CS 181 Spring 2020 Section 5
Margin-Based Classification, SVMs

1 Motivation

In the past, with binary linear classifiers, we found a hyperplane that separated the data (or in cases when this was not possible separated a large amount of the data).

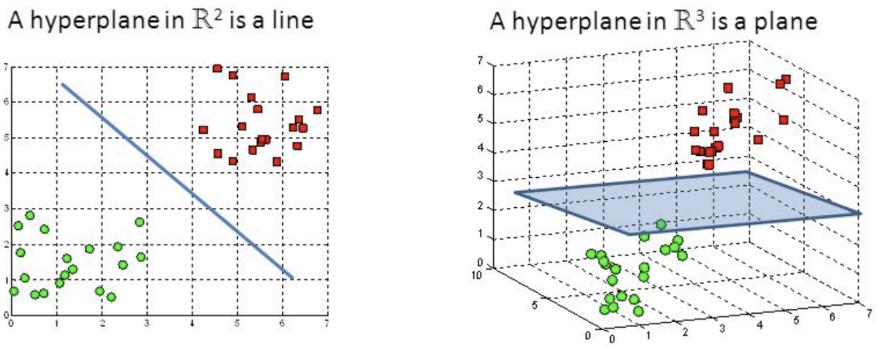
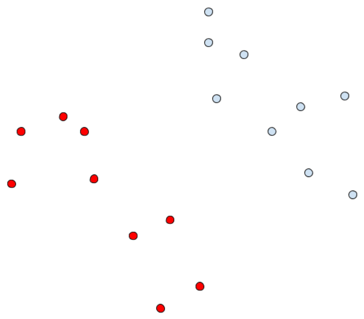


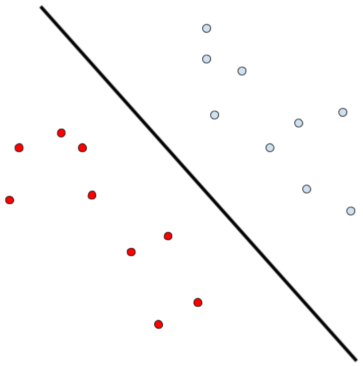
Figure 1: Source: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

However, if data is linearly separable, there may be many boundaries that work, so how can we know which is best?

Concept Question: Draw the "best" decision boundary for the data below (using your own definition of best). What would be a worse decision boundary? Why?



Solution: A best line would look something like the below. An important property of this line is that it "splits the difference" between the points, leaving a large boundary on both sides. We don't have any reason to put our line closer to either class of data points, so it would be illogical to do so.



As your intuition told you above, the idea for Support Vector Machines is that, for all the linear hyperplanes that exist, we want one that will create the largest distance, or "margin", with the training data. At a high level, we define the marign as the minimum distance between a point and our boundary. Larger margins tend to improve generalization error.

Now we'll put this into math. To find a mathematical formula for the margin, we consider a hyperplane of the form

$$\mathbf{w}^\top \mathbf{x} + w_0 = 0$$

For two points $\mathbf{x}_1$ and $\mathbf{x}_2$ on the hyperplane, consider the projection with $\mathbf{w}$:

$$\mathbf{w}^\top (\mathbf{x}_1 - \mathbf{x}_2) = \mathbf{w}^\top \mathbf{x}_1 - \mathbf{w}^\top \mathbf{x}_2 = -w_0 - (-w_0) = 0$$

