A Statistical Binary Classifier: Probabilistic Vector Machine

Mihai Cimpoeşu, Andrei Sucilă, and Henri Luchian

Alexandru Ioan Cuza University, Faculty of Computer Science, Iasi, Romania
mihai.cimpoesu@info.uaic.ro

Abstract. A binary classification algorithm, called Probabilistic Vector Machine – PVM, is proposed. It is based on statistical measurements of the training data, providing a robust and lightweight classification model with reliable performance. The proposed model is also shown to provide the optimal binary classifier, in terms of probability of error, under a set of loose conditions regarding the data distribution. We compare PVM against GEPSVM and PSVM and provide evidence of superior performance on a number of datasets in terms of average accuracy and standard deviation of accuracy.

Keywords: binary classification, hyperplane classifier, SVM, probabilistic, statistical model, kernel.

1 Introduction

In the field of supervised classification, linear binary discriminative classifiers have long been a useful decision tool. Two representatives of this class are the Perceptron, introduced in [7], and Support Vector Machines (SVM), introduced in [19], [5], [20], which have been highly successful in tackling problems from diverse fields, such as bioinformatics, malware detection and many others.

Binary classification problems have as input a training set, which we will denote as $S = \{(x_i, y_i) | x_i \in \mathbb{R}^n, y_i \in \{\pm 1\}, i \in \overline{1..m}\}$, consisting of points $x_i \in \mathbb{R}^n$ and their labels $y_i \in \{\pm 1\}$. We will also denote by $S_+ = \{x_i \in S | y_i = 1\}$ and by $S_- = \{x_i \in S | y_i = -1\}$ the positively and, respectively, negatively labeled training points.

For an input training set a decision function, $f : \mathbb{R}^n \to \{\pm 1\}$, is obtained, for which the parameters are chosen according to the training set. This decision function is later used to classify new examples, referred to as test examples.

Linear binary classifiers search for a hyperplane defined by its normal vector, $w \in \mathbb{R}^n$, and its offset $b \in \mathbb{R}$. This hyperplane is used to define the decision function as $label(x) = sgn(< w, x > +b)$, thus assigning positive labels to the points found in the positive semispace and negative labels to those found in the negative semispace. The hyperplane is refered to as a separating hyperplane.

The focus of this paper is to introduce a binary classifier - named Probabilistic Vector Machine (PVM) - which is based on statistical information derived from the training data. This will provide a strong link to the generalization ability

L. Correia, L.P. Reis, and J. Cascalho (Eds.): EPIA 2013, LNAI 8154, pp. 211–222, 2013.

and a robust model that would be resilient to outliers. The mathematical modeling should also allow for complete and efficient resolution of the optimization problem.

The paper is structured as follows: related work in Section 2; model and motivation in Section 3; kernel usage for nonlinear classification in Section 4; solution for the proposed optimization problem in Section 5; test results in Section 6 and conclusions in Section 7.

2 Related Work

Since the first linear binary classifier, the perceptron, was first introduced in [7] many other classifiers using the same type of decision function have been introduced. The differences among linear binary classifiers are given by the criteria for choosing the separating hyperplane. SVM, introduced in [19], [5], [20], searches for the separating hyperplane which maximizes the distance to the closest point from the S_+ and S_- sets. It was originally developed only for the separable case, in which $conv(S_+) \cap conv(S_-) = \emptyset$, also referred to as the hard margin SVM. However, this rather stringent constraint on the input was later relaxed by allowing training errors, for which the objective is altered. This introduces a tradeoff which is sometimes hard to balance and requires a grid search for the parameters, where every combination of parameters is evaluated via a fold procedure. This can be very time consuming.

SVMs are shown by Vapnik to be built upon the Structural Risk Minimization (SRM) principle. Shortly after its introduction, it quickly outperformed the existing classifiers as shown in domains such as pattern classification [15], text processing [12], bioinformatics [17] and many others. The choice of the hyperplane as done by SVM has been shown to be relatable to the probability of error in [3] and [4]. Some of the shortcomings of SVM stem from the fact that it bases the hyperplane construction on the border elements of S_+ and S_-.

