set of saddle-points for for different values of t defines the *central path* for saddle-point problem (A60):

$$C := \left\{ \left(x_t, y_t\right) : t \geq 0, \left(x_t, y_t\right) \textit{is a saddle point of } L_t\left(x, y\right) \right\}. \tag{A64}$$

The central path is the main tool of *path-following* algorithms, i.e., algorithms that try to reach a solution by generating iterates around the central path for progressively larger values of t.

Lemma A2. *For a point $(x_t,y_t) \in C$ the following inequality for the measure of proximity to saddle-point (A61) holds:*

$$v\left(x_t, y_t\right) \leq \frac{\left(n^2 + 4n + m\right)}{t}.$$

This lemma is motivation for developing an algorithm that follows the central path to solve the saddle-point problem. For points on the central path, the measure of proximity converges to zero as the parameter t is increased.

It is often very hard to find points that are exactly on the central path.

For $(x,y)\in X\times Y$ and well defined Δ-measure of proximity to the central path (see [21] for details) the following inequality holds:

$$v\left(x, y\right) \leq \frac{\left(n^2 + 4n + m\right)}{t}\left(1 + \frac{\Delta}{\sqrt{n^2 + 4n + m}}\right).$$

Given $(\hat{x},\hat{y})\in X\times Y$ we define the following restriction functions:

$$L^t_{\hat{y}}\left(x\right) = tL\left(x, \hat{y}\right) + F\left(x\right), \forall x \in X^0, \tag{A65}$$

$$L^t_{\hat{x}}\left(y\right) = tL\left(\hat{x}, y\right) - G\left(y\right), \forall y \in Y^0, \tag{A66}$$

$L^t_{\hat{y}}\left(x\right)$ and $\left(-L^t_{\hat{x}}\left(y\right)\right)$ are (strongly) self-concordant functions in their domains.

The magnitude of the progress made by a Newton step for minimizing a given convex function $f(z)$ can serve as a measure of proximity to the minimizer of the function.

Given a strictly convex function $f(.)$ and a vector z in its domain, consider the following function:

$$\eta\left(f, z\right) := \sqrt{\nabla f^T(z)\left[\nabla^2 f\left(z\right)\right]^{-1}\nabla f\left(z\right)}. \tag{A67}$$

The quantity in the square root on the right-hand-side of equation above is the quadratic Taylor series approximation to the decrease of the function f by taking a *full* Newton step from point z. This quantity is called *Newton decrement*. In [22] Nemirovski considers a generalization of the Newton decrement for convex-concave functions. In our case it is:

$$\eta\left(L_t, x, y\right) := \sqrt{\eta^2\left(L^t_{\hat{y}}, x\right) + \eta^2\left(-L^t_{\hat{x}}, y\right)} \tag{A68}$$

where $t>0$ and $(x,,y)\in X^0\times Y^0$.

We will use $\eta(L_t,x,y)$ as a measure of proximity to the central path *C*. This measure vanishes only on the central path.If we are close to the central path with respect to measure of proximity $\eta(L_t,x,y)$ then we are close to a saddle-point according to the following.lemma:

Lemma A.3 *If* $(\bar{x},\bar{y}) \in X \times Y$ *satisfies* $\eta(L_t,x,y) \le \beta$ *with* $\beta \le \frac{1}{2}$, *then*

$$v(\bar{x},\bar{y}) = \psi(\bar{x}) - \varphi(\bar{y}) \le \left(1 + \frac{6\beta}{\sqrt{n^2+4n+m}}\right)\frac{(n^2+4n+m)}{t}.$$

Now we can describe the short step algorithm to find a saddle-point of the problem (A60).

A1.3.5. An Interior-Point Algorithm

This algorithm can be viewed as a specialization of the short-step algorithm proposed in [22]. The method in [38] updates the central path parameter *t* according to the formula $t_+ = \left(1 + \frac{\delta}{\sqrt{\theta}}\right)t$ where θ is the parameter of the barrier function for the domain of the problem (n^2+4n+m in our case) and $\delta \le 0.001$ in our case. Then, the method of [38] uses a single Newton step to find the new iterate satisfying proximity bound.

In [38], Nemirovski develops an alternate method that can replace δ above with a larger constant such as 1, but requires an inner iteration procedure -the so-called *saddle Newton* method, which may take several, but a bounded number of steps to generate the next iterate. Our algorithm improve the constant δ to at least 0.1 from 0.01 but still use a single Newton step between parameter updates.

The saddle-point algorithm

Step 1**. Initialization:**

Choose α *and* β *that satisfy the relationship*

$$\gamma := \frac{13}{10}\left((1+\alpha)\beta + \left(\alpha\sqrt{n^2+4n+m}\right)\right) < 1,$$

$$\gamma^2\frac{1+\gamma}{1-\gamma} \le \frac{13}{10}\beta$$

Find $t_0 > 0$ *and* $(x_0,y_0) \in X^0 \times Y^0$ *that satisfies* $\eta(L_{t_0},x_0,y_0) \le \beta$.

