
Jugowice, 11th–15th September 2017

# A NEW SIMPLE LINEAR CLASSIFICATION METHOD

**Krzysztof Fujarewicz, Katarzyna Pojda**

Silesian University of Technology
Akademicka 16, 44-100 Gliwice
krzysztof.fujarewicz@polsl.pl, katarzyna.pojda@polsl.pl

## ABSTRACT

Parameters of a linear classification function may be found using several existing methods. One of the most effective is a support vector machines (SVM) technique with linear kernel. Unfortunately, this method has some drawbacks. It is relatively computationally intensive because it solves quadratic programming (QP) problem and requires the value of regularisation parameter from the user. In this article we propose a new method, which is similar in spirit to the SVM technique, but is free of mentioned disadvantages. In particular it requires solving linear programming (LP) problems instead of QP problems and it does not require the regularisation parameter, even for linearly non-separable datasets. Numerical examples confirm usefulness and efficiency of the proposed method.

## INTRODUCTION

Linear classification is the most popular method of classification. Despite the fact that it is relatively simple, in many practical situations, for real datasets, it turns out to be better than other, more complicated approaches. Parameters of linear classifier can be found/tuned using many existing methods having statistical or heuristic background. Among others it is worth to mention: Fisher linear discriminant analysis (LDA), perceptron algorithm, linear regression, support vector machines (SVM) with linear kernel and logistic regression. The latter gives in fact a non-linear classification function, but divide the feature space linearly by hyperplane. Most methods require from the user values of some additional parameters, the methods differ in computational effort and, of course, in accuracy, when they are trained with particular learning datasets. In practice one of the best is linear SVM, but it is computationally intensive and require from the user the regularisation term. We propose in this article a method that is similar to SVM, but in some respects it has some new and unique properties. We start the next section with recalling the basic idea of SVM classification.

## LINEAR SVM — A LINEARLY SEPARABLE CASE

Let us consider the set of $M$ vectors $\{x_i\}_{i=1}^{M},\ x_i \in R^n$. Each vector represents one and only one class $A$ or $B$. In standard linear classification problem we are looking for a weight vector $w$ and scalar bias $b$ of a linear classifying (discriminant) function

$$f(x) = w^T x + b \tag{1}$$

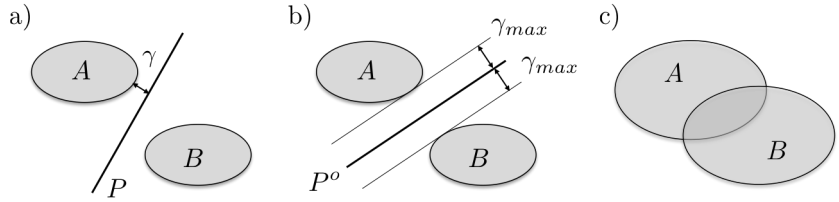


Figure 1. Linear SVM idea explanation; a) a non-optimal hyperplane $P$, b) the optimal SVM hyperplane $P^o$ with maximised margin of separation $\gamma$, c) linearly non-separable case

which satisfies the following set of inequalities

$$\begin{array}{lll} w^T x_i + b > 0 & \text{for} & x_i \in A \\ w^T x_i + b < 0 & \text{for} & x_i \in B \end{array} \tag{2}$$

for $i = 1, 2, \ldots, M$.

When the training set is linearly separable then there exist such a function.

For the simplicity of notation let us introduce a set of labels (desired responses or target outputs): $\{d_i\}_{i=1}^M$ defined based on class membership:

$$d_i = \begin{cases} +1 & \text{when} \quad x_1 \in A \\ -1 & \text{when} \quad x_1 \in B \end{cases} \tag{3}$$

Using labels the set of inequalities (2) can be rewritten

$$d_i(w^T x_i + b > 0) > 0 \quad i = 1, 2, \ldots, M \tag{4}$$

Discriminant function (1) determines in the $n$-dimensional input space, a $n-1$-dimensional hyperplane $P$ called the *decision surface*, for which the discriminant function is equal to zero:

$$P = \{x : \ w^T x + b = 0\} \tag{5}$$

An example of such hyperplane is shown in Fig. 1a. For the sake of simplicity and clarity of explanation subsets $A$ and $B$ are presented rather as regions instead of a sets of points $x_i$.