To address some of the limitations of SVMs, [9] introduces a bound on the expected generalization error and subsequently uses this bound in [10] to develop a classifier which minimizes this measure. This allows the separating hyperplane to be built using information derived from the distribution of the entire training set.

The bounds approach, however, produces optimization problems with nonconvex objectives. It leads to models which are difficult to solve efficiently. Currently, the methods based on error bounds resort to simple hill climbing, as the optimization process can become very complicated if one aims at fully solving the underlying problems.

A different approach is taken by [8], where the proximal support vector machines (PSVM) are introduced. PSVM uses two parallel hyperplanes in order to label points. The label is attributed based on which of the two hyperplanes is closer. This, of course, is equivalent to having a single separating hyperplane. This approach eliminates the quadratic program and provides similar classification ability. [16] later introduces the generalized eigenvalue proximal SVM

(GEPSVM), a classification process based on two nonparallel hyperplanes. This idea is further built upon in [11] by introducing the twin SVM (TSVM), where two hyperplanes are used to represent the data. Later, [14] use the same idea to introduce the least squares TSVM (LSTSVM) which also searches for two nonparallel hyperplanes that minimize the distances introduced in [11] in a least squares sense. [18] present a noise resistant variant of TSVM entitled robust twin SVM (R-TWSVM) designed to correctly classify data that contains measurement errors. The various versions that have been introduced in previous years have also provided a incremental increase in clasiffication ability.

A detailed overview of some of the most popular classification techniques can be found in [21] and [13].

3 Probabilistic Vector Machine

For a motivation of our model, let $h = (w, b) \in \mathbb{R}^n \times \mathbb{R}$ be a hyperplane in the problem space. Let $x \in \mathbb{R}^n$ be a point whose actual label is $y \in \{\pm 1\}$. The probability of error may then be expressed as:

$$\begin{aligned} P(err) =& P(y \neq sgn(< w, x > +b)) \\ =& P((y = 1) \cap (< w, x > +b < 0)) \\ &+ P((y = -1) \cap (< w, x > +b > 0)) \end{aligned}$$

So the probability of error is the sum of the probabilities of false negatives (FN) and false positives (FP). The objective proposed by PVM is the minimization of the maximum between these two components. Specifically, the hyperplane sought is:

$$\begin{aligned} (w, b) = \arg_{(w,b)} \min \max \{ & P((y = 1) \cap (< w, x > +b < 0)), \\ & P((y = -1) \cap (< w, x > +b > 0)) \} \end{aligned}$$

This choice is motivated by several arguments:

- In practical settings, the accuracy of a classifier is not the only measure. In order to be of use, both FN and FP probabilities have to be low.
- The reduction of the two also leads to the reduction of their sum, although, admittedly, sometimes not the lowest possible value. However, there is clearly a strong link between minimizing the maximum of the two and minimizing their sum.
- Minimizing the maximum of the two leads, under certain conditions, to a mathematical model which can be solved with convex optimizers.

During the training stage of the problem we only have the S_+ and S_- to base the choice of the hyperplane upon. Therefore, the probabilities for FN and FP have to be related to these sets. The objective expressed using only S_+ and S_- is expressed as:

$$\begin{aligned} (w, b) = \arg_{(w,b)} \min \max \{ & P((< w, x > +b < 0) | x \in S_+), \\ & P((< w, x > +b > 0) | x \in S_-) \} \end{aligned} \tag{3.1}$$

Assume that the signed distances to the hyperplane are normally distributed. Let E_+, E_- be the averages over the positive and, respectively, negative training set. Let σ_+, σ_- be the respective standard deviations. Note that these are in fact induced by the choice of the hyperplane. The probabilities can be expressed as a function of the distance between the hyperplane and the average divided by the corresponding $\sigma_\pm$. The hyperplane coresponds to a signed distance of 0, so if we denote by $\lambda_+ = \frac{E_+}{\sigma_+}$ and $\lambda_- = \frac{-E_-}{\sigma_-}$, then we get:

$$P((< w, x > +b < 0)|x \in S_+) = \int_{-\infty}^{-\lambda_+} f(s)ds$$
$$P((< w, x > +b > 0)|x \in S_-) = \int_{-\infty}^{-\lambda_-} f(s)ds$$

where $f : \mathbb{R} \to [0, 1]$ is the gaussian density of probability function. The objective can then be stated as:

$$\begin{aligned}(w, b) &= \arg_{(w,b)} \min \max\{-\lambda_+, -\lambda_-\} \\ &= \arg_{(w,b)} \max \min\{\lambda_+, \lambda_-\}\end{aligned}$$

