1 Computational Aspects of SVM

We have a dataset containing M points in N dimensions (i.e. each datapoint is described by N floats). After investigating the data and finding optimal hyperparameters, we train a Support Vector Machine (SVM) classifier with an RBF Kernel that results in S support vectors.

A) How many parameters are required to store the trained model?

Solution

To store a trained SVM model, it is necessary to store S support vectors, each of which has a dimensionality of N. Moreover, the corresponding coefficients $\alpha_i y_i$ (one for each support vector) and the scalar parameter b should be kept. Therefore, the total number of parameters to store is

$$S \times N + S + 1 = S(N+1) + 1. \tag{1}$$

Note that, due to the constraint $\sum_i^M \alpha_i y_i = \sum_i^S \alpha_i y_i = 0$, an option is to store one less parameter. However, such choice is rarely used in practice.

■

B) Assume that each parameter has a float data type that takes 8 bytes in memory. What is the required memory to store the model if dataset is 100-dimensional (i.e., $N = 100$) and the number of support vectors is $S = 10,000$?

Solution

Using eq. (1) with $S = 10,000$ and $N = 100$ will amount to

$$10,000 \times (100 + 1) = 1,010,000$$

float-type parameters. Note that the $+1$ term corresponding to b is omitted since it does not make a significant difference. Thus, given that each float number takes up 8B (8 bytes), the required memory is

$$\frac{1,010,000 \times 8\text{B}}{1024 \times 1024} \approx 7.71\text{MB}.$$

■

C) Assume that, in the previous question, only 1% of all datapoints became support vectors, meaning that total number of datapoints is $M = 1,000,000$. The training time complexity for SVM is $O(MN^2)$. Furthermore, assume that training in a smaller problem with $M = 1000$ and $N = 10$ takes 0.1 second. How much time do we approximately need to train the classifier for the initial problem?

Solution

The initial problem with $M = 1,000,000$ has 1000 times more datapoints than the smaller dataset, and 10 times larger dimensionality. As such, training time of the larger dataset will be $1000 \times 10^2 = 100,000$ of that of the smaller dataset, and is equal to

$$100,000 \times 0.1\text{s} = 10,000\text{s} = 166\frac{2}{3}\text{min} \approx 2.78\text{h} \approx 2\text{h}47\text{min}.$$

■

D) Average modern laptop CPU requires 50W of power under full load. Using the time from the previous question, how much energy (in Wh) does one need to train such SVM? For comparison, note that a regular kettle draws 1500W, and it takes 5 minutes to boil approximately 2 liters of water. Training this SVM model is equivalent to boiling how many liters of water?

Solution

If CPU draws 50W, then 2.78h of training requires 2.78h $\times$ 50W = 139Wh of energy. On the other hand, boiling the regular kettle of water for 5 minutes demands $\frac{5}{60}$h $\times$ 1500W = 125Wh. Hence, training this SVM model is roughly equivalent to boiling 2 liters of water.

■

2 Classification with SVM

A) Consider a 2-dimensional classification problem with only 2 datapoints, including $\boldsymbol{x}^1 = [0.5, 0.5]^\top$ and $\boldsymbol{x}^2 = [-0.5, -0.5]^\top$ with $+1$ and -1 class labels, respectively (see fig. 1). Compute the coefficients α_i and the bias term b for a SVM classifier run on this problem with an RBF kernel where $k(\boldsymbol{x}^1, \boldsymbol{x}^2) = 0.5$ ($\phi(\cdot)$ is the corresponding feature map). Moreover, draw the isolines of the classifier function and the classifier hyperplane.

Hint: Recall that the SVM classifier function is given by

$$f(\boldsymbol{x}) = \text{sign}\left(\sum_i \alpha_i y_i k(\boldsymbol{x}, \boldsymbol{x}^i) + b\right). \tag{2}$$

Furthermore, the necessary conditions for optimality are provided below.

$$\begin{cases} \boldsymbol{w} = \sum_i \alpha_i y_i \phi(\boldsymbol{x}^i) \\ \sum_i \alpha_i y_i = 0 \quad \text{(appearing in the dual problem)} \\ y_i\left(\sum_j \alpha_j y_j k(\boldsymbol{x}^j, \boldsymbol{x}^i) + b\right) \geq 1, \quad \forall i = 1, \dots, M \quad \text{(primal feasibility)} \\ \alpha_i \geq 0, \quad \forall i = 1, \dots, M \quad \text{(dual feasibility)} \\ \alpha_i\left(y_i\left(\sum_j \alpha_j y_j k(\boldsymbol{x}^j, \boldsymbol{x}^i) + b\right) - 1\right) = 0, \quad \forall i = 1, \dots, M \quad \text{(KKT condition)} \end{cases}. \tag{3}$$

