# Support vector machines: Primal/dual formulations

When studying Fisher Discriminants, we have seen how to use the notion of Lagrange multipliers when optimizing an objective with an equality constraint. In the formulation of support vector machines (both in the linearly separable case formulated on Mar 11 and in the general case we will see in this handout), we are interested in minimizing an objective subject to inequality constraints. We will adapt the method of Lagrangians to handle inequality constraints in this handout.

In addition, our Lagrangian approach allows us to the support vector machines from a complete different light, the so-called dual approach, which will give us insights into how the training data is used to learn a model. In addition, this also reveals to us a structured way to introduce non-linearity into our models, and we will exploit these insights to understand simple feedforward neural networks.

## 1 Linearly separable case

Recall the support vector formulation for the linearly separable case. We have examples $\mathbf{z}_1, \ldots, \mathbf{z}_n$ with labels $y_1, \ldots, y_n$. The examples come from a $p-$dimensional real vector space and the labels can be either 1 or $-1$. If the examples are linearly separable, the maximum margin classifier (or our linear svm) is obtained using the following optimization problem:

$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2$$

$$\text{subject to the constraints } y_i(\mathbf{w}^T\mathbf{z}_i - b) \geq 1 \text{ for all } 1 \leq i \leq n. \quad (1)$$

To write the Lagrangian for this problem, we rewrite each inequality constraint above so that it looks like $f_i(\mathbf{w}, b) \leq 0$, namely

$$1 - y_i(\mathbf{w}^T\mathbf{z}_i - b) \leq 0.$$

Each inequality gets its own Lagrange multiplier $\lambda_i$, so our Lagrangian is (letting $\Lambda = (\lambda_1, \ldots, \lambda_n)$)

$$\mathcal{L}(\mathbf{w}, b, \Lambda) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \lambda_i\big(1 - y_i(\mathbf{w}^T\mathbf{z}_i - b)\big).$$

Now consider the following problem for a specific choice of $\mathbf{w}$ and $b$,

$$\max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda),$$

where $\Lambda \geq 0$ is shorthand for $\lambda_1 \geq 0, \lambda_2 \geq 0, \cdots, \lambda_n \geq 0$. Now if $\mathbf{w}$ and $b$ satisfy all constraints, we will have for all $1 \leq i \leq n$ that

$$1 - y_i(\mathbf{w}^T\mathbf{z}_i - b) \leq 0,$$

therefore

$$\begin{aligned}\mathcal{L}(\mathbf{w}, b, \Lambda) &= \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \lambda_i\big(1 - y_i(\mathbf{w}^T\mathbf{z}_i - b)\big) \\ &\leq \frac{1}{2}||\mathbf{w}||^2,\end{aligned}$$

with equality in the second equation iff we choose $\lambda_i = 0$ for all $i$. Therefore, for any $\mathbf{w}$ and $b$ satisfying all constraints,

$$\max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda) = \frac{1}{2}||\mathbf{w}||^2.$$

On the other hand if $\mathbf{w}$ and $b$ are such that there is even a single constraint violated, that is for some $j$,

$$1 - y_j(\mathbf{w}^T\mathbf{z}_j - b) \geq 0.$$

Then to maximize $\mathcal{L}(\mathbf{w}, b, \Lambda)$, we can choose $\lambda_j \to \infty$, therefore

$$\lambda_j(1 - y_j(\mathbf{w}^T\mathbf{z}_j - b)) \to +\infty,$$

therefore

$$\max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda) = \infty.$$

Putting it together

$$\max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda) = \begin{cases} \frac{1}{2}||\mathbf{w}||^2 & \mathbf{w}, b \text{ satisfy all } n \text{ constraints} \\ \infty & \text{else.} \end{cases}$$

Let us call

$$g(\mathbf{w}, b) \overset{\text{def}}{=} \max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda)$$

for convenience. Now there is definitely at least one $\mathbf{w}$, $b$ that satisfies all constraints (since the points are linearly separable). Therefore, the smallest value $g(\mathbf{w}, b)$ can take is definitely not infinity (so any $\mathbf{w}, b$ that violates any constraint will never minimize $g(\mathbf{w}, b)$). That means that if we look for

$$\arg\min_{\mathbf{w}, b} g(\mathbf{w}, b),$$

the solution must be $\mathbf{w}^*, b^*$ from (1), since we are minimizing $\frac{1}{2}||\mathbf{w}||^2$ but only among such $\mathbf{w}, b$ that satisfy every constraint given.