The condition that the distributions are normal can be replaced by a weaker one.

Definition 1. *Two random variables, D_+ and D_-, with means E_+, E_- and standard deviations σ_+, σ_-, are called similarly negatively distributed if:*

$$P(\frac{D_+ - E_+}{\sigma_+} \leq -\lambda) = P(\frac{D_- + E_-}{\sigma_-} \leq -\lambda), \forall \lambda \in [0, \infty) \tag{3.2}$$

This property obviously holds if D_+ and D_- are each normally distributed. It also holds for random variables which are distributed simetrically and identically, but with different means.

If the distances to the separating hyperplane have the property (3.2), then the optimal hyperplane in terms of (3.1) can be found by optimizing:

$$(w, b) = \arg_{(w,b)} \max \min\{\frac{E_+}{\sigma_+}, -\frac{E_-}{\sigma_-}\}$$

Making the choice of the separating hyperplane in this way would yield signed distance distributions as shown in Figure 1.

The standard deviation is then replaced by the average deviation, in order to obtain a model to which convex optimization can be applied. This will be the case for all σ definitions. The system that defines our problem is then be expressed as:

$$\begin{cases} minmax\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\} \\ \frac{1}{|S_+|}\sum_{x_i \in S_+} d(x_i, h) = E_+ \\ -\frac{1}{|S_-|}\sum_{x_i \in S_-} d(x_i, h) = E_- \\ E_+ > 0, E_- > 0 \\ \sigma_+ = \frac{1}{|S_+|-1}\sum_{x_i \in S_+} |d(x_i, h) - E_+| \\ \sigma_- = \frac{1}{|S_-|-1}\sum_{x_i \in S_-} |d(x_i, h) + E_-| \end{cases}$$

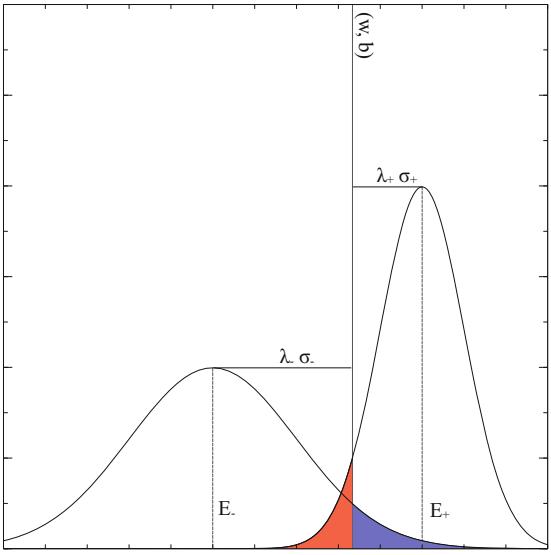

Fig. 1. Induced distance distributions. The hyperplane is sought such as to minimize the maximum between the red and blue areas, which correspond to the FN and FP probabilities. Equivalently, the hyperplane has to maximize the minimum between λ_+ and λ_-.

