INFORMATICA, 2009, Vol. 20, No. 2, 173–186

# Stochastic Optimization Algorithms for Support Vector Machines Classification

Vaida BARTKUTĖ-NORKŪNIENĖ

*Institute of Mathematics and Informatics and Utena College*
*Akademijos 4, LT-08663 Vilnius, Lithuania*
*e-mail: vaidaba@ukolegija.lt*

Received: September 2008; accepted: January 2009

**Abstract.** In this paper, we consider the problem of semi-supervised binary classification by Support Vector Machines (SVM). This problem is explored as an unconstrained and non-smooth optimization task when part of the available data is unlabelled. We apply non-smooth optimization techniques to classification where the objective function considered is non-convex and non-differentiable and so difficult to minimize. We explore and compare the properties of Simulated Annealing and of Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms (SPSA with the Lipschitz Perturbation Operator, SPSA with the Uniform Perturbation Operator, Standard Finite Difference Approximation) for semi-supervised SVM classification. Numerical results are given, obtained by running the proposed methods on several standard test problems drawn from the binary classification literature. The performance of the classifiers were evaluated by analyzing Receiver Operating Characteristics (ROC).

**Keywords:** support vector machine, semi-supervised classification, stochastic approximation, simulated annealing.

## 1. Introduction

Multidimensional data constantly appear in engineering, medicine, economics, ecology, and many other areas. Modern technologies in many scientific fields allow us to produce and store large data sets with ever increasing sample sizes and dimensions. The increasing amount of data used for analysis naturally leads to the search for efficient approaches. The common data mining technique for finding hidden patterns in data is classification analysis. There are a lot of classification problems in webpage classification, bioinformatics, image classification, business risk identification, credit scoring where data labelling is hard or expensive, while unlabeled data is often abundant and cheap to collect. If classification results can be directly seen in a two-dimensional space, it will be great help for classification users. A simple and direct idea for visualizing high-dimensional data is to reduce its dimensionality to two or three dimensions by using some dimension reduction algorithms (Bernatavičiene *et al.*, 2006, 2007). When data points consist of exactly two sets: one set that has been labelled by a decision maker and the other that is not classified, but belongs to one known category we have a traditional semi-supervised classification problem. The goal of semi-supervised classification is to use unlabeled data to improve the performance of standard supervised learning algorithms. Because semi-supervised classification requires less human effort and generally achieves higher accuracy this kind of classification is very important both in theory and in practice. Support Vector Machines (SVMs) are an efficient approach for semi-supervised classification as well as for clustering, time series, and regression analysis. In the standard binary classification problem, a set of training data $(u^i, y^i), \dots, (u^m, y^m)$ is observed, where the input set of points is $u^i \in U \subset \Re^n$, the $y^i$ is either $+1$ or $-1$, indicating the class to which the point $u^i$ belongs, $y^i \in \{+1, -1\}$. The learning task is to create the classification rule $f$: $U \to \{+1, -1\}$ that will be used to predict the labels for new inputs. The basic idea of Support Vector Machine classification is to find a maximal margin separating hyperplane between two classes. It was described by Cortes and Vapnik (1995). The standard binary SVM classification problem is shown visually in Fig. 1.

We denote by $\|\cdot\|$ the Euclidean norm in $\Re^n$ and by $\langle w, u\rangle$ the inner product of the vectors $w$ and $u$ throughout the paper. For a linearly separable case, the support vector algorithm simply looks for the separating hyperplane with the largest margin. The distance between two hyperplanes $H_1$ and $H_2$ is called a margin and equal to $\frac{2}{\|w\|}$, where $w$ is the normal vector of a separating hyperplane. Therefore the goal of classification is to maximize the margin width $\frac{2}{\|w\|}$ which is equivalent to minimizing $\frac{\|w\|^2}{2}$. Now we can formulate our problem as a standard quadratic programming problem (Cristianini and Shawe-Taylor, 1987). It was described by Cortes and Vapnik (1995):

$$\min_{w,b} \frac{1}{2}\|w\|^2, \tag{1}$$

subject to

$$y^i\big(\langle w, u^i\rangle + b\big) \geqslant 1, \quad i = 1, \dots, m.$$

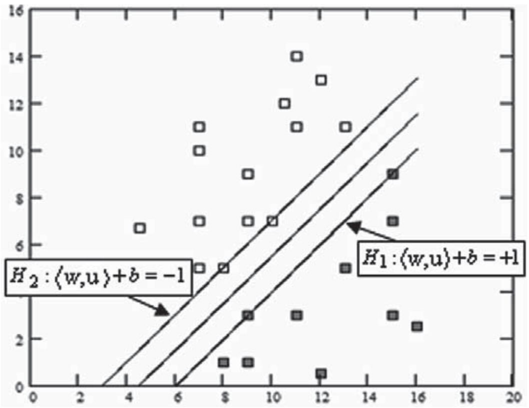

Fig. 1. Separating hyperplanes for a linearly separable case.