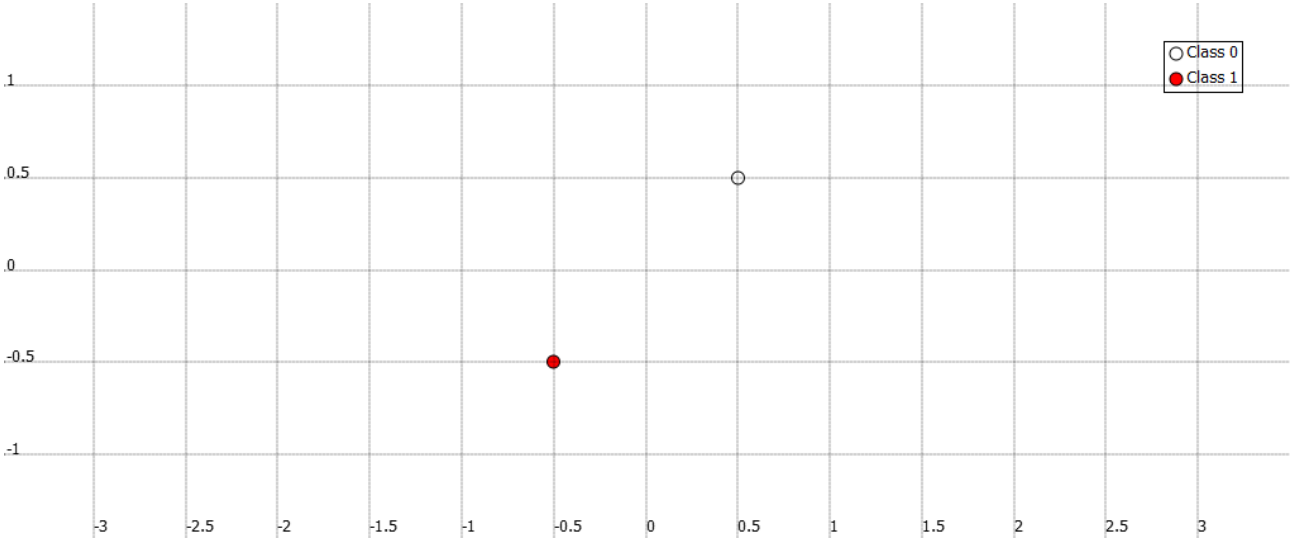


Figure 1: Question 2.A

Solution

Since there are only two data points, both datapoints must be support vectors in order to satisfy the constraint $\sum_i \alpha_i y_i = 0$ mentioned in eq. (3). Each point is located exactly on either side of the margin. Hence, the value of the classifier function at each support vector $\boldsymbol{x}^i$ is equal to ± 1 depending on its label y_i. In other words, it follows from eq. (2) that

$$\begin{cases} \sum_{i=1}^{M} \alpha_i y_i k(\boldsymbol{x}^1, \boldsymbol{x}^i) + b = 1 \\ \sum_{i=1}^{M} \alpha_i y_i k(\boldsymbol{x}^2, \boldsymbol{x}^i) + b = -1 \end{cases}. \tag{4}$$

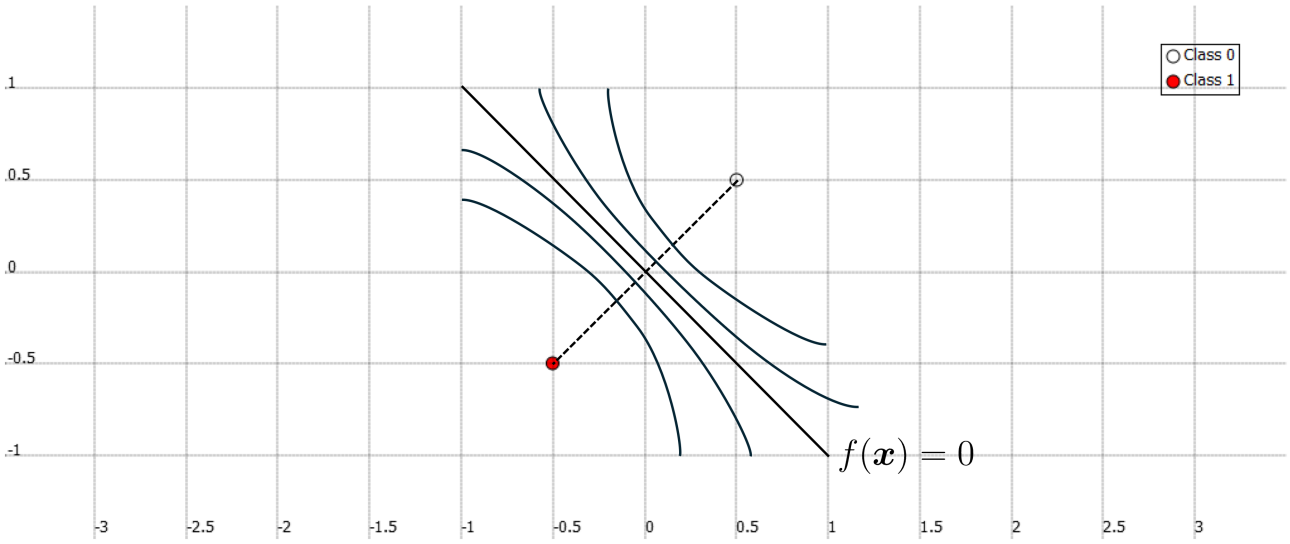


Figure 2: Question 2.A - Solution

The constraint $\sum_i \alpha_i y_i = 0$ reduces to $\alpha_1 y_1 + \alpha_2 y_2 = 0$. As such, with $y_1 = +1$ and $y_2 = -1$, we obtain $\alpha_1 = \alpha_2$. Combining this with $k(\boldsymbol{x}^1, \boldsymbol{x}^1) = k(\boldsymbol{x}^2, \boldsymbol{x}^2) = 1$ and $k(\boldsymbol{x}^1, \boldsymbol{x}^2) = k(\boldsymbol{x}^2, \boldsymbol{x}^1) = 0.5$, eq. (4) will be expanded into

$$\begin{cases} \alpha_1 - 0.5\alpha_1 + b = 1 \\ 0.5\alpha_1 - \alpha_1 + b = -1 \end{cases}. \tag{5}$$

Further simplification yields

$$\begin{cases} 0.5\alpha_1 + b = 1 \\ -0.5\alpha_1 + b = -1 \end{cases}. \tag{6}$$

Summing the above equations will result in $b = 0$. Putting the value of b back into one of the equations listed in eq. (6) will yield $\alpha_1 = 2$. Therefore, the parameters of this SVM classifier are $\alpha_1 = \alpha_2 = 2$ and $b = 0$. The separating hyperplane and the isolines are illustrated in fig. 2.

■

B) Two more points are added to this dataset in different ways as illustrated in figs. 3 and 4. How would the α_i and b parameters change in each case? Draw the support vectors and the classification boundary for each case.