Note that $d(x_i, h) = \frac{<w,x>+b}{||w||}$, so:

$$\begin{array}{l}
\frac{1}{||w||\cdot|S_+|}\sum_{x_i\in S_+} < w, x_i > +b = E_+ \\
-\frac{1}{||w||\cdot|S_-|}\sum_{x_i\in S_-} < w, x_i > +b = E_- \\
\sigma_+ = \frac{1}{||w||\cdot|S_+|-1}\sum_{x_i\in S_+} | < w, x_i > +b - ||w|| \cdot E_+| \\
\sigma_- = \frac{1}{||w||\cdot|S_-|-1}\sum_{x_i\in S_-} | < w, x_i > +b + ||w|| \cdot E_-|
\end{array}$$

Since the objective depends upon $\frac{\sigma_+}{E_+}$ and $\frac{\sigma_-}{E_-}$, the system may be rewritten as:

$$\begin{cases}
minmax\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\} \\
\frac{1}{|S_+|}\sum_{x_i\in S_+} < w, x_i > +b = E_+ \\
-\frac{1}{|S_-|}\sum_{x_i\in S_-} < w, x_i > +b = E_- \\
E_+ \geq 1, E_- \geq 1 \\
\sigma_+ = \frac{1}{|S_+|-1}\sum_{x_i\in S_+} | < w, x_i > +b - E_+| \\
\sigma_- = \frac{1}{|S_-|-1}\sum_{x_i\in S_-} | < w, x_i > +b + E_-|
\end{cases} \tag{3.3}$$

Note that, in order for the system to be feasible, all that is required is that $\frac{1}{|S_+|}\sum_{x_i\in S_+} x_i \neq \frac{1}{|S_-|}\sum_{x_i\in S_-} x_i$, because then one can choose $w_0 \in \mathbb{R}^n$ such that:

$$\frac{1}{|S_+|}\sum_{x_i\in S_+} w_0 \cdot x_i \neq \frac{1}{|S_-|}\sum_{x_i\in S_-} w_0 \cdot x_i$$

. If $\frac{1}{|S_+|}\sum_{x_i \in S_+} w_0 \cdot x_i < \frac{1}{|S_-|}\sum_{x_i \in S_-} w_0 \cdot x_i$, then $w_1 = -w_0$; else, $w_1 = w_0$. We now have:

$$\frac{1}{|S_+|}\sum_{x_i \in S_+} w_0 \cdot x_i > \frac{1}{|S_-|}\sum_{x_i \in S_-} w_0 \cdot x_i$$

and may choose the offset $b \in \mathbb{R}$ such that:

$$b + \frac{1}{|S_+|}\sum_{x_i \in S_+} w_0 \cdot x_i > 0 > b + \frac{1}{|S_-|}\sum_{x_i \in S_-} w_0 \cdot x_i$$

which, after scaling, gives a feasible solution to System (3.3).

If the system is not feasible, then, from the viewpoint of a normal distribution of distances, it would be pointless to search for a separation, because the accuracy rate would be at most 50%.

Lemma 1. *The optimization problem* (3.3) *is equivalent to solving:*

$$\begin{cases} minmax\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\} \\ b + \frac{1}{|S_+|}\sum_{x_i \in S_+} < w, x_i > = E_+ \\ -b - \frac{1}{|S_-|}\sum_{x_i \in S_-} < w, x_i > = E_- \\ E_+ \geq 1, E_- \geq 1 \\ | < w, x_i > + b - E_+| \leq \sigma_+^i, \forall x_i \in S_+ \\ | < w, x_i > + b + E_-| \leq \sigma_-^i, \forall x_i \in S_- \\ \sigma_+ = \frac{1}{|S_+|-1}\sum_{x_i \in S_+} \sigma_+^i \\ \sigma_- = \frac{1}{|S_-|-1}\sum_{x_i \in S_-} \sigma_-^i \end{cases} \tag{3.4}$$

Note that, apart from the objective function, system (3.4) uses only linear equations. This will lead to an easy solution, as detailed in Section 5.

The important properties of the model proposed thus far are that it has a direct connection to the generalization error embedded in the objective function and that it is likely to be resilient to outliers; the latter results from the fact that it is based on a statistical model of the training data which provides a built–in mechanism for dealing with outliers.

Also note that, because of the way the system is built, it does not require for S_+ and S_- to be linearly separable, as hard margin SVM would require, and does not require special treatment for classification errors, thus avoiding the introduction of a tradeoff term in the objective function.

