Liet. mat. rink. LMD darbai, **48/49**, 2008, 299–305

Stochastic approximation algorithms for support vector machines semi-supervised binary classification

Vaida BARTKUTĖ-NORKŪNIENĖ (MII)*
e-mail: vaidaba@ukolegija.lt

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 problems where the objective function considered is non-convex and non-differentiable and so difficult to minimize. We explore and compare the properties of Stochastic Approximation algorithms (Simultaneous Perturbation Stochastic Approximation (SPSA) with the Lipschitz Perturbation Operator, SPSA with the Uniform Perturbation Operator, and Standard Finite Difference Approximation) for semi-supervised SVM classification. We present some numerical results obtained by running the proposed methods on several standard test problems drawn from the binary classification literature.

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

1. Introduction

Support Vector Machines (SVMs) are well-known data mining methods for classification, regression and time serries analysis problems. In the standard binary classification problem, a set of training data $(u^i, y^i), \ldots, (u^m, y^m)$ is analysed, 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 $f\colon U \to \{+1, -1\}$ that will be used to predict the labels for new inputs. The main idea of SVM classification is to find a maximal margin separating hyperplane between classes [4]. The standard binary SVM classification problem is shown visually in Fig. 1.

$\langle w, u\rangle$ is the scalar product of two vectors. 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 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 [4, 5]:

*The research is partialy supported by The Lithuanian State Science and Studies Foundation project System for interbank settlement simulation, modelling and optimization (No T-08072)

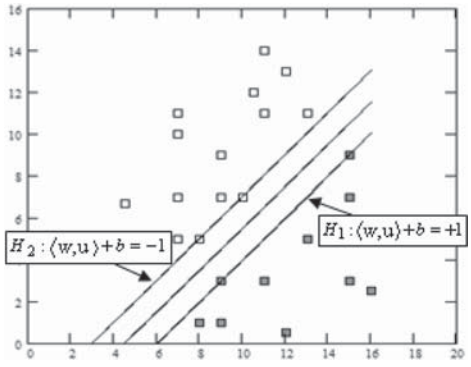

Fig. 1. Linear separating hyperplanes for the separable case.

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

subject to

$$y^i \cdot \left((w^T \cdot u^i) + b\right) \geqslant 1, \quad i = 1, \ldots, m.$$

There are a lot of classification problems where data labelling is expensive and difficult, or labelling is often unreliable. When data points consist of two sets exactly: 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. To solve that we may rewrite problem (1) in the following unconstrained form [1]:

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

where

$$f(w,b) = \frac{1}{2}\|w\|^2 + C_1 \cdot \sum_{i=1}^{p} \max\left(0, 1 - y^i \cdot (w^T \cdot u^i + b)\right)$$

$$+ C_2 \cdot \sum_{i=p+1}^{m+p} \max\left(0, 1 - |w^T \cdot u^i + b|\right),$$

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, and the third one incorporates unlabelled (testing) data. The error 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. 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. In [1] authors applied a bundle type optimization method for semi-supervised classification problems. In this paper, we implement and compare three SA algorithms: Simultaneous Perturbation Stochastic Approximation (*SPSA*) algorithm with Lipschitz Perturbation Operator (*SPSAL*), SPSA with Uniform Perturbation Operator (*SPSAU*), and Standard Finite Difference Approximation (*SFDA*) algorithm.

2. Stochastic approximation techniques

Application of Stochastic Approximation (SA) algorithms to non-differentiable functions is of particular theoretical and practical interest in classification. The methods of SA use the ideas of smoothing and stochastic gradient. Thus, the solution of (2) is provided by minimizing smoothed function, while the smoothing parameter is changed in an appropriate way [3].

Let us introduce an SA sequence:

$$x^{t+1} = x^t - \rho_t \cdot g^t, \quad t = 1, 2, \ldots, \tag{3}$$

where g^t is the value of the stochastic gradient estimator at the current point x^i, ρ_t is a scalar multiplier in iteration t, and x^0 is the initial point. This scheme is the same for various 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:

1) SPSAL – SPSA algorithm with a Lipschitz perturbation operator, the stochastic gradient is as follows [2]:

$$g(x, \sigma, \xi) = \frac{(f(x + \sigma\xi) - f(x)) \cdot \xi}{\sigma \cdot \|\xi\|}, \tag{4}$$

where ξ is a vector uniformly distributed in the unit ball.

2) SPSAU – SPSA algorithm with a uniform perturbation operator, the stochastic gradient is expressed as follows [6]:

$$g(x, \sigma, \xi) = \frac{(f(x + \sigma \cdot \xi) - f(x - \sigma \cdot \xi)) \cdot \xi}{2\sigma}, \tag{5}$$

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

3) FDSA – Finite Difference Stochastic Approximation algorithm and the stochastic gradient is a vector with components [6]:

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

where ξ is the same as in (4), $\varepsilon_t = (0, 0, 0, \ldots, 1, \ldots, 0)$, $t = \overline{1, n}$, is the vector with zero components except the ith one, which is equal to 1, $\upsilon > 0$ and $\sigma > 0$ are the values of finite difference and perturbation parameters, respectively.