Therefore, we can pose (1) as follows:

$$\mathbf{w}^*, b^* = \arg\min_{\mathbf{w},b} \max_{\Lambda \geq 0} \mathcal{L}(\mathbf{w}, b, \Lambda). \tag{2}$$

We will call the above the *primal* formulation of the constrained optimization problem in (1)—where we write the Lagrangian, and observe that the solution of (1) is obtained by the minmax formulation in (2).

As the name "primal" suggests, we will also have a *dual* formulation of the optimization problem in (1). But before we get into the dual formulation, we have a little segue into elementary game theory.

## 1.1 Essential game theory

The above formulation (minmax) is very common across multiple areas, and game theory is one of the proto-problems that deals with these sorts of questions. What we say in this subsection applies more generally than the Lagrange optimization above. To emphasize this, we will change the notation in this subsection to a generic function $f$ of two (each could possibly be vector-valued) arguments, $\mathbf{x}$ and $\mathbf{y}$.

Suppose we have two (possibly vector-valued) variables $\mathbf{x}$ and $\mathbf{y}$, and we seek

$$\min_{\mathbf{x}} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}).$$

The above is callled the minmax formulation, where player 1 maximizes $f(\mathbf{x}, \mathbf{y})$ over $\mathbf{y}$ and returns a function of $\mathbf{x}$ to player 2, who then minimizes the function given by player 1 over $\mathbf{x}$ to achieve the value in the displayed expression above. Interchanging the order of min and max gives us

$$\max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}).$$

Now player 1 goes first, and returns a function of $\mathbf{y}$ to player 2 obtained by minimizing $f(\mathbf{x}, \mathbf{y})$ over all $\mathbf{x}$. Player 2 then maximizes the function given over all $\mathbf{y}$.

How do the above two problems relate to each other? Regardless of $f$, we will see below that minmax is $\geq$ maxmin . First notice that for all $\mathbf{x}'$ and $\mathbf{y}$

$$f(\mathbf{x}', \mathbf{y}) \geq \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}). \tag{3}$$

The above is true for all values of $\mathbf{y}$, and let $\mathbf{y}^-$ to be the value that maximizes the right side, namely

$$y^- = \arg\max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y})$$

so that

$$\max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}) = \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^-). \tag{4}$$

Then we have for all $\mathbf{x}'$, by combining (3) and (4)

$$f(\mathbf{x}', \mathbf{y}^-) \geq \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}^-) = \max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}).$$

Furthermore, we have for all $\mathbf{x}'$,

$$\max_{\mathbf{y}} f(\mathbf{x}', \mathbf{y}) \geq f(\mathbf{x}', \mathbf{y}^-) = \max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}).$$

Since the above inequality is true for all $\mathbf{x}'$, it must be true when we set $\mathbf{x}'$ to be the value $\mathbf{x}^-$ that minimizes the left side. So we have

$$\max_{\mathbf{y}} f(\mathbf{x}^-, \mathbf{y}) \geq \max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}),$$

or that

$$\min_{\mathbf{x}} \max_{\mathbf{y}} f(\mathbf{x}, \mathbf{y}) \geq \max_{\mathbf{y}} \min_{\mathbf{x}} f(\mathbf{x}, \mathbf{y}).$$

### 1.2 Dual formulation of the Lagrangian

The dual formulation the minmax formulation (2) is the maxmin version, namely

$$\max_{\Lambda \geq 0} \min_{\mathbf{w}, b} \mathcal{L}(\mathbf{w}, b, \Lambda). \tag{5}$$

As we saw, generally speaking the maxmin is $\geq$ minmax. However, when $\mathcal{L}(\mathbf{w}, b, \Lambda)$ is the Lagrangian obtained from convex objectives and inequality constraints as in our case, we have an interesting theorem that says that the minmax and maxmin formulations are related by an equality rather than the more general $\geq$. In addition, the values of the Lagrangian coefficients obtained from (5)and the value of $\mathbf{w}, b$ corresponding to those multipliers are the optimal values we are looking for.

