

## Lecture 10 — February 11

*Scribe: Avi Mehta**Lecturer: Deva Ramanan*

**Note:** These lecture notes are rough, and have only have been mildly proofread.

## 10.1 Lagrangian duality

### 10.1.1 Recall

We are working with a dataset that is linearly separable. We want to maximize the minimum distance of a data points from the separator. Mathematically, it can be formulated as:

$$\min \frac{1}{2} \|w\|^2 \quad (10.1)$$

$$\text{s.t. } y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad (10.2)$$

These equations are of the form:

$$\min f(w) \quad (10.3)$$

$$\text{s.t. } g_i(w) \leq 0 \quad (10.4)$$

In last lecture, we had claimed that the equations 10.3 and 10.4 above can be rewritten in the form:

$$\min_w \max_{\alpha \geq 0} \mathcal{L}(w, \alpha), \text{ where} \quad (10.5)$$

$$\mathcal{L}(w, \alpha) = f(w) + \sum \alpha_i g_i(w) \quad (10.6)$$

### 10.1.2 Weak versus strong duality

For all  $f(w)$  and  $g_i(w)$ , the following holds:

$$\max_{\alpha \geq 0} \min_w \mathcal{L}(w, \alpha) \leq \min_w \max_{\alpha \geq 0} \mathcal{L}(w, \alpha) \quad (10.7)$$

The above property, called weak duality, was proven in the previous lecture notes. Under certain conditions on  $f(w)$  and  $g(w)$ , equality holds and the problem is said to exhibit strong duality. If  $f(w)$  is convex and  $g_i(w)$  are affine then strong duality holds. Alternatively, if  $f(w)$  is convex and  $g_i(w)$  are convex and there exists at least one feasible  $w$ , then equality also holds.

## 10.2 Support Vector Machines

Applying above equations to the case of Support Vector Machines, we have:

$$\max_{\alpha \geq 0} \left[ \min_{w, b} \mathcal{L}(w, b, \alpha) \right] \quad (10.8)$$

$$(10.9)$$

In the above equation, the term inside square brackets can then be easily minimized over  $w$  and  $b$  for each value of  $\alpha$ . Thus, optimizing based on these principles, we get:

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow w = \sum_i \alpha_i y^{(i)} x^{(i)} \quad (10.10)$$

$$\frac{\partial \mathcal{L}}{\partial w} = 0 \Rightarrow \sum_i \alpha_i y^{(i)} = 0 \quad (10.11)$$

Now we have the final problem of maximization of expression over all  $\alpha \geq 0$ . This maximization can be written as:

$$\max_{\alpha \geq 0} \sum_i \alpha_i - \frac{1}{2} \sum_{i, j} \alpha_i \alpha_j y^{(i)} y^{(j)} x^{(i)T} x^{(j)} \quad (10.12)$$

$$s.t. \sum_i \alpha_i y^{(i)} = 0 \quad (10.13)$$

### 10.2.1 KKT Conditions

For the  $i^{th}$  point in a given dataset, we have  $\alpha_i g_i(w) = 0$ . This means either  $\alpha_i = 0$  or  $g_i(w) = 0$ . This implies the following conditions

$$if \alpha_i = 0 \quad then \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 \quad (10.14)$$

$$if \alpha_i > 0 \quad then \quad y^{(i)}(w^T x^{(i)} + b) = 1 \quad (10.15)$$

In the above formulation, The  $\alpha_i$ 's  $> 0$  are called support vectors because they are the active constraints that determine the final decision boundary. Intuitively, these are points at the margin whose constraints are active at the final solution.

Consider a given a dataset  $\{x^{(i)}, y^{(i)}\}$  and the  $w^*$  that maximizes the margin for that data. One could remove all the points strictly beyond the margin  $y^{(i)}(w^T x^{(i)} + b) > 1$ , recompute the solution, and will obtain the same  $w^*$ .

