

DIMACS Technical Report 2009-18
June 2009

Experimental Study of Support Vector Machines Based
on Linear and Quadratic Optimization Criteria¹

by

Alexey Nefedov
Dep. of Veterinary Clinical Sciences
University of Liverpool
alexeyn@dimacs.rutgers.edu

Jiankuan Ye
Dep. of Computer Science
Rutgers University
jiye@cs.rutgers.edu

Casimir Kulikowski
Dep. of Computer Science
Rutgers University
kulikows@cs.rutgers.edu

Ilya Muchnik
DIMACS
Rutgers University
muchnik@dimacs.rutgers.edu

Kenton Morgan
Dep. of Veterinary Clinical Sciences
University of Liverpool
k.l.morgan@liverpool.ac.uk

¹This work was supported in part by grant BB/D012627/1 from the UK Biotechnology and Biosciences Research Council.

ABSTRACT

We present results from a comparative empirical study on the performance of two methods for constructing support vector machines (SVMs). The first method is the conventional one based on the quadratic programming approach, which builds the optimal separating hyperplane maximizing the margin between two classes (optimal SVM). The second method is based on the linear programming approach suggested by Vapnik to build a separating hyperplane with the minimum number of support vectors (heuristic SVM). Using synthetic data from two classes, we compare the classification performance of these SVMs, with an in-depth geometrical comparison of their separating hyperplanes and support vectors. We show that both classifiers achieve practically identical classification accuracy and generalization performance. However, the heuristic SVM has many fewer support vectors than the optimal SVM. In addition, in contrast to the optimal SVM, its support vectors lie on the furthestmost borders of the classes, at the maximum distance from the opposite class. In our future work, we will seek to find a theoretical basis to explain these geometrical patterns of the heuristic SVM. We will also compare these classifiers using real benchmark data.

Key words: support vector machines, quadratic programming, linear programming, geometrical patterns of separating hyperplanes and support vectors.

1 Introduction

In this paper, we present a comparative experimental study of two methods for constructing support vector machines (SVMs). One is the conventional method based on quadratic programming (QP), while the other is based on linear programming (LP). Statistical learning theory shows that SVM classifier based on the solution of QP problem has the best generalization ability¹ among all hyperplane classifiers [12]. This classifier and its separating hyperplane are, therefore, called optimal. In [12], Vapnik proposed SVM based on the LP problem aimed at minimizing the number of support vectors of the separating hyperplane. He called this classifier heuristic since no theoretical estimates of its generalization performance were obtained at that time.

The goal of our work is to carry out an experimental investigation of similarities and differences between the optimal and heuristic SVM classifiers. We wanted to compare the accuracy of the classifiers, their separating hyperplanes, the number and spatial location of their support vectors. In particular, we wanted to see how many fewer support vectors a heuristic SVM would have in relation to the optimal SVM for the same data. We limited our study to considering classification problems with two classes, using synthetic data, and SVMs with linear kernels.

There are several related works which compare SVMs based on linear and quadratic optimization, including [13, 9, 11, 2, 14]. However, we do not know any work that has studied SVM based on Vapnik's LP problem suggested in [12]. Besides, [13, 9, 11, 2, 14] focus on the convergence rate and performance of the methods, and do not take into account geometrical properties of the classifiers. The novelty of our study is in the geometrical comparison of the separating hyperplanes and support vectors obtained by the different methods. This comparison allows us to make important conclusions about differences between the heuristic and optimal SVMs.

Our experiments have shown that classifiers constructed using QP and LP are practically identical in terms of generalization ability. However, there are two important differences between them. First, they have different number of support vectors. In line with the idea used to formulate the LP problem, the heuristic classifier has many fewer support vectors than the optimal one. Second, their support vectors have very different distribution in space. As is known, for the optimal classifier its support vectors lie on the margin between two classes. In other words, they lie on the inner borders between two classes, among those vectors that are closest to the vectors of the opposite class. In contrast to this, for the heuristic classifier its support vectors turned out to lie on the furthestmost borders of the two classes, at the maximum distance from the vectors of the opposite class.

This leads us to an important methodological conclusion. Over the last 15 years, the view about special importance of the training vectors lying on the margin between the classes has become commonplace in the SVM community. The importance of vectors lying on the distant, furthestmost borders of two classes, has been downgraded and viewed as negligible.

¹The generalization ability (performance) of a classifier is defined as its ability to correctly classify new objects that are not included in the training set.

Our experiments show the issue about the relative importance of different vectors in the training set is more complex than it might conventionally appear. It turns out that most distinctive vectors of the training set can be used to build a classifier which has practically the same performance as the optimal one, built using vectors from the margin. This suggests that new criteria for building good classifiers should be considered. Beyond this, combining classifiers that are related to different parts of the training set allows us to endow the latter with a meaningful structure and interpretation. For instance, it may be split into three subsets (Figure 1):

- 1) the most distinctive (contrasting, obvious) members of classes,
- 2) typical members, located in the central areas of the classes,
- 3) in-margin or margin-adjacent members (the most challenging for classification).

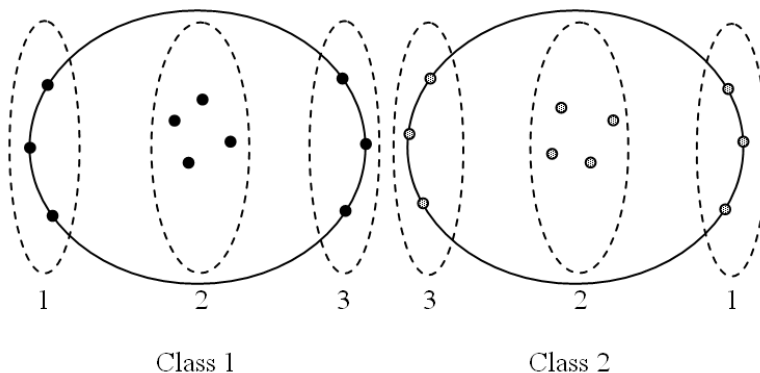


Figure 1: Contrast between distinctive (1), typical (2), and margin-adjacent (3) members of two classes.

This is particularly important in situations when a classifier is needed not to classify new observations, but rather to analyze collected data and, for instance, to evaluate features used to describe observations.

This paper is organized as follows. In Section 2, we review the formulations of QP and LP problems that are at the core of the optimal and Vapnik’s heuristic SVM classifiers. In Section 3, we describe the specific data models which were used to test and compare SVM classifiers. Section 4 describes our approach to the comparison of two classifiers. Section 5 summarizes the results of our experiments with the classifiers: their performance on different data, and comparison of their principal features. In section 6, we describe additional experiments with heuristic SVM which give us more insight into geometrical properties of its support vectors. In Section 7, we state the conclusions that can be drawn from our work.

2 Quadratic and Linear Programming in SVM Learning

Classification problems involving two classes can be formulated in the following way. We are given a training set of l objects x_i whose classification labels y_i are known: $(x_1, y_1), (x_2, y_2), \dots, (x_l, y_l)$. Objects x_i are n -dimensional numerical vectors, $x_i = (x_i^1, x_i^2, \dots, x_i^n) \in R^n$, class labels y_i are taken from set $\{-1, 1\}$, $i = 1, 2, \dots, l$. Using the given training set, we want to find a decision function $f : R^n \rightarrow \{-1, 1\}$, that will correctly classify any vector from R^n .

A SVM classifier (with a linear kernel) uses a decision function of the form

$$f(x) = \text{sgn}((w, x) + b), \quad (2.1)$$

where $w \in R^n$, $b \in R$, and (w, x) is the inner product of vectors w and x . Geometrically, this decision function corresponds to a hyperplane defined by equation $(w, x) + b = 0$, which separates R^n into two half-spaces. It then assigns label 1 to all vectors from one of the half-spaces, and label -1 to all vectors from the other half-space.

The optimal SVM classifier uses a decision function of the form (2.1), where parameters $w = (w_1, w_2, \dots, w_n)$ and b are obtained as the optimal solution of the following QP problem:

$$\frac{1}{2} \sum_{k=1}^n w_k^2 + C \sum_{i=1}^l \xi_i \rightarrow \min, \quad (2.2)$$

subject to

$$y_i \left(\sum_{k=1}^n w_k x_i^k + b \right) \geq 1 - \xi_i, \quad i = 1, 2, \dots, l, \quad (2.3)$$

$$\xi_i \geq 0, \quad i = 1, 2, \dots, l, \quad (2.4)$$

where $C > 0$ is a characteristic parameter of the classifier. The set of constraints (2.3) implies that the decision function (2.1) should classify correctly all vectors from the given training set, up to some admissible errors (called slack variables). Optimization criterion (2.2) means we want to maximize the margin between the two classes and minimize the overall error on the training vectors.

Standard practice for obtaining a solution for this problem is to solve its dual problem, formulated by introducing a Lagrangian and written in terms of dual variables $\alpha_1, \alpha_2, \dots, \alpha_l$:

$$\sum_{i=1}^l \alpha_i - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l y_i y_j \alpha_i \alpha_j (x_i, x_j) \rightarrow \max, \quad (2.5)$$

subject to

$$\begin{aligned} \alpha_i &\geq 0, \quad i = 1, 2, \dots, l, \\ \alpha_i &\leq C, \quad i = 1, 2, \dots, l, \\ \sum_{i=1}^l \alpha_i y_i &= 0. \end{aligned} \quad (2.6)$$

If a set $\alpha_1^*, \alpha_2^*, \dots, \alpha_l^*$ is the optimal solution of dual problem (2.5)-(2.6), then the optimal solution w, b of primary problem (2.2)-(2.4) may be found according to the following equations:

$$w = \sum_{i=1}^l \alpha_i^* y_i x_i, \quad (2.7)$$

$$b = y_t - (w, x_t), \quad (2.8)$$

where t is an index of arbitrary $\alpha_t^* < C$. Decision function (2.1) can thus be written as

$$f(x) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i^*(x_i, x) + b\right). \quad (2.9)$$

Expansion (2.7) has an important geometrical interpretation. In practical applications, most of the α_i^* are generally equal to zero, which means that the vector w (and parameter b) is defined by a small number of vectors from the training set. These training vectors are called support vectors. Geometrically, they lie on the margins between the two classes, and in SVM model they are the only training examples important in designing an optimal classifier.