Since in many applications to label data is not an easy job we may rewrite problem (1) in the following unconstrained form using penalty function (Astorino and Fuduli, 2007):

$$\min_{w\in\Re^n, b\in\Re} f(w,b), \tag{2}$$

where

$$\begin{aligned} f(w,b) &= \frac{1}{2}\|w\|^2 + C_1 \sum_{i=1}^{p} \max\bigl(0, 1 - y^i(\langle w, u^i\rangle + b)\bigr) \\ &\quad + C_2 \sum_{i=p+1}^{m+p} \max\bigl(0, 1 - |\langle w, u^i\rangle + b|\bigr), \end{aligned}$$

where $C_1 \geqslant C_2 \geqslant 0$ are certain penalty coefficients, $p$ is the size of training set, $m$ is the size of testing set. The first two terms in the objective function $f(w,b)$ define the standard SVM. The third term incorporates unlabelled data. The loss over labelled and unlabelled examples is weighted by two parameters $C_1$ and $C_2$. This form seems advantageous especially when the input dataset is very large.

On the other hand, the function $f(w,b)$ is non-differentiable and, moreover, due to the third term involving the unlabelled points, it is even non-convex and multiextremal in general. Since the objective function of the unconstrained SVM model is a non-smooth function, most of powerful methods of smooth optimization cannot be used to solve it. Astorino and Fuduli (2007) applied a bundle type optimization method for semi-supervised classification problems. For the unsupervised and supervised data classification Bagirov *et al.* (2002, 2003) described the combination of discrete gradient methods with the cutting angle method for global minimization. In this paper, we implement and compare several stochastic optimization algorithms for minimizing the non-differentiable objective function (2). Stochastic Approximation and Simulated Annealing are well-known approaches for non-smooth optimization. Although the Stochastic Approximation is created as a local optimization approach, it also distinguishes itself by some global search properties that are very useful for semi-supervised SVM classification (Maryak and Chin, 2001; Chen, 2002). To this end we explore and compare properties of Simulated Annealing (SA) and of the Simultaneous Perturbation Stochastic Approximation (SPSA) algorithms (SPSA with the Lipschitz Perturbation Operator (SPSAL), SPSA with the Uniform Perturbation Operator (SPSAU), Standard Finite Difference Approximation (FDSA)).

We judge on the applicability of these algorithms to solve problem (2) by the training error and analysis of Receiver Operating Characteristics (ROC). The training error is just the measured error rate on the training data and expressed as follows:

$$R_{emp}(b) = \frac{1}{2p}\sum_{i=1}^{p} \bigl|y^i - h(w,b)\bigr|, \tag{3}$$

where $h(w,b) = \langle w, u\rangle + b$. The "loss" is the term $\frac{1}{2}|y^i - h(w,b)|$.

The evaluation is based on the following criteria:

$$\text{sensitivity} = \frac{TP}{TP+FN}, \quad \text{specificity} = \frac{TN}{TN+FP},$$

$$\text{overall accuracy} = \frac{TP+TN}{TP+TN+FP+FN},$$

$$\text{test error} = \frac{FP+FN}{TP+TN+FP+FN},$$

where $TP$ (true positive) means number of $+1$ class assigned to $+1$ class; $FP$ (false positive) stands for the number of $-1$ class assigned to $+1$ class; $TN$ (true negative) – the number of $-1$ class assigned to $-1$ class; $FN$ (false negative) means the number of $+1$ class assigned to $-1$ class.

## 2. Stochastic Techniques for Non-Smooth Optimization

The methods of stochastic approximation based on the ideas of smoothing and the stochastic gradient. The solution of (2) is obtained by minimizing the smoothed function, and changing the smoothing parameter in an appropriate way (Robins and Monro, 1951; Kiefer and Wolfowitz, 1952; Blum, 1954; Dvoretzky, 1956; Yudin, 1965; Wasan, 1969; Ermoliev, 1976; Mikhalevitch *et al.*, 1987; Ermoliev *et al.*, 1995; Sakalauskas, 2002; Kushner and Yin, 2003; Bartkute and Sakalauskas, 2007; etc.).

Let us introduce the stochastic approximation sequence:

$$x^{t+1} = x^t - \rho_t \cdot g^t, \quad t = 1, 2, \dots, \tag{4}$$

where $g^t$ is the value of the stochastic gradient estimator at the current point $x^t, \rho_t$ is a scalar multiplier in the iteration $t$, and $x^0$ is the initial point. This scheme is the same for different stochastic approximation algorithms that differ only by the approach to stochastic gradient estimation.