Let us introduce $\gamma$ quantity, called *margin of separation*, which is defined as Euclidean distance $\rho$ between hyperplane $P$ and the closest training vector

$$\gamma = \min_i \rho(P, x_i), \ i = 1, 2, \ldots, M \tag{6}$$

The margin of separation $\gamma$ is also presented in Fig. 1a.

Now, the basic linear SVM problem for linearly separable case can be formulated as follows:

**Problem 1.** *Find optimal $w^o$ and $b^o$ that maximise $\gamma$ subject to constraints (2)*

Results of such maximisation is presented in Fig. 1b. The optimal hyperplane $P^o$ corresponds to maximal value of margin of separation $\gamma_{max}$.

It can be shown that Problem 1 can be transformed into quadratic programming (QP) problem [1]. For more details we refer the reader to the article [1] or books [3, 4].

Unfortunately, such elegant and simple explanation exists only for linearly separable case. In case of linearly non-separable case, presented in Fig. 1c, Problem 1 and corresponding QP problem have no solutions. To make the problem numerically tractable the QP problem is modified by introducing *slack variables* leading to the concept of *soft margin* which is controlled by a regularisation term $C$, a parameter that must be specified by the user.

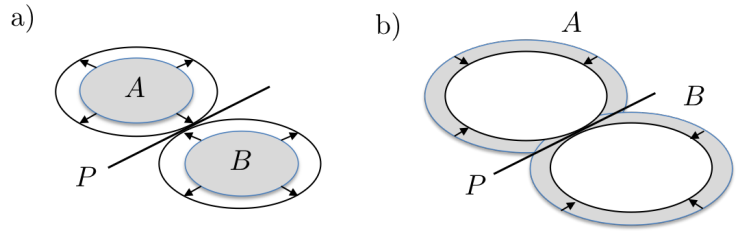


Figure 2. Scaling training sets to obtain marginally separable sets: a) expanding linearly separable set: $\beta > 1$, b) shrinking linearly non-separable sets: $\beta < 0$.

## SCALING OF THE TRAINING SET

To avoid the fundamental qualitative difference between a separable and non-separable training sets in this work a scaling operation is introduced and utilised.

Let $x_C$ be a centroid of all training vectors belonging to class (subset of points) $C$. The result of scaling of any point $x_i \in C$ around the centroid $x_C$ with the scaling factor $\beta$ is defined as follows:

$$S(x_i, x_C, \beta) = \beta x_i + (1 - \beta) x_C \tag{7}$$

For $\beta > 1$ the set $C$ is growing (around $x_C$), for $\beta < 1$ the set $C$ shrinks (towards $x_C$), and for $\beta = 1$ the set $C$ stays unchanged.

The whole two-class training set $\{A \cup B\}$ after scaling operation by factor $\beta$ becomes a new set $\{S(A, x_A, \beta) \cup S(B, x_B, \beta)\}$.

Two linearly non-separable sets $A$ and $B$ with different centroids $x_A$ and $x_B$ can be always shrinked ($\beta < 1$) such that they become linearly separable. And *vice versa*. Two linearly separable sets $A$ and $B$ can always be enlarged ($\beta > 1$) to obtain two linearly non-separable sets. Consequently, we can always achieve a boundary *marginally separable* positions of two sets. We call two sets $A$ and $B$ marginally linearly separable when they are linearly non-separable but there exists a linear function (1) satisfying a set of inequalities

$$d_i(w^T x_i + b > 0) \geq 0 \quad i = 1, 2, \ldots, M \tag{8}$$

Both cases: expanding and shrinking are presented graphically in Fig. 2.

## THE PRIMARY PROBLEM

Now, let us state the following problem which can be formulated for any, linearly separable or linearly non-separable, training sets.

**Problem 2.** *For given training set* $\{A \cup B\}$ *find the value of the scaling factor* $\beta_0$ *for which the scaled set* $\{S(A, x_A, \beta) \cup S(B, x_B, \beta)\}$ *is marginally linearly separable.*

Note that the philosophy of Problem 2 is fundamentally different from all existing classification methods. While existing methods tries to find, or *manipulate*, the classification function with unchanged training set, the proposed approach tries to manipulate the training set in order to obtain one specific i.e. marginally separating classification function.

The final linear classification function is then a function (1) that marginally separate the training set scaled by $\beta_0$ factor. The working name for our approach is *zero margin* (ZM) classifier.