In [12], Vapnik formulated an alternative problem to find parameters w, b of the decision function (2.1), aimed at minimizing the number of its support vectors:

$$\sum_{i=1}^l \alpha_i + C \sum_{i=1}^l \xi_i \rightarrow \min, \quad (2.10)$$

subject to

$$\begin{aligned} y_i \left(\sum_{j=1}^l \alpha_j y_j (x_j, x_i) + b \right) &\geq 1 - \xi_i, \quad i = 1, 2, \dots, l, \\ \alpha_i &\geq 0, \quad i = 1, 2, \dots, l, \\ \xi_i &\geq 0, \quad i = 1, 2, \dots, l. \end{aligned} \quad (2.11)$$

In this problem, the optimization variables are $\alpha_1, \alpha_2, \dots, \alpha_l, \xi_1, \xi_2, \dots, \xi_l, b$. When the optimal solution of (2.10)-(2.11) is found, equation (2.7) is used to calculate the vector w . The decision function is still of the form (2.9). Similar to a quadratic SVM, the training vectors x_i with nonzero coefficients α_i^* in expansion (2.7) are called support vectors.

In contrast to the optimal SVM classifier, statistical properties of the classifier obtained from the solution of problem (2.10)-(2.11) (VC-bounds, generalization performance estimates) are not known. This is why this classifier is called heuristic SVM. In the present study we compared the generalization performance of heuristic and optimal classifiers experimentally.

Let us note that the heuristic SVM can be used with kernels, too. That is, if the decision function

$$f(x) = \text{sgn}\left(\sum_{i=1}^l y_i \alpha_i^* k(x_i, x) + b\right)$$

is used instead of (2.1), where $k(\cdot, \cdot)$ is a kernel function, then the generalized heuristic classifier may be obtained by solving the same LP problem (2.10)-(2.11) where in the first set of constraints the $k(x_j, x_i)$ term is used instead of (x_j, x_i) .

We should also note that many alternative LP formulations for the linear separation of two classes have been proposed in the literature (see, for example, [1, 6, 7, 8, 10, 4, 5]). In particular, some works compare SVMs based on different linear and quadratic optimization problems [13, 9, 11, 2, 14]. In the Introduction we briefly noted the main differences between these earlier approaches and the present study. A detailed discussion of other works lies outside the scope of this paper.

3 Data Description

We compared the optimal and heuristic classifiers in a series of experiments with different synthetic data. These data were deliberately endowed with simple geometrical structure to allow easy analysis and clear interpretation of their classification. Different parameters of the data (dimensionality, degree of separability, boundaries of the classes, etc.) have been varied to obtain a range of classification situations wide enough to formulate convincing conclusions about the classifiers being tested.

We generated two classes of points in n -dimensional Euclidian space using two slightly different data models. In both models, a mixture of three n -variable Gaussian distributions (with standard deviation σ for every variable) was used to generate each class². In data model I, the means of the Gaussian distributions are points A_1, A_2 and A_3 for the first class, and B_1, B_2 and B_3 for the second class (Figure 2). These points lie on a single plane, symmetrically about the origin O , $\rho(A_1, A_2) = \rho(A_1, A_3) = \rho(B_1, B_2) = \rho(B_1, B_3) = r$, and $\rho(A_1, B_1) = d$, where $\rho(\cdot, \cdot)$ is the Euclidean distance between two points. The angle between the line, passing through points A_1, B_1 , and segments $A_1A_2, A_1A_3, B_1B_2, B_1B_3$, is γ . For the first class, the probability of choosing A_1 as the mean of a Gaussian distribution is p , $0 < p < 1$, while the probability of choosing A_2 (or A_3) as the mean is $\frac{1-p}{2}$. The same holds for the second class. For both classes, $2l$ points are generated using this model, l points in each class. Thus, data model I has six parameters n, d, r, γ, p, l , and parameter σ .

Figure 3 shows a slightly modified geometrical structure which was used as data model II. All parameters of this model are similar to those of model I. These two models produce classes which have different types of boundaries: model I produces classes with more contrast and fewer marginal points, whereas model II produces classes with fewer contrast and more marginal points.

Six series of experiments were carried out for each data model. In each experiment, we chose values for parameters n, d, r, γ, p, l , and generated $2l$ points for a training set (l points in each class) and 2000 points for a testing set (1000 points in each class). Then, we trained, tested and compared heuristic and optimal classifiers. Each series of experiments consisted in varying one of six parameters n, d, r, γ, p, l (Table 1). For example, the first series consisted

²Thus, each distribution has n by n covariance matrix of diagonal form with σ on the diagonal.

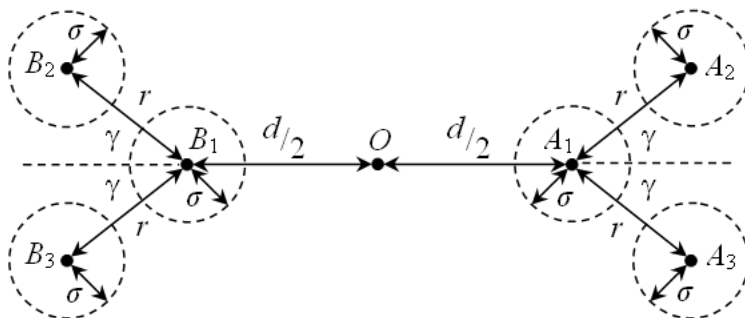


Figure 2: Geometrical structure of data model I.

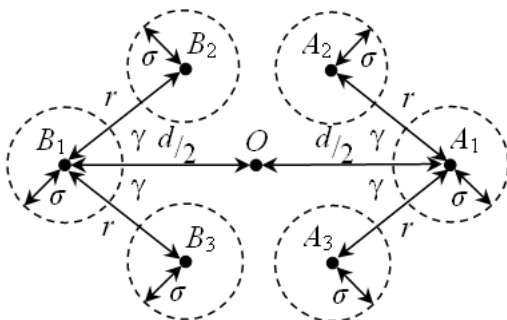


Figure 3: Geometrical structure of data model II.

of eight experiments with n variation. In every series, one parameter was varied according to Table 1, whereas all other parameters were fixed at values from the following set which we called primary: $n = 2$, $d = 5$, $r = 2.5$, $\gamma = 45^\circ$, $p = 0.34$, $l = 100$. The second series consisted of five experiments with d variation. Note that experiment with the underscored value five was not conducted in the second series because this was the experiment with the primary set of parameters that had been already made as the very first one. This remark holds for every series of experiments but the first one.

Thus, a total of 25 experiments were conducted for each data model. We used $\sigma = 2$ in all experiments. Note that for the given values of parameters d , r , γ and σ , the classes are likely to be linearly nonseparable, and their intersection should be larger for data model II. In fact, in experiments 10-13 ($d = 3.5, 3, 2.5, 2$) with data model II, points B_2 and B_3 lie on the right side of points A_2 and A_3 , so in some cases classes have considerable overlap and are very mixed up.

Table 1: Parameter values in the experiments. Values of the primary set are underscored.

Parameter	Brief description	Values	Experiment number
n	Space dimensionality	<u>2</u> 3 4 5 10 20 30 50	1-8
d	Distance between A_1, B_1	<u>5</u> 4 3.5 3 2.5 2	9-13
r	Distance between A_1, A_2	<u>2.5</u> 2 1.5 1	14-16
γ	Half of the angle $A_2A_1A_3$	11.25° 22.5° <u>45°</u>	17-18
p	Probability of choosing $A_1(B_1)$	0.17 <u>0.34</u> 0.5 0.75	19-21
l	Number of training points in each class	<u>100</u> 200 300 400 500	22-25

4 Comparison of SVM Classifiers

Comparative analysis of the classifiers was done on the basis of

- 1) their training and testing performance (accuracy), and their errors on a testing set;
- 2) geometrical similarity between their separating hyperplanes (angles and distances between hyperplanes);
- 3) comparison of their support vectors.

We give more technical details on this analysis in the following subsections, but before that, let us discuss one additional issue. Both optimal and heuristic classifiers have parameter C which should be set before learning (see problems (2.5)–(2.6) and (2.10)–(2.11)). Clearly, this parameter affects properties of the optimal and heuristic classifiers differently, and comparing optimal and heuristic classifiers with the same value of C seems to be inappropriate. Therefore, we decided to compare classifiers that had optimal value of the parameter C in a given range of values, namely, in the set $H = \{10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 1, 10, 10^2, 10^3, 10^4\}$. The optimal value of parameter C was selected using five-fold cross validation. The training set was split into five equal parts, a classifier was trained on four parts, and the number of its correct answers on the remaining part was calculated. This step was repeated four more times for different parts of the training set, and the total number of correct answers was calculated for the given value of C . Value $C^* \in H$ that gave maximum number of correct answers in five-fold cross validation was chosen as the optimal for parameter C .

4.1 Performance of classifiers

In addition to the accuracy of classifiers on training and testing data, we also used recall, precision, and F_1 measures to characterize their performance. Assume that Table 2 describes results of classification of $a + b + c + d$ objects into two classes.

Table 2: Classification results summary.

Predicted classes	True classes	
	1	2
1	a	b
2	c	d

Then, recall, precision, and F_1 measures are defined as follows:

$$\text{recall} = \frac{a}{a+c}, \quad \text{precision} = \frac{a}{a+b}, \quad F_1 = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

The Tanimoto measure was used to evaluate similarity in classifiers’ testing errors. The Tanimoto measure of similarity between two sets M_1 and M_2 is defined as

$$\tau(M_1, M_2) = \frac{|M_1 \cap M_2|}{|M_1 \cup M_2|}.$$

It may be seen that $\tau(M_1, M_2)$ varies from zero to one; it equals to zero when M_1 and M_2 have no common elements, and to one when M_1 and M_2 are the same.

4.2 Coefficients of similarity between two hyperplanes

To evaluate the similarity between separating hyperplanes, we calculated weighted angles between hyperplanes and three quantities characterizing their proximity (distance) based on the training set. Thus, seven different characteristics were used to capture as much useful information about the geometry of the classifiers as possible.