4 Using Kernels

The system introduced thus far focuses on simply the linear aspect of the problem. However, many practical applications require nonlinear separation. The way to achieve this is by using kernel functions. In order to do so, we first project the points into a Hilbert space, H, via a projection function $\Phi : \mathbb{R}^n \to H$. In the H space we train our classifier in the linear manner described in Section 3. Since the constraint equations are linear and the separation is linear as well, the

search for w can be restricted to the linear subspace generated by the training points.

Consider $w = \sum_{i=1}^{m} \alpha_i \Phi(x_i)$, where $\alpha_i \in \mathbb{R}$. The scalar products can be expressed as:

$$< w, \Phi(x) >=< \sum_{i=1}^{m} \alpha_i \Phi(x_i), \Phi(x) >= \sum_{i=1}^{m} \alpha_i < \Phi(x_i), \Phi(x) >$$

By defining $K : \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ as $K(u, v) =< \Phi(u), \Phi(v) >$, the projection function does not require an explicit definition. Indeed, using Mercer's theorem, one only needs to define the kernel function, K, and have the projection function Φ implicitly defined.

Replacing the scalar product accordingly in system (3.4), we obtain the following system:

$$\begin{cases} \min \max\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\} \\ b + \sum_{i=1}^{m} [\alpha_i \cdot \frac{1}{|S_+|} \sum_{x_j \in S_+} K(x_i, x_j)] = E_+ \\ -b - \sum_{i=1}^{m} [\alpha_i \cdot \frac{1}{|S_-|} \sum_{x_j \in S_-} K(x_i, x_j)] = E_- \\ |\sum_{x_i \in S} \alpha_i K(x_i, x_j) + b - E_+| \leq \sigma_+^j \ , \forall x_j \in S_+ \\ |\sum_{x_i \in S} \alpha_i K(x_i, x_j) + b + E_-| \leq \sigma_-^j \ , \forall x_j \in S_- \\ \frac{1}{|S_+|-1} \sum_{x_i \in S_+} \sigma_+^i = \sigma_+ \\ \frac{1}{|S_-|-1} \sum_{x_i \in S_-} \sigma_-^i = \sigma_- \\ \sigma_+ \leq t \cdot E_+ \\ \sigma_- \leq t \cdot E_- \\ E_+ \geq 1, \ E_- \geq 1 \end{cases} \tag{4.1}$$

5 Solving the PVM Problem

While thus far we have proposed a model for the problem, we have not discussed yet the way this system can be solved.

In the current formulation, since fractions do not preserve convexity, the objective $\min \max\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\}$ is not a convex function. However, each of the fractions used has linear factors. Hence, one deals with a quasilinear function (see [1], Chapter 3, pg. 95 for details), meaning that each sublevel and superlevel set is a convex set. Moreover, the maximum of two quasiconvex functions is a quasiconvex function. To see this, let $t \in [0, +\infty)$. Restricting our domain to $E_+ > 0, E_- > 0$, we get:

$$\begin{aligned} &\max\{\tfrac{\sigma_+}{E_+}, \tfrac{\sigma_-}{E_-}\} \leq t \Leftrightarrow \\ &\Leftrightarrow \begin{cases} \frac{\sigma_+}{E_+} \leq t \\ \frac{\sigma_-}{E_-} \leq t \end{cases} \Leftrightarrow \begin{cases} \sigma_+ \leq t \cdot E_+ \\ \sigma_- \leq t \cdot E_- \end{cases} \end{aligned}$$

which, for a fixed t, is a set of linear equations, the intersection of which is a convex domain.