Set k=0.

Step 2. **Iteration:**

While $\left(t_k < \frac{1}{\varepsilon}\left(1 + \frac{6\beta}{\sqrt{n^2+4n+m}}\right)(n^2+4n+m)\right)$ do

$$Set\ t_{k+1} = (1+\alpha)t_k,$$

Take a full step of Newton method for function (A62)

$$(x_{k+1}, y_{k+1}) = (x_k, y_k) - \left[\nabla^2 L_{t_{k+1}}(x_k, y_k)\right]^{-1} L_{t_{k+1}}(x_k, y_k)$$

Set k=k+1

End while.

Remark A6. $\alpha = \dfrac{0.1}{\sqrt{n^2 + 4n + m}}$ *and* β*=0.1 can be chosen to satisfy the condition in the initialization step.*

Polynomial complexity for the algorithm given by following theorem:

Theorem A14. *The saddle-point algorithm finds a feasible point (x,y) with* $v(x,y) \le \varepsilon$ *in* $O\left(ln\frac{1}{\varepsilon}\sqrt{n^2 + 4n + m}\right)$ *iterations.*

Algorithm requires finding $t_0 > 0$ and $(x_0, y_0) \in X^0 \times Y^0$ that satisfies $\eta\left(L_{t_0}, x_0, y_0\right) \le \beta$. This can be done by approximately solving the analytic center problems over *X* and *Y*, in $O\left(ln\frac{1}{\varepsilon}\sqrt{n^2 + 4n + m}\right)$ time. This will give an approximation; say (x_0, y_0) to the saddle-point of the pure barrier function $L_0(x,y)$. Then, t_0 can be chosen as the largest *t* satisfying $\eta\left(L_{t_0}, x_0, y_0\right) \le \beta$.

Appendix 2. SP-SVM and MOP RSVM

As we already mentioned a very close setting to the feature selection problem was done by Bi [31] for multi-objective program (MOP). Let us briefly describe this work.

Associating each feature x_j with a scaling factor performs the feature selection s_j: $\mathbf{x} = (x_1, \ldots, x_M) \to \left(x_1\sqrt{s_1}, \ldots, x_M\sqrt{s_M}\right)$.

The master multi-objective programming (MMOP) formulation of the problem is following:

$$\begin{aligned}
&\min && \sum_{i=1}^{N} \xi_i, \\
&\min && RW + c\sum_{i=1}^{M} s_i \\
&s.t. && y_i\left(\left(\sum_{j=1}^{M} s_j w^j x_i^j\right) + b\right) \ge 1 - \xi_i, \\
& && \xi_i \ge 0, \quad i = 1, \ldots, N, \\
& && \mathbf{w} \cdot \mathbf{w} \le W, \\
& && \sum_{j=1}^{M} s_j x_i^j x_i^j \le R, \quad i = 1, \ldots, N.
\end{aligned} \tag{A69}$$

This is a very hard to find a Pareto-optimal solution of this problem. The RSVM setting of the problem is following:

$$\begin{aligned} \min \quad & \sum_{i=1}^{M} \xi_i \\ s.t. \quad & \mathbf{w} \cdot \mathbf{w} \le W, \\ & y_i \left(\left(\sum_{j=1}^{M} s_j w^j x_i^j \right) + b \right) \ge 1 - \xi_i, \\ & \xi_i \ge 0, \quad i = 1,...,N. \end{aligned} \tag{A70}$$

Here W is no longer a variable, but a user-specified parameter.

In [31] proposed the following two-step procedure of feature selection.

The first step focuses on improving the performance of the classification model by minimizing the empirical risk with a fixed VC dimension. By optimizing on s, the second step seeks a feature space, for which a smaller VC dimension can be possibly achieved with the empirical risk preserved.

The optimization algorithm for feature selection is following:

1. Initialize s^0 with (1,…,1), W^0 with appropriate value.
2. Solve the dual formulation of RSVM with the fixed s^{t-1} and W^{t-1},

$$\begin{aligned} \min_{\lambda} \quad & \sqrt{W^{t-1} \sum_{i,j=1}^{N} \lambda_i \lambda_j y_i y_j \sum_{l=1}^{M} s_l^{t-1} x_i^l x_j^l} - \sum_{i=1}^{N} \lambda_i \\ & \sum_{i=1}^{N} \lambda_i y_i = 0, \\ & 0 \le \lambda_i \le 1, \ \mathrm{i} = 1,..., \mathrm{N}. \end{aligned} \tag{A71}$$

and compute the optimal $b^t, w_l^t = \sum_{i=1}^{N} \lambda_i^t y_i \sqrt{s_l} x_i^l$, $l=1,...,M$, where λ^t is constructed by dividing the optimal solution $\tilde{\lambda}$ of (A72) by

$$\gamma = \frac{\sum_{i,j=1}^{N} \tilde{\lambda}_i \tilde{\lambda}_j y_i y_j \sum_{l=1}^{M} s_l^{t-1} x_i^l x_j^l}{\sqrt{W^{t-1}}}$$

(see [2, 30]). Calculate the corresponding optimal objective value E^t of Primal (A70).