From numerical point of view Problem 2 is not an easy task. Of course one can propose an iterative algorithm for searching optimal $\beta$ by sequential solving standard (hard margin) linear SVM problems. Such an approach should work but would not be elegant (from mathematical point of view) and would be very computationally intensive. Instead of this we propose to reformulate Problem 2 into slightly simpler form, which can be transformed to linear programming (LP) problem. It will be the topic of the next section.

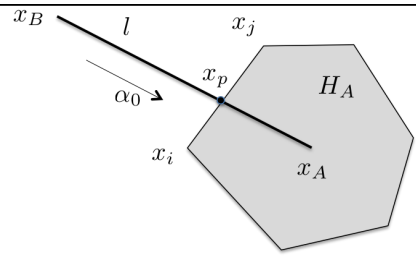


Figure 3. Illustration of the equivalence of Problem 3 to LP problem.

## THE SIMPLIFIED PROBLEM AND ITS SOLUTION

Instead of solving Problem 2 we formulate two simpler problems, one for one class:

**Problem 3.** *For given set $A$ and the centroid $x_B$ find the value of the scaling factor $\beta_A$ for which the scaled set $\{S(A, x_A, \beta) \cup x_B\}$ is marginally linearly separable.*

**Problem 4.** *For given set $B$ and the centroid $x_A$ find the value of the scaling factor $\beta_B$ for which the scaled set $\{S(B, x_B, \beta) \cup x_A\}$ is marginally linearly separable.*

Let us assume that first $M_A$ vectors belongs to class $A$. The convex hull of the set $A$ is denoted as $H_A$ and is presented in Fig. 3. Problem 3 corresponds to the problem of finding the point $x_p$. On the one hand, $x_p$ belongs to $H_A$:

$$x_p = \sum_{i=1}^{M_A} \alpha_i x_i \tag{9}$$

$$\sum_{i=1}^{M_A} \alpha_i = 1 \tag{10}$$

$$\alpha_i \geq 0; \quad i = 1, 2, \ldots, M_A \tag{11}$$

On the other hand, $x_p$ belongs to the line passing through points $x_A$ and $x_B$:

$$x_p = \alpha_0 x_A + (1 - \alpha_0) x_B; \quad \alpha_0 \in \mathbb{R} \tag{12}$$

Combining (9) and (12) we obtain

$$\alpha_0 x_A + (1 - \alpha_0) x_B = \sum_{i=1}^{M_A} \alpha_i x_i \tag{13}$$

In addition, $x_p$ is the closest point to $x_B$ satisfying (9) and (12), so it can be found by minimising $\alpha_0$. Therefore $x_p$ can be found by solving the following LP problem.

**Problem 5.** *Find $\alpha_0, \alpha_1, \ldots \alpha_{M_A}$ minimising the objective function $J = \alpha_0$ subject to constraints (10), (11) and (13).*

After solving Problem 5, $n$ of $M_A$ coefficients $\alpha_i$ are non-zero (positive). Let us denote by $\Omega_A$ the set on these $n$ coefficients. They indicate *support* vectors indicated in Fig. 3 as $x_i$ and $x_j$. Now, let us calculate a first classification function $f_A(x) = {w_A}^T x + b_A$ by solving $n + 1$ linear equations:

$${w_A}^T x_i + b_A = 0; \quad i \in \Omega_A \tag{14}$$

$${w_A}^T x_A + b_A = 1; \tag{15}$$

A symmetric function $f_B(x) = {w_B}^T x + b_B$ is found by solving a problem analogous to Problem 5 and equations analogous to (14) and (15).

The final linear classification function separating $A$ from $B$ is calculated as a difference of these two functions

$$f(x) = f_A(x) - f_B(x) = (w_A - w_B)^T x + (b_A - b_B) \tag{16}$$

**A further modification**

We noted that the classification function (16) gives satisfactory accuracy but it is possible to get even better classification quality by sequentially finding $f_A(x)$ and $f_B(x)$ and removing vectors with indices $i \in \{\Omega_A \cup \Omega_B\}$ until some percent $\eta$ of all vectors from the training set is removed. Then, the final classification function is a mean of functions (16) obtained during all such iterations.

## NUMERICAL EXAMPLES

We did simulations on three datasets, one artificial and two real problems.

An artificial data set consists of 200 samples (100 samples per class) and 2 features drawn from Gaussian distribution. Parameters of the distribution for the first feature are $\mu_{11} = 1$, $\sigma_{11} = 1.5$ for the first class and $\mu_{12} = 4$, $\sigma_{12} = 1.5$ for the second class. Parameters of the distribution for the second feature is $\mu_{21} = 1$, $\sigma_{21} = 3.7$ for the first class and $\mu_{22} = 4$, $\sigma_{22} = 1.5$ for the second class.