In the other words, solving (5) above will solve (1) as well. We will not prove this equivalence in this course however, just use the result. Keep in mind that in a more general problem, the value of the dual formulation lower bounds the value of the optimization you are looking for, rather than solve the optimization right away as in this case.

Let us solve the dual. In particular, recall that

$$\mathcal{L}(\mathbf{w}, b, \Lambda) = \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \lambda_i \big(1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)\big).$$

Fix any $\Lambda \geq 0$. Let us solve the inner minimization first, *i.e.,* treating $\Lambda$ as fixed, we minimize

$$\min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b, \Lambda) = \min_{\mathbf{w},b} \frac{1}{2}||\mathbf{w}||^2 + \sum_{i=1}^{n} \lambda_i \big(1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)\big).$$

To do so, we take the gradient and partial derivatives with respect to $\mathbf{w}$ and $b$ respectively and set them to 0. Now,

$$\nabla_{\mathbf{w}} \mathcal{L}(\mathbf{w}, b, \Lambda) = \mathbf{w} - \sum_{i=1}^{n} \lambda_i y_i \mathbf{z}_i,$$

so we obtain by setting the gradient to 0 that

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i y_i \mathbf{z}_i. \tag{6}$$

Similarly,

$$\frac{\partial \mathcal{L}(\mathbf{w}, b, \Lambda)}{\partial b} = -\sum_{i=1}^{n} \lambda_i y_i$$

and setting the above to zero yields

$$\sum_{i=1}^{n} \lambda_i y_i = 0. \tag{7}$$

We now obtain several insights into support vector machines.

**Representation in terms of examples** The equation (6) is very important. It tells us that our solution is just a linear combination of the examples $\mathbf{z}_i$. This is a phenomenon we see in many problems, and not by accident. In this undergraduate level, we will not explore the complete range of problems with this property though EE 645 does. But for now, we are highlighting this property of support vector machines.

**Only pairwise dot products matter** Let us substitute for $\mathbf{w}$ from (6) and that $\sum \lambda_i y_i = 0$ into the expression for $\mathcal{L}(\mathbf{w}, b, \Lambda)$. We then get

$$\mathcal{L}(\mathbf{w}, b, \Lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \lambda_j \lambda_k y_j y_k \mathbf{z}_j^T \mathbf{z}_k.$$

Now we can rewrite the dual formulation as

$$\max_{\Lambda \geq 0} \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{j=1}^{n} \sum_{k=1}^{n} \lambda_j \lambda_k y_j y_k \mathbf{z}_j^T \mathbf{z}_k. \tag{8}$$

Without bothering about the mechanics of solving the problem above, we can immediately conclude that the solution only depends on the pariwise dot products $\mathbf{z}_j^T \mathbf{z}_k$, $1 \leq j, k \leq n$. This is because the only way the examples enter the formulation is via their dot products. Or in other words, if you had two different sets of examples but with the same pairwise dot products, they would both yield the same values Lagrange coefficients.

What about predicting the label of a test point $\mathbf{z}$? Well, we know that once we find the optimal coefficients $\Lambda^*$ from the maximization in (8), we will have the optimal separating hyperplane direction is

$$\mathbf{w} = \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{z}_i.$$

In order to predict on the test data point $\mathbf{z}$, we have to compute

$$\mathbf{w}^T \mathbf{z} - b.$$

We predict the label to be 1 if the quantity above is $> 0$, -1 else.

But we will have

$$\mathbf{w}^T \mathbf{z} = \left( \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{z}_i \right)^T \mathbf{z} = \sum_{i=1}^{n} \lambda_i^* y_i \mathbf{z}_i^T \mathbf{z},$$

or in other words, we only need the dot product of $\mathbf{z}$ with the examples. Similarly, the exercise below asks you to show that $b$ is just a function of the pairwise dot products between the examples as well.