### 10.2.2 Observations

We can divide the set of all  $\alpha$  into two parts corresponding to positive and negative training examples. Thus we have,

$$w = \sum_i \alpha_i y^{(i)} x^{(i)} \quad (10.16)$$

$$= \sum_{i \in pos} \alpha_i x^{(i)} - \sum_{i \in neg} \alpha_i x^{(i)} \quad (10.17)$$

The above implies the final weight vector  $w$  lies in the span of the data points  $x^{(i)}$ . We can similarly split up the terms for the following constraint

$$\sum_i \alpha_i y^{(i)} = 0 \quad (10.18)$$

$$\Rightarrow \sum_{i \in pos} \alpha_i = \sum_{i \in neg} \alpha_i \quad (10.19)$$

The above equations imply that the influence of the points on the positive and the negative side of the decision boundary are equal.

It is useful to relate an SVM to a class-conditional gaussian model. Recall that, for spherically-distributed classes, a class-conditional gaussian model fit a decision boundary whose normal  $w$  is a line connecting the average class 0 point to the average class 1 point. An SVM does something similar, but computes a weighted average of points at the boundary. Specifically, the weights are given by  $\alpha_i$  and the boundary points are the support vectors.

### 10.2.3 Summarizing

We rewrite our optimization problem in dual form because:

1) We can examine useful properties of the final solution, such as sparsity and the balanced influence of positives and negatives. 2) We can exploit sparsity during the optimization. We will later discuss an algorithm (SMO) that does this. 3) We can apply kernel "tricks" to learn linear boundaries in very high dimensional feature spaces. We discuss it in following section.

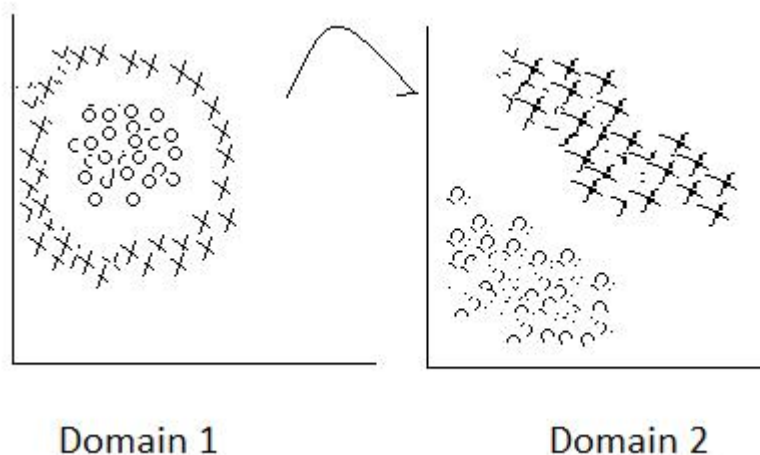
## 10.3 Kernels

Recall that we have:

$$h(w) = w^T x + b \quad (10.20)$$

$$= (\sum_i \alpha_i y^{(i)} x^{(i)T}) x + b \quad (10.21)$$

$$= \sum_i \alpha_i y^{(i)} \langle x^{(i)T}, x^{(i)} \rangle + b \quad \text{where } \langle x, z \rangle = x^T z. \quad (10.22)$$



**Observation:** To both train the model and use it, we don't need the actual data points. We just need to be able to compute inner products between vectors  $\langle x^{(i)T}, x^{(i)} \rangle$ .

**Kernel trick:** Replace  $\langle x, z \rangle$  with  $k(x, z)$  where  $k$  is a *kernel function* that corresponds to an inner product in some feature space - i.e.,  $k(x, z) = \langle \phi(x), \phi(z) \rangle$ . For example, here is one valid kernel function

$$k(x, z) = (x^T z)^2 \quad (10.23)$$

What is the feature space that corresponds to this kernel? We can answer that by expanding the RHS (assuming  $x, z \in R^2$ )

$$\begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} = (x_1 z_1 + x_2 z_2)^2 \quad (10.24)$$

$$= x_1^2 z_1^2 + 2x_1 x_2 z_1 z_2 + x_2^2 z_2^2 \quad (10.25)$$

$$= \begin{bmatrix} x_1^2 & \sqrt{2}x_1 x_2 & x_2^2 \end{bmatrix} \begin{bmatrix} z_1^2 \\ \sqrt{2}z_1 z_2 \\ z_2^2 \end{bmatrix} \quad (10.26)$$

