Lecture 18

An important task in machine learning is classification: given a training set of points $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_p$, with $\boldsymbol{x}_i \in \mathbb{R}^p$, and associated labels y_i (for example, -1 and 1, or more labels), use this data to estimate a function $f(\boldsymbol{x})$ that assigns to each new vector $\boldsymbol{x}$ a label. As we have seen earlier, example include spam filters, letter recognition, or text classification. In this lecture we discuss a very influential method for classification, **Support Vector Machines (SVMs)**, from the point of view of convex optimization.

18.1 Linear Support Vector Machines

The simplest case is when the set of labels is $\mathcal{Y} = \{-1, 1\}$ and the set of training points $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$ is *linearly separable*: this means that there exists an affine hyperplane $h(\boldsymbol{x}) = \boldsymbol{w}^\top \boldsymbol{x} + b$ such that $h(\boldsymbol{x}_i) > 0$ if $y_i = 1$ and $h(\boldsymbol{x}_j) < 0$ if $y_j < 0$. We call the points for which $y_i = 1$ *positive*, and the ones for which $y_j = -1$ *negative*. The problem of finding such a hyperplane can be posed as a linear programming feasibility problem as follows: we look for a vector of *weights* $\boldsymbol{w}$ and a *bias term* b (together a $(p+1)$-dimensional vector) such that

$$\boldsymbol{w}^\top \boldsymbol{x}_i + b \geq 1, \text{ for } y_i = 1, \quad \boldsymbol{w}^\top \boldsymbol{x}_j + b \leq -1, \text{ for } y_j = -1.$$

Note that we can replace the $+1$ and -1 with any other positve or negative quantity by rescaling the $\boldsymbol{w}$ and $\boldsymbol{b}$, so this is just convention. We can also describe the two inequalities concisely as

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 \geq 0. \tag{18.1}$$

A hyperplane separating the two point sets will in general not be unique. As we want to use the linear classifier on new, yet unknown data, we want to find a separating hyperplane with best possible **margin**. Let d_+ and d_- denote the distance of a separating hyperplane to the closest positive and closest negative point, respectively. The quantity $d = d_+ + d_-$ is then called the margin or the classifier, and we want to find a hyperplane with largest possible margin.

Given a hyperplane H described in (18.1) and a point $\boldsymbol{x}$ such that we have the equality $\boldsymbol{w}^\top \boldsymbol{x}_i + b = 1$ (the point is as close as possible to the hyperplane, also called a **support vector**), the distance of that point to the hyperplane can be computed by

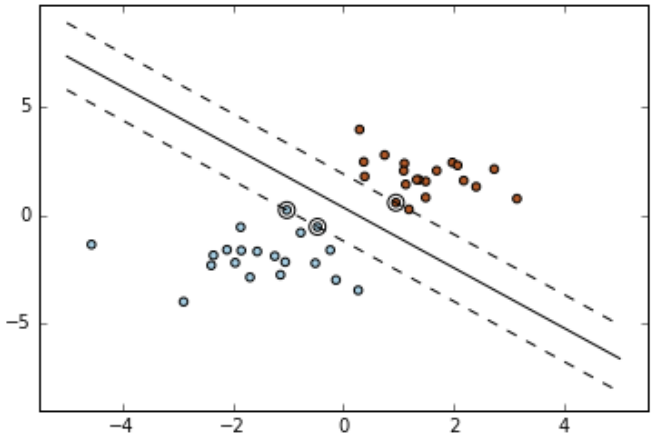

Figure 18.1: A hyperplane separating two sets of points with margin and support vectors.

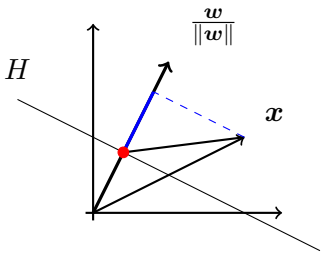

Figure 18.2: Computing the distance to the hyperplane

first taking the difference of $\boldsymbol{x}$ with a point $\boldsymbol{p}$ on H (an *anchor*), and then computing the dot product of $\boldsymbol{x} - \boldsymbol{p}$ with the unit vector $\boldsymbol{w}/\|\boldsymbol{w}\|$ orthogonal to H.