Therefore, $\mathbf{w}$ is orthogonal to the hyperplane. So to get the distance from a hyperplane and an arbitrary example $\mathbf{x}$, we just need the length in the direction of $\mathbf{w}$ between the point and the hyperplane (this gives us the perpendicular distance between $\mathbf{x}$ and the hyperplane–why do we want the perpendicular distance?). We let r signify the distance between a point and the hyperplane. Then $\mathbf{x}_\perp$ is the projection of the point onto the hyperplane, so that we can decompose a point $\mathbf{x}$ as

$$\mathbf{x}_\perp + r\frac{\mathbf{w}}{||\mathbf{w}||} = \mathbf{x}$$

Left multiply by $\mathbf{w}^\top$:

$$\mathbf{w}^\top \mathbf{x}_\perp + r\frac{\mathbf{w}^\top \mathbf{w}}{||\mathbf{w}||} = \mathbf{w}^\top \mathbf{x} \Rightarrow r = \frac{\mathbf{w}^\top \mathbf{x} + w_0}{||\mathbf{w}||}$$

Scalar r then gives the signed, normalized distance between a point and the hyperplane. For correctly classified data, we have $y_i = +1$ when this distance is positive and $y_i = -1$ when it is negative. Based on this, we can obtain a positive distance for both kinds of examples by multiplying by y_i (and y_i will not change the magnitude). We define the margin of the dataset as the minimum such distance over all examples:

$$\min_i \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{||\mathbf{w}||}$$

Concept Question: Before optimizing a model, it's important to make sure you understand how it works. Give a new data point $\mathbf{x}$, what would a SVM predict (assume you know w^*)?

Solution: For a new data point $\mathbf{x}$ we would predict class 1 if $\mathbf{w}^{*T}\mathbf{x} + \mathbf{w}_0^* > 0$ and class -1 otherwise.

We now want to figure out how to find the optimal $\mathbf{w}$. As we discussed in motivating this model, we want the $\mathbf{w}$ and w_0 that maximize the margin:

$$\text{argmax}_{\mathbf{w},w_0} \frac{1}{||\mathbf{w}||} \min_i y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)$$

When discussing margin classifiers, we consider both hard and soft margin classifiers. In the hard-margin training problem, we know that the data is linearly separable and therefore any margin (including the optimal) must be greater than 0 (in the soft-margin problem, we do not make this assumption; this case is more complex so we deal with it later):

$$\min_i \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{||\mathbf{w}||} > 0$$

We can observe that $\mathbf{w}$ and w_0 are invariant to changes of scale. Because of this, it is without loss of generality to impose $\min_i \frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{||\mathbf{w}||} > 1$ (we prove this in the exercises of these notes). This lets us write the optimization problem as:

$$\text{argmax}_{\mathbf{w},w_0} \frac{1}{||\mathbf{w}||} \quad \text{s.t. } \forall i \; y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1$$

We can invert $\mathbf{w}$ to change the max to a min:

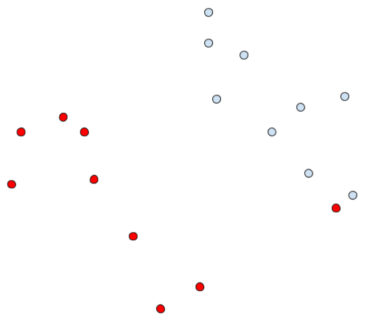
$$\arg\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||^2 \quad \text{s.t. } \forall i \; y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1$$

Informally, this is the same because the constraint is binding for the examples closest to the decision boundary. For these examples we have $y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) = 1$. The distance on these examples is $1/||\mathbf{w}||$, and is maximized by minimizing $||\mathbf{w}||^2$. Mathematically, the min of $\frac{1}{||w||}$ must be the max of $\frac{1}{2}||w||^2$ since otherwise, we could just pick the proposed better maximum of w^* and find something eve smaller than our preexisting min. We discuss exactly how to solve this problem in the third part of these notes, but first we discuss the soft margin problem.

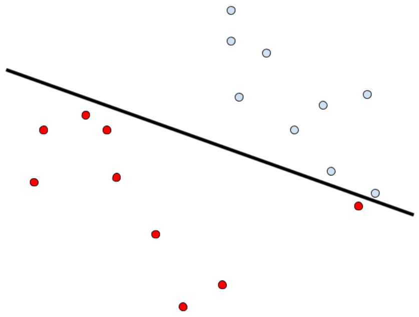
2 Soft Margin Formulation

For the hard margin formulation, we have been assuming that the data is linearly separable. However, this is not always true, and even if the data is linearly separable, it may not be best to find a separating hyperplane. In optimizing generalization error, there is a tradeoff between the size of the margin and the number of mistakes on the training data.