$$= \phi(x)^T \phi(z) \quad (10.27)$$

A linear boundary in the feature space  $\phi(x)$  can represent a quadratic boundary in the original space. For example, a circle can be represented as  $w^T \phi(x) + b = 0$ . See figure (10.3). We can also use kernels that map into a feature space  $\phi(x)$  that is a large number, or even an infinite number of dimensions. Some common kernels are given below

$$k(x, z) = (1 + x^T z)^n \text{ Polynomial kernel} \quad (10.28)$$

$$k(x, z) = e^{-\frac{1}{2\sigma^2} \|x - z\|^2} \text{ Gaussian kernel} \quad (10.29)$$

The Gaussian kernel uses a feature space that is infinite dimensional. The best way to think about that is for any given dataset the gaussian kernel, with an appropriate  $\sigma$ , can find a

linear separator. This is because it has an infinite choice of dimensions to split the data with.

A given function  $k(x, z)$  is a valid kernel if and only if it corresponds to an inner product in some space. This is equivalent to satisfying Mercer's conditions

1.  $k$  is symmetric -  $k(x, z) = k(z, x)$
2. For any set of points  $x^{(i)} \dots x^{(m)}$ , the associated kernel matrix  $K$  where  $K_{ij} = k(x^{(i)}, x^{(j)})$  is positive semidefinite -  $v^T K v \geq 0$  for any  $v$ .

One can show the second condition is necessary by

$$v^T K v = \sum_i \sum_j v_i^T K_{ij} v_j \quad (10.30)$$

$$= \sum_i \sum_j v_i^T \phi(x^{(i)})^T \phi(x^{(j)}) v_j \quad (10.31)$$

$$= \sum_i v_i^T \phi(x^{(i)})^T \sum_j \phi(x^{(j)}) v_j \quad (10.32)$$

$$= (\sum_i v_i \phi(x^{(i)}))^2 \geq 0 \quad (10.33)$$

We can define kernels on things that do not even live in vector space. For example, consider strings.

$x_1$  = "The boy ran home."

$x_2$  = "Boys and girls."

$k(x_1, x_2)$  = number of substrings in common between  $x_1, x_2$ .

We can show that its a valid kernel since it is a dot product between vectors of indicator for all possible strings.

**Note:** Many algorithms that studied in this course can be kernelized, including linear and logistic regression. We will also discuss linear subspace methods such as principle component analysis and linear discriminant analysis (which we have briefly described). In all these cases the learned model can be represented as a weight vector that lies in the span of the original data points - i.e.,  $w = \sum_i \alpha_i x^{(i)}$  for some  $\alpha_i$ . Also in these cases, the prediction for a new point enters in the model through an inner product  $h(x) = g(w^T x)$  for some function  $g$ . This suggests the algorithms can be kernelized with  $w^T x = \sum_i \alpha_i k(x^{(i)}, x)$ .

But one disadvantage of kernelizing an algorithm is that now the model is nonparametric. It grows with the size of the dataset. We need to keep around all our original data points  $x^{(i)}$  to make a prediction. This is roughly the same storage and computational requirement as nearest neighbors, so this can get expensive for large datasets. An SVM, however, yeilds a sparse solution in terms of the data points. So we only need to retain the *support vectors* instead of all the points. This makes kernels and SVMs a nice fit.

## 10.4 Next class - soft margin SVM

In general, we also find outliers along with the sample points. These outliers can exist within margin from the separator or even be misclassified. To prevent outliers from affecting the

separator, we introduce a slack variable for each point. The equation can be rewritten with slack as:

$$\min \frac{1}{2} \|w\|^2 + C \sum \xi_i \quad \& \quad y^{(i)}(w^T x^{(i)} + b) \geq 1 - \xi_i \quad (10.34)$$

$$\xi_i \geq 0 \quad (10.35)$$

Thus the new lagrangian can be rewritten as:  $\mathcal{L}(w, b, \xi_i, \alpha_i, r_i)$ . Where  $r_i$  is the lagrangian multiplier corresponding to the  $\xi_i \geq 0$  constraints.