CS 189 Introduction to Machine Learning
Spring 2018

Note 20

1 Support Vector Machines

So far we've explored **generative classifiers** (LDA) and **discriminative classifiers** (logistic regression), but in both of these methods, we tasked ourselves with modeling some kind of probability distribution. One observation about classification is that in the end, if we only care about assigning each data point a class, all we really need to know do is find a "good" decision boundary, and we can skip thinking about the distributions. **Support Vector Machines (SVMs)** are an attempt to model decision boundaries directly in this spirit.

Here's the setup for the problem. We are given a training dataset $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{-1, +1\}$. Our goal is to find a $d-1$ dimensional **hyperplane** decision boundary H which separates the $+1$'s from the -1's.

1.1 Motivation for SVMs

In order to motivate SVMs, we first have to understand the simpler **perceptron** classifier and its shortcomings. Given that the training data is **linearly separable**, the perceptron algorithm finds a $d-1$ dimensional hyperplane that perfectly separates the $+1$'s from the -1's. Mathematically, the goal is to learn a set of parameters $\mathbf{w} \in \mathbb{R}^d$ and $b \in \mathbb{R}$, that satisfy the linear separability constraints:

$$\forall i, \quad \begin{cases} \mathbf{w}^\top \mathbf{x}_i - b \geq 0 & \text{if } y_i = 1 \\ \mathbf{w}^\top \mathbf{x}_i - b \leq 0 & \text{if } y_i = -1 \end{cases}$$

Equivalently,

$$\forall i, \quad y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 0$$

The resulting decision boundary is a hyperplane $H = \{\mathbf{x} : \mathbf{w}^\top \mathbf{x} - b = 0\}$. All points on the positive side of the hyperplane are classified as $+1$, and all points on the negative side are classified as -1.

Note that perceptrons have two major shortcomings that as we shall see, SVMs can overcome. First of all, if the data is not linearly separable, the perceptron fails to find a stable solution. As we shall see, soft-margin SVMs fix this issue by allowing best-fit decision boundaries even when the data is not linearly separable. Second, if the data is linearly separable, the perceptron could find infinitely many decision boundaries — if $(\mathbf{w}, b)$ is a pair that separates the data points, then the perceptron could also end up choosing a slightly different $(\mathbf{w}, b + \epsilon)$ pair. Some hyperplanes are better than others, but the perceptron cannot distinguish between them. This leads to generalization issues.

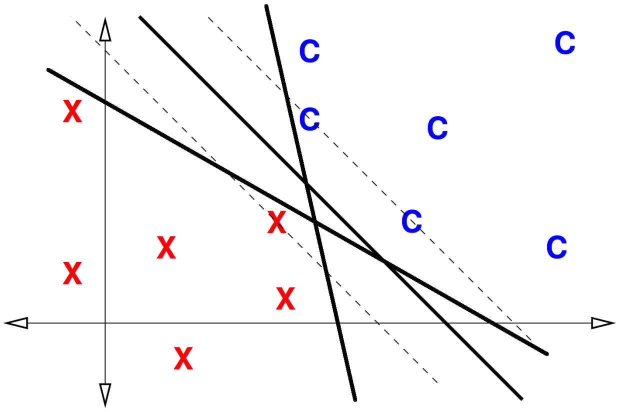

Figure 1: Several possible decision boundaries under the perceptron. The X's and C's represent the $+1$'s and -1's respectively.

In the figure above, we consider three potential linear separators that satisfy the constraints. One could imagine that if we observed new test points that are nearby the region of C's in the training data, they should also be of class C. One separator would incorrectly classify some of these new test points, while the others would most likely still be able to classify them correctly. To the eyes of the perceptron algorithm, all lines are perfectly valid decision boundaries. Therefore, the perceptron may not be able to generalize well to unseen data.

1.2 Hard-Margin SVMs

Hard-Margin SVMs solve the generalization problem of perceptrons by maximizing the **margin**, formally known as the minimum distance from the decision boundary to any of the training points.

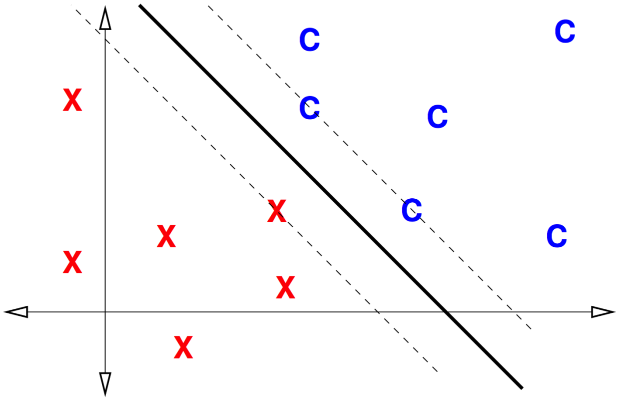

Figure 2: The optimal decision boundary (as shown) maximizes the margin.