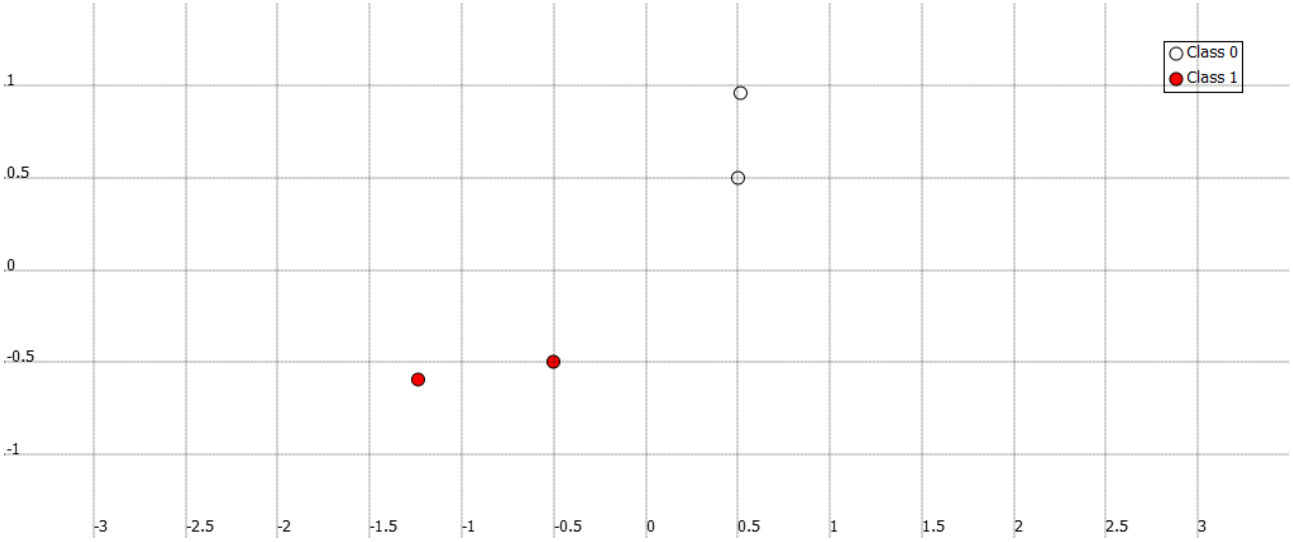


Figure 3: Question 2.B - Case (1)

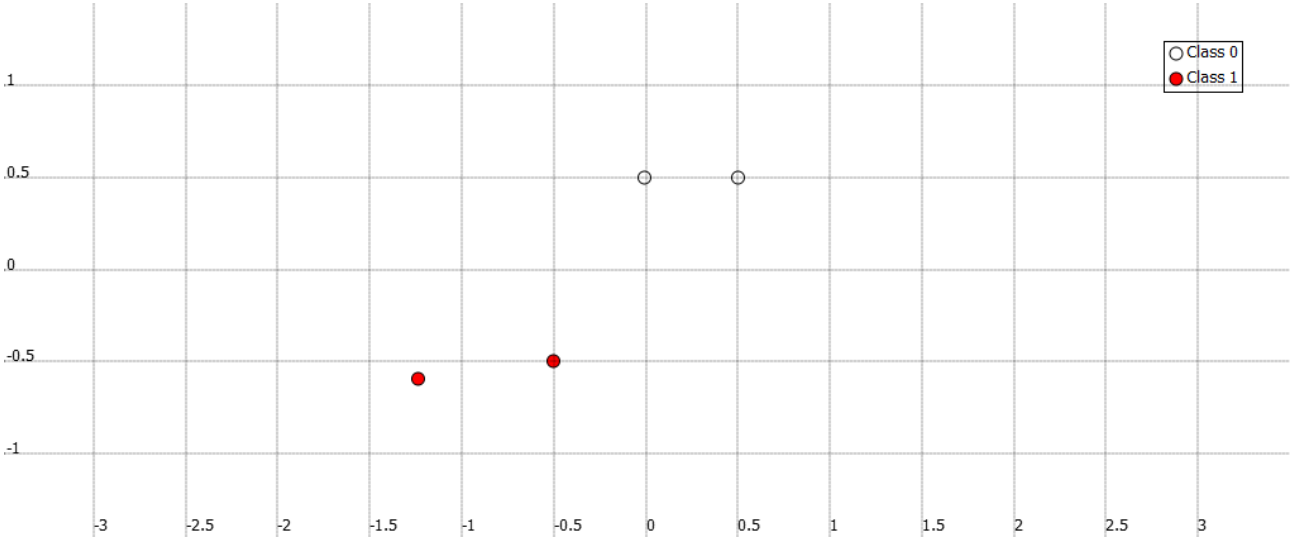


Figure 4: Question 2.B - Case (2)

Solution

Case (1): Since the new points lay outside the margin, the separating hyperplane and the support vectors will remain unchanged (see fig. 5).

Case (2): The point added to the white class is inside the original margin; hence, it becomes a support vector instead the original point in the white class (see fig. 6).