Let π_1, π_2 be the hyperplanes used by two SVM classifiers, and w_1, w_2 are their normal vectors, respectively. Let K_1 is a subset of all points from the training set S_{train} , that belong to the first class, and K_2 is a subset of all points from the training set, that belong to the second class. For two vectors u_1, u_2 from R^n , we denote by $\alpha(u_1, u_2)$ the angle between them, i.e.,

$$\alpha(u_1, u_2) = \arccos \frac{(u_1, u_2)}{\|u_1\| \cdot \|u_2\|}.$$

For vector u and hyperplane π we denote by $u(\pi)$ the orthogonal projection of u onto π .

We calculated the following quantities.

1. The angle between hyperplanes is equal to the angle between their normal vectors:

$$\alpha_1(\pi_1, \pi_2) = \alpha(w_1, w_2). \tag{4.1}$$

2. Angle between hyperplanes divided by angle β , where β is defined for each data configuration as shown in Figure 4:

$$\alpha_2(\pi_1, \pi_2) = \frac{\alpha_1(\pi_1, \pi_2)}{\beta}. \tag{4.2}$$

This quantity takes into account the spread between two classes of points. Note that in experiments 10–13 ($d = 3.5, 3, 2.5, 2$) with data model II angle β is not defined, since points B_2, B_3 lie on the right side of points A_2, A_3 , and so parameter (4.2) is not defined either.

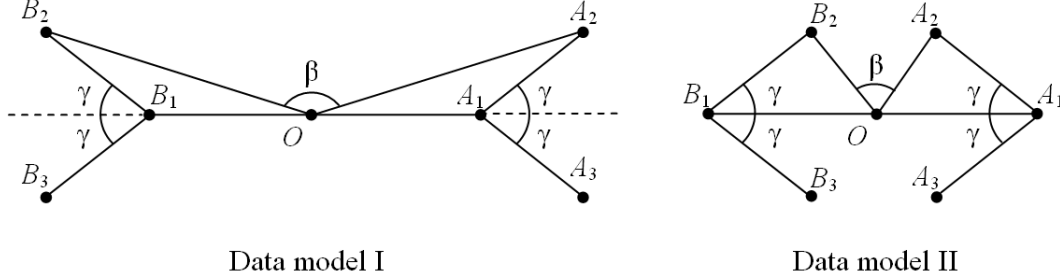


Figure 4: Angle β .

3. The angle between hyperplanes, normalized by the average angle between vectors from the different classes:

$$\alpha_3(\pi_1, \pi_2) = \frac{\alpha_1(\pi_1, \pi_2)}{\frac{1}{|K_1||K_2|} \sum_{u_1 \in K_1} \sum_{u_2 \in K_2} \alpha(u_1, u_2)}. \quad (4.3)$$

4. Let us assume $K_1 = \{a_1, a_2, \dots, a_m\}$, $K_2 = \{b_1, b_2, \dots, b_s\}$, and define sets K_1^* and K_2^* :

$$K_1^* = \{(a_i, a_j) : 1 \leq i < j \leq m\}, \quad K_2^* = \{(b_i, b_j) : 1 \leq i < j \leq s\}.$$

Note that

$$|K_1^*| = |K_1| \frac{|K_1| - 1}{2}, \quad |K_2^*| = |K_2| \frac{|K_2| - 1}{2}.$$

We calculated the angle between hyperplanes using one more normalization factor:

$$\alpha_4(\pi_1, \pi_2) = \frac{\alpha_1(\pi_1, \pi_2)}{|K_1||K_2|} \left(\frac{1}{|K_2| - 1} \sum_{u_3 \in K_1} \sum_{(u_1, u_2) \in K_2^*} \alpha(u_1 - u_3, u_2 - u_3) + \frac{1}{|K_1| - 1} \sum_{u_3 \in K_2} \sum_{(u_1, u_2) \in K_1^*} \alpha(u_1 - u_3, u_2 - u_3) \right). \quad (4.4)$$

5. Average distance between projections of training vectors onto the hyperplanes normalized by the average norm of training vectors:

$$d_1(\pi_1, \pi_2) = \frac{\sum_{u \in S_{train}} \|u(\pi_1) - u(\pi_2)\|}{\sum_{u \in S_{train}} \|u\|}. \quad (4.5)$$

6. Average distance between projections of training vectors onto the hyperplanes normalized by the average distance between vectors from different classes:

$$d_2(\pi_1, \pi_2) = \frac{\frac{1}{|S_{train}|} \sum_{u \in S_{train}} \|u(\pi_1) - u(\pi_2)\|}{\frac{1}{|K_1||K_2|} \sum_{u_1 \in K_1} \sum_{u_2 \in K_2} \|u_1 - u_2\|}. \quad (4.6)$$

7. Average distance between projections of training vectors onto the hyperplanes normalized by the average norm of their projections onto hyperplanes:

$$d_3(\pi_1, \pi_2) = \frac{2}{|S_{train}|} \sum_{u \in S_{train}} \frac{\|u(\pi_1) - u(\pi_2)\|}{\|u(\pi_1)\| + \|u(\pi_2)\|}. \quad (4.7)$$

4.3 Support vectors

For two classifiers, we compared the number of their support vectors, the number of common support vectors, and their spatial location. We used the following approach to analyze spatial location of support vectors. For each vector in a class, we calculated the sum of Euclidean distances (cumulative distance) between this vector and all vectors in the opposite class:

$$D(a_i) = \sum_{j=1}^s \rho(a_i, b_j), \quad i = 1, 2, \dots, m,$$

$$D(b_j) = \sum_{i=1}^m \rho(b_j, a_i), \quad j = 1, 2, \dots, s,$$

where $K_1 = \{a_1, a_2, \dots, a_m\}$ is the first class of vectors, $K_2 = \{b_1, b_2, \dots, b_s\}$ is the second class of vectors, and $\rho(\cdot, \cdot)$ is the Euclidian distance between two vectors (Figure 5). Then, we created two ordered lists of all vectors in each class using calculated values:

$$D(a_{i_1}) \leq D(a_{i_2}) \leq \dots \leq D(a_{i_m}), \quad (4.8)$$

$$D(b_{j_1}) \leq D(b_{j_2}) \leq \dots \leq D(b_{j_s}). \quad (4.9)$$

Take, for example, the first class. We put vector a_{i_1} with the minimum value of $D(a_{i_1})$ (closest to the opposite class) in the first position in list (4.8), vector a_{i_2} with the minimum value of $D(a_{i_2})$ among the rest of the vectors (next closest to the opposite class) – in the second position, and so on, until we put vector a_{i_m} with the maximum value of $D(a_{i_m})$ (most distant from the opposite class) in the last position in the list. The same was done for the second class. Finally, for each class, we calculated minimum, maximum, and average positions of support vectors in lists (4.8), (4.9).

5 Experimental Results

Throughout this section, a heuristic classifier based on the LP problem (2.10)–(2.11) is denoted as SVM-L, and an optimal classifier based on the QP problem (2.5)–(2.6) is denoted as SVM-Q. In our study, SVM-Q classifier was taken from LIBSVM library [3], and SVM-L classifier was written by the authors in C using the MOSEK³ optimization software.

³<http://www.mosek.com>

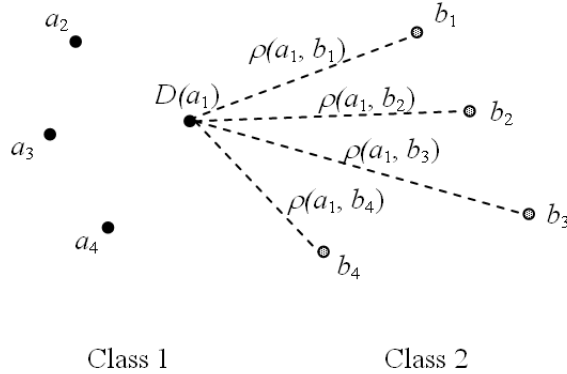


Figure 5: Calculation of cumulative distance between a vector from one class and vectors from the other class.

5.1 Performance of classifiers

As Tables 4 and 5 show, the performance of classifiers is practically identical in all experiments. Let us recall that in experiments 1–21 the size of the training set is 200 vectors, and in experiments 22, 23, 24, and 25 it is 400, 600, 800, and 1000 vectors, respectively (equal number of vectors in each class). The size of the testing set is 2000 vectors in all experiments (1000 vectors in each class). As it was mentioned in Section 3, vectors generated using data model II are more difficult to classify than those generated using data model I, because the intersection of two classes tends to be larger for data model II. This results in lower average performance of both classifiers on data set II.

Table 4 provides Tanimoto coefficient which measures the number of common errors of two classifiers on a testing set. It follows from this table that SVM-L and SVM-Q misclassify very similar subsets of vectors.

Overall, Tables 4 and 5 suggest that both classifiers build very similar separating hyperplanes. This observation will be further substantiated below.

5.2 Separating hyperplanes

Similarity between the separating hyperplanes of SVM-L and SVM-Q, as measured by the geometrical characteristics defined in equations (4.1)–(4.7), is shown in Tables 6 and 7. In experiments with data model I, the average value of the angle between hyperplanes (parameter α_1) is 6.38° , with standard deviation of 7.73° (min and max values are 0.00° and 24.98°). In experiments with data model II, the average value of the angle between hyperplanes is 5.57° , with standard deviation of 6.45° (min and max values are 0.00° and 29.50°). In fact, parameters (4.1)–(4.7) are highly correlated with each other, as Table 3 shows. Therefore, parameter α_1 gives quite a complete picture of similarity between the hyperplanes. It may be seen that the angle between the hyperplanes was rather large in experiments 2–8, where the

dimensionality of vector space was varied. Note that in experiments 10–13 on data model II parameter α_2 is not defined (see remark after equation (4.2)).

In general, Tables 6 and 7 prove that both classifiers have very similar separating hyperplanes.

Table 3: Correlation between parameters (4.1)–(4.7).