The first real data set is the benchmark problem for Wisconsin Breast Cancer Data classification. The data consists of 9 medical attributes (10 attributes in total, but the last one is the id number) and 699 instances including missing values. The aim of the problem is to decide whether the cancer is malignant or benign. After excluding the instances with missing values, 683 instances left in data set.

The second data set is set of diagnostic measurements of female patients of Pima Indian heritage. The objective is to predict whether a patient has diabetes. Originally dataset consist of 9 attributes and 768 observations. After excluding the instances with missing values, 392 instances left in data set. Both data sets are available in [7], but we used its adaptation to R language in [5].

The SVM solver we used is the C-SVM with linear kernel. We used libSVM package [2] suited to R language in e1071 R package [6]. We also used *tune* function from that package, which perform optimising of SVM classifier accuracy over the specified parameter vectors.

The experiment goes as follows: for each data set we use k-fold cross validation with $k = 10$ to assess accuracy of classification. In each iteration of validation we normalise train data using z-score method and perform tuning of C-SVM C parameter. Parameter C is selected from range $2^x$, where $x$ changes from $-1$ to 8 with 0.3 step. Best trained model is tested on normalised test data set. We do not perform any tuning of the ZM method. Percent parameter $\eta$ is constant and equal 5%. The comparison of accuracy rate in Table 1 proves the advantage of the ZM classifier.

Table 1. Accuracy of classification obtained by classifiers for the three investigated data sets.

| | ZM classifier | C-SVM |
|---|---|---|
| artificial | 86.5% | 84.5% |
| breast | 97.072% | 96.779% |
| indian | 78.061% | 77.806% |

## CONCLUSIONS

In this paper we proposed a new linear classification method. The idea of the approach is to scale (up or down) the original training set in order to obtain linearly marginally separable set. Then the linear classification function that marginally separate the scaled set is the final classification function. Thanks to the scaling operation, any training set, linearly separable and linearly non-separable, is treated in the same way. In particular it is not necessary to incorporate a parameter $C$ (used in standard SVM method) that plays a regularisation role but in the same time is necessary to handle linearly non-separable sets. Provided numerical examples illustrates how

the proposed method works on artificial and real biomedical data. The results obtained by our approach appeared slightly better than classic C-SVM even with optimised parameter $C$. The presented preliminary results encourage us to further modification and improvement of the method and testing it in other classification problems.

## ACKNOWLEDGEMENTS

This work was supported by the NCBiR under Grant Strategmed2/267398/4/NCBR/2015. Calculations were performed using the infrastructure supported by the computer cluster Ziemowit (www.ziemowit.hpc.polsl.pl) funded by the Silesian BIO-FARMA project No. POIG.02.01.00-00-166/08 and expanded in the POIG.02.03.01-00-040/13 in the Computational Biology and Bioinformatics Laboratory of the Biotechnology Centre at the Silesian University of Technology.

## REFERENCES

[1] B. E. Boser, I. M. Guyon, and V. N. Vapnik: *A training algorithm for optimal margin classifiers*, Proceedings of the fifth annual workshop on Computational learning theory, 1992, pp. 144–152.

[2] C. Chang and C. Lin: *LIBSVM: A library for support vector machines*, ACM Transactions on Intelligent Systems and Technology **2** (2011), 27:1–27:27, Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[3] N. Cristianini and J. Shawe-Tylor: *An introduction to support vector machines and other kernel-based learning methods*, Cambridge University Press, UK, 2000.

[4] S. Haykin: *Neural networks — a comprehensive foundation*, Prentice Hall, 1999.

[5] F. Leisch and E. Dimitriadou: *mlbench: Machine Learning Benchmark Problems*, 2010, R package version 2.1-1.

[6] D. Meyer, E. Dimitriadou, K. Hornik, A. Weingessel, and F. Leisch: *e1071: Misc Functions of the Department of Statistics, Probability Theory Group (Formerly: E1071), TU Wien*, 2017, R package version 1.6-8.

[7] D. J. Newman, S. Hettich, C. L. Blake, and C. J. Merz: *UCI Repository of machine learning databases*, University of California, Irvine, Dept. of Information and Computer Sciences, 1998.