Concept Question: Let's illustrate the tradeoff between size of the margin and number of mistakes on training data. Assuming that you have sufficient basis transformations to draw any boundary (since a boundary linear on the transformed data may be nonlinear on the original dimensions), draw what you feel would be the most generalizable SVM classifier for the below data?

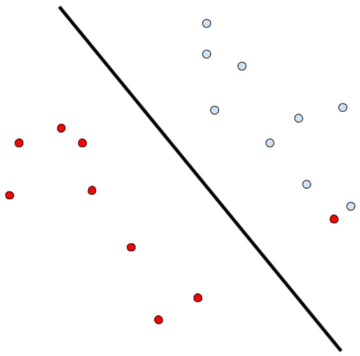


Solution: Our initial idea might be to separate the data as below



However, it might be that the outlying red point is measurement error (or just a storng outlier) that should not skew our predictive plane.

We might prefer something similar to



The relative merits of these two hyperplanes would depend on the application, and this concept question is designed to make you think about the relative tradeoffs in choosing our model's objective function.

For the soft margin formulation, we introduce a slack variable $\xi_i \geq 0$ for each i to relax the constraints on each example.

$$\xi_i \begin{cases} = 0 & \text{if correctly classified} \\ \in (0,1] & \text{correctly classified but inside margin region} \\ > 1 & \text{if incorrectly classified} \end{cases}$$

We can now rewrite the training problem for a soft margin formulation to be

$$\arg\min_{\mathbf{w},w_0} \frac{1}{2}||\mathbf{w}||^2 + C\sum_{i=1}^{n} \xi_i$$

$$\text{s.t. } \forall i \; y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1 - \xi_i$$

$$\xi_i \geq 0$$

We add a regularization parameter C, that controls how much we penalize violating the hard margin constraints. A large C penalizes these violations and thus "respects" the data closely and has small regularization. A small C does not penalize the sum of slack variables as heavily, relaxing the constraint. This is increasing the regularization.

3 Dual Form of the Support Vector Machine Training Problem

Let's return to our training problem, from the first part. Our original hard-margin training problem, which looks for weights that maximize the margin on the training data is quite difficult. Recall the training problem below

$$\mathbf{w}^\star, w_0^\star = \arg\min_{\mathbf{w}, w_0} \frac{1}{2}||\mathbf{w}||^2 \quad \text{s.t.} \quad y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) \geq 1, \quad \forall i \in \{1, \ldots, n\}\,. \tag{1}$$

We introduce *Lagrange multipliers*, $\alpha_1, \ldots, \alpha_n \geq 0$, one for each inequality in Equation 1, i.e., one per example, to obtain the Lagrangian function (Bishop appendix E does a good job of explaining Lagrange multipliers):

$$L(\mathbf{w}, w_0, \alpha) = \frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i (y_i(\mathbf{w}^\top \mathbf{x}_i + w_0) - 1) \tag{2}$$

Based on this, we now equivalently solve:

$$\mathbf{w}^*, w_0^* = \arg\min_{\mathbf{w}, w_0} \max_{\boldsymbol{\alpha} \geq 0} L(\mathbf{w}, w_0, \alpha)$$

By strong duality (out of scope!), we can equivalently write this as:

$$\max_{\boldsymbol{\alpha} \geq 0} \min_{\mathbf{w}, w_0} L(\mathbf{w}, w_0, \alpha)$$

This is useful because we can now solve analytically for the $\min_{\mathbf{w}, w_0}[\cdot]$ part of this expression. Taking derivatives, we see:

$$\frac{\partial L}{\partial w_j} = w_j - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0 \Rightarrow \mathbf{w}^* = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i$$

$$\frac{\partial L}{\partial w_0} = -\sum_{i=1}^{n} \alpha_i y_i = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y_i = 0$$