	Data I		Data II
	α_1		α_1
α_2	1.00	α_2	0.77
α_3	0.99	α_3	1.00
α_4	0.88	α_4	0.98
d_1	0.99	d_1	1.00
d_2	0.99	d_2	0.99
d_3	0.66	d_3	0.75

5.3 Support vectors

Despite the fact that the hyperplanes of SVM-L and SVM-Q turned out to be very similar in terms of geometry, the number of support vectors differed radically for these classifiers, as Tables 8 and 9 show. Generally, SVM-L had several support vectors (rarely more than three), while SVM-Q – dozens and hundreds of support vectors. Table 8 shows that nearly in a half of experiments (10 out of 25 for data I, and 17 out of 25 for data II) the number of support vectors for SVM-Q is larger than 75% of the training set size (marked in bold in Table 8). Conversely, for SVM-L the number of support vectors varies from 0.1% to 12.5% of the training set size.

In all experiments on data I and II, excluding those where n was varied (experiments 1–8), the number of support vectors for SVM-L is equal to one or two. In experiments 1–8 on data I, it changes from one to 13, and in experiments 1–8 on data II – from one to 25.

It is interesting to review experiments 1, 22–25 with l variation, where the number of support vectors for SVM-Q significantly changes with l growth, while the number of support vectors for SVM-L practically does not change.

As it may be seen in Table 8, in experiment 22 with data I the number of support vectors for SVM-Q is 96% of the 400-vector training set, and is only 9.8% of the 1000-vector training set in experiment 25. This situation may seem to be confusing, but the explanation here is that in these experiments SVM-Q had different values of the parameter C (see Figure 7 in Appendix).

From the last two rows of Table 8 we can see that, due to less linear separability of classes in data model II, both classifiers have larger average number of support vectors in

experiments with data model II.

Overall, Tables 8 and 9 clearly show how efficiently the idea used to build SVM-L works: the heuristic SVM has many fewer support vectors, than the optimal SVM.

Table 10 shows the distribution of support vectors over the two classes for each classifier. While this distribution is almost perfectly symmetric for SVM-Q, SVM-L tends to prefer second class to the first when choosing support vectors. To see whether this asymmetry is attributed to chance, we conducted nine more series of 25 experiments with each data model, and produced the averaged distribution of support vectors in ten series of experiments, which is given in Table 11. We can see from this table that averaged numbers still show a bias in distribution of SVM-L support vectors over the two classes in some experiments (e.g., experiments number 5, 11, 15, and 17, data I), and this bias is not always one-directional (cf. experiments 15 and 17, data I). We used t-test to see whether the average number of support vectors in each class is the same for each experiment. In all experiments except experiments 5, 11, 15, and 17 with data model I, this was proved to be the case with significance level 0.01.

Tables 12 and 13 describe the spatial location of support vectors with respect to the opposite class. As described in subsection 4.3, vectors of the training set were ordered according to their cumulative distance to the vectors of the opposite class. For a set of support vectors, their minimum, maximum, and average positions in this order were calculated, to see where they lay with respect to the inner and outer boundaries of the classes.

Let us go through the first row of Table 12 with some explanations. First part of this row says that in the first experiment SVM-L had one support vector in the second class and no support vectors in the first class. Maximum position of its support vectors from the second class in ordering (4.9) is, therefore, equal to the position of that single support vector, which happened to be 100. Similarly, average and minimum position of support vectors from the second class is equal to 100. Now, in this experiment, there were 200 vectors in the training set, 100 vectors in each class. Thus, among all training vectors of the second class, our single support vector is the one most distant from the opposite class.

The second part of the first row in Table 12 says that SVM-Q had 24 support vectors in each class. In ordering (4.8), support vectors from the first class had maximum position equal to 40, average position equal to 15, and minimum position equal to 1. Thus, among total 100 training vectors of the first class, 24 support vectors of SVM-Q were positioned in the range from 40 to 1, with average position being equal to 15. That is, they were distributed among vectors closest to the opposite class. Similar interpretation is valid for SVM-Q support vectors from the second class.

From Tables 12 and 13 we can see that the average position of SVM-L support vectors is always greater than the average position of SVM-Q support vectors, the maximum position of SVM-L support vectors usually corresponds to the most distant borders of the classes, while the minimum position of SVM-Q corresponds to the most close borders of classes. This shows that support vectors of SVM-L are selected from vectors lying further from the opposite class, while support vectors of SVM-Q are selected from vectors lying closer to the opposite class.

6 Heuristic SVM: Test on Data with an Outlier

To better understand geometrical properties of support vectors used by heuristic SVM, we conducted additional tests of this classifier on simple 2-D data containing an outlier. Using data model shown on Figure 6, we generated two classes of points with Gaussian distribution. In the first class, the mean of the distribution was A_1 , in the second class, it was B_1 . Points A_1 and B_1 lied symmetrically about the origin, at distance d from each other, which varied in two series of experiments. Both distributions had standard deviation $\sigma = 2$. Each class consisted of 100 vectors.

We used this setup to test heuristic SVM in the following way. First, we built a hyperplane separating two generated classes. Then, we substituted one point in the first class with an outlier C_1 at distance $r = 10$ from A_1 (Figure 6), and built a new hyperplane for these data. We continued by rotating the outlier about the center of the first class (point A_1) anticlockwise by angle $\gamma = \pi/8$ at a time, and building a hyperplane for each position of the outlier. Overall, we conducted 8 experiments – one without the outlier, and seven with the outliers C_1 – C_7 . In each experiment, we were interested which vectors would become support vectors of the hyperplane.

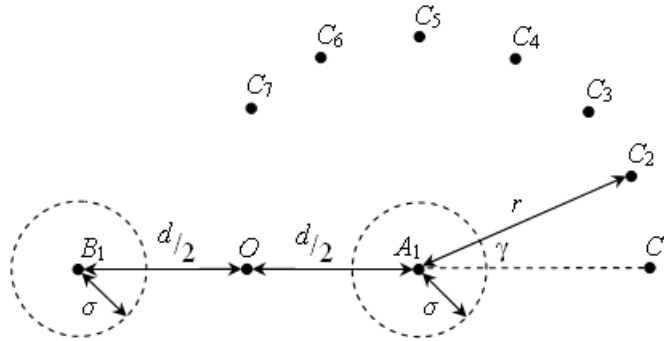


Figure 6: Data model used in experiments with an outlier.

Figures 8 and 9 show results of these experiments when $d = 10$ and two classes are linearly separable, and when $d = 6$ and two classes are overlapped. We can see from these figures that in all experiments support vectors are always those vectors lying at the largest distance from the opposite class. In the first series of experiments (linearly separable classes), support vectors also lie at the largest distance from the margin between two classes. However, it is not the case for the second series of experiments (overlapped classes). This observation gives us an important detail of the geometrical properties of support vectors of the heuristic SVM: these vectors lie at the maximum distance from the opposite class, but not necessarily at the maximum distance from the margin between two classes.

7 Conclusions and Future Work

We have shown that the heuristic and optimal SVMs have almost identical performance on our synthetic training and testing data. This result is similar to the conclusions reported in [13, 9, 11, 2, 14] about the performance of other LP-based SVMs.

The most interesting results of this study come from the geometrical comparison of separating hyperplanes and support vectors used by heuristic and optimal SVMs. We have shown that hyperplanes of these SVMs are very similar in terms of geometry, which is a more fundamental property of these classifiers than their similar performance. We have no theoretical explanation of this observation but we believe it can be found. We also have shown that there are two important differences between support vectors of heuristic and optimal SVMs. These methods differ significantly in the number of their support vectors, and in the location of their support vectors in space. While, as is well known, the optimal SVM selects support vectors from the margin between two classes (margin-adjacent vectors on Figure 1), it turned out that heuristic SVM selects support vectors from the most distant borders of two classes, at the maximum distance from the opposite class (distinctive vectors on Figure 1).

This observation is a very interesting outcome of our study. It shows that most distinctive vectors of the training set can be used to build classifiers almost identical to optimal one which uses marginal vectors. This observation may provide a useful clue to the theoretical analysis of the heuristic SVM. Apparently, some property related to the distance between support vectors from opposite classes may be used to get an upperbound VC-dimension for the heuristic SVM (similar to the margin between two classes which is used to get an upperbound VC-dimension for the optimal SVM, [12]). The fact that the heuristic SVM selects the most distinctive vectors of two classes, and the number of these vectors is small, may also allow practitioners to obtain a more meaningful interpretation and a deeper understanding of the solution of the classification problem at hand.

In our future work, we are interested in getting theoretical insights into the traits of heuristic SVM – both related to its geometrical similarity to optimal SVM, and to the geometrical location of its support vectors in space. We would like to compare heuristic and optimal SVM on real, large benchmark data. We are also interested in testing these classifiers with non-linear kernels and in their application to regression analysis.

References

- [1] K.P. Bennet and O.L. Mangasarian. Robust linear programming discrimination of two linearly inseparable sets. *Optimization methods and software*, 1:23–34, 1992.
- [2] A.B. Chan, N. Vasconcelos, and G. R. G. Lanckriet. Direct convex relaxations of sparse SVM. In *Proceedings of the ACM 24th international conference of machine learning*, volume 227, pages 145–153, 2007.

- [3] C.C. Chang and C.J. Lin Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [4] M. T. Cover. Geometrical and statistical properties of systems of linear inequalities with applications in pattern recognition. *IEEE transactions on electronic computers*, 14:326–334, 1965.
- [5] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern classification*. Wiley-Interscience, 2 edition, 2000.
- [6] F. Glover. Improved linear programming models for discriminant analysis. *Decision sciences*, 21:771–785, 1990.
- [7] R.C. Grinold. Mathematical programming methods of pattern classification. *Management science*, 19:272–289, 1972.
- [8] O.L. Mangasarian. Linear and nonlinear separation of patterns by linear programming. *Operations research*, 13(3):444–452, 1965.
- [9] E. Prankeviciene, R. Somorjai, and M.N. Tran. Feature/model selection by the linear programming SVM combined with state-of-art classifiers: what can we learn about the data. In *Proceedings of the 20th international joint conference on neural networks*, volume 1, pages 1627–1632, 2007.
- [10] F.W. Smith. Pattern classification design by linear programming. *IEEE transactions on computers*, 17(4):367–372, 1968.
- [11] S. Sra. Efficient large scale linear programming support vector machines. *Lecture notes in computer science*, pages 767–774, 2006.
- [12] V. Vapnik. *Statistical learning theory*. John Wiley, 1998.
- [13] Q. Wu and D.X. Zhou. SVM soft margin classifiers: linear programming versus quadratic programming. *Neural computation*, 17(5):1160–1187, 2005.
- [14] W. Zhou, L. Zhang, and L. Jiao. Linear programming support vector machines. *Pattern recognition*, 35(12):2927–2936, 2002.