3. Substitute the w^t and b^t into the MMOP, and restrict the first objective to be no more than E^t. Solve the resulting optimization problem:

$$\min_{s,R,W,\xi} \quad RW + c\sum_{i=1}^{M} s_i^t$$

$$s.t. \quad y_i\left(\left(\sum_{j=1}^{M} s_j w^j x_i^j\right) + b\right) \geq 1 - \xi_i,$$

$$\xi_i \geq 0, \quad i = 1,...,N,$$

$$\sum_{i=1}^{N} \xi_i \leq E^t,$$

$$\sum_{i,j=1}^{N} \lambda_i^t \lambda_j^t y_i y_j \sum_{l=1}^{M} s_l^t x_i^l x_j^l \leq W,$$

$$\sum_{j=1}^{M} s_j^t x_i^j x_i^j \leq R, \quad i = 1,...,N.$$

to obtain s^t and W^t.

4. Determine if more iterations are needed, for instance, if either E^t or $H^t=R^tW^t$ is decreased, set $t=t+1$, and go to Step 2; otherwise, stop.

This scheme does not guarantee to achieve a Pareto-optimal solution. In [31] proved only that both E^t and H^t not increasing from iteration to iteration.

The comparison of this work with our present one says that from one side the settings of [30, 31] give more possibilities to control VC dimension because it does not include a hardly estimated parameter C. From other side our approach is strict and has an exact polynomial algorithm for its solution.

Acknowledgements.

Ilya Muchnik thanks to NSF for partial support of this work with grant No 0325398.

Thanks to Thorsten Joachims for letting us use his program and to make small changes in it for our needs.

References.

1. V. N. Vapnik, "The Nature of Statistical Learning Theory" Springer-Verlag, New York, 1995.
2. V. N. Vapnik, "Statistical Learning Theory" New York, John Wiley and sons, 1998.
3. V.F.Dem'yanov, V.N Malozemov, “Introduction to minimax” , Moscow (in Russian), 1972.
4. V.F.Dem'yanov, L.V Vasil'ev, “Nondierentiable Optimization”, Optimization Software, New York, NY, 1985.
5. B.T.Polyak, “Introduction to optimization”, Moscow (in Russian), 1983.
6. R.T. Rockafellar, “Convex Analysis”, Princeton University Press, Princeton, 1970

7. A.S.Antipin, Controlled Proximal Differential Systems for Saddle Problems. *Differentsial'nye Uravneniya*. 1992. T.28. No.11. pp.1846-1861. English transl.: Differential Equations. 1992. Vol.28. No.11. pp.1498-1510.
8. A.S.Antipin, E.S.Hamraeva, Controlled saddle differential gradient methods of second order, 1996, Computing Center, Russian Academy of Sciences, Moscow (in Russian)
9. A.S.Antipin, Feedback-Controlled Saddle Gradient Processes, 1994,Avtomatika i telemehanika, 3,p.12-23 (in Russian), English transl: *Automation and Remote Control*. 1994. Vol.55. No.3. pp. 311-320.
10. V.I.Venetsz, Continuous algorithm of saddle point search for convex-concave function, 1984, Avtomatika i telemehanika, 1, pp. 42-47 (in Russian)
11. S.V.Emelyanov, S.K.Korovin, Principles of building and base properties of closed dynamic systems with various types of feedbacks. 1992, Russian academy of sciences, Institute for Systems Analysis.(in Russian)
12. V.F.Dem'yanov, A.B.Pevnui, Numerical methods for search of saddle points , 1972, Jurnal vuichislitel noi matematiki i matematicheskoi fiziki,T12,5.(in Russian)
13. G.M. Korpelevich, The extragradient method for finding saddle points and other problems, Ekonomika i Matematicheskie Metody 12, 1976, 747-756.(in Russian)
14. E.G.Gol'shtein, Generalized saddle variant of levels method, Ekonomika i matematicheskie metodui, T.37, 3, 2001. (in Russian)
15. E.G.Gol'shtein, Nemirovskii A., Nesterov Yu., Levels method, its generalizaions, and applications, Ekonomika i matematicheskie metodui, T.31,3, 1995. (in Russian)
16. Lemarechal C.,Nemirovskii A., Nesterov Yu, New variants of bundle methods, Math.Program.Ser.B. 1995. V.69. 1.P.111-147.
17. Kallio M., Ruszczynski A. Perturbation methods for saddle point computations, Working Paper WP-94-38, IIASA, Laxenburg, 1994.
18. Ruszczynski A. A Partial Regularization Method For Saddle Point Seeking, Working Paper WP–94–20, IIASA, Laxenburg, 1994.
19. Kallio M., Ruszczynski A. Parallel solution of linear programs via Nash equilibria, Working Paper WP–94–15, IIASA, Laxenburg, 1994.
20. Larsson, T., Patriksson, M., and Stromberg, A.-B., On the Convergence of Conditional epsilon-Subgradient Methods for Convex Programs and Convex-Concave Saddle-Point Problems, *European Journal of Operational Research* 151,461-473, 2003.
21. Halldoorsson V., Tutuncu H. An interior-point method for a class of saddle-point problems, 2002
22. Nemirovski, A .,On Self-Concordant Convex-Concave Functions, Optimization Methods and Software, Vol. 11/12, p. 303-384, 1999.
23. L. Molina, L. Belanche and A. Nebot, Feature Selection Algorithms: A Survey and Experimental Evaluation IEEE Internat. Conf. on Data Mining, Japan
24. Somol, P., Pudil, P., Feature selection toolbox, Pattern Recognition, vol. 35, No. 12: 2749-2759, 2002.
25. Sebban, M., Nock, R., A hybrid filter/wrapper approach of feature selection using information theory, Pattern Recognition, vol. 35, No. 4: 835-846, 2002.