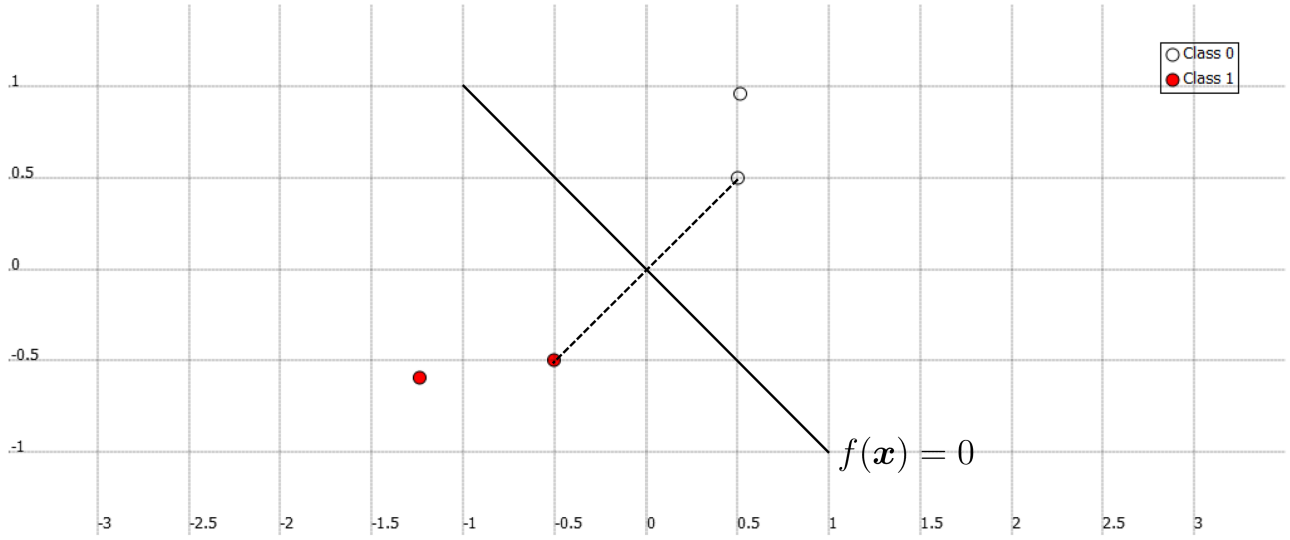


Figure 5: Question 2.B - Case (1) - Solution

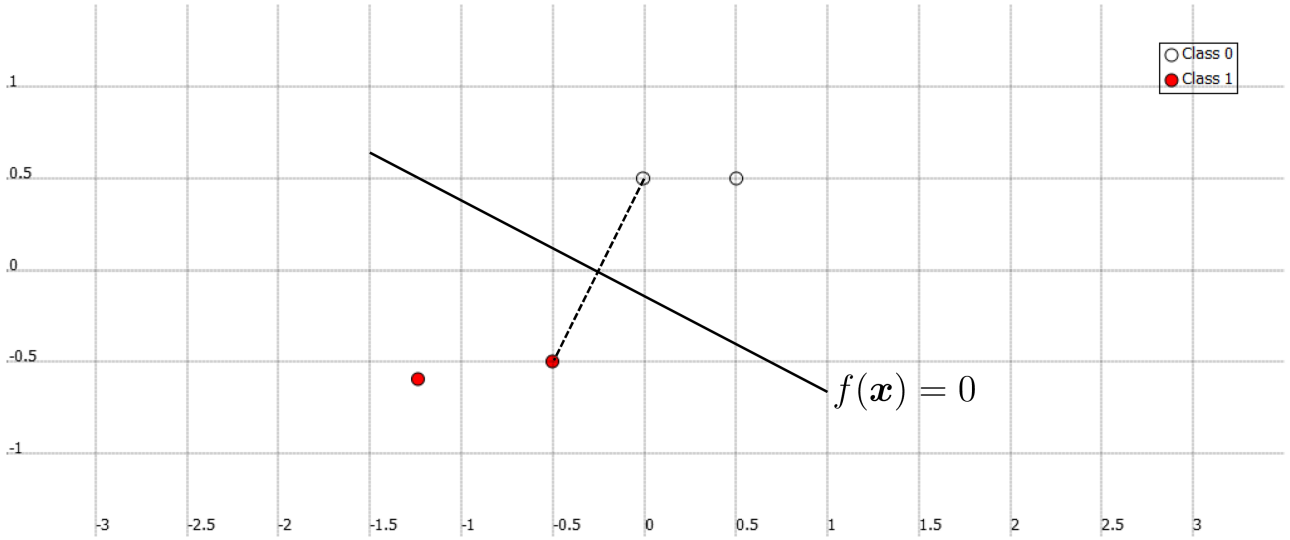
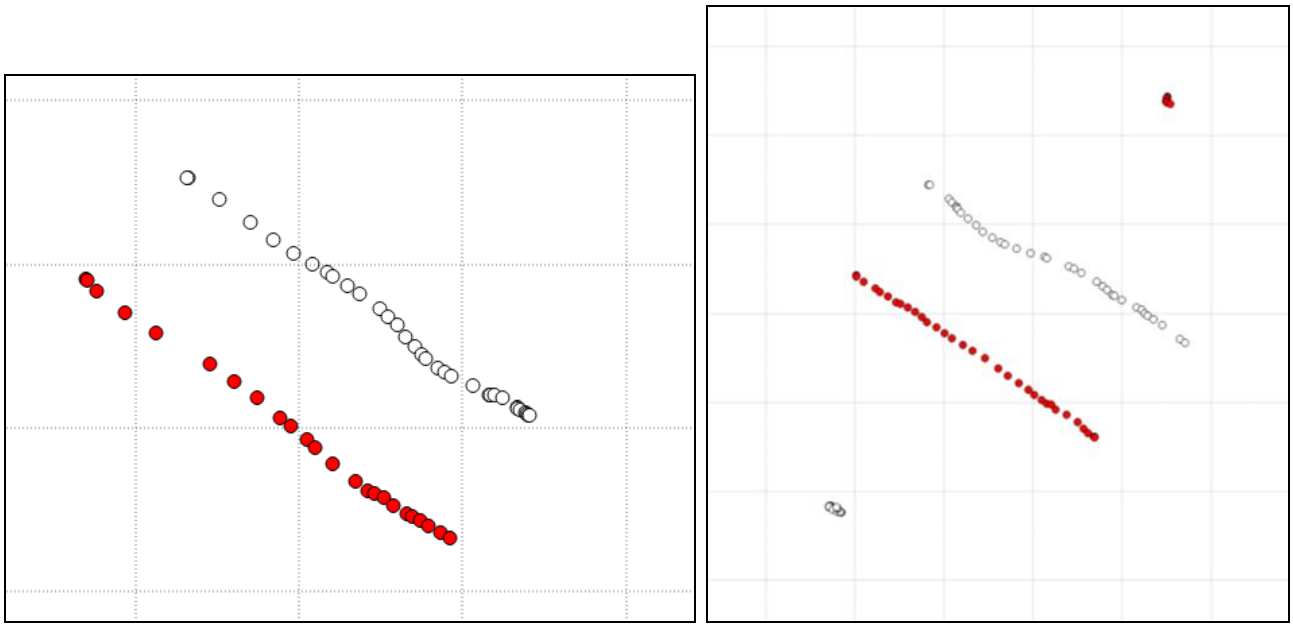


Figure 6: Question 2.B - Case (2) - Solution

C) Consider the binary classification problem among red and white classes shown in fig. 7. For the case of SVM with an RBF kernel, draw the separating line in each case. Do not compute it nor run MLDemos; instead, infer what the line would look like from your intuition. Discuss how this line changes as a function of the penalty factor C and the kernel width σ.



(i) **(ii)**

Figure 7: Question 2.C

Solution

Case (i): The separating line is unaffected by the value of the penalty C since both classes are perfectly separable. The separating line is a straight line passing in-between the two classes. The kernel width affects only the number of support vectors. More specifically, the smaller the kernel width, the more support vectors (see fig. 8).