Intuitively, maximizing the margin allows us to generalize better to unseen data, because the decision boundary with the maximum margin is as far away from the training data as possible and the boundary cannot be violated unless the unseen data contains outliers.

Simply put, the goal of hard-margin SVMs is to find a hyperplane H that maximizes the margin m. Let's formalize an optimization problem for hard-margin SVMs. The variables we are trying to optimize over are the margin m and the parameters of the hyperplane, $\mathbf{w}$ and b. The objective is to maximize the margin m, subject to the following constraints:

- All points classified as $+1$ are to the positive side of the hyperplane and their distance to H is greater than the margin
- All points classified as -1 are to the negative side of the hyperplane and their distance to H is greater than the margin
- The margin is non-negative.

Let's express the first two constraints mathematically.

First, note that the vector $\mathbf{w}$ is perpendicular to the hyperplane $H = \{\mathbf{x} : \mathbf{w}^\top \mathbf{x} - b = 0\}$.

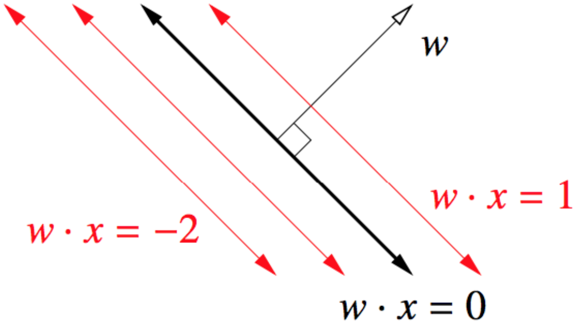

Figure 3: Image courtesy Professor Shewchuk's notes.

Proof: consider any two points on H, $\mathbf{x}_0$ and $\mathbf{x}_1$. We will show that $(\mathbf{x}_1 - \mathbf{x}_0) \perp \mathbf{w}$. Note that

$$(\mathbf{x}_1 - \mathbf{x}_0)^\top(\mathbf{w}) = (\mathbf{x}_1 - \mathbf{x}_0)^\top((\mathbf{x}_1 + \mathbf{w}) - \mathbf{x}_1) = \mathbf{x}_1{}^\top \mathbf{w} - \mathbf{x}_0{}^\top \mathbf{w} = b - b = 0$$

Since $\mathbf{w}$ is perpendicular to H, the (shortest) distance from any arbitrary point $\mathbf{z}$ to the hyperplane H is determined by a scaled multiple of $\mathbf{w}$. If we take any point on the hyperplane $\mathbf{x}_0$, the distance from $\mathbf{z}$ to H is the length of the projection from $\mathbf{z} - \mathbf{x}_0$ to the vector $\mathbf{w}$, which is

$$D = \frac{|\mathbf{w}^\top(\mathbf{z} - \mathbf{x}_0)|}{\|\mathbf{w}\|_2} = \frac{|\mathbf{w}^\top \mathbf{z} - \mathbf{w}^\top \mathbf{x}_0|}{\|\mathbf{w}\|_2} = \frac{|\mathbf{w}^\top \mathbf{z} - b|}{\|\mathbf{w}\|_2}$$

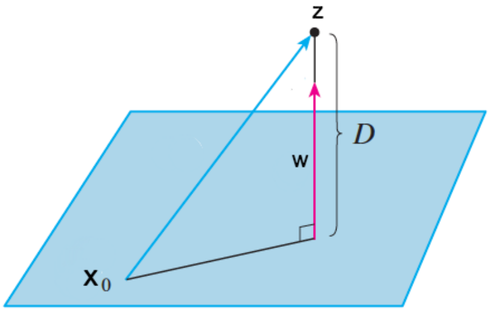

Figure 4: Shortest distance from z to H is determined by projection of $z - \mathbf{x}_0$ onto $\mathbf{w}$

Therefore, the distance from any of the training points $\mathbf{x}_i$ to H is

$$\frac{|\mathbf{w}^\top \mathbf{x}_i - b|}{\|\mathbf{w}\|_2}$$

In order to ensure that positive points are on the positive side of the hyperplane outside a margin of size m, and that negative points are on the negative side of the hyperplane outside a margin of size m, we can express the constraint

$$y_i \frac{(\mathbf{w}^\top \mathbf{x}_i - b)}{\|\mathbf{w}\|_2} \geq m$$

Putting everything together, we have the following optimization problem:

$$\begin{aligned} \max_{m,\mathbf{w},b} \quad & m \\ \text{s.t.} \quad & y_i \frac{(\mathbf{w}^\top \mathbf{x}_i - b)}{\|\mathbf{w}\|_2} \geq m \quad \forall i \\ & m \geq 0 \end{aligned} \tag{1}$$

Maximizing the margin m means that there exists at least one point on the positive side of the hyperplane and at least one point on the negative side whose distance to the hyperplane is exactly equal to m. These points are the **support vectors**, hence the name “support vector machines.”