26. Klimesova, D., Saic, S., Feature Selection Algorithm and Cobweb Correlation, Pattern Recognition Letter, vol. 19, No. 8: 681-685, 1998.
27. Shapira, Y., Gath, I.., Feature selection for multiple binary classification problems, Pattern Recognition Letter, vol. 20, No. 8: 823-832, 1999.
28. Kudo, M., Sklansky, J., Comparison of algorithms that select features for pattern classifiers, Pattern Recognition, vol. 33, No. 1: 25-41, 2000.
29. Chen, X.W. An improved branch and bound algorithm for feature selection, Pattern Recognition Letter, vol. 24, No. 12: 1925-1933, 2003.
30. J. Bi and V.Vapnik, Learning with Rigorous Support Vector Machines. COLT 2003: Conference on Learning Theory, Washington D.C. August 24-27 2003.
31. J. Bi, Multi-Objective Programming in SVMs. *Proceedings of the Twentieth International Conference on Machine Learning (ICML-2003)*, Washington DC 2003.
32. J. Weston, S, Mukherjee, O. Chapelle, M. Pontil, T. Poggio, V. Vapnik, Feature Selection for SVMs. In Advances in Neural Information Proccesing Systems, No. 13, MIT Press 668-674, 2000.
33. J. Weston et. al., Feature Selection and Transduction for Prediction of Molecular Bioactivity for Drug Design. Bioinformatics Vol. 1, No. 1 2002: 1-8
34. J. Bi et al., Dimensionality Reduction via Sparse SVMs. Journal of Machine Learning Research, No. 1:1-48, 2002.
35. Draper, N. and H. Smith., Applied Regression Analysis. Wiley, New York. 1966.
36. Larsson T., Patriksson M., Stromberg A., Conditional subgradient optimization-theory and applications, European Journal of Operations Research, 151, 461-473, 2003.
37. Petersson J., Patriksson M. Topology optimization of sheets in contact by a subgradient method, International Journal of Numerical Methods in Engineering, 40, p.1295-1321, 1997.
38. Nesterov, Yu., Nemirovski, A , Interior-Point Polynomial Algorithms in Convex Programming, SIAM, Philadelphia, Pennsylvania, 1994.
39. Joachims, T., Optimizing Search Engines using Clickthrough Data**,** Proceedings of the ACM Conference on Knowledge Discovery and Data Mining (KDD), 2002.
40. Joachims, T., Estimating the Generalization Performance of an SVM Efficiently**,** Proceedings of ICML-00, 17th International Conference on Machine Learning, 1999.
41. Andrei V. Anghelescu and Ilya B. Muchnik, Optimization of SVN in a Space of Two Parameters: Weak Margin and Intercept. DIMACS Working Group on Monitoring Message Streams, May 2003.
42. Nash J.F., Two-person cooperative games, Econometrica, 21, 1953, p.128-140.
43. Antipin A., Gradient Approach of Computing Fixed Points of Equilibrium Problems, Journal of Global Optimization 24, pp. 285-309, 2002
44. Antipin A., Differential Equations for Equilibrium Problems with Coupled Constraints, Nonlinear Analysis 47, pp. 1833-1844, 2001
45. C.J.C. Burges A Tutorial on Support Vector Machines for Pattern Recognition, Data Mining and Knowledge Discovery, No. 2:121-167, 1998