The regulation conditions of step length and the perturbation operator that guarantee the convergence of the SA algorithm $\sum_{t=1}^{\infty}\rho_t = \infty$, $\sum_{t=1}^{\infty}\rho_t^2 < \infty$, $\sigma_t \to 0$, $\frac{|\sigma_t-\sigma_{t-1}|}{\rho_t} \to 0$, $\frac{\rho_t}{\sigma_t} \to 0$ are determined and the rate of convergence of SA $E\|x^t - x^*\|^2 = \mathrm{O}(t^{-\beta})$ for $1 \leqslant \beta < 2$ is proved [3].

3. Experimental results

To study the applicability of SA algorithm (SPSAL, SPSAU, FDSA) to solve problem (2) several standard examples drawn from the binary classification literature where chosen. Each test function was minimized $M = 100$ times by SA algorithms described above. Since the error over labelled and unlabelled examples is weighted by two parameters C_1 and C_2 and $C_1 \geqslant C_2 \geqslant 0$, penalty coefficients C_1 and C_2 in function (2) are chosen equal to 2.0 and 0.5, respectively. The coefficients of sequence (4) were chosen according to the convergence conditions [3]: $\rho_t = n \cdot \min(a; \frac{b}{t})$, $\sigma_t = \sqrt{\frac{(n+2)\cdot(n+3)}{n\cdot(n+1)}}\min(c; \frac{d}{t^{\beta}})$, $\beta = 0.75$, where a, b, c, d are different for various stochastic approximation algorithms.

Example 3.1. Linear example. Data set [8]:

Table 1. 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 2. 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 3.1) are demonstrated in Fig. 2. Fig. 3 illustrates that the SPSAL classifier for training and testing datasets is 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 $-0.7458u_1 + 0.6873u_2 + 3.6499 = 0$. The linear separating hyperplanes in Fig. 2 are obtained solving problem (1) for training set by MathCad Software.

Example 3.2. High dimensional case. The dataset consists of 200 vectors. The covariate vectors x is 20-dimensional and generated uniformly from the unit cube $[0, 1]^{20}$. The boundary between two classes is a linear function of only first three variables: $f(x) = 2u_1 + 4u_2 + 4u_3 - 4.8$. Therefore the important set is $\{u_1, u_2, u_3\}$ and the remaining seventeen variables are redundant [7].

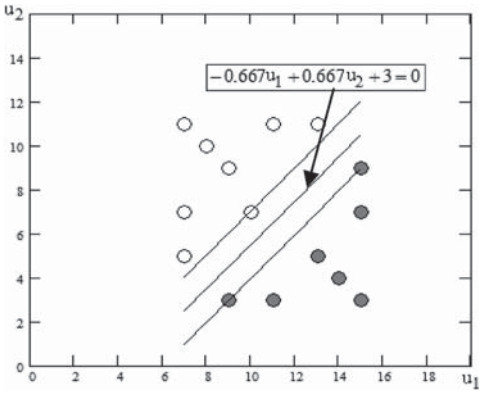

Fig. 2. Linear separating hyperplanes of training.

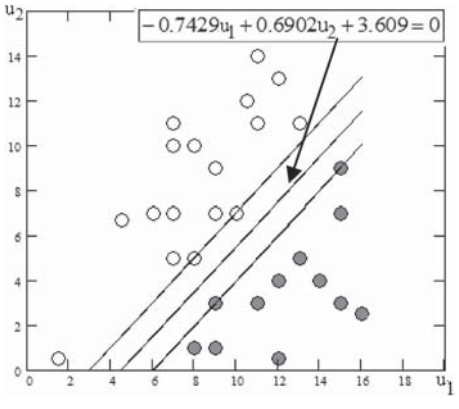

Fig. 3. Linear separating hyperplanes of the training and testing data (SPSAL).

We judge by the training error on the applicability of SA algorithms to solve problem (2). The training error is just the measured error rate on the training data and expressed as follows: $R_{emp}(b) = \frac{1}{2p}\sum_{i=1}^{p} |y_i - h(w,b)|$, where $h(w,b) = \langle w, u\rangle + b$. The "loss" is the term $\frac{1}{2}|y_i - h(w,b)|$. Fig. 4 depicts how the averaged training error rate changes for each algorithm as the training sample size p is increasing. For all SA algorithms their training error decreases significantly.

Figs. 5 and 6 show the value of the objective function during the SA iterations on two datasets described above. Dependences of averaged objective function on the number of iterations confirm convergence of the SA algorithms described above. The theoretical and empirical least squares estimates of the rate of convergence by the Monte-Carlo method are presented in Table 3. As we can see from the table, computer simulation corroborates very well the theoretically defined convergence rates.