**Karush-Kuhn-Tucker conditions** For the support vector machine problem, we will have the following additional property. For every example,

$$\lambda_i \big(1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)\big) = 0. \tag{9}$$

The above is derived from a series of arguments in convex optimization we will not go into in this class, but part of the arguments are based on what is commonly known as the Karush-Kuhn-Tucker conditions for optimality. You may have encountered this in other classes too since this is another widely used pricinple.

But here we pay attention to (9) in particular. This immediately means the following: if $1 - y_i(\mathbf{w}^T\mathbf{z}_i - b) \neq 0$, *i.e.*,

$$y_i(\mathbf{w}^T\mathbf{z}_i - b) > 1,$$

then (9) forces $\lambda_i = 0$. Any example with $y_i(\mathbf{w}^T\mathbf{z}_i - b) > 1$ is a point in the "interior", that is it is not one of the points that determines the margin. Since $\mathbf{w} = \sum \lambda_i y_i \mathbf{z}_i$, such points are irrelevant in determining $\mathbf{w}$ (such a point has coefficient $\lambda_i = 0$, hence makes no contribution to $\mathbf{w}$.). Even if you took out all points in the "interior" like this, the solution will not change

**Support vectors** These are points for which $\lambda_i \neq 0$. Note that (9) does NOT imply that every point on the margin ($y_i(\mathbf{w}^T\mathbf{z}_i - b) = 1$) is a support vector. Rather (9) implies that every support vector must be a point on the margin. These are, in a sense, the only points that matter. While predicting, these points are the only points used to make a prediction as well.

**Generalization** The primal and dual optimization are all solved to find the separating hyperplane from the training data. What about test data? How does this approach generalize to test data?

We will see using an argument from leave-one-out cross-validation a few classes hence that the number of support vectors is an indication of the test error. Fewer the number of support vectors as a fraction of the training data, lower the expected prediction error on test examples.

**Exercise 1** Using the fact that all support vectors satisfy

$$y_i(\mathbf{w}^T\mathbf{z}_i - b) = 1,$$

write $b$ in terms of the support vectors in the linearly separable case.

Hint: multiply both sides of the displayed equation by $\lambda_i y_i$ and add over all support vectors. What is $\sum \lambda_i y_i$, where the sum is taken over all support vectors? Use (7). $\square$

## 2 General case

It is quite simple actually to extend the linearly separable case to a general setup. Here, we allow for points to be misclassified. As before, our examples are $\mathbf{z}_1, \ldots, \mathbf{z}_n$ with labels $y_1, \ldots, y_n$ respectively.

If the points are not linearly separable, no combination of $\mathbf{w}$ and $b$ can satisfy for all $1 \le i \le n$,

$$y_i(\mathbf{w}^T\mathbf{z}_i - b) \ge 1.$$

Rather for any $\mathbf{w}$ and $b$, we will have some examples such that

$$y_i(\mathbf{w}^T\mathbf{z}_i - b) < 1.$$

Therefore, we allow a slack $\xi_i \ge 0$ to account for such examples. Namely, we ask that the $i$'th example

$$y_i(\mathbf{w}^T\mathbf{z}_i - b) \ge 1 - \xi_i,$$

but try to ensure that $\sum_i \xi_i$ is also minimized as best as possible. This means that for any example that can be classified properly, we set $\xi_i = 0$ (therefore eliminating its contribution to $\sum_i \xi_i$). When unavoidable, we try to ensure that $\xi_i$ is as small as possible.

Recall from our Jan 29th class that minimizing the *number* of misclassified examples is actually very difficult (NP-hard). Here, we are minimizing not the number of misclassified examples, but the sum of their slacks—and this is a simple convex optimization problem very similar to the linearly separable case.