To solve problem (2), we consider and compare three stochastic approximation methods:

*SPSAL* – SPSA algorithm with a Lipschitz perturbation operator and the stochastic gradient is as follows (Bartkutė and Sakalauskas, 2006):

$$g(x, \sigma, \xi) = \frac{(f(x+\sigma\xi) - f(x))\xi}{\sigma\|\xi\|}, \tag{5}$$

where $\xi$ is a vector uniformly distributed in the unit ball.

*SPSAU* – SPSA algorithm with a uniform perturbation operator and the stochastic gradient is expressed as follows (Mikhalevitch *et al.*, 1987):

$$g(x, \sigma, \xi) = \frac{(f(x+\sigma\cdot\xi) - f(x-\sigma\cdot\xi))\xi}{2\sigma}, \tag{6}$$

where $\xi$ is a vector uniformly distributed in the hypercube $[-1;1]^n$.

*FDSA* – Finite Difference Stochastic Approximation algorithm and the stochastic gradient is a vector with the components (Mikhalevitch *et al.*, 1987):

$$g_i(x,\sigma,\xi,\upsilon) = \frac{f(x+\sigma\cdot\xi+\upsilon\cdot\varepsilon_i) - f(x+\sigma\cdot\xi-\upsilon\cdot\varepsilon_i)}{2\upsilon}, \tag{7}$$

where $\xi$ is the same as in (5), $\varepsilon_t = (0,0,0,\ldots,1,\ldots,0),\ t = \overline{1,n}$, is the vector with zero components except the $i$th one, which is equal to 1, $\upsilon > 0$ and $\sigma > 0$ are the values of the finite difference and perturbation parameters, respectively.

The regulation conditions of step length and the perturbation operator that guarantee the convergence of the stochastic approximation algorithm $\sum_{t=1}^{\infty}\rho_t = \infty$, $\sum_{t=1}^{\infty}\rho_t^2 < \infty$, $\sigma_t \to 0$, $\frac{|\sigma_t-\sigma_{t-1}|}{\rho_t} \to 0$, $\frac{\rho_t}{\sigma_t} \to 0$ have been determined. The rate of convergence of stochastic approximation for objective functions with sharp minimum $E\|x^t - x*\|^2 = \mathrm{O}(t^{-\beta})$, where $1 \leqslant \beta < 2$, have been proved by theoretical and experimental ways (see, e.g., Bartkute and Sakalauskas, 2007). Although the stochastic approximation converges locally, it stands out by the property of global convergence (Maryak and Chin, 2001; Chen, 2002). The probability to hit to the attraction zone of global minimum can be improved by algorithm parameter regulation. SPSA algorithm with the Metropolis rule was applied in the optimization of the automated recognition system of ischemic stroke in computed tomography images (Grigaitis *et al.*, 2007).

Simulated Annealing (SA) is widely applied in multiextremal problems. Conditions of global convergence of SA are studied by many authors (Granville *et al.*, 1994; Yang, 2000; etc.). We use the modification of SA, developed by Yang (2000), where the function regulating the neighbourhood depth of solution is introduced together with the temperature regulation function. The latter property enables us to improve the rate of convergence (Felinskas and Sakalauskas, 2003). The procedure of the SA algorithm consists of the following steps. Let $D \subset \Re^n$ be a bounded set and an initial point $x^0 \in D$, an initial temperature value $T_0 > 0$, a kind of temperature-dependent generation probability density function, a corresponding temperature updating function, and a sequence $\{\rho_t, t \geqslant 0\}$ of monotonically decreasing positive numbers, describing neighbouring states. Calculate $f(x^0)$. In the current iteration $t$ we generate a random vector $z^t$ by using the generation probability density function. The points $z_i^t$are rejected if $q^t \in D$ or there exists $i$ such that $\|z_i^t\| < \rho_t,\ 1 \leqslant i \leqslant n$, where $z_i^t$ is the $i$th component of the vector $z^t$. Otherwise, generate a new trial point $q^t$ by adding the random vector $z^t$ to the current iteration point $x^t$,

$$q^t = x^t + z^t. \tag{8}$$

The Metropolis acceptance criterion is used to determine a new iteration point $x^{t+1}$ (Metropolis *et al.*, 1953). Specifically, generate a random number $\kappa$ with a uniform distribution over [0,1], and then calculate the probability $P(q^t, x^t, T_t)$ of accepting the trial point $q^t$ as the new iteration point $x^{t+1}$, given $x^t$ and $T_t$,

$$P(q^t, x^t, T_t) = \min\Big\{1, \exp\Big(\frac{f(x^t) - f(q^t)}{T_t}\Big)\Big\}. \tag{9}$$

If $\kappa \leqslant P(q^t, x^t, T_t)$, set $x^{t+1} = q^t$ and $f(x^{t+1}) = f(q^t)$; otherwise, set $x^{t+1} = x^t$ and $f(x^{t+1}) = f(x^t)$. The optimization procedure is terminated, if the prescribed termination condition is satisfied.