Plugging these optimal values into the Lagrangian function, we get (you should understand but not memorize this):

$$\begin{aligned} L(\mathbf{w}, w_0, \alpha) &= \frac{1}{2}||\sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i||^2 - \sum_{i=1}^{n} \alpha_i y_i (\sum_{i'=1}^{n} \alpha_{i'} y_{i'} \mathbf{x}_{i'})^\top \mathbf{x}_i - \sum_{i=1}^{n} \alpha_i y_i w_0 + \sum_{i=1}^{n} \alpha_i \\ &= \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'} - \sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'} - w_0 \sum_{i=1}^{n} \alpha_i y_i + \sum_{i=1}^{n} \alpha_i \\ &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{i'=1}^{n} \alpha_i \alpha_{i'} y_i y_{i'} \mathbf{x}_i^\top \mathbf{x}_{i'} \end{aligned}$$

We then solve for α, maximizing L, subject to $\alpha \geq 0$. The support vectors are defined to be $Q = \{i : \alpha_i > 0\}$. For the soft-margin variation, these include examples that lie in the margin region in addition to on the margin boundary.

4 Why bother with the dual form? *important*

To classify a new example $\mathbf{x}$, we compute

$$\sum_{i=1}^{n} \alpha_i^\star \, y_i \mathbf{x}_i^\top \mathbf{x} + w_0^\star$$

where $\alpha^\star$ and $w_0^\star$ solve the training problem. Based on this, we classify the example as $+1$ if this discriminant value is > 0, and -1 otherwise. The dual form is useful because the number of variables to maximize is linear in n (one for each training example), but there tends to be a small number of support vectors (data points that define the boundary) and thus the trained classifier (which only depends on the support vectors, not all the data points–can you see why?) can be interpretable and easier to calculate.

In addition, the dual has the very nice property that if we use a basis function to map $\mathbf{x}$ to a higher dimensional space, this only comes in through the "kernel function." The training problem is as explained earlier, except where $\mathbf{x}_i^\top \mathbf{x}_{i'}$ appears, we use

$$K(\mathbf{x}_i, \mathbf{x}_{i'}) = \phi(\mathbf{x}_i)^\top \phi(\mathbf{x}_{i'})$$

in its place. Similarly, we classify a new example $\mathbf{x}$ based on the value of discriminant $\sum_{i=1}^{n} \alpha_i^\star \, y_i K(\mathbf{x}_i, \mathbf{x}) + w_0^\star$. The reason that this is interesting is because we can directly compute the dot product $\phi(\mathbf{x})^\top \phi(\mathbf{x}')$ *without projecting to the higher-dimensional space!* This is known as the "kernel trick." As long as $K()$ is a valid kernel (see practice question below) the dual training problem can be solved without actually computing ϕ. Note that we can use this trick specifically because the dual gives us an equation for $\mathbf{w}^*$ in terms of $\phi(\mathbf{x}_i)^T \phi(\mathbf{x}_{i'})$

5 Practice Problems

1. **Removing Support Vectors and Retraining (Berkeley, Fall '11)**

   Suppose that we train two SVMs, the first containing all of the training data and the second trained on a data set constructed by removing some of the support vectors from the first training set. How does the size of the optimal margin change between the first and second training data? What is a downside to doing this?

   Solution:

2. **Proof that margin is invariant to scalar multiplication**

   In reformulating our max margin

   $$\frac{y_i(\mathbf{w}^\top \mathbf{x}_i + w_0)}{||\mathbf{w}||}$$

   training problem, we use the fact that the margin is invariant to multiplying $(\mathbf{w}, w_0)$ by any $\beta > 0$. Show that this property is true.