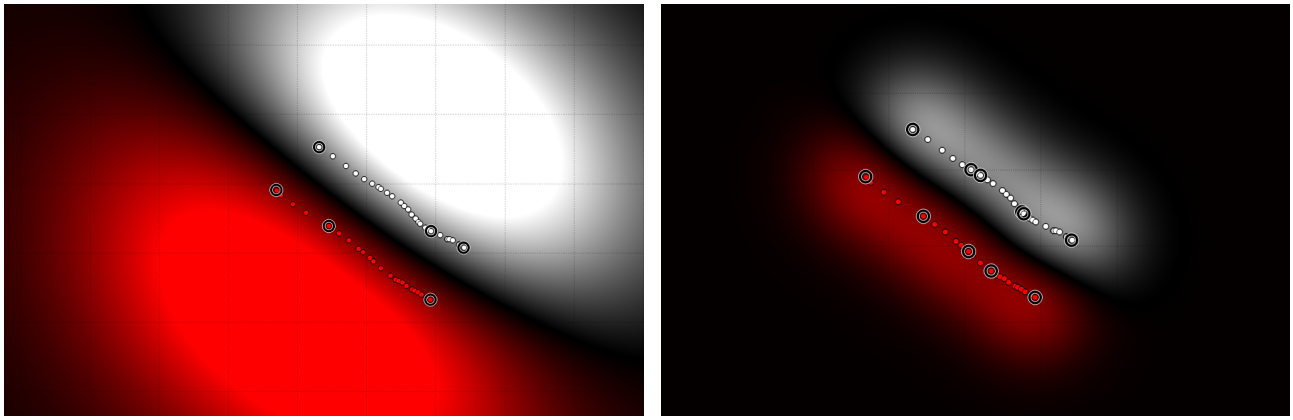


Figure 8: Solution found for case (i) with $\sigma = 0.1$ (left) and $\sigma = 0.01$ (right).

Case (ii): Here, the separating line changes as a function of both the penalty factor C and kernel width σ as presented in figs. 9 and 10.

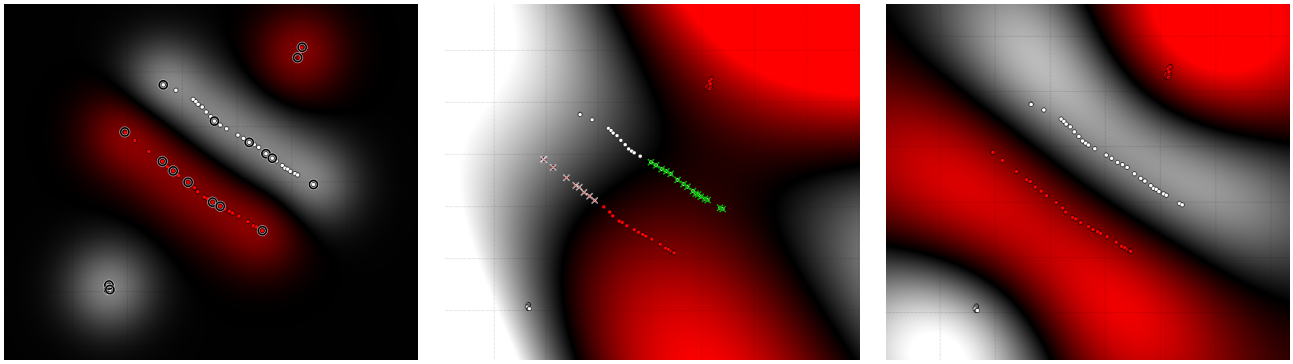
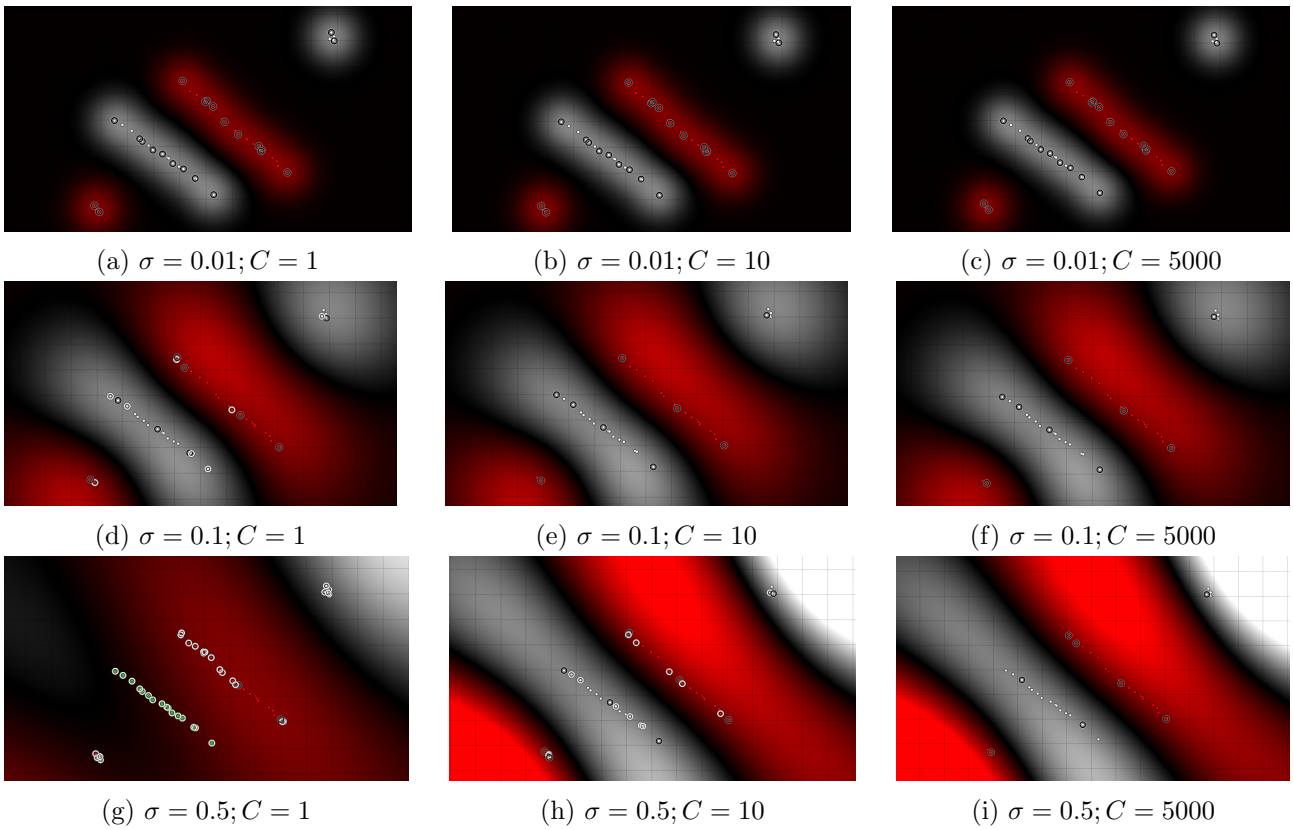