Thus, by applying the generation mechanism and the Metropolis acceptance criterion, the SA algorithm produces two sequences of random points. These are the sequence $\{q^t, t \geqslant 0\}$ of trial points generated by (8), and the sequence $\{x^t, t \geqslant 0\}$ of iteration points determined by applying the Metropolis acceptance criterion. These two sequences of random variables depend all on the temperature sequence $\{T_t, t \geqslant 0\}$ determined by the temperature updating function, the state neighbouring sequence $\{\rho_t, t \geqslant 0\}$, and the approach of random vector generation.

The sequence $\{\rho_t, t \geqslant 0\}$ of positive numbers is used to impose a lower bound on the random vector, generated at each iteration, for obtaining the random trial point. This lower bound should be small enough and monotonically decreasing as the annealing proceeds. Since the temperature-dependent generation probability density function is used to generate random trial points and since only one trial point is generated at each temperature value, the SA algorithm considered is characterized by a non-homogeneous continuous-state Markov chain. The convergence conditions of SA were studied by Yang (2000) who suggested several updating functions of method parameters, which ensure the convergence of the method.

We apply the next updating functions in testing our approach. Let $\rho_t = \rho_0 \cdot t^{-\frac{\lambda}{u \cdot n}}$ for all $t \geqslant 1$ be the sequence used to impose lower bounds on the random vectors generated in the SA algorithm, where $u > 1,\ 0 < \lambda < u,\ 0 < \rho_0 < \min_{1 \leqslant i \leqslant n} \max_{x,q \in D} |x_i - q_i|$. Let the temperature-dependent generation probability density function $p(\cdot\,, T_t)$ be given by

$$p(z, T_t) = \prod_{i=1}^{n} \frac{(a-1)}{2T_t} \Big( \frac{|z_i|}{T_t} + 1 \Big) \left( \log \Big( \frac{|z_i|}{T_t} + 1 \Big) \right)^d, \quad z \in \Re^n, \ \ d > 1. \tag{10}$$

Then, for any initial point $x^0 \in D$, the sequence $\{f(x^t); t \geqslant 0\}$ of objective function values converges in probability to the global minimum $f^*$, if the temperature sequence $\{T_t, t \geqslant 0\}$ determined by the temperature updating function satisfies the following condition:

$$T_t = T_0 \exp\big( - l \cdot t^{\frac{1}{d \cdot n}} \big), \quad i = 1, 2, \ldots\,, \tag{11}$$

where $T_0 > 0$ is the initial temperature value and $l > 0$ is the given real number (Yang, 2000). Typically a different form of the temperature updating function has to be used with respect to a different kind of the generation probability density function in order to ensure the global convergence of the corresponding SA algorithm. Furthermore, a flatter tail of the generation probability function implies a faster decrement of the temperature sequence determined by the temperature updating function.

Table 1

Datasets used in the experiments

| No. | Dataset | Dimension ($n$) | Points |
|---|---|---|---|
| 1 | Linear example (Ziv and Motore, 2004) | 2 | 30 |
| 2 | Iris (Asuncion and Newman, 2007) | 4 | 150 |
| 3 | High dimensional case (Zhang *et al.*, 2004) | 20 | 200 |
| 4 | Ionosphere (Asuncion and Newman, 2007) | 34 | 351 |

## 3. Experimental Results

The empirical evidence of our approach, using synthetic and real datasets, and the achieved results are provided and discussed in this section.

To study the applicability of the stochastic optimization algorithm (SPSAL, SPSAU, FDSA, SA) to solve problem (2), several standard examples drawn from the classification literature were explored. Table 1 summarizes the characteristics of the datasets used.

Each test function was minimized $M = 100$ times by stochastic optimization algorithms described above. Penalty coefficients $C_1$ and $C_2$ in function (2) are equal to 2.0 and 0.5, respectively, if not said otherwise. The coefficients of sequence (4) were chosen according to the convergence conditions (Bartkute and Sakalauskas, 2007): $\rho_t = n\min(a; \frac{b}{t}),\ \sigma_t = \sqrt{\frac{(n+2)(n+3)}{n(n+1)}}\min(c; \frac{d}{t^{\beta}}),\ \beta = 0.75$, where $a, b, c, d$ are different for various stochastic approximation algorithms.

EXAMPLE 1 (linear example).