As anchor point $\boldsymbol{p}$ we can just choose a multiple $c\boldsymbol{w}$ that is on the plane, i.e., that satisfies $\langle \boldsymbol{w}, c\boldsymbol{w} \rangle + b = 0$. This implies that $c = -b/\|\boldsymbol{w}\|^2$, and consequently $\boldsymbol{p} = -(b/\|\boldsymbol{w}\|^2\boldsymbol{w}$. The distance is then

$$d_+ = \langle \boldsymbol{x} + \frac{b}{\|\boldsymbol{w}\|^2}\boldsymbol{w}, \frac{\boldsymbol{w}}{\|\boldsymbol{w}\|} \rangle = \frac{1}{\|\boldsymbol{w}\|}.$$

Similarly, we get $d_- = 1/\|\boldsymbol{w}\|$. The margin of this particular separating hyperplane is thus $d = 2/\|\boldsymbol{w}\|$. If we want to find a hyperplane with *smallest* margin, we thus have to solve the quadratic optimization problem

$$\begin{aligned} &\text{minimize} && \frac{1}{2}\|\boldsymbol{w}\|^2 \\ &\text{subject to} && y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) - 1 \geq 0, \quad 1 \leq i \leq n. \end{aligned}$$

Note that b is also an unknown variable in this problem! The factor $1/2$ in the objective function is just to make the gradient look nicer. The Lagrangian of this problem is

$$\begin{aligned}\mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \frac{1}{2}\|\boldsymbol{w}\|^2 - \sum_{i=1}^{m} \lambda_i y_i \boldsymbol{w}^\top \boldsymbol{x}_i - \lambda_i y_i b + \lambda_i \\ &= \frac{1}{2}\boldsymbol{w}^\top \boldsymbol{w} - \boldsymbol{\lambda}^\top \boldsymbol{X}\boldsymbol{w} - b\boldsymbol{\lambda}^\top \boldsymbol{y} + \sum_{i=1}^{m} \lambda_i,\end{aligned}$$

where we denote by $\boldsymbol{X}$ the matrix with the $y_i \boldsymbol{x}_i^\top$ as rows. We can then write the conditions on the gradient with respect to $\boldsymbol{w}$ and b of the Lagrangian as

$$\begin{aligned}\nabla_{\boldsymbol{w}} \mathcal{L}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \boldsymbol{w} + \boldsymbol{X}^\top \boldsymbol{\lambda} = \boldsymbol{0} \\ \frac{\partial \mathcal{L}}{\partial b}(\boldsymbol{w}, b, \boldsymbol{\lambda}) &= \boldsymbol{y}^\top \boldsymbol{\lambda} = 0.\end{aligned} \tag{18.2}$$

Replacing $\boldsymbol{w}$ by $-\boldsymbol{X}^\top \boldsymbol{\lambda}$ and $\boldsymbol{\lambda}^\top \boldsymbol{y}$ by 0 in the Lagrangian function then gives the expression for the Lagrange dual $g(\boldsymbol{\lambda})$,

$$g(\boldsymbol{\lambda}) = -\frac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\lambda} - \sum_{i=1}^{m} \lambda_i.$$

Finally, changing the sign and the maximum with a minimum, we can formulate the Lagrange dual optimization problem as

$$\text{minimize} \quad \frac{1}{2}\boldsymbol{\lambda}^\top \boldsymbol{X}\boldsymbol{X}^\top \boldsymbol{\lambda} + \boldsymbol{\lambda}^\top \boldsymbol{e} \quad \text{subject to} \quad \boldsymbol{\lambda} \geq \boldsymbol{0}, \tag{18.3}$$

where $\boldsymbol{e}$ is the vector of all ones.

Note that there is one dual variable λ_i per data point $\boldsymbol{x}_i$. We can find the optimal value by solving the dual problem (18.3), but that does not give us automatically the weights $\boldsymbol{w}$ and the bias b. We can find the weights by $\boldsymbol{w} = -\boldsymbol{X}^\top \boldsymbol{\lambda}$. As for b, this is best determined from the KKT conditions of the problem. These can be written by combining the constraints of the primal problem with the conditions on the gradient of the Lagrangian (18.2), the condition $\boldsymbol{\lambda} \geq \boldsymbol{0}$, and complementary slackness as

$$\begin{aligned}\boldsymbol{X}\boldsymbol{w} + b\boldsymbol{y} - \boldsymbol{e} &= \boldsymbol{0} \\ \boldsymbol{\lambda} &\geq \boldsymbol{0} \\ \lambda_i(1 - y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b)) &= 0 \text{ for } 1 \leq i \leq n \\ \boldsymbol{w} + \boldsymbol{X}^\top \boldsymbol{\lambda} &= \boldsymbol{0} \\ \boldsymbol{y}^\top \boldsymbol{\lambda} &= 0.\end{aligned}$$

To get b, we can choose one of the equations in which $\lambda_i \neq 0$, and then find b by setting $b = y_i(1 - y_i \boldsymbol{w}^\top \boldsymbol{x}_i)$. With the KKT conditions written down, we can go about solving the problem of finding a maximum margin linear classifier using methods such as the barrier method.

Example 18.1. To be written.

18.2 Extensions

So far we looked at the particularly simple case where (a) the data falls into two classes, (b) the points can actually be well separated, and (c) they can be separated by an affine hyperplane. In reality, these three assumptions may not hold. We briefly discuss extensions of the basic model to account for the three situations just mentioned.

Non-exact separation

Non-linear separation and kernels

Multiple classes