With the above change in mind, we therefore look for $\mathbf{w}^*, b^*$ and a set of slacks $\{\xi_i^* : 1 \le i \le n\}$, one for each example, such that

$$\begin{aligned}\mathbf{w}^*, b^*, \{\xi_i^*\} &= \arg\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n}\xi_i \\ &\text{subject to } y_i(\mathbf{w}^T\mathbf{z}_i - b) \ge 1 - \xi_i \\ &\quad\text{and } \xi_i \ge 0 \text{ for all } 1 \le i \le n. \end{aligned} \tag{10}$$

In the above, $C$ is a number we choose that controls the balance between the margin ($\frac{1}{2}||\mathbf{w}||^2$) and the misclassification penalty ($\sum_{i=1}^{n}\xi_i$). It is like a knob we tune with and see what works best. When $C$ is large, any misclassification (where $\xi_i \ne 0$) gets multiplied by $C$ and the penalty is severe. When $C$ is small, we ignore misclassifications and instead focus primarily on the margin of the properly classified points. In practice, choose $C$ around 1 to balance the two (unless you have some other motivation specific to the problem that makes you want to focus on one or the other).

## 2.1 Minimizing hinge loss with $\ell_2$ regularization

The particular form of the optimization in (10) can be parsed in a different way that allows us to see the support vector machine as a single neuron with linear activation, minimizing what is known as the hinge loss. Note that from the constraints in (10), we have

$$\xi_i \geq 1 - y_i(\mathbf{w}^T \mathbf{z}_i - b) \text{ and } \xi_i \geq 0.$$

Putting the two together, we conclude that if $1-y_i(\mathbf{w}^T\mathbf{z}_i-b) > 0$, we should set (to minimize $\sum \xi_i$

$$\xi_i = 1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)$$

and if $1 - y_i(\mathbf{w}^T\mathbf{z}_i - b) \leq 0$, we set $\xi_i = 0$. Therefore,

$$\xi_i = \max\big(0, 1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)\big).$$

If we think of the support vector machine as a neuron with weights $\mathbf{w}$ and bias $b$, with a linear activation function, then the predicted value on the example $\mathbf{z}_i$ is

$$\hat{y}_i = \mathbf{w}^T \mathbf{z}_i - b,$$

so that

$$\xi_i = \max(0, 1 - y_i \hat{y}_i).$$

We can treat $\xi_i$ as the loss incurred on the example $\mathbf{z}_i$. This form of loss, $\max(0, 1 - y_i\hat{y}_i)$ is called the *hinge loss.* Therefore the hinge loss on the training data is

$$\sum_i \xi_i.$$

So now, the support vector machine formulation can be rewritten in a different way: we minimize is (dividing by $C$ to phrase it in a more convenient form)

$$\min_{\mathbf{w},b} \sum_i \xi_i + \frac{1}{2C}||\mathbf{w}||^2. \tag{11}$$

where

$$\xi_i = \max\big(0, 1 - y_i(\mathbf{w}^T \mathbf{z}_i - b)\big).$$

Now, we view the problem as minimizing the hinge loss, but to which the penalty term $\frac{1}{2C}||\mathbf{w}||^2$ has been added. The length-squared of $\mathbf{w}$, *i.e.,* $||\mathbf{w}||^2$ is called the $\ell_2$ penalty. We read (11) as "minimize the hinge loss subject to a $\ell_2$ penalty" or "minimize the hinge loss with $\ell_2$ regularization".

Consider what would have happened if we had the following constrained optimization problem

$$\min_{\mathbf{w},b} \sum_i \xi_i \text{ subject to } ||\mathbf{w}||^2 \leq k \tag{12}$$

The Lagrangian for the above problem is exactly

$$\sum_i \xi_i + \lambda(||w||^2 - k),$$

and in the dual formulation, we would have solved

$$\max_{\lambda \geq 0} \min_{\mathbf{w},b} \sum_i \xi_i + \lambda(||w||^2 - k). \tag{13}$$

Now, (12) minimizes a convex function (hinge loss) subject to convex constraints, and in this case the dual formulation above is completely equivalent to the original (12).

When we do the inner minimization in (13), we will set the gradient with respect to $\mathbf{w}$ to 0. That is exactly what we do to solve (11). Now when we subsequently solve the outer maximization, we get a value of $\lambda$ that depends on $k$.