Datasets:

Table 2

Training set

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | 7 | 7 | 11 | 13 | 8 | 9 | 15 | 7 | 15 | 13 | 14 | 9 | 11 | 15 | 10 |
| $u_2$ | 5 | 11 | 11 | 11 | 10 | 9 | 9 | 7 | 7 | 5 | 4 | 3 | 3 | 3 | 7 |
| y | 1 | 1 | 1 | 1 | 1 | 1 | − 1 | 1 | − 1 | − 1 | − 1 | − 1 | − 1 | − 1 | 1 |

Table 3

Testing set

| | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $u_1$ | 4.5 | 8 | 7 | 9 | 9 | 16 | 6 | 12 | 10.5 | 12 | 12 | 11 | 1.5 | 6 | 8 |
| $u_2$ | 6.7 | 5 | 10 | 7 | 1 | 2.5 | 7 | 0.5 | 12 | 13 | 4 | 14 | 0.5 | 7 | 1 |

The linear separating hyperplanes of training data (Example 1) are demonstrated in Fig. 2. Fig. 3 illustrates that the SPSAL classifier for training and testing datasets is

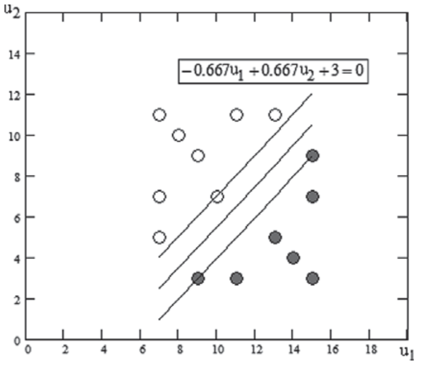

Fig. 2. Linear separating hyperplanes of training data.

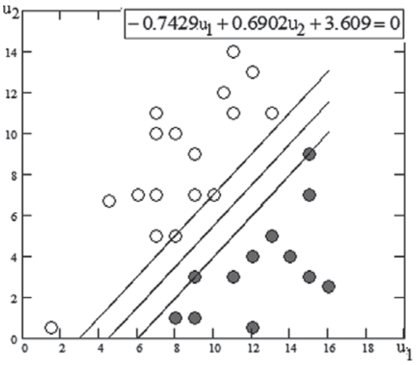

Fig. 3. Linear separating hyperplanes of the training and testing data (SPSAL).

close to an optimal decision boundary. Corresponding averaged separating hyperplanes for other algorithms are similar: for SPSAU is $-0.8861u_1 + 0.9469u_2 + 3.6577 = 0$, for FDSA is $-0.7458u_1 + 0.6873u_2 + 3.6499 = 0$, and for SA is $-0.67094u_1 + 0.6701u_2 + 3.0231 = 0$.

EXAMPLE 2 (high dimensional case). The dataset consists of 200 vectors. The covariate vectors $x$ are 20-dimensional and generated uniformly from a unit cube $[0, 1]^{20}$.

The boundary between two classes is a linear function of the first three variables only: $f(u) = 2u_1 + 4u_2 + 4u_3 - 4.8$. The boundary between two classes is a linear function of the first three variables only: $\{u_1, u_2, u_3\}$ and the remaining seventeen variables are redundant (Zhang *et al.*, 2004).

Fig. 4 depicts how the averaged training error rate changes for each algorithm as the training sample size $p$ is increasing. For all stochastic optimization algorithms their training error decreases significantly.

Fig. 5 illustrates dependence of test error rate on penalty parameter $C_2$ changing the size of training set. Dependencies are similar for other algorithms. These results show

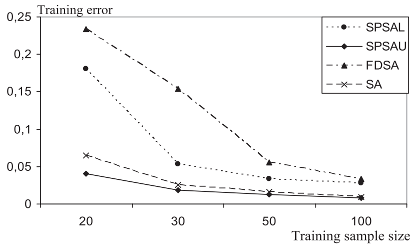

Fig. 4. The averaged training error rate as the training sample size $p$ is increasing.

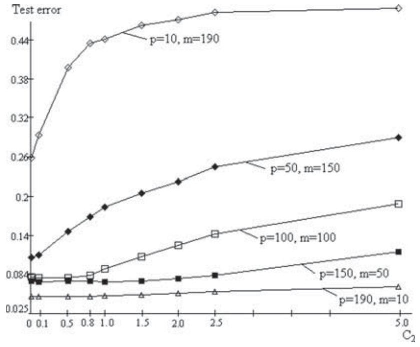

Fig. 5. Overall accuracy rate against the weight parameter $C_2$ (weight parameter $C_1 = 2.0$, stochastic optimization algorithm is SPSAU).

importance of the penalty parameters $C_1$ and $C_2$, and confirm that $C_2$ is recommended to be chosen not large (Astorino and Fuduli, 2007).