Appendix

This section contains Tables 4-13 and Figures 7-9.

Table 4: SVM-L and SVM-Q performance in percents of correctly classified objects. The Tanimoto coefficient is given multiplied by 100.

Experiment number	Data I			Data II		
	Training performance SVM-L / SVM-Q	Testing performance SVM-L / SVM-Q	Tanimoto coefficient for testing errors	Training performance SVM-L / SVM-Q	Testing performance SVM-L / SVM-Q	Tanimoto coefficient for testing errors
1	95.0 / 95.0	95.1 / 95.0	92.3	73.5 / 73.5	73.6 / 73.7	97.4
2	95.5 / 95.5	96.3 / 96.3	94.7	75.0 / 75.5	72.3 / 72.3	97.7
3	97.5 / 97.0	95.1 / 94.8	69.7	73.5 / 74.0	71.9 / 71.7	91.0
4	95.0 / 95.0	95.5 / 95.7	72.5	74.0 / 73.0	73.0 / 73.5	78.5
5	95.5 / 95.5	95.3 / 95.3	73.4	76.5 / 78.0	71.3 / 70.6	93.7
6	96.5 / 95.0	93.6 / 94.3	57.8	78.0 / 79.0	71.1 / 71.4	80.8
7	93.5 / 94.0	94.4 / 95.4	55.3	74.5 / 75.0	67.7 / 67.1	97.7
8	96.5 / 96.0	92.3 / 94.2	47.3	78.5 / 80.5	66.8 / 69.3	64.6
9	92.5 / 92.5	92.9 / 92.1	72.4	65.5 / 64.5	66.2 / 65.7	87.4
10	91.5 / 91.5	91.6 / 90.2	64.1	60.0 / 61.5	61.5 / 61.0	91.0
11	90.0 / 90.0	89.1 / 89.4	91.2	53.5 / 53.5	55.4 / 55.7	99.4
12	89.0 / 88.5	86.9 / 86.8	94.8	53.0 / 53.5	49.9 / 49.5	95.2
13	86.5 / 86.5	84.3 / 84.0	90.4	52.5 / 53.0	51.2 / 51.1	94.2
14	95.0 / 95.0	94.5 / 94.5	99.1	75.0 / 77.5	76.6 / 77.5	80.0
15	94.5 / 95.5	94.0 / 93.7	83.6	81.0 / 81.0	80.7 / 80.6	99.7
16	94.0 / 92.0	93.0 / 91.8	60.0	83.0 / 82.5	84.1 / 83.9	88.2
17	96.5 / 95.5	96.3 / 95.7	64.9	64.5 / 65.5	65.0 / 65.0	81.0
18	96.5 / 95.5	96.1 / 96.1	82.6	67.5 / 66.5	67.3 / 67.2	90.3
19	97.5 / 97.5	97.0 / 97.0	100.0	69.5 / 68.5	69.1 / 69.2	97.4
20	95.0 / 95.5	93.8 / 93.9	68.0	77.5 / 78.0	76.5 / 77.0	82.7
21	92.0 / 91.5	91.7 / 91.7	96.4	84.5 / 83.5	82.5 / 82.6	77.9
22	96.3 / 96.3	94.9 / 95.3	86.8	77.0 / 76.3	73.3 / 73.1	96.2
23	96.3 / 96.3	94.9 / 95.1	68.9	73.0 / 73.2	73.1 / 73.6	93.3
24	95.4 / 95.5	95.5 / 95.3	78.8	72.1 / 72.5	73.5 / 73.9	94.8
25	95.7 / 95.7	95.4 / 95.4	83.2	72.8 / 73.0	73.7 / 73.6	95.0
Average	94.4 / 94.2	93.6 / 93.5	77.9	71.4 / 71.7	69.9 / 70.0	89.8
Standard deviation	2.7 / 2.7	3.0 / 3.1	15.0	8.9 / 8.9	8.5 / 8.6	8.7

Table 5: Recall, precision, F_1 for testing data.

Experiment number	Data I			Data II		
	Recall	Precision	F_1	Recall	Precision	F_1
	SVM-L / SVM-Q	SVM-L / SVM-Q	SVM-L / SVM-Q	SVM-L / SVM-Q	SVM-L / SVM-Q	SVM-L / SVM-Q
1	0.94 / 0.94	0.96 / 0.96	0.95 / 0.95	0.73 / 0.73	0.75 / 0.74	0.74 / 0.74
2	0.97 / 0.97	0.96 / 0.96	0.96 / 0.96	0.72 / 0.71	0.74 / 0.74	0.73 / 0.73
3	0.95 / 0.93	0.96 / 0.97	0.95 / 0.95	0.69 / 0.69	0.79 / 0.79	0.74 / 0.74
4	0.95 / 0.95	0.96 / 0.96	0.96 / 0.96	0.71 / 0.73	0.77 / 0.75	0.74 / 0.74
5	0.95 / 0.95	0.96 / 0.95	0.95 / 0.95	0.71 / 0.70	0.71 / 0.71	0.71 / 0.71
6	0.93 / 0.94	0.94 / 0.95	0.94 / 0.94	0.71 / 0.70	0.72 / 0.74	0.71 / 0.72
7	0.95 / 0.97	0.94 / 0.94	0.94 / 0.95	0.69 / 0.68	0.65 / 0.64	0.67 / 0.66
8	0.93 / 0.92	0.92 / 0.97	0.92 / 0.94	0.68 / 0.71	0.63 / 0.64	0.65 / 0.68
9	0.92 / 0.89	0.94 / 0.96	0.93 / 0.92	0.65 / 0.64	0.69 / 0.73	0.67 / 0.68
10	0.91 / 0.87	0.92 / 0.95	0.92 / 0.91	0.61 / 0.60	0.64 / 0.65	0.62 / 0.62
11	0.87 / 0.88	0.92 / 0.91	0.89 / 0.90	0.55 / 0.55	0.60 / 0.60	0.57 / 0.58
12	0.87 / 0.87	0.87 / 0.87	0.87 / 0.87	0.50 / 0.49	0.55 / 0.53	0.52 / 0.51
13	0.83 / 0.84	0.86 / 0.84	0.84 / 0.84	0.51 / 0.51	0.54 / 0.51	0.52 / 0.51
14	0.94 / 0.94	0.95 / 0.95	0.95 / 0.94	0.75 / 0.78	0.80 / 0.76	0.77 / 0.77
15	0.93 / 0.94	0.95 / 0.93	0.94 / 0.94	0.81 / 0.80	0.81 / 0.81	0.81 / 0.81
16	0.92 / 0.88	0.94 / 0.96	0.93 / 0.92	0.83 / 0.83	0.85 / 0.85	0.84 / 0.84
17	0.96 / 0.94	0.97 / 0.98	0.96 / 0.96	0.64 / 0.66	0.70 / 0.63	0.67 / 0.64
18	0.95 / 0.96	0.97 / 0.96	0.96 / 0.96	0.65 / 0.66	0.73 / 0.70	0.69 / 0.68
19	0.98 / 0.98	0.96 / 0.96	0.97 / 0.97	0.69 / 0.69	0.71 / 0.71	0.70 / 0.70
20	0.94 / 0.95	0.94 / 0.93	0.94 / 0.94	0.76 / 0.76	0.76 / 0.79	0.76 / 0.77
21	0.91 / 0.91	0.92 / 0.92	0.92 / 0.92	0.82 / 0.82	0.84 / 0.84	0.83 / 0.83
22	0.94 / 0.94	0.96 / 0.96	0.95 / 0.95	0.72 / 0.72	0.76 / 0.77	0.74 / 0.74
23	0.95 / 0.94	0.95 / 0.96	0.95 / 0.95	0.72 / 0.73	0.76 / 0.75	0.74 / 0.74
24	0.96 / 0.95	0.95 / 0.96	0.95 / 0.95	0.72 / 0.73	0.76 / 0.76	0.74 / 0.74
25	0.95 / 0.96	0.95 / 0.95	0.95 / 0.95	0.73 / 0.73	0.76 / 0.75	0.74 / 0.74
Average	0.93 / 0.93	0.94 / 0.94	0.94 / 0.94	0.69 / 0.69	0.72 / 0.72	0.71 / 0.70
Standard deviation	0.03 / 0.04	0.03 / 0.03	0.03 / 0.03	0.08 / 0.09	0.08 / 0.09	0.08 / 0.08

Table 6: Angles between hyperplanes (in degrees).

Experiment number	α_1		$100 \cdot \alpha_2$		α_3		α_4	
	Data I	Data II	Data I	Data II	Data I	Data II	Data I	Data II
1	1.69	1.01	1.25	2.24	0.01	0.01	0.07	0.02
2	1.05	0.68	0.78	1.51	0.01	0.01	0.04	0.01
3	8.86	5.31	6.56	11.80	0.07	0.06	0.27	0.10
4	9.73	13.79	7.21	30.64	0.09	0.15	0.25	0.25
5	9.26	3.49	6.86	7.76	0.09	0.04	0.20	0.06
6	23.90	14.21	17.70	31.58	0.25	0.16	0.46	0.24
7	24.98	1.47	18.50	3.27	0.26	0.02	0.45	0.02
8	24.91	29.50	18.45	65.56	0.27	0.32	0.45	0.50
9	2.01	5.29	1.55	35.34	0.02	0.06	0.08	0.09
10	1.87	5.00	1.48	n/a	0.02	0.05	0.06	0.09
11	1.00	0.00	0.81	n/a	0.01	0.00	0.03	0.00
12	1.12	3.70	0.94	n/a	0.01	0.04	0.03	0.06
13	1.73	2.99	1.51	n/a	0.02	0.03	0.05	0.05
14	0.70	3.23	0.50	4.30	0.01	0.03	0.03	0.07
15	0.81	0.31	0.55	0.29	0.01	0.00	0.03	0.01
16	3.64	4.86	2.35	3.55	0.03	0.04	0.15	0.13
17	3.83	8.38	2.27	74.49	0.03	0.09	0.21	0.15
18	4.22	4.52	2.68	20.09	0.03	0.05	0.22	0.08
19	0.00	1.19	0.00	2.64	0.00	0.01	0.00	0.02
20	10.44	9.43	7.73	20.96	0.08	0.09	0.45	0.21
21	1.58	12.51	1.17	27.80	0.01	0.11	0.06	0.33
22	1.36	1.50	1.01	3.33	0.01	0.01	0.06	0.03
23	12.95	1.72	9.59	3.82	0.10	0.02	0.56	0.03
24	3.00	1.90	2.22	4.22	0.02	0.02	0.13	0.04
25	4.85	3.37	3.59	7.49	0.04	0.03	0.21	0.07
Average	6.38	5.57	4.69	17.27	0.06	0.06	0.18	0.11
Standard deviation	7.73	6.45	5.73	20.90	0.08	0.07	0.17	0.12

Table 7: Distances between hyperplanes.