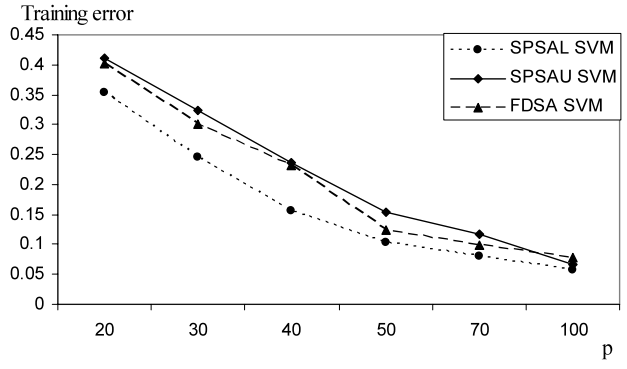

Fig. 4. The averaged training error rate as the training sample size p is increasing.

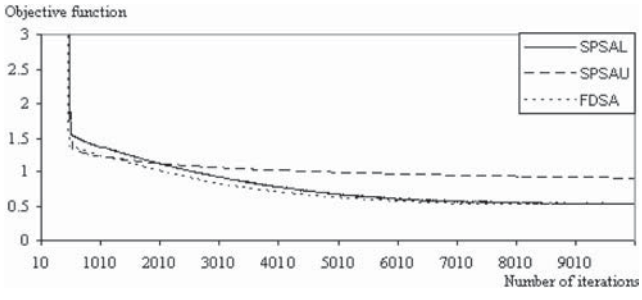

Fig. 5. Value of the averaged objective function during the SA iterations (Example 3.1, number of iterations $N = 10000$, number of trials $M = 100$).

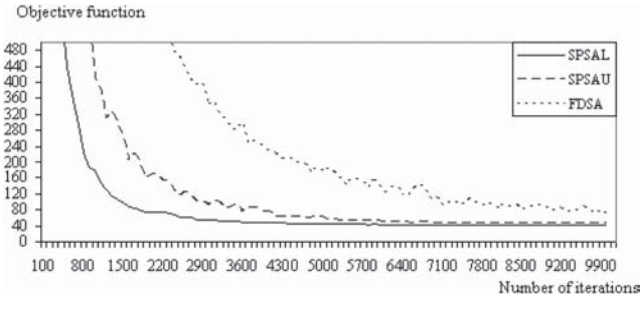

Fig. 6. Value of the averaged objective function during the SA iterations (Example 3.2, number of iterations $N = 10000$, number of trials $M = 100$).

4. Conclusions

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.

Table 3. The least squares estimates of the rate of convergence of SA algorithms by the Monte-Carlo method

Theoretical: $\gamma = 1 + \beta = 1.75$, $\beta = 0.75$	**Empirical**		
	SPSAL	**SPSAU**	**FDSA**
Example 1	1.7262	1.7137	1.7543
Example 2	1.7733	1.7388	1.7411

We explore three SA algorithms (SPSAL, SPSAU, FDSA). The applicability of SA algorithms in such problems has been studied by computer simulation. Computer simulation results corroborate very well the theoretically defined SA convergence rates. The simulation studies with proposed datasets show that these algorithms can be successfully applied to optimizing non-differentiable loss functions in the classification problems. 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. The choice of an appropriate interval for penalty coefficients might be the subject of future research.

References

1. A. Astorino, A. Fuduli, Nonsmooth optimization techniques for semisupervised classification, *IEEE Trans. Pattern Anal. Mach. Intell.*, **29**(12), 2135–2142 (2007).
2. V. Bartkutė, L. Sakalauskas, Application of stochastic approximation in technical design, *Series on Computers and Operations Research, Computer Aided Methods in Optimal Design and Operations*, **7**, 29–38 (2006).
3. V. Bartkutė, L. Sakalauskas, Simultaneous perturbation stochastic approximation for nonsmooth functions, *European Journal on Operational Research*, **181**(3), 1174–1188 (2007).
4. C. Cortes, V. Vapnik, Support-vector networks, *Machine Learning*, **20**(3), 273–297 (1995).
5. N. Cristianini, J. Shawe-Taylor, Support vector and kernel methods, in: M. Berthold, D.J. Hand (Eds.), *Intelligent Data Analysis*: *An Introduction*, Springer-Verlag (2003), pp. 169–197.
6. В.С. Михалевич, А.М. Гупал, В.И. Норкин, *Методы невыпуклой оптимизации*, Наука, Москва (1987).
7. H.H. Zhang, J. Ahn, X. Lin, C. Park, Variable selection for SVM via shrinkage methods, in: *Discussion session, SAMSI*: *Data Mining and Machine Learning Workshop*, Research Triangle Park, NC, January (2004).
8. B.-J. Ziv, A.W. Motore, Machine learning, in: *School of Computer Science*, Carnegie Mellon University (2004), pp. 10–781.

REZIUMĖ

V. Bartkutė-Norkūnienė. Stochastinės aproksimacijos metodai atraminių vektorių klasifikavimo algoritmuose

Straipsnyje pasiūlyti trys stochastinės aproksimacijos (SA) metodai binarinio klasifikavimo uždaviniams spręsti naudojant atraminių vektorių klasifikatorių (S*upport 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 du standartiniai klasifikavimo uždaviniai.

Raktiniai žodžiai: atraminių vektorių mašinos, pusiau prižiūrėtas klasifikavimas, stochastinė aproksimacija.