EXAMPLE 3 (Database of Iris Plants; Asuncion and Newman, 2007). The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other two, the latter are not linearly separable from each other. In our approach for the binary classification we use only two classes of iris plant: iris Setosa (the class $+1$) and iris Virginica (the class $-1$).

The data calculated for sensitivity, specificity, and overall accuracy rates of the classifiers are summarized in Tables 4 and 5. When the size of the training set is equal to 20, the SA algorithm has the highest sensitivity, specificity, and overall accuracy rates of 86.0%, 100.0%, and 99.3%, respectively. When the size of the training set was increased to 70, the SA, SPSAU, FDSA algorithms acquired 100% sensitivity, specificity, and overall accuracy rates.

Linear separating hyperplanes for two-dimensional Iris Plant data are illustrated in Fig. 6. These results illustrate the applicability of stochastic algorithms in the optimization of non-differentiable objective function (2).

EXAMPLE 4 (Johns Hopkins University Ionosphere database; Asuncion and Newman, 2007). The Johns Hopkins University Ionosphere database consists of radar data collected by a system in Goose Bay, Labrador. This system consists of a phased array of 16 highfrequency antennas with the total transmitted power on the order of 6.4 kilowatts. The targets were free electrons in the ionosphere. "Good" radar returns are those showing

Table 4

Average correct classification rates ($p = 20$, $m = 80$)

| | SPSAL | | SPSAU | | FDSA | | SA | |
|---|---|---|---|---|---|---|---|---|
| | Setosa | Virginica | Setosa | Virginica | Setosa | Virginica | Setosa | Virginica |
| Setosa | 11.5 | 0.0 | 32.52 | 0.01 | 50.0 | 0.0 | 50.0 | 0.0 |
| Virginica | 38.5 | 50.0 | 17.48 | 49.99 | 0.0 | 50.0 | 0.0 | 50.0 |
| Overall accuracy | 61.50% | | 82.51% | | 100% | | 100% | |

Table 5

Average correct classification rates ($p = 70$, $m = 30$)

| | SPSAL | | SPSAU | | FDSA | | SA | |
|---|---|---|---|---|---|---|---|---|
| | Setosa | Virginica | Setosa | Virginica | Setosa | Virginica | Setosa | Virginica |
| Setosa | 50.0 | 0.0 | 49.99 | 0.05 | 50.0 | 0.0 | 50.0 | 0.0 |
| Virginica | 0.0 | 50.0 | 0.01 | 49.95 | 0.0 | 50.0 | 0.0 | 50.0 |
| Overall accuracy | 100% | | 99.94% | | 100% | | 100% | |

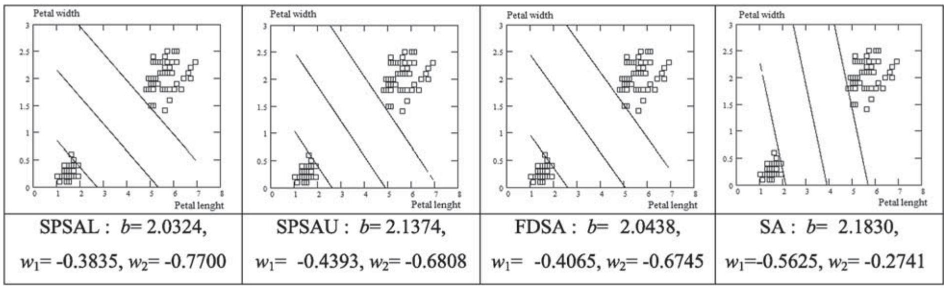

Fig. 6. Linear separating hyperplanes for two-dimensional Iris Plant data.

evidence of some type of structure in the ionosphere. “Bad” returns are those that do not; their signals pass through the ionosphere. The data contain 351 observation, 34 attributes and one class attribute. All the 34 attributes are continuous and the target attribute is either “good” or “bad” according to the definition summarized above (see, Asuncion and Newman, 2007).

The calculated sensitivity, specificity, and overall accuracy rates of the classifiers are summarized in Tables 6 and 7. The SA algorithm has the highest sensitivity, specificity, and overall accuracy rates. Yao and Yu (2007) for this dataset proposed a new model of the support vector machines, which is called oblique support vector machines (OSVMs). To apply stochastic approximation algorithms for Oblique Support Vector Machines and compare achieved results might be one of the subjects of future research.