A quasiconvex function has only one (strict) local optimum, which is also the global one. As a consequence, one can solve system (4.1) via a set of feasibility problems. To see this, let us denote by $Feas(t)$ the feasibility problem obtained by enforcing the condition $\max\{\frac{\sigma_+}{E_+}, \frac{\sigma_-}{E_-}\} \leq t$:

$$\begin{cases} \sigma_+^j - \sum_{x_i \in S} \alpha_i (K(x_i, x_j) - K_+^i) \geq 0 \ , \forall x_j \in S_+ \\ \sigma_+^j + \sum_{x_i \in S} \alpha_i (K(x_i, x_j) - K_+^i) \geq 0 \ , \forall x_j \in S_+ \\ \sigma_-^j - \sum_{x_i \in S} \alpha_i (K(x_i, x_j) - K_-^i) \geq 0 \ , \forall x_j \in S_- \\ \sigma_-^j + \sum_{x_i \in S} \alpha_i (K(x_i, x_j) - K_-^i) \geq 0 \ , \forall x_j \in S_- \\ (|S_+| - 1)t \cdot \left(b + \sum_{x_i \in S} \alpha_i K_+^i\right) - \sum_{x_i \in S_+} \sigma_+^i \geq 0 \\ (|S_-| - 1)t \cdot \left(-b - \sum_{x_i \in S} \alpha_i K_-^i\right) - \sum_{x_i \in S_-} \sigma_-^i \geq 0 \\ b + \sum_{x_i \in S} \alpha_i K_+^i \geq 1 \\ -b - \sum_{x_i \in S} \alpha_i K_-^i \geq 1 \end{cases} \tag{5.1}$$

The optimal solution to system (4.1) is, then:

$$\begin{aligned} t_{optimal} &= \inf\{t \in \mathbb{R}_+ | Feas(t) \text{ is feasible}\} \\ &= \sup\{t \in \mathbb{R}_+ | Feas(t) \text{ is infeasible}\} \end{aligned}$$

The $t_{optimal}$ value can thus be easily found using a simple bisection procedure. Initialize $0 = t_{left} < t_{optimal} < t_{right} = \infty$. Each iteration, let $t = 0.5 \cdot (t_{left} + t_{right})$. If $Feas(t)$ is feasible, then $t_{right} = t$, otherwise $t_left = t$.

The feasibility problems formulated during this bisection procedure can be solved using one of the many linear programming solvers freely available. Note, however, that as the bounds on $t_{optimal}$ get closer together, the feasible region of $Feas(t)$ approaches a single point and this can lead to numerical problems in the linear solvers.

6 Results

For the linear solver required by PVM, we have used the GNU Linear Programming Kit (GLPK) which is freely available. The CPLEX library can also be used with the code. A version of the algorithm, with a few extra features which are not discussed in this paper, can be found at `https://code.google.com/p/dpvm/`.

6.1 Testing on Aritificial Data

In the figures shown in this section the stars have positive labels, the circles have negative labels and the obtained separating hyperplane is represented by a single black line.

In Figure 2 we compared our method with the soft margin SVM. For training the SVM we use a well known package, libSVM, presented in [6], [2]. The comparisons where done using artificial datasets. This is for illustrating the way in which PVM works. As can be seen, PVM takes into account the distribution

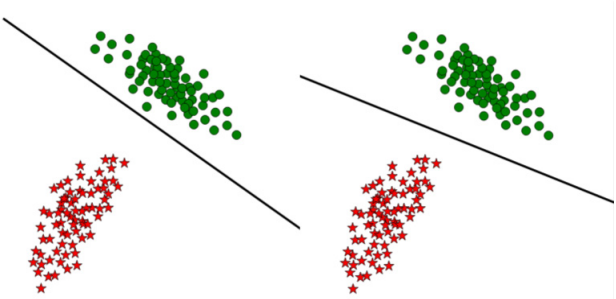

Fig. 2. PVM Separation on the left; SVM Separation on the right; PVM separation takes into account the distributions of the two sets, not just the closest members

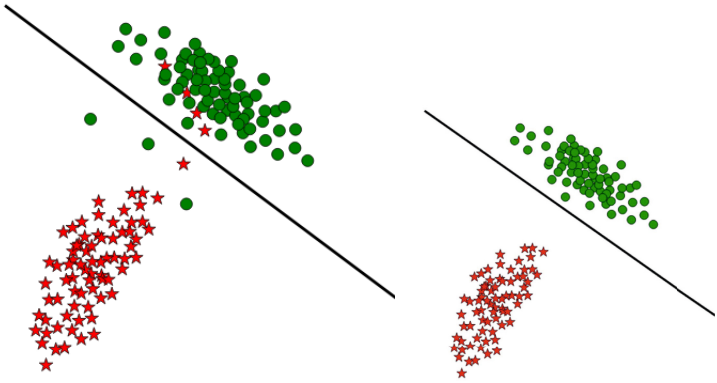

Fig. 3. Comparing the result of the training process on a set of points with the same training result on the same set to which some outliers have been added

of the entire training dataset, while SVM takes into account only the bordering elements of S_+ and S_-.