Figure 9: Solution found for case (ii) with (left) small kernel width ($\sigma = 0.01$) and large penalty ($C = 5000$) leads to perfect classification; however, this can also be viewed as overfitting. (Middle) Large kernel width ($\sigma = 0.5$) and small penalty ($C = 10.0$) yields incorrect classification. (Right) Correct values of kernel width ($\sigma = 0.1$) and penalty ($C = 1000$) bringing about a good classification with no overfitting.



(a) $\sigma = 0.01; C = 1$ (b) $\sigma = 0.01; C = 10$ (c) $\sigma = 0.01; C = 5000$

(d) $\sigma = 0.1; C = 1$ (e) $\sigma = 0.1; C = 10$ (f) $\sigma = 0.1; C = 5000$

(g) $\sigma = 0.5; C = 1$ (h) $\sigma = 0.5; C = 10$ (i) $\sigma = 0.5; C = 5000$

Figure 10: Effect of having different values of kernel width σ and penalty factor C in case (ii).

3 Optimization of SVM

A: Convex Optimization - Multiplicity of Solutions in SVM) SVM is based on solving a convex optimization problem, where the objective function $\|\boldsymbol{w}\|^2$ is strictly convex. As discussed in the lecture, while the convex problem admits a single global optimum and hence leads to a unique vector $\boldsymbol{w} \in \mathbb{R}^N$, there can be multiple ways in which $\boldsymbol{w}$ is constructed. Indeed, $\boldsymbol{w}$ is constructed as a linear combination of support vectors. If one has at disposal a set of K support vectors with $K > N$, these vectors are linearly dependent. Therefore, there exists more than one combination of scalars $\alpha_i,\ \forall i = 1, \ldots, K$, yielding the same $\boldsymbol{w}$ constructed as $\boldsymbol{w} = \sum_{i=1}^{K} \alpha_i y_i \boldsymbol{x}^i$.

Convince yourself that this is the case when considering linear SVM for binary classification, assuming that $N = 2$ and that you have at your disposal 3 non-zero and non-collinear datapoints $\boldsymbol{x}^i,\ i = 1, 2, 3$ that satisfy the constraint $y_i(\boldsymbol{w}^\top \boldsymbol{x}^i + b) = 1,\ \forall i$. Show that there exists another combination of points that can construct the same $\boldsymbol{w}$.

Solution

The variables of the dual SVM optimization are the Lagrange parameters α_i, with one Lagrange parameter per datapoint, i.e., $i = 1, \ldots, M$. As per the KKT conditions, the Lagrange parameters represent the weight given to each datapoint to construct $\boldsymbol{w} = \sum_i \alpha_i y_i \boldsymbol{x}^i$. The question is whether we can find different sets of α_i that lead to the same optimum.

Let $\boldsymbol{w} = \alpha_1 y_1 \boldsymbol{x}^1 + \alpha_2 y_2 \boldsymbol{x}^2$ be the optimal $\boldsymbol{w}$. Since none of the datapoints are collinear, any pair of two points is linearly independent. Hence, each point can be expressed as the linear combination of two other points. We can hence construct $\boldsymbol{x}^2 = \beta_1 \boldsymbol{x}^1 + \beta_3 \boldsymbol{x}^3$ with appropriate scalars β_1 and β_3. Replacing $\boldsymbol{x}^2$ in $\boldsymbol{w}$ formulation, we obtain a new set of coefficients α_i' for the same optimal $\boldsymbol{w}$, namely

$$\boldsymbol{w} = \underbrace{(\alpha_1 y_1 + \alpha_2 y_2 \beta_1)}_{\alpha_1' y_1} \boldsymbol{x}^1 + \underbrace{\alpha_2 y_2 \beta_3}_{\alpha_3' y_3} \boldsymbol{x}^3 = \alpha_1' y_1 \boldsymbol{x}^1 + \alpha_3' y_3 \boldsymbol{x}^3.$$

■

B: Margin) The constraints of the SVM problem specify that all support vectors should lie on either of the two hyperplanes parallel to the separating hyperplane with equations $\boldsymbol{w}^\top \boldsymbol{x} + b = \pm 1$. Show that the constant 1 is arbitrary and does not affect the solution.

Solution

The KKT condition $\sum_i \alpha_i y_i = 0$ implies that we have at least two support vectors, one in each class. Hence, there exist two points, which we denote as $\boldsymbol{x}^1$ and $\boldsymbol{x}^2$ with $y_1 = 1$ and $y_2 = -1$, for which the constraints $y_i(\boldsymbol{w}^\top \boldsymbol{x}^i + b) = 1$ are satisfied.

We modify the constraints such that all support vectors lie on a plane with equation $y_i(\boldsymbol{w}^\top \boldsymbol{x}^i + b) = a$, with $a > 0$. We then have

$$\begin{cases} \boldsymbol{w}^\top \boldsymbol{x}^1 + b = a \\ \boldsymbol{w}^\top \boldsymbol{x}^2 + b = -a \end{cases}. \tag{7}$$

Subtracting the two lines, we get $\boldsymbol{w}^\top (\boldsymbol{x}^1 - \boldsymbol{x}^2) = 2a$. Expanding the inner product, we obtain $\|\boldsymbol{w}\| = \frac{2a}{\|(\boldsymbol{x}^1 - \boldsymbol{x}^2)\| \cos(\theta)}$, where θ is the angle between $\boldsymbol{w}$ and the vector $\boldsymbol{x}^1 - \boldsymbol{x}^2$. We see that the factor a only scales the norm of the vector $\boldsymbol{w}$, but it does not affect the choice of support vectors. It also does not change the direction of $\boldsymbol{w}$ and hence does not affect the orientation of the hyperplane.