Table 6

Average correct classification rates ($p = 50$, $m = 301$)

| | SPSAL | | SPSAU | | FDSA | | SA | |
|---|---|---|---|---|---|---|---|---|
| | Good | Bad | Good | Bad | Good | Bad | Good | Bad |
| Good | 91.45 | 48.14 | 96.27 | 40.17 | 83.31 | 44.35 | 89.81 | 30.43 |
| Bad | 34.55 | 176.86 | 29.73 | 184.83 | 42.69 | 180.65 | 36.19 | 194.57 |
| Overall accuracy | 76.44% | | 80.09% | | 75.20% | | 81.02% | |

Table 7

Average correct classification rates ($p = 150$, $m = 201$)

| | SPSAL | | SPSAU | | FDSA | | SA | |
|---|---|---|---|---|---|---|---|---|
| | Good | Bad | Good | Bad | Good | Bad | Good | Bad |
| Good | 110.2 | 27.44 | 106.62 | 12.6 | 109.55 | 30.08 | 110.28 | 12.68 |
| Bad | 15.8 | 197.56 | 19.38 | 212.4 | 16.45 | 194.92 | 15.72 | 212.32 |
| Overall accuracy | 87.68% | | 90.89% | | 86.74% | | 91.91% | |

## 4. Conclusions and Future Work

In this paper, the problem of semi-supervised binary classification by Support Vector Machines (SVM) has been considered when a part of the available data is unlabelled. The applicability of stochastic optimization algorithms in such problems has been studied by computer simulation. We explore three stochastic approximation algorithms (SPSAL, SPSAU, FDSA) and the Simulated Annealing (SA) algorithm with the regulation of solution neighbourhood depth. The simulation studies with several synthetic and real-life datasets show that these algorithms can be successfully applied to optimizing non-differentiable loss functions in the classification problems. On the base of the simulation results a conclusion can be drawn that SA has higher sensitivity, specificity, and overall accuracy rates as compared with the stochastic approximation algorithms. The main advantage of the methods proposed is the possibility to train the classifier on the basis of a large number of labelled and unlabelled points. This could be useful especially in many real-life classification problems. It may be concluded that the classification error depends on labelled and unlabelled data weighting in the objective function (2). The choice of an appropriate interval for penalty coefficients might be the subject of future research.

There are many cases in the real-life problems when data in the input space is not linearly separable and non-linear classifiers are needed. Kernel methods use an approach which increases the flexibility of linear functions by applying a nonlinear mapping from the input space into a higher dimensional vector space called feature space. To compare the efficiency of stochastic approximation algorithms described with other nonsmooth optimization algorithms computing also the CPU time and to use stochastic approximation methods for solving non-linearly classification problems would be a future research direction, too.

## References

Astorino, A., and A. Fuduli (2007). Nonsmooth optimization techniques for semisupervised classification. *IEEE Trans. Pattern Anal. Mach. Intell*, **29**(12), 2135–2142.

Asuncion, A., and D.J. Newman (2007). *UCI Machine Learning Repository*. School of Information and Computer Science, University of California, Irvine, CA.
`http://www.ics.uci.edu/~mlearn/MLRepository.html`

Bagirov, A.M., A.M. Rubinov and J. Yearwood (2002). Global optimization approach to classification. *Optimization and Engineering*, **3**, 129–155.

Bagirov, A.M., A.M. Rubinov, N.V. Soukhoroukova and J. Yearwood (2003). Unsupervised and supervised data classification via nonsmooth and global optimization. *TOP*: *Spanish Operations Research Journal*, **11**(1), 1–93.

Bartkute, V., and L. Sakalauskas (2006). Application of stochastic approximation in technical design. In *Computer Aided Methods in Optimal Design and Operations*, *Series on Computers and Operations Research*, Vol. 7. pp. 29–38.

Bartkute, V., and L. Sakalauskas (2007). Simultaneous perturbation stochastic approximation for nonsmooth functions. *European Journal on Operational Research*, **181**(3), 1174–1188.

Bernataviciene, J., G. Dzemyda, O. Kurasova, V. Marcinkevicius (2006). Optimal decisions in combining the SOM with nonlinear projection methods. *European Journal of Operation Research*, **173**, 729–745.

Bernataviciene, J., G. Dzemyda, V. Marcinkevicius (2007). Conditions for optimal efficiency of relative MDS. *Informatica*, **18**(2), 187–202.

Blum, J. (1954). Multidimensional stochastic approximation procedures. *Annals of Mathematical Statistics*, **25**(4).

Chen, H.-F. (2002). *Stochastic Approximation and its Application*. Series: Nonconvex Optimization and Its Applications, Vol. 64.

Cortes, C., and V. Vapnik (1995). Support-vector networks. *Machine Learning*, **20**(3), 273–297.

Cristianini, N., and J. Shawe-Taylor (2003). Support vector and kernel methods. In M. Berthold and D.J. Hand (Eds.), *Intelligent Data Analysis*: *An Introduction*. Springer-Verlag. pp. 169–197.

Dvoretzky, A. (1956). On stochastic approximation. In J. Neumann (Ed.), *Proc. 3rd Berkeley Symposium of Mathematical Statistics and Probability*, Vol. 1. University of California Press, Berkeley. pp. 39–55.