Therefore, solving (11) is *exactly* what we would have done for solving the dual formulation (13) for that value of $k$ which would yield $\lambda = \frac{1}{2C}$. In other words, a different way to see the entire general support vector machine formulation is as the constrained minimization (12): minimize the hinge loss subject to $\ell_2$ regularization.

## 2.2 Comparison with the linearly separable case

The linear separable case is actually the limit as $C \to \infty$. You may be tempted to see the linear separable case as the situation where $C = 0$, but it is not. To see why, if $C = 0$, any misclassification incurs no penalty at all. Then we hit the smallest value of the objective ($\frac{1}{2}||\mathbf{w}||^2$) by picking $\mathbf{w} = 0$, $b = 1$ and setting $\xi_i = 1 + y_i b$ (note that $1 + y_i b = 1 + y_i \geq 0$ always since $y_i$ the label can only take on values +1 or -1).

In the limiting case where $C \to \infty$, we force $\xi_i = 0$ for all $i$ (even a single $\xi_i > 0$ makes $\arg\min \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^n \xi_i$ blow up to infinity. Then the problem reduces to the linearly separable setup.

**Exercise 2** As we have seen, in the general (non-linearly separable) case, given training examples $\mathbf{z}_1, \ldots, \mathbf{z}_n$ with labels $y_1, \ldots, y_n$, we solve the optimization by introducing a slack $\xi_i$ for each example $\mathbf{z}_i$. Specifically, we solve (10)

1. Derive the Lagrangian for this constrained optimization problem.
2. Derive the primal (minmax) form of this optimization optimization.
3. Show that the dual form of the optimization in terms of only the Lagrange coefficients and the examples is

$$\max_{0\le\Lambda\le C} \sum_{i=1}^{n} \lambda_i - \frac{1}{2}\sum_{j=1}^{n}\sum_{k=1}^{n} \lambda_j\lambda_k y_j y_k \mathbf{z}_j^T \mathbf{z}_k, \tag{14}$$

   where $0 \le \Lambda \le C$ is shorthand that means $0 \le \lambda_i \le C$ for all $1 \le i \le n$, and $\lambda_i$ is the Lagrange multiplier of the constraint

$$1 - \xi_i - y_i(\mathbf{w}^T\mathbf{z}_i - b) \le 0$$

   corresponding to the example $\mathbf{z}_i$. Namely, eliminate $\mathbf{w}, b$ and the slacks $\xi_i$ to get the above just as we eliminated $\mathbf{w}, b$ to get to (8). □

From (14), we conclude that in the general case,

- $\lambda_i$ depends on the examples only through their pairwise dot products, as is evident from the dual formulation (14);
- (You need to show this) The weights $\mathbf{w}$ is a linear combination $\sum \lambda_i y_i \mathbf{z}_i$ just as before, only difference is that $\lambda_i$ is obtained from (14) instead of (8). Once again, $b$ only depends on the pairwise dot products just as in Exercise 1;
- Suppose the Lagrange coefficient of the constraint $-\xi_i \le 0$ is $\nu_i$ and that of $y_i(\mathbf{w}^T\mathbf{z}_i - b) \ge 1 - \xi_i$ is $\lambda_i$. Then $\nu_i + \lambda_i = C$;
- The Karush-Kuhn-Tucker condition now requires $\nu_i\xi_i = 0$ in addition. So if $\nu_i > 0$, we must have $\xi_i = 0$. The examples such that $0 \le \lambda_i < C$ therefore will have $\nu_i = C - \lambda_i > 0$ and hence $\xi_i = 0$. These points are guaranteed to be classified properly;
- If a point has non-zero slack ($\xi_i > 0$) (including misclassified points), the Karush-Kuhn-Tucker condition $\nu_i\xi_i = 0$ will therefore ensure that $\nu_i = 0$ and therefore $\lambda_i = C$;
- The examples that have $\lambda_i > 0$ are the support vectors. There are two kinds of support vectors now—the ones on the margin (in the correct side) and the misclassified examples (these have $\lambda = C$ from the previous observation).