■

C: Convexity and Optimality of the Relaxed Problem) The introduction of slack variables in the SVM optimization problem allows to find a solution to the problem that would otherwise been deemed infeasible. The drawback is that the slack leads to solutions that are suboptimal in a sense that it allows violations of the strict constraints in the unrelaxed problem. Note that the problem remains convex, but the slacks shift the optimum to a value different from the true optimum.

Prove first that the relaxed problem remains convex. Recall the conditions for convexity and strict convexity: a convex function f is such that $f(\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \leq f(\lambda \boldsymbol{x}) + f((1-\lambda)\boldsymbol{y})$ for $0 \leq \lambda \leq 1$. Strict convexity arises when the inequality is replaced by a strict inequality ($<$ in place of $\leq$) for $0 < \lambda < 1$ and $\boldsymbol{x} \neq \boldsymbol{y}$.

Secondly, explain under which conditions the optimum in the relaxed problem is identical to the original problem for linear SVM.

Solution

The relaxed problem for linear SVM is mentioned below.

$$\begin{array}{ll} \min\limits_{\boldsymbol{w},b,\boldsymbol{\xi}} & \frac{1}{2}\|\boldsymbol{w}\|^2 + \frac{C}{M}\sum_{i=1}^{M}\xi_i \\ \text{s.t.} & y_i(\boldsymbol{w}^\top \boldsymbol{x}^i + b) \geq 1 - \xi_i \quad \forall i = 1,\ldots,M \\ & \xi_i \geq 0 \quad \forall i = 1,\ldots,M \end{array} \tag{8}$$

For two arbitrary vectors $\boldsymbol{x}, \boldsymbol{y}$ and $\lambda \in [0,1]$,

$$\begin{aligned} \|\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\|^2 &= (\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y})^\top (\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}) \\ &= \lambda^2\|\boldsymbol{x}\|^2 + (1-\lambda)^2\|\boldsymbol{y}\|^2 + 2\lambda(1-\lambda)\boldsymbol{x}^\top \boldsymbol{y}. \end{aligned}$$

We now compare this with $\lambda\|\boldsymbol{x}\|^2 + (1-\lambda)\|\boldsymbol{y}\|^2$:

$$\begin{aligned} \lambda\|\boldsymbol{x}\|^2 + (1-\lambda)\|\boldsymbol{y}\|^2 - \|\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\|^2 &= \lambda(1-\lambda)\big(\|\boldsymbol{x}\|^2 + \|\boldsymbol{y}\|^2 - 2\boldsymbol{x}^\top \boldsymbol{y}\big) \\ &= \lambda(1-\lambda)\|\boldsymbol{x} - \boldsymbol{y}\|^2 \geq 0. \end{aligned}$$

Hence,

$$\|\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\|^2 \leq \lambda\|\boldsymbol{x}\|^2 + (1-\lambda)\|\boldsymbol{y}\|^2, \quad \forall\, \lambda \in [0,1],$$

which proves that $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$ is convex.

Moreover, if $\boldsymbol{x} \neq \boldsymbol{y}$ and $0 < \lambda < 1$, then $\|\boldsymbol{x} - \boldsymbol{y}\|^2 > 0$, and the inequality becomes strict:

$$\|\lambda \boldsymbol{x} + (1-\lambda)\boldsymbol{y}\|^2 < \lambda\|\boldsymbol{x}\|^2 + (1-\lambda)\|\boldsymbol{y}\|^2.$$

Therefore, $f(\boldsymbol{x}) = \|\boldsymbol{x}\|^2$ is strictly convex.

Note that the inequality constraints in eq. (8) are affine in the decision variables $\boldsymbol{w}$, b, and ξ_i $\forall i = 1,\ldots,M$. As to the objective function, note that $\frac{1}{2}\|\boldsymbol{w}\|^2$ is strictly convex (proven above) and $\sum_{i=1}^{M}\xi_i$ is convex. Since the quadratic term is strictly convex and grows faster than the linear terms, the objective function is strictly convex. It hence admits a single global optimum.

The addition of the slack variables, however, can shift the optimum of the objective function to a solution that is not the true optimum (without relaxation of constraints). The relaxed optimization finds an optimal solution that is a tradeoff between augmenting the margin across the two classes (reducing the first term of the cost function) and reducing the cost of violating one or more constraints (reducing the second term of the cost function).

The penalty associated to the violation of the constraint is conveyed through the choice of the constant C. A large C will tend to force the optimization to find a solution close to that of the unrelaxed

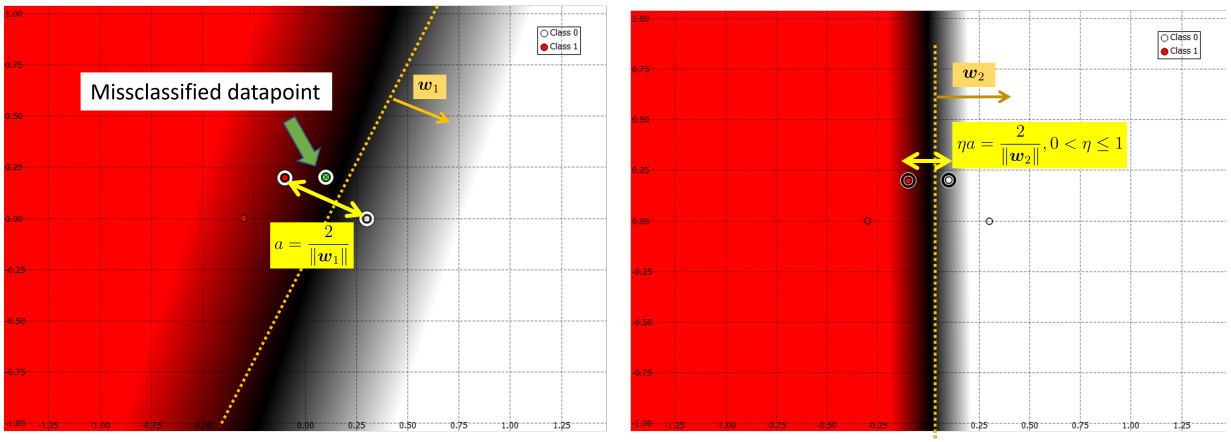


Figure 11: Optimal solution of the relaxed SVM optimization when using a low penalty on slacks such as $C = 5$ versus a high penalty such as $C = 100$.

problem. This is illustrated in fig. 11. When applying a small penalty such as $C = 5$ for the violation of the constraints, the optimization finds a separating hyperplane with a larger margin compared with a high penalty such as $C = 100$.