Figure 3 compares the training result when using the same data as in Figure 2 with the training result obtained when adding some outliers to this data. As is evident, the outliers bear little influence on the training process.

6.2 Testing on Public Datasets

We have conducted a set of tests on datasets from the University of California Irvine Machine Learning (UCI ML) repository. The datasets used can be found at `http://archive.ics.uci.edu/ml/`. Table 1 describes the datasets used. Preprocessing consisted in normalizing the features in each database. PVM has been compared to GEPSVM, presented in [16] and PSVM, presented in [8], which both proved similar or superior performance to SVMs. The parameters of each algorithm were obtained via grid searches and the evaluation of a combination of parameters was done using tenfold cross validation. The optimal parameters

Table 1. UCI ML datasets description

Dataset	Records	Features	Positive	Negative
Hepatitis	156	19	32	123
WPBC	199	34	47	151
Sonar	209	60	111	97
Heart-statlog	270	14	150	120
Heart-c	303	14	139	164
Bupa Liver	345	7	200	145
Ionosphere	351	34	126	225
Votes	434	16	267	167
Australian	690	14	307	383
Pima-Indian	768	8	500	268

where then used for a set of 100 tenfold cross validation tests, or equivalently 1000 tests. Table 2 shows the results in terms of average accuracy and standard deviation obtained for the RBF kernel for PVM, GEPSVM and PSVM. The average accuracy and its standard deviation where computed over the 1000 tests. As the results presented in Table 2 suggest, PVM outperforms the other classfiers in terms of average accuracy, winning on 5 of the datasets, with a large improvement on 3 of these, namely Hepatitis, Heart–c and Votes. It is important to note that PVM outperforms the other algorithms especially on the larger datasets. One possibility why this happens is that the statistical measurements used offer more relevant and stable information once the number of training points is large

Table 2. Comparison between PVM, GEPSVM and PSVM on the UCI ML datasets on the RBF kernel. The results indicate the average accuracy over 100 tenfold runs. The accuracy is measured in percentages. The best results for a dataset are shown in bold.

Dataset	PVM	GEPSVM	PSVM
Hepatitis	**87.151**±0.94	79.28±5.2	78.57±0.24
WPBC	78.853±0.64	80±5.97	**80.55**±3.92
Sonar	86.995±1.37	80±5.97	**90**±7.21
Heart–statlog	77.548±1.84	**86.52**±7.36	70.74±6.86
Heart–c	**77.119**±1.21	70.37±8.90	70.68±7.66
Bupa Liver	73.021±0.92	68.18±6.2	**74.84**±9.04
Ionosphere	92.903±1.36	84.41±6.2	**95**±4.17
Votes	**96.54**±0.39	94.5±3.37	95.95±2.25
Australian	**83.623**±0.94	69.55±5.37	73.97±6.16
Pima-Indian	**77.133**±0.31	75.33±4.91	76.8±3.83

enough. One other important observation is that, of the algorithms tested here, PVM has a distinctly lower standard deviation of the accuracy. This implies that the derived separating hyperplane is more stable than the one derived by PSVM or the two hyperplanes of GEPSVM. The datasets used in this comparison have not been chosen such that the condition (3.2) is satisfied. However, PVM proves to be competitive. This suggests that, although the setting in which PVM provides an optimal separation hyperplane requires a special condition, practical examples do not always stray from the proposed model more than they stray from the models proposed by GEPSVM or PSVM.

7 Conclusion

We have introduced a new linear binary classifier designed to use statistical measurements of the training datasets. The underlying model of PVM is robust to outliers and shows good generalization ability on the datasets tested. PVM can also use linear programming tools, which are well established in the literature.