Through a series of optimization steps, we can simplify the problem by removing the margin variable and just optimizing the parameters of the hyperplane. In order to do so, we have to first introduce two new variables $\mathbf{w}'$ and b' that capture the relationship between the three original variables m, $\mathbf{w}$, and b.

$$\begin{aligned} \max_{m,\mathbf{w},b,\mathbf{w}',b'} \quad & \frac{1}{\|\mathbf{w}'\|_2} \\ \text{s.t.} \quad & y_i(\mathbf{w}'^\top \mathbf{x}_i - b') \geq 1 \quad \forall i \\ & m \geq 0 \\ & \mathbf{w}' = \frac{\mathbf{w}}{\|\mathbf{w}\|_2 m} \\ & b' = \frac{b}{\|\mathbf{w}\|_2 m} \end{aligned} \tag{2}$$

Having introduced the new variables $\mathbf{w}'$ and b', the old variables m, $\mathbf{w}$, and b are no longer relevant to the optimization problem, and we can remove them. The previous optimization problem is equivalent to

$$\begin{aligned} \max_{\mathbf{w}',b'} \quad & \frac{1}{\|\mathbf{w}'\|_2} \\ \text{s.t.} \quad & y_i(\mathbf{w}'^\top \mathbf{x}_i - b') \geq 1 \quad \forall i \end{aligned} \tag{3}$$

Let's verify that (2) and (3) are equivalent. We will show that

1. The optimal value of (2) is at least as good as the optimal value of (3). Assume that the optimal values for (3) are $\mathbf{w}'^*$ and b'^*. One feasible point for (2) is $(m, \mathbf{w}, b, \mathbf{w}', b') = (\frac{1}{\|\mathbf{w}'\|_2}, \mathbf{w}'^*, b'^*, \mathbf{w}'^*, b'^*)$, which leads to the same objective value as (3). Therefore, the optimal value of (2) is at least as good as that of (3).

2. The optimal value of (3) is at least as good as the optimal value of (2). Assume that the optimal values for (2) are $(m^*, \mathbf{w}^*, b^*, \mathbf{w}'^*, b'^*)$. One feasible point for (3) is $(\mathbf{w}', b') = (\mathbf{w}'^*, b'^*)$ which leads to the same objective value as (2). Therefore, the optimal value of (3) is at least as good as that of (2).

We can rewrite objective so that the problem is a minimization rather than a maximization:

$$\begin{aligned} \min_{\mathbf{w}',b'} \quad & \frac{1}{2}\|\mathbf{w}'\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}'^\top \mathbf{x}_i - b') \geq 1 \quad \forall i \end{aligned} \tag{4}$$

At last, we have formulated the hard-margin SVM optimization problem! Using the notation $\mathbf{w}$ and b, the objective of hard-margin SVMs is

$$\begin{aligned} \min_{\mathbf{w},b} \quad & \frac{1}{2}\|\mathbf{w}\|_2^2 \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 \quad \forall i \end{aligned} \tag{5}$$

1.3 Soft-Margin SVMs

The hard-margin SVM optimization problem has a unique solution only if the data are linearly separable, but it has no solution otherwise. This is because the constraints are impossible to satisfy if we can't draw a hyperplane that separates the $+1$'s from the -1's. In addition, hard-margin SVMs are very sensitive to outliers – for example, if our data is class-conditionally distributed Gaussian such that the two Gaussians are far apart, if we witness an outlier from class $+1$ that crosses into the typical region for class -1, then hard-margin SVM will be forced to compromise a more generalizable fit in order to accommodate for this point. Our next goal is to come up with a classifier that is not sensitive to outliers and can work even in the presence of data that is not linearly separable. To this end, we'll talk about **Soft-Margin SVMs**.

A soft-margin SVM modifies the constraints from the hard-margin SVM by allowing some points to violate the margin. Formally, it introduces **slack variables** ξ_i, one for each training point, into the constraints:

$$\begin{aligned} y_i(\mathbf{w}^\top \mathbf{x}_i - b) &\geq 1 - \xi_i \\ \xi_i &\geq 0 \end{aligned}$$

which, is a less-strict, *softer* version of the hard-margin SVM constraints because it says that each point $\mathbf{x}_i$ need only be a "distance" of $1-\xi_i$ of the separating hyperplane instead of a hard "distance" of 1.

(By the way, the Greek letter ξ is spelled "xi" and pronounced "zai." ξ_i is pronounced "zai-eye.")