Ermoliev, Yu. M. (1976). *Methods of Stochastic Programming*. Nauka, Moscow (in Russian).

Ermoliev, Yu.M., V.I. Norkin and R.J.-B. Wets (1995). The minimization of semicontinuous functions: mollifier subgradients. *Control and Optimization*, **33**(1), 149–167.

Felinskas, G., and L. Sakalauskas (2003). Pareto type models in simulated annealing algorithms. *Lith. Math. J.*, **43** (spec. issue), 573–578 (in Lithuanian).

Grigaitis, D., V. Bartkutė, L. Sakalauskas (2007). An optimization of system for automatic recognition of ischemic stroke areas in computed tomography images. *Informatica*, **18**(4), 603–614.

Granville, V., M. Krivanek and J.P. Rasson (1994). Simulated annealing: a proof of convergence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **16**(6), 652–656.

Kiefer, J., and J. Wolfowitz (1952). A stochastic estimation of the maximum of a regression function. *Annals of Mathematical Statistics*, **23**(3), 462–466.

Kushner, H.J., and G.G. Yin (2003). *Stochastic Approximation and Recursive Algorithms and Applications*. Springer, New York/Heidelberg/Berlin.

Maryak, J.L., and D.C. Chin (2001). Global random optimization by simultaneous perturbation stochastic approximation. In *Proceedings of the 33nd Conference on Winter Simulation*, Arlington, Virginia. pp. 307–312.

Metropolis, N., A. Rosenbluth, M. Rosenbluth, A. Teller and E. Teller (1953). Equation of state calculations by fast computing machines, *J. Chem. Phys.*, **21**(6), 1087–1092.

Mikhalevitch, V.S., A.M. Gupal and V.I. Norkin (1987). *Methods of Nonconvex Optimization*. Nauka, Moscow (in Russian).

Robins, H., and S. Monro (1951). A stochastic approximation method. *Annals of Mathematical Statistics*, **22**(3), 400–407.

Sakalauskas, L. (2000). Nonlinear stochastic optimization by Monte-Carlo method. *Informatica*, **11**(4), 455–468.

Yao, C.-C., and P.-T. Yu (2007). Oblique support vector machines. *Informatica*, **18**(1), 137–157.

Yang, R.L. (2000). Convergence of the simulated annealing algorithm for continuous global optimization. *JOTA*, **104**(3), 691–716.

Yudin, D.B. (1965). Qualitative methods for analysis of complex systems. *Izv. AN SSSR*.

Zhang, H.H., J. Ahn, X. Lin and C. Park (2004). Variable selection for SVM via shrinkage methods. In *Discussion Session*, *SAMSI*: *Data Mining and Machine Learning Workshop*, January 2004. Research Triangle Park, NC.

Ziv, B.-J., and A.W. Motore (2004). Machine learning. In *School of Computer Science*, Carnegie Mellon University. pp. 10–781.

Wasan, M.T. (1969). Stochastic approximation. In *Transactions in Mathematics and Mathematical Physics*. Cambridge University Press, Cambridge.

**V. Bartkutė-Norkūnienė** is a researcher in the Department of Operational Reasearch, Institute of Mathematics and Informatics and an associated professor in Utena College. V. Bartkute-Norkuniene received the PhD degree from this Institute and Vytautas Magnus University (2007), First degree (1997) and Master of Science (1999) in applied mathematics from Kaunas University of Technology. Researcher in International Project "Critical thinking development in higher education", supported by Open Society Institute, New York, International Reading Association. Research visits to International Centre of Theoretical Physics (ICTP; Italy, 2004) and High Performance Computing center CINECA (Bologna, Italy, 2007, 2008, 2009). Member of the European Working Group on Continuous Optimization since 2004. Author of about 18 scientific articles. Research interests: continuous optimization, extreme value theory, order statistics, Monte-Carlo method, optimal design.

## Stochastinio optimizavimo metodai atraminių vektorių klasifikavimo algoritmuose

Vaida BARTKUTĖ-NORKŪNIENĖ

Straipsnyje pasiūlyti keturi stochastinio optimizavimo metodai (SPSAL, SPSAU, FDSA ir SA) binarinio klasifikavimo uždaviniams spręsti naudojant atraminių vektorių klasifikatorių (Support Vector Machines). Tokiuose uždaviniuose kvadratinio programavimo uždavinys yra suvedamas į nediferencijuojamo optimizavimo uždavinį be ribojimų taikant nediferencijuojamas baudos funkcijas. Norint įsitikinti šių metodų tinkamumu straipsnyje aptariamai problemai spręsti nagrinėjami keturi klasifikavimo uždaviniai.