   Solution:

3. **(Berkeley, Fall '11)**

   Consider $\mathbf{x}, \mathbf{x}' \in \mathbb{R}^2$. Show that the polynomial kernel of degree 2, $K(\mathbf{x}, \mathbf{x}') = (1 + \mathbf{x}^\top \mathbf{x}')^2$ is equivalent to a dot product $\boldsymbol{\phi}(\mathbf{x})^\top \boldsymbol{\phi}(\mathbf{x}')$ where $\boldsymbol{\phi}(\mathbf{x}) = (x_1^2, x_2^2, x_1, x_2, x_1 x_2, 1)$.

   Solution:

4. **Composing Kernels**

A key benefit of SVM training is the ability to use kernel functions $K(\mathbf{x}, \mathbf{x}')$ as opposed to explicit basis functions $\phi(\mathbf{x})$. Kernels make it possible to implicitly express large or even infinite dimensional basis features. We do this by computing $\phi(\mathbf{x})^\top \phi(\mathbf{x}')$ directly, without ever computing $\phi(\mathbf{x})$.

When training SVMs, we begin by computing the kernel matrix $\mathbf{K}$, over our training data $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$. The kernel matrix, defined as $K_{i,i'} = K(\mathbf{x}_i, \mathbf{x}_{i'})$, expresses the kernel function applied between all pairs of training points.

In class, we saw Mercer's theorem (maybe), which tells us that any function K that yields a positive semi-definite kernel matrix forms a valid kernel, i.e. corresponds to a matrix of dot-products under *some* basis ϕ. Therefore instead of using an explicit basis, we can build kernel functions directly that fulfill this property.

A particularly nice coralary of this theorem is that it allows us to build more expressive kernels by composition. In this problem, you are tasked with using Mercer's theorem and the definition of a kernel matrix to prove that the following compositions are valid kernels, assuming $K^{(1)}$ and $K^{(2)}$ are valid kernels. Recall that a positive semi-definite matrix $\mathbf{K}$ requires $\mathbf{z}^\top \mathbf{K} \mathbf{z} \geq 0, \ \forall \, \mathbf{z} \in \mathbb{R}^n$.

(a) $K(\mathbf{x}, \mathbf{x}') = c\, K^{(1)}(\mathbf{x}, \mathbf{x}') \quad \text{for } c > 0$

(b) $K(\mathbf{x}, \mathbf{x}') = K^{(1)}(\mathbf{x}, \mathbf{x}') + K^{(2)}(\mathbf{x}, \mathbf{x}')$

(c) $K(\mathbf{x}, \mathbf{x}') = f(\mathbf{x})\, K^{(1)}(\mathbf{x}, \mathbf{x}')\, f(\mathbf{x}')$ where f is any function from $\mathbb{R}^m$ to $\mathbb{R}$

(d) $K(\mathbf{x}, \mathbf{x}') = K^{(1)}(\mathbf{x}, \mathbf{x}')\, K^{(2)}(\mathbf{x}, \mathbf{x}')$
[Hint: Use the property that for any $\phi(\mathbf{x})$, $K(\mathbf{x}, \mathbf{x}') = \phi(\mathbf{x})^\top \phi(\mathbf{x}')$ forms a positive semi-definite kernel matrix.]

(e) i. The $\exp$ function can be written as,

$$\exp(x) = \lim_{i \to \infty} \left(1 + x + \cdots + \frac{x^i}{i!} \right).$$

Use this to show that $\exp(xx')$ (here $x, x' \in \mathbb{R}$)) can be written as $\phi(x)^\top \phi(x')$ for some basis function $\phi(x)$. Derive this basis function, and explain why this would be hard to use as a basis in standard logistic regression.

ii. Using the previous identities, show that $K(\mathbf{x}, \mathbf{x}') = \exp(K^{(1)}(\mathbf{x}, \mathbf{x}'))$ is a valid kernel.