Experiment number	d_1		d_2		d_3	
	Data I	Data II	Data I	Data II	Data I	Data II
1	0.02	0.02	0.01	0.01	0.05	0.03
2	0.01	0.01	0.01	0.01	0.02	0.01
3	0.09	0.09	0.05	0.06	0.08	0.09
4	0.12	0.18	0.07	0.12	0.10	0.17
5	0.11	0.05	0.07	0.03	0.07	0.04
6	0.30	0.21	0.20	0.15	0.18	0.16
7	0.33	0.02	0.22	0.01	0.20	0.02
8	0.35	0.40	0.24	0.28	0.18	0.28
9	0.02	0.09	0.01	0.06	0.07	0.17
10	0.02	0.08	0.01	0.06	0.07	0.13
11	0.01	0.00	0.01	0.00	0.03	0.00
12	0.02	0.06	0.01	0.04	0.04	0.08
13	0.02	0.05	0.01	0.03	0.06	0.05
14	0.01	0.05	0.00	0.03	0.03	0.10
15	0.01	0.00	0.00	0.00	0.05	0.01
16	0.04	0.06	0.02	0.04	0.16	0.17
17	0.03	0.12	0.02	0.08	0.11	0.26
18	0.04	0.07	0.02	0.04	0.14	0.14
19	0.00	0.02	0.00	0.01	0.00	0.04
20	0.11	0.13	0.06	0.08	0.32	0.29
21	0.02	0.16	0.01	0.10	0.05	0.35
22	0.01	0.02	0.01	0.02	0.06	0.06
23	0.15	0.03	0.08	0.02	0.30	0.06
24	0.03	0.03	0.02	0.02	0.12	0.06
25	0.05	0.05	0.03	0.04	0.15	0.12
Average	0.08	0.08	0.05	0.05	0.11	0.11
Standard deviation	0.10	0.09	0.07	0.06	0.08	0.10

Table 8: Number of support vectors for SVM-L and SVM-Q in percents of the training set size. Numbers larger than 75% are in bold.

Experiment number	Size of the training set	Data I		Data II	
		Number of support vectors for SVM-L (% of training set)	Number of support vectors for SVM-Q (% of training set)	Number of support vectors for SVM-L (% of training set)	Number of support vectors for SVM-Q (% of training set)
1	200	0.5	24.0	0.5	63.0
2	200	1.5	11.5	1.5	61.5
3	200	1.5	100.0	1.5	93.0
4	200	2.0	29.5	1.0	73.0
5	200	1.0	58.5	5.0	57.0
6	200	6.5	62.0	7.5	97.0
7	200	4.0	100.0	12.5	68.0
8	200	4.5	100.0	4.0	100.0
9	200	0.5	100.0	1.0	83.5
10	200	0.5	100.0	0.5	89.5
11	200	1.0	39.0	1.0	92.0
12	200	0.5	79.0	0.5	100.0
13	200	0.5	83.0	1.0	100.0
14	200	0.5	59.0	0.5	100.0
15	200	0.5	14.0	1.0	100.0
16	200	0.5	100.0	0.5	84.0
17	200	0.5	100.0	1.0	100.0
18	200	0.5	52.0	1.0	81.0
19	200	1.0	6.5	0.5	77.5
20	200	0.5	13.5	0.5	54.5
21	200	1.0	68.0	0.5	83.0
22	400	0.5	96.0	0.3	58.8
23	600	0.2	37.5	0.3	82.7
24	800	0.3	34.9	0.1	80.8
25	1000	0.2	9.8	0.2	62.9
	Average	1.2	59.1	1.8	81.7
	Standard deviation	1.5	35.0	2.8	15.6

Table 9: Number of common support vectors for SVM-L and SVM-Q. Numbers larger than 75% of the training set size are in bold.

Experiment number	Size of the training set	Data I			Data II		
		Number of support vectors			Number of support vectors		
		SVM-L /	SVM-Q /	common	SVM-L /	SVM-Q /	common
1	200	1 /	48 /	-	1 /	126 /	-
2	200	3 /	23 /	-	3 /	123 /	-
3	200	3 /	200 /	3	3 /	186 /	-
4	200	4 /	59 /	-	2 /	146 /	-
5	200	2 /	117 /	-	10 /	114 /	-
6	200	13 /	124 /	-	15 /	194 /	4
7	200	8 /	200 /	4	25 /	136 /	2
8	200	9 /	200 /	1	8 /	200 /	2
9	200	1 /	200 /	1	2 /	167 /	-
10	200	1 /	200 /	1	1 /	179 /	-
11	200	2 /	78 /	-	2 /	184 /	-
12	200	1 /	158 /	-	1 /	200 /	1
13	200	1 /	166 /	-	2 /	200 /	2
14	200	1 /	118 /	-	1 /	200 /	1
15	200	1 /	28 /	-	2 /	200 /	2
16	200	1 /	200 /	-	1 /	168 /	-
17	200	1 /	200 /	1	2 /	200 /	2
18	200	1 /	104 /	-	2 /	162 /	-
19	200	2 /	13 /	-	1 /	155 /	-
20	200	1 /	27 /	-	1 /	109 /	-
21	200	2 /	136 /	-	1 /	166 /	-
22	400	2 /	384 /	-	1 /	235 /	-
23	600	1 /	225 /	-	2 /	496 /	-
24	800	2 /	279 /	-	1 /	646 /	-
25	1000	2 /	98 /	-	2 /	629 /	-
	Average	2.6 / 143.4 / 1.8			3.7 / 220.8 / 2.0		
	Standard deviation	3.0 / 89.1 / 1.3			5.6 / 144.7 / 0.9		

Table 10: Distribution of support vectors of SVM-L and SVM-Q over classes.

Experiment number	Size of the training set	Data I		Data II	
		Number of support vectors for SVM-L Class 1 / Class 2	Number of support vectors for SVM-Q Class 1 / Class 2	Number of support vectors for SVM-L Class 1 / Class 2	Number of support vectors for SVM-Q Class 1 / Class 2
1	200	- / 1	24 / 24	1 / -	63 / 63
2	200	1 / 2	12 / 11	- / 3	61 / 62
3	200	- / 3	100 / 100	1 / 2	93 / 93
4	200	- / 4	30 / 29	1 / 1	74 / 72
5	200	- / 2	59 / 58	5 / 5	57 / 57
6	200	3 / 10	61 / 63	6 / 9	97 / 97
7	200	5 / 3	100 / 100	14 / 11	70 / 66
8	200	2 / 7	100 / 100	3 / 5	100 / 100
9	200	- / 1	100 / 100	1 / 1	84 / 83
10	200	- / 1	100 / 100	1 / -	89 / 90
11	200	1 / 1	39 / 39	1 / 1	92 / 92
12	200	- / 1	79 / 79	- / 1	100 / 100
13	200	- / 1	83 / 83	- / 2	100 / 100
14	200	- / 1	59 / 59	1 / -	100 / 100
15	200	- / 1	14 / 14	1 / 1	100 / 100
16	200	- / 1	100 / 100	- / 1	84 / 84
17	200	- / 1	100 / 100	1 / 1	100 / 100
18	200	- / 1	52 / 52	1 / 1	81 / 81
19	200	- / 2	7 / 6	- / 1	78 / 77
20	200	1 / -	14 / 13	- / 1	55 / 54
21	200	1 / 1	68 / 68	- / 1	83 / 83
22	400	- / 2	192 / 192	- / 1	118 / 117
23	600	1 / -	112 / 113	- / 2	248 / 248
24	800	- / 2	139 / 140	- / 1	323 / 323
25	1000	- / 2	49 / 49	- / 2	314 / 315
	Average	1.9 / 2.2	71.7 / 71.7	2.7 / 2.5	110.6 / 110.3
	Standard deviation	1.5 / 2.2	44.4 / 44.7	3.6 / 2.7	72.2 / 72.5

Table 11: Averaged distribution of support vectors of SVM-L and SVM-Q over classes in 10 series of 25 experiments.