We show next that the quadratic and linear terms in the objective function of the relaxed linear SVM, mentioned in eq. (8), vary with the width of the margin, which we denote as a.

Consider the group of four points in fig. 11. The two hyperplanes generated by $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$ are both optimal solutions for different values of C. For the first hyperplane defined by $\boldsymbol{w}_1$, it holds that $\|\boldsymbol{w}_1\| = \frac{2}{a}$. In this case, one of the two points from the white class is misclassified. The cost associated with the constraint's violation for this point is entailed in the associated slack ξ. Next, we show that the slack varies linearly with the distance to the hyperplane. Without loss of generality, we can assume $b = 0$ (shift of the origin). The constraints are satisfied at equality for the two datapoints on the margin and for the point inside the margin with slack ξ. For the latter, we have

$$\xi = 1 - y_i \boldsymbol{w}_1^\top \boldsymbol{x}^i = 1 - y_i \|\boldsymbol{w_1}\| \|\boldsymbol{x}^i\| \cos(\theta), \tag{9}$$

where θ is the angle between $\boldsymbol{w}$ and the vector $\boldsymbol{x}^i$. We can also describe slack ξ as

$$\xi = 1 - y_i \boldsymbol{w}_1^\top \boldsymbol{x}^i = 1 - y_i \|\boldsymbol{w}_1\| \underbrace{\frac{\boldsymbol{w}_1^\top \boldsymbol{x}^i}{\|\boldsymbol{w}_1\|}}_{d_i} = 1 - \|\boldsymbol{w}_1\| y_i d_i, \tag{10}$$

where d_i is the signed distance of the point $\boldsymbol{x}^i$ to the hyperplane. It can be observed in both eqs. (9) and (10) that the slack varies linearly with $\|\boldsymbol{w}_1\|$ and distance to the hyperplane.

The second hyperplane with $\boldsymbol{w}_2$ satisfies all constraints; hence, $\xi_i = 0 \; \forall i$. It also holds that $\|\boldsymbol{w}_2\| = \frac{2}{\eta a}$, where $0 < \eta \leq 1$ is an scaling factor with respect to the margin width a for the first hyperplane. To determine if a solution with slack can lead to a value on the objective function that is equal or better than the solution without slack, one must check whether

$$\frac{1}{2}\|\boldsymbol{w}_1\|^2 + \frac{C}{M}\sum_i \xi_i \overset{?}{\leq} \frac{1}{2}\|\boldsymbol{w}_2\|^2 \xrightarrow{\sum_i \xi_i = \xi} \frac{2}{a^2} + \frac{C}{M}\xi \overset{?}{\leq} \frac{2}{(\eta a)^2}$$

holds. Many cases will arise depending on the values of C and η. Observe that the cost in the objective function associated with enlarging the margin is privileged over violating constraints, as the former grows quadratically with $\|\boldsymbol{w}\|$ whereas the latter varies linearly. The solver will hence tend to privilege solutions with small violation of constraints if these lead to an increase in the margin. The shift of the optimum is illustrated in fig. 12 for a simple 2-point separation.

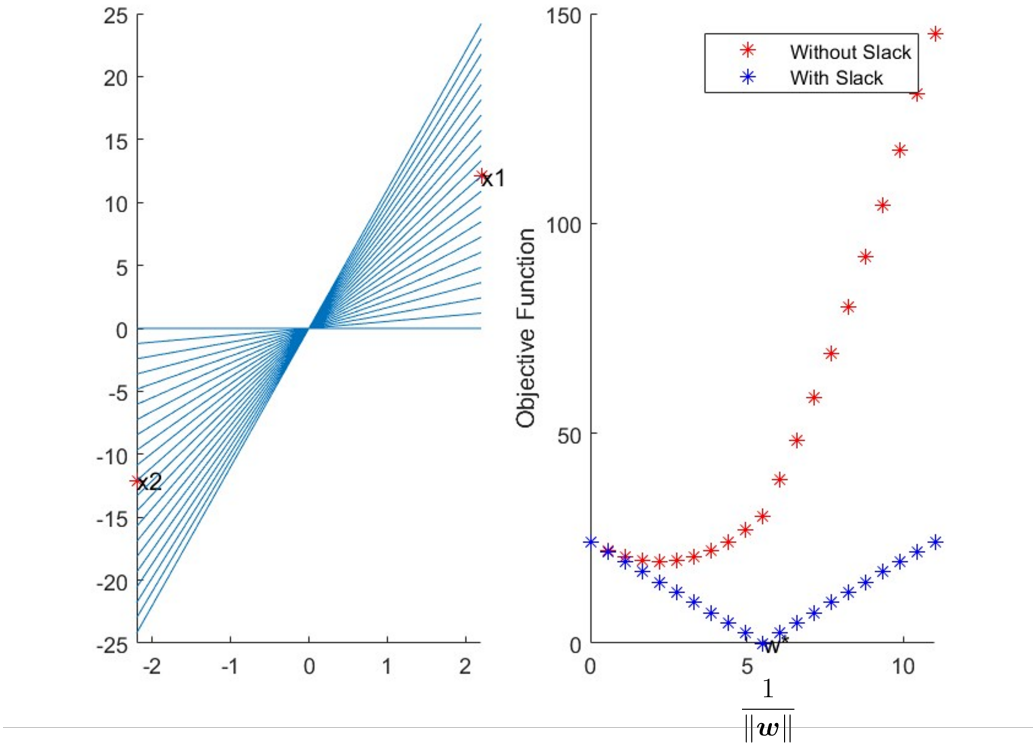


Figure 12: (Left) distribution of separating hyperplanes across a pair of datapoint. (Right) evolution of optimum of the SVM objective function for the distribution of hyperplane with and without slack.