These constraints would be fruitless if we didn't bound the values of the ξ_i's, because by setting them to large values, we are essentially saying that any point may violate the margin by an arbitrarily large distance...which makes our choice of $\mathbf{w}$ meaningless. We modify the objective function to be:

$$\min_{\mathbf{w},b,,\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i$$

Where C is a hyperparameter tuned through cross-validation. Putting the objective and constraints together, the soft-margin SVM optimization problem is

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{6}$$

The table below compares the effects of having a large C versus a small C. As C goes to infinity, the penalty for having non-zero ξ_i goes to infinity, and thus we force the ξ_i's to be zero, which is exactly the setting of the hard-margin SVM.

	small C	large C
Desire	maximize margin	keep ξ_i's small or zero
Danger	underfitting	overfitting
Outliers	less sensitive	more sensitive

1.4 SVMs as Tikhonov Regularization Learning

Consider the following regularized regression problem:

$$\min_{\mathbf{w},b} \frac{1}{n}\sum_{i=1}^{n} L(y_i, \mathbf{w}^\top \mathbf{x}_i - b) + \lambda\|\mathbf{w}\|^2$$

In the context of classification, the loss function that we would like to optimize is 0-1 **step loss**:

$$L_{\text{STEP}}(y, \mathbf{w}^\top \mathbf{x} - b) = \begin{cases} 1 & y(\mathbf{w}^\top \mathbf{x} - b) < 0 \\ 0 & y(\mathbf{w}^\top \mathbf{x} - b) \geq 0 \end{cases}$$

The 0-1 loss is 0 if $\mathbf{x}$ is correctly classified and 1 otherwise. Thus minimizing $\frac{1}{n}\sum_{i=1}^n L(y_i, \mathbf{w}^\top \mathbf{x}_i - b)$ directly minimizes classification error on the training set. However, the 0-1 loss is difficult to optimize: it is neither convex nor differentiable (see Figure 5).

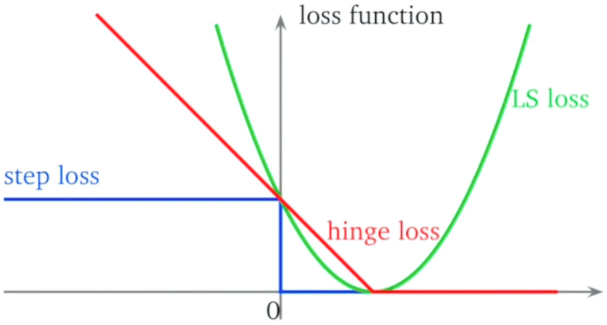

Figure 5: Step (0-1) loss, hinge loss, and squared loss. Squared loss is convex and differentiable, hinge loss is only convex, and step loss is neither.

We can try to modify the 0-1 loss to be convex. The points with $y(\mathbf{w}^\top \mathbf{x} - b) \geq 0$ should remain at 0 loss, but we may consider allowing a linear penalty "ramp" for misclassified points. This leads us to the **hinge loss**, as illustrated in Figure 5:

$$L_{\text{HINGE}}(y, \mathbf{w}^\top \mathbf{x} + b) = \max(1 - y(\mathbf{w}^\top \mathbf{x} - b), 0)$$

Thus the regularized regression problem becomes

$$\min_{\mathbf{w},b} \frac{1}{n}\sum_{i=1}^n \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0) + \lambda\|\mathbf{w}\|^2$$

Recall that the original soft-margin SVM optimization problem is

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & y_i(\mathbf{w}^\top \mathbf{x}_i - b) \geq 1 - \xi_i \quad \forall i \\ & \xi_i \geq 0 \quad \forall i \end{aligned} \tag{7}$$

We claim these two formulations are actually equivalent. Manipulating the first constraint, we have that

$$\xi_i \geq 1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b)$$

Combining with the constraint $\xi_i \geq 0$, we have that

$$\xi_i \geq \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0)$$

At the optimal value of the optimization problem, these inequalities must be tight. Otherwise, we could lower each ξ_i to equal $\max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0)$ and decrease the value of the objective function. Thus we can rewrite the soft-margin SVM optimization problem as

$$\begin{aligned} \min_{\mathbf{w},b,\xi_i} \quad & \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^n \xi_i \\ \text{s.t.} \quad & \xi_i = \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0) \quad \forall i \end{aligned} \tag{8}$$

Simplifying further, we can remove the constraints:

$$\min_{\mathbf{w},b} \quad \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{n} \max(1 - y_i(\mathbf{w}^\top \mathbf{x}_i - b), 0) \tag{9}$$

If we divide by Cn (which does not change the optimal solution of the optimization problem), we can see that this formulation is equivalent to the regularized regression problem, with $\lambda = \frac{1}{2Cn}$. Thus we have two interpretations of soft-margin SVM: either as finding a max-margin hyperplane that is allowed to make some mistakes via slack variables ξ_i, or as regularized empirical risk minimization with the hinge loss.