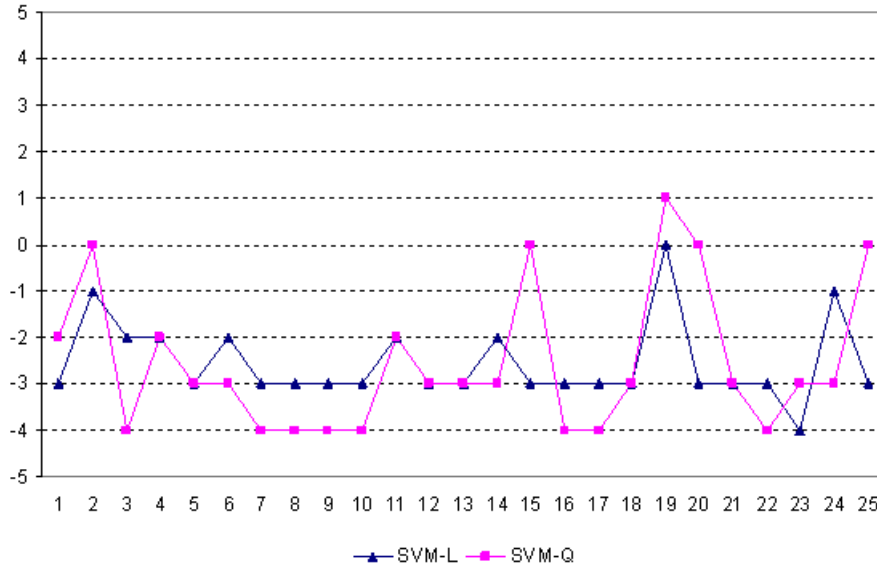
Experiment number	Size of the training set	Data I		Data II	
		Average number of support vectors for SVM-L	Average number of support vectors for SVM-Q	Average number of support vectors for SVM-L	Average number of support vectors for SVM-Q
		Class 1 / Class 2	Class 1 / Class 2	Class 1 / Class 2	Class 1 / Class 2
1	200	0.9 / 0.9	62.5 / 62.5	1.3 / 0.7	84.3 / 84.3
2	200	1.4 / 0.9	32.4 / 32.5	1.4 / 1.3	75.2 / 75.3
3	200	1.4 / 1.7	59.0 / 58.8	1.5 / 1.7	79.9 / 79.7
4	200	1.0 / 1.9	41.8 / 41.0	2.0 / 2.3	85.0 / 84.4
5	200	4.4 / 1.7	61.7 / 61.6	3.3 / 3.9	80.8 / 80.4
6	200	5.3 / 5.6	71.9 / 72.0	5.0 / 6.9	82.8 / 82.9
7	200	5.8 / 5.1	75.7 / 76.5	7.1 / 7.5	90.5 / 90.0
8	200	6.8 / 7.2	81.2 / 80.8	17.4 / 16.5	87.1 / 86.6
9	200	1.0 / 0.7	79.9 / 79.9	0.9 / 0.8	86.3 / 86.3
10	200	1.0 / 0.7	58.4 / 58.6	1.2 / 0.7	92.2 / 92.3
11	200	1.3 / 0.6	63.0 / 62.7	0.6 / 1.0	94.8 / 94.7
12	200	0.8 / 1.0	62.8 / 62.8	1.3 / 0.9	99.0 / 98.8
13	200	0.5 / 1.2	60.7 / 60.7	0.9 / 0.8	98.5 / 98.3
14	200	0.7 / 1.0	64.7 / 64.6	1.0 / 0.7	80.5 / 80.4
15	200	1.3 / 0.3	62.7 / 62.6	0.6 / 1.2	74.3 / 74.3
16	200	0.8 / 0.6	77.2 / 77.1	0.5 / 1.0	87.0 / 86.9
17	200	0.4 / 1.3	51.2 / 51.2	1.1 / 0.8	86.6 / 86.6
18	200	0.8 / 0.5	73.0 / 73.0	1.2 / 0.9	81.7 / 81.8
19	200	0.9 / 1.0	39.8 / 39.8	0.9 / 0.6	88.7 / 88.5
20	200	0.6 / 0.9	39.5 / 39.4	0.8 / 0.7	64.5 / 64.4
21	200	1.0 / 0.7	77.9 / 77.9	0.7 / 0.7	65.4 / 65.3
22	400	0.9 / 1.1	73.9 / 74.1	0.9 / 1.0	128.2 / 128.1
23	600	0.8 / 0.9	158.2 / 158.2	0.7 / 1.2	238.9 / 238.9
24	800	0.9 / 0.9	150.5 / 150.4	0.9 / 0.8	307.0 / 306.9
25	1000	1.0 / 0.9	151.4 / 151.7	0.9 / 1.1	373.1 / 373.4
	Average	1.7 / 1.6	73.2 / 73.2	2.2 / 2.2	112.5 / 112.4
	Standard deviation	1.6 / 2.0	37.4 / 37.6	3.3 / 3.1	63.9 / 64.1

Table 12: Location of support vectors (SV) with respect to the opposite class. Data I.

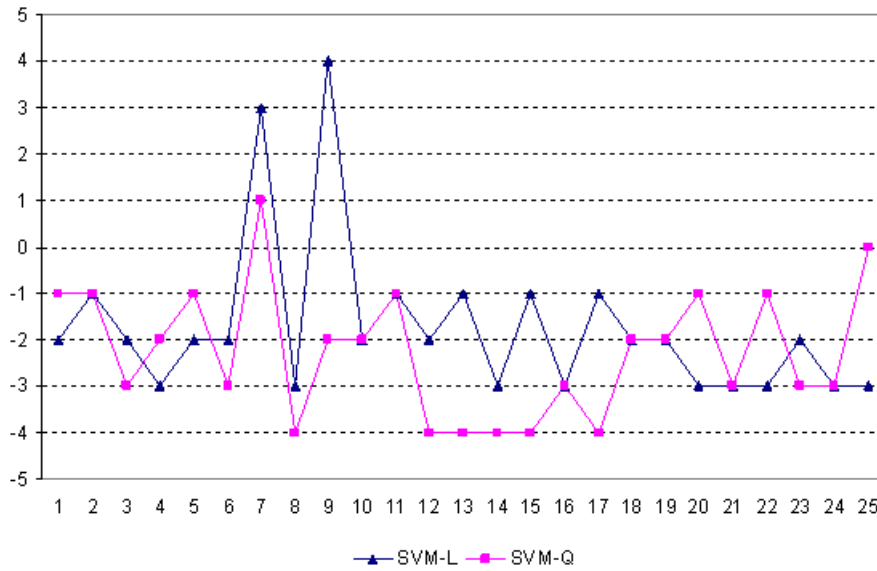
Experiment number / Size of training set	Number of SV	SVM-L			Number of SV	SVM-Q		
		Support vectors max Class1 / Class2	position average	min		Support vector max Class1 / Class2	position average	min
1 / 200	- / 1	- / 100	- / 100	- / 100	24 / 24	40 / 32	15 / 13	1 / 1
2 / 200	1 / 2	99 / 99	99 / 96	99 / 92	12 / 11	14 / 25	7 / 8	1 / 1
3 / 200	- / 3	- / 100	- / 99	- / 98	100 / 100	100 / 100	51 / 51	1 / 1
4 / 200	- / 4	- / 100	- / 94	- / 81	30 / 29	67 / 54	20 / 18	1 / 1
5 / 200	- / 2	- / 98	- / 93	- / 88	59 / 58	87 / 83	33 / 32	1 / 1
6 / 200	3 / 10	94 / 100	91 / 87	89 / 69	61 / 63	91 / 96	35 / 36	1 / 1
7 / 200	5 / 3	94 / 92	92 / 84	89 / 75	100 / 100	100 / 100	51 / 51	1 / 1
8 / 200	2 / 7	96 / 100	89 / 85	81 / 61	100 / 100	100 / 100	51 / 51	1 / 1
9 / 200	- / 1	- / 100	- / 100	- / 100	100 / 100	100 / 100	51 / 51	1 / 1
10 / 200	- / 1	- / 100	- / 100	- / 100	100 / 100	100 / 100	51 / 51	1 / 1
11 / 200	1 / 1	99 / 100	99 / 100	99 / 100	39 / 39	44 / 72	20 / 22	1 / 1
12 / 200	- / 1	- / 100	- / 100	- / 100	79 / 79	83 / 94	40 / 41	1 / 1
13 / 200	- / 1	- / 100	- / 100	- / 100	83 / 83	91 / 94	42 / 42	1 / 1
14 / 200	- / 1	- / 100	- / 100	- / 100	59 / 59	77 / 63	31 / 30	1 / 1
15 / 200	- / 1	- / 100	- / 100	- / 100	14 / 14	25 / 33	10 / 9	1 / 1
16 / 200	- / 1	- / 100	- / 100	- / 100	100 / 100	100 / 100	51 / 51	1 / 1
17 / 200	- / 1	- / 100	- / 100	- / 100	100 / 100	100 / 100	51 / 51	1 / 1
18 / 200	- / 1	- / 100	- / 100	- / 100	52 / 52	76 / 67	27 / 27	1 / 1
19 / 200	- / 2	- / 100	- / 100	- / 99	7 / 6	8 / 7	4 / 4	1 / 1
20 / 200	1 / -	100 / -	100 / -	100 / -	14 / 13	43 / 32	10 / 9	1 / 1
21 / 200	1 / 1	100 / 100	100 / 100	100 / 100	68 / 68	73 / 84	35 / 35	1 / 1
22 / 400	- / 2	- / 200	- / 200	- / 199	192 / 192	196 / 197	97 / 97	1 / 1
23 / 600	1 / -	300 / -	300 / -	300 / -	112 / 113	156 / 224	58 / 63	1 / 1
24 / 800	- / 2	- / 400	- / 400	- / 399	139 / 140	229 / 329	73 / 79	1 / 1
25 / 1000	- / 2	- / 500	- / 500	- / 499	49 / 49	171 / 119	35 / 30	1 / 1

Table 13: Location of support vectors (SV) with respect to the opposite class. Data II.

Experiment number / Size of training set	Number of SV	SVM-L			Number of SV	SVM-Q		
		Support vector position max Class1 / Class2	average	min		Support vectors position max Class1 / Class2	average	min
1 / 200	1 / -	100 / -	100 / -	100 / -	63 / 63	97 / 91	35 / 34	1 / 1
2 / 200	- / 3	- / 99	- / 97	- / 93	61 / 62	97 / 92	37 / 36	1 / 1
3 / 200	1 / 2	96 / 100	96 / 100	96 / 99	93 / 93	99 / 98	47 / 47	1 / 1
4 / 200	1 / 1	100 / 100	100 / 100	100 / 100	74 / 72	93 / 98	41 / 42	1 / 1
5 / 200	5 / 5	100 / 97	85 / 95	73 / 93	57 / 57	98 / 100	37 / 39	1 / 1
6 / 200	6 / 9	87 / 97	48 / 73	22 / 25	97 / 97	100 / 100	50 / 49	1 / 1
7 / 200	14 / 11	94 / 95	61 / 64	5 / 2	70 / 66	100 / 100	47 / 47	1 / 1
8 / 200	3 / 5	92 / 99	59 / 87	42 / 64	100 / 100	100 / 100	51 / 51	1 / 1
9 / 200	1 / 1	100 / 99	100 / 99	100 / 99	84 / 83	98 / 94	44 / 42	1 / 1
10 / 200	1 / -	100 / -	100 / -	100 / -	89 / 90	98 / 100	46 / 46	1 / 1
11 / 200	1 / 1	100 / 97	100 / 97	100 / 97	92 / 92	99 / 100	47 / 47	1 / 1
12 / 200	- / 1	- / 99	- / 99	- / 99	100 / 100	100 / 100	51 / 51	1 / 1
13 / 200	- / 2	- / 100	- / 100	- / 99	100 / 100	100 / 100	51 / 51	1 / 1
14 / 200	1 / -	100 / -	100 / -	100 / -	100 / 100	100 / 100	51 / 51	1 / 1
15 / 200	1 / 1	100 / 100	100 / 100	100 / 100	100 / 100	100 / 100	51 / 51	1 / 1
16 / 200	- / 1	- / 100	- / 100	- / 100	84 / 84	95 / 84	43 / 43	1 / 1
17 / 200	1 / 1	100 / 96	100 / 96	100 / 96	100 / 100	100 / 100	51 / 51	1 / 1
18 / 200	1 / 1	100 / 96	100 / 96	100 / 96	81 / 81	93 / 86	42 / 41	1 / 1
19 / 200	- / 1	- / 99	- / 99	- / 99	78 / 77	100 / 96	42 / 40	1 / 1
20 / 200	- / 1	- / 100	- / 100	- / 100	55 / 54	90 / 74	31 / 30	1 / 1
21 / 200	- / 1	- / 100	- / 100	- / 100	83 / 83	91 / 91	42 / 42	1 / 1
22 / 400	- / 1	- / 200	- / 200	- / 200	118 / 117	180 / 188	66 / 64	1 / 1
23 / 600	- / 2	- / 300	- / 300	- / 299	248 / 248	283 / 291	125 / 127	1 / 1
24 / 800	- / 1	- / 400	- / 400	- / 400	323 / 323	377 / 395	164 / 167	1 / 1
25 / 1000	- / 2	- / 500	- / 500	- / 499	314 / 315	479 / 466	172 / 179	1 / 1



Data I



Data II

Figure 7: C values (logarithmic scale) for SVM-L and SVM-Q in experiments 1-25.

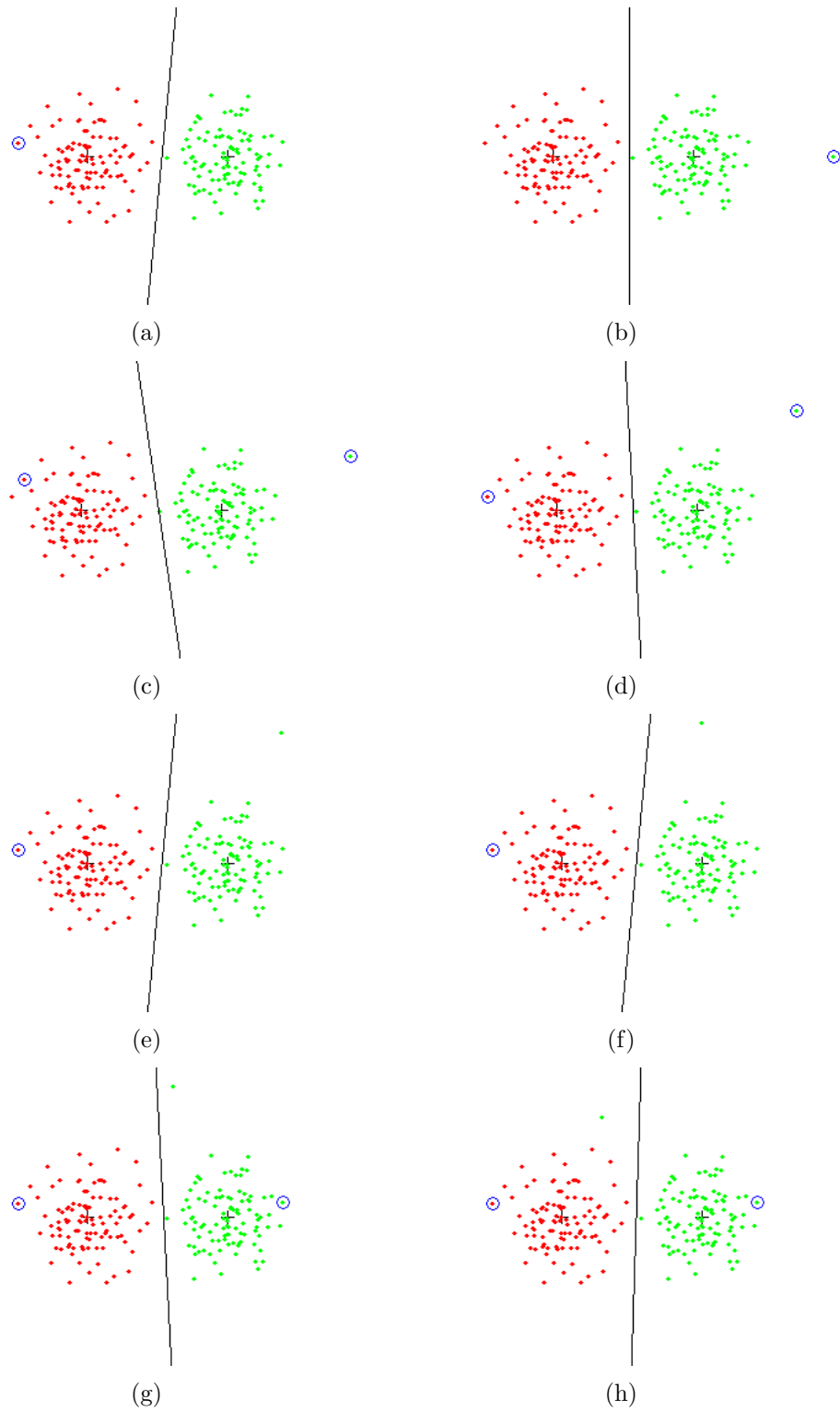


Figure 8: Test of heuristic classifier on data with an outlier. Distance d between class centers (black crosses) is equal to 10. On figure (a) both classes have no outliers, on figures (b)–(h) first class (green points) has one outlier which is rotating around the center of the first class towards the second class. Black line is the separating hyperplane, blue circles mark its support vectors.

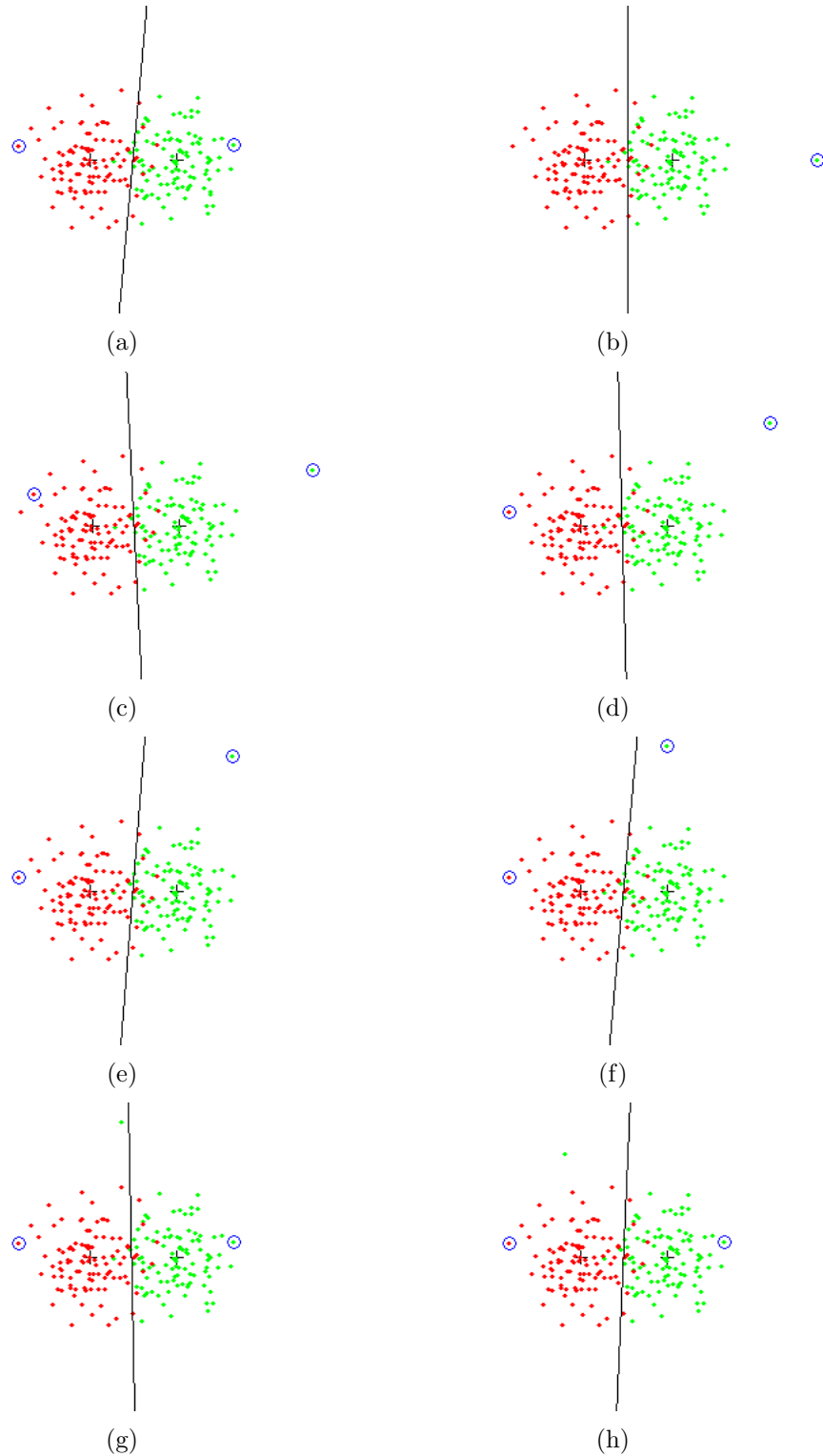


Figure 9: Test of heuristic classifier on data with an outlier. Distance d between class centers (black crosses) is equal to 6. On figure (a) both classes have no outliers, on figures (b)–(h) first class (green points) has one outlier which is rotating around the center of the first class towards the second class. Black line is the separating hyperplane, blue circles mark its support vectors.