Future work will include the study of dedicated linear programming tools and will focus on developing a stable distributed feasibility solver to tackle the optimization problem proposed by the theoretical foundation of the algorithm.

Acknowledgements. This research has been supported through financing offered by the POSDRU/ 88/1.5/S/47646 Project for PHDs. This work has also been supported by the European Social Fund in Romania, under the responsibility of the Managing Authority for the Sectoral Operational Programme for Human Resources Development 2007-2013 [grant POSDRU/CPP 107/DMI 1.5/S/ 78342].

References

1. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press (2004)
2. Chang, C.-C., Lin, C.-J.: Libsvm: a library for support vector machines. ACM Transactions on Intelligent Systems and Technology 2(3) (2011)
3. Chapelle, O., Vapnik, V.N.: Bounds on error expectation for support vector machines. Neural Computation 12(9), 2012–2036 (2000)
4. Chapelle, O., Vapnik, V.N.: Choosing multiple parameters for support vector machines. Machine Learning 46(1-3), 131–159 (2001)
5. Cortes, C., Vapnik, V.N.: Support-vector networks. Machine Learning 20(3), 273–297 (1995)
6. Fan, R.-E., Chen, P.-H., Lin, C.-J.: Working set selection using second order information for training support vector machines. The Journal of Machine Learning Research 6, 1889–1918 (2005)
7. Frank, R.: The perceptron a perceiving and recognizing automaton. Technical Report 85-460-1, Cornell Aeronautical Laboratory (1957)
8. Fung, G., Mangasarian, O.L.: Proximal support vector machine classifiers. In: Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, pp. 77–86. ACM, New York (2001)

9. Garg, A., Har-Peled, S., Roth, D.: On generalization bounds, projection profile, and margin distribution. In: ICML 2002 Proceedings of the Nineteenth International Conference on Machine Learning, Sydney, Australia, pp. 171–178. Morgan Kaufmann Publishers Inc. (2002)
10. Garg, A., Roth, D.: Margin distribution optimization. Computational Imaging and Vision 29, 119–128 (2005)
11. Jayadeva, Khemchandani, R., Chandra, S.: Twin support vector machines for pattern classification. IEEE Transactions on Pattern Analysis and Machine Intelligence 29(5), 905–910 (2007)
12. Joachims, T., Ndellec, C., Rouveriol, C.: Text categorization with support vector machines: learning with many relevant features. In: Nédellec, C., Rouveirol, C. (eds.) ECML 1998. LNCS, vol. 1398, pp. 137–142. Springer, Heidelberg (1998)
13. Kotsiantis, S.B.: Supervised machine learning: A review of classification techniques. Informatica 31, 3–24 (2007)
14. Kumar, A.M., Gopal, M.: Least squares twin support vector machines for pattern classification. Expert Systems with Applications: An International Journal 36(4), 7535–7543 (2009)
15. Lee, S., Verri, A.: SVM 2002. LNCS, vol. 2388. Springer, Heidelberg (2002)
16. Mangasarian, O.L., Wild, W.: Multisurface proximal support vector machine classification via generalized eigenvalues. IEEE Transaction on Pattern Analysis and Machine Intelligence 28(1), 69–74 (2005)
17. Noble, W.S.: Support vector machine applications in computational biology. In: Kernel Methods in Computational Biology, pp. 71–92 (2004)
18. Qi, Z., Tian, Y., Shi, Y.: Robust twin support vector machine for pattern classification. Pattern Recognition (June 27, 2012) (accepted)
19. Vapnik, V.N., Boser, B.E., Guyon, I.: A training algorithm for optimal margin classifiers. In: COLT 1992 Proceedings of the Fifth Annual Workshop on Computational Learning, Pittsburgh, PA, USA, vol. 5, pp. 144–152. ACM, New York (1992)
20. Vapnik, V.N.: Statistical learning theory. John Wiley and Sons Inc. (1998)
21. Witten, I.H., Frank, E., Hall, M.A.: Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann (2011)