(f) Finally use this analysis and previous identities to prove the validity of the Gaussian kernel:

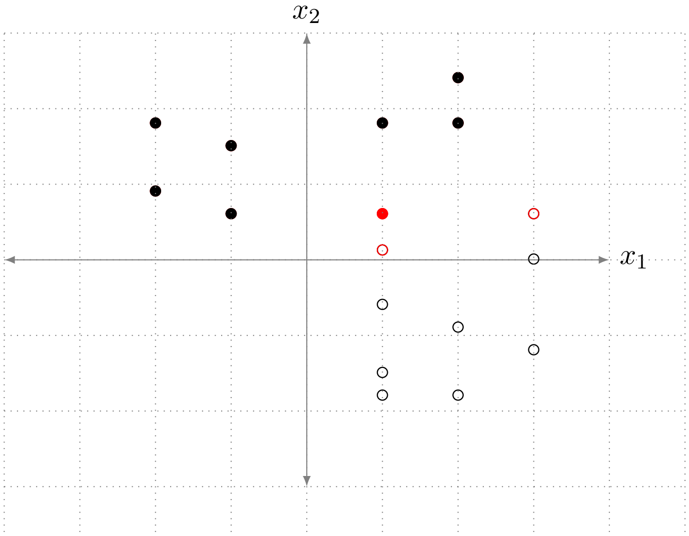
$$K(\mathbf{x}, \mathbf{x}') = \exp\left(\frac{-||\mathbf{x} - \mathbf{x}'||_2^2}{2\sigma^2} \right)$$

Solution:

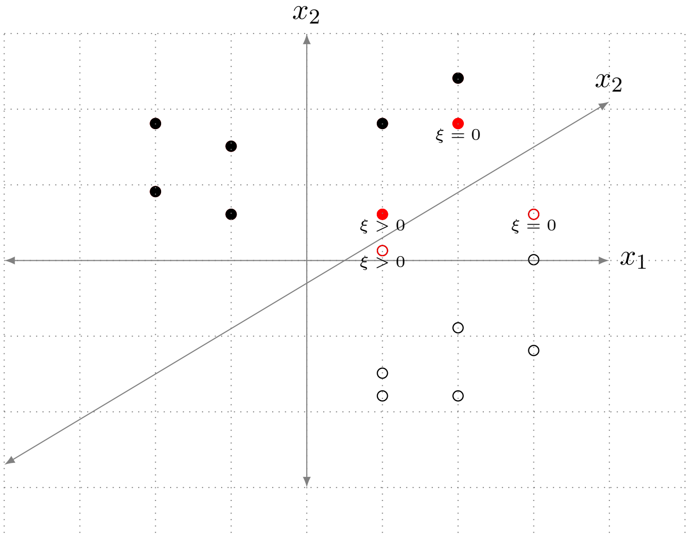
5. **Draw Margin Boundary**

In the figures below, the red examples represent the support vectors. All other examples can be assumed to have $\alpha_i = 0$. Draw the margins for the boundary given this information.
i) For the first example, you can assume the hard margin formulation. Draw the decision boundary as well as the two margin boundaries given the support vector.
ii) For the second example, you can assume the soft margin formulation and that all points are correctly classified with the optimal decision boundary. The decision boundary is already given. Draw the two margin boundaries given the support vector.

i)



ii)



6. **String Kernel**

Let s and s′ be strings. To measure how similar s and s′ are, consider the "string kernel" $K(\mathrm{s}, \mathrm{s}')$, which returns the total number of distinct substrings (of any length) that s and s′ have in common. For example, $K(\texttt{'aa'}, \texttt{'aab'}) = 3$ because the substrings '', 'a', and 'aa' are in common.
(i) Compute $K(\texttt{'aza'}, \texttt{'zaz'})$.
(ii) What is the number of possible substrings of length 1, 2, and 3 in strings that are composed from a 26-letter alphabet?
(iii) Suppose we wanted to project a string into a higher-dimensional space such that we could represent via a 0 or 1 each of all possible substrings of length ≤ 3. How many dimensions would we need?

Solution: