Lecture #18: Support Vector Classifiers

Data Science 1

CS 109A, STAT 121A, AC 209A, E-109A

Pavlos Protopapas   Kevin Rader
Rahul Dave   Margo Levine

Lecture Outline

Classifying Linear Separable Data

Classifying Linear Non-Separable Data

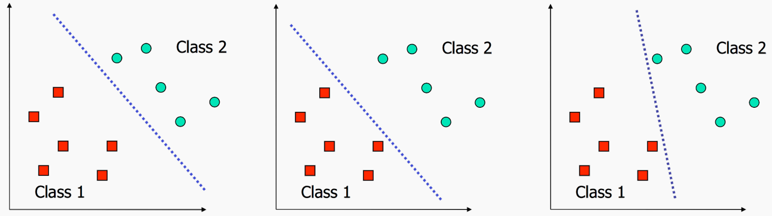
Classifying Linear Separable Data

Decision Boundaries Revisited

In logistic regression, we learn a ***decision boundary*** that separates the training classes in the feature space.

When the data can be perfectly separated by a linear boundary, we call the data ***linearly separable***.

In this case, multiple decision boundaries can fit the data. How do we choose the best?



Question: What happens to our logistic regression model when training on linearly separable datasets?

Decision Boundaries Revisited

Constraints on the decision boundary:

- In logistic regression, we typically learn an ℓ_1 or ℓ_2 regularized model.

  So, when the data is linearly separable, we choose a model with the 'smallest coefficients'.

  The purpose of regularization is to prevent overfitting.

Decision Boundaries Revisited

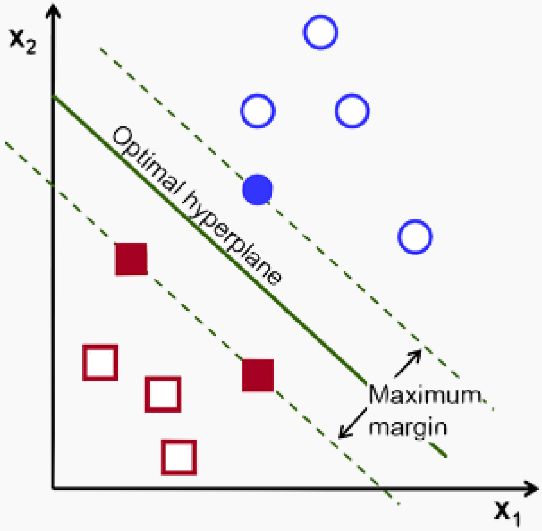
Constraints on the decision boundary:

- We can consider alternative constraints that prevent overfitting.

  For example, we may prefer a decision boundary that does not ‘favor’ any class (esp. when the classes are roughly equally populous).

  Geometrically, this means choosing a boundary that maximizes the distance or ***margin*** between the boundary and both classes.

Decision Boundaries Revisited

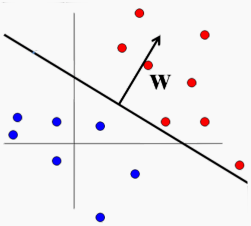


Geometry to Decision Boundaries

Recall that the decision boundary is defined by some equation in terms of the predictors. A linear boundary is defined by

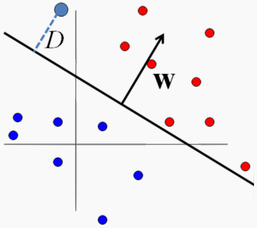
$w^\top x + b = 0$ (General equation of a hyperplane)

Recall that the non-constant coefficients, w, represent a ***normal vector***, pointing orthogonally away from the plane



Geometry to Decision Boundaries

Now, using some geometry, we can compute the distance between any point to the decision boundary using w and b.



The signed distance from a point $x \in \mathbb{R}^n$ to the decision boundary is

$$D(x) = \frac{w^\top x + b}{\|w\|} \quad \text{(Euclidean Distance Formula)}$$

Maximizing Margins

Now we can formulate our goal - find a decision boundary that maximizes the distance to both classes - as an optimization problem

$$\begin{cases} \max_{w,b} M \\ \text{such that } |D(x_n)| = \frac{y_i(w^\top x_n + b)}{\|w\|} \geq M,\ n = 1, \dots, N \end{cases}$$

where M is a real number representing the width of the ‘margin’. The inequalities $|D(x_n)| \geq M$ are called ***constraints***.

The constrained optimization problem as present here looks tricky. Let’s simplify it with a little geometric intuition.

Maximizing Margins

Notice that maximizing the distance of ***all points*** to the decision boundary, is the exactly the same as maximizing the distance to the ***closest points***.

The points closest to the decision boundary are called ***support vectors***.

For any plane, we can always scale the equation

$$w^\top x + b = 0$$

so that the support vectors lie on the planes

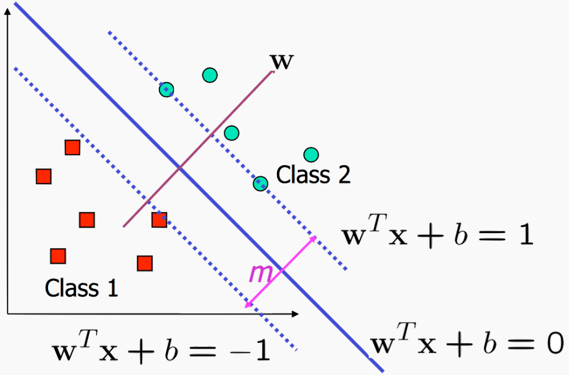
$$w^\top x + b = \pm 1,$$

depending on their classes.

Maximizing Margins

For points on planes $w^\top x + b = \pm 1$, their distance to the decision boundary is $\pm \frac{1}{\|w\|}$.

So we can define the ***margin*** of a decision boundary as the distance to its support vectors, $m = \frac{2}{\|w\|}$


$\mathbf{w}$
Class 2
$\mathbf{w}^T\mathbf{x} + b = 1$
m
Class 1
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Support Vector Classifier: Hard Margin

Finally, we can reformulate our optimization problem - find a decision boundary that maximizes the distance to both classes - as the maximization of the margin, m, ***while maintaining zero misclassifications***,

$$\begin{cases} \max\limits_{w,b} \dfrac{2}{\|w\|} \\ \text{such that } y_n(w^\top x_n + b) \geq 1,\ n = 1, \dots, N \end{cases}$$

The classifier learned by solving this problem is called ***hard margin support vector classification***.

Often SVC is presented as a minimization problem:

$$\begin{cases} \min\limits_{w,b} \|w\|^2 \\ \text{such that } y_n(w^\top x_n + b) \geq 1,\ n = 1, \dots, N \end{cases}$$

SVC and Convex Optimization

As a convex optimization problem SVC has been extensively studied and can be solved by a variety of algorithms

- **(Stochastic)** libLinear

  Fast convergence, moderate computational cost
- **(Greedy)** libSVM

  Fast convergence, moderate computational cost
- **(Stochastic)** Stochastic Gradient Descent

  Slow convergence, low computational cost per iteration
- **(Greedy)** Quasi-Newton Method

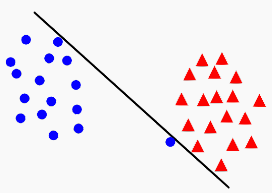
  Very fast convergence, high computational cost

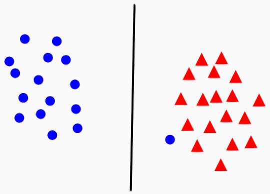
Classifying Linear Non-Separable Data

The Margin/Error Trade-Off

Maximizing the margin is a good idea as long as we assume that the underlying classes are linear separable and that the data is noise free.

If data is noisy, we might be sacrificing generalizability in order to minimize classification error with a very narrow margin





With every decision boundary, there is a trade-off between maximizing margin and minimizing the error.

Support Vector Classifier: Soft Margin

Since we want to balance maximizing the margin and minimizing the error, we want to use an objective function that takes both into account:

$$\begin{cases} \min\limits_{w,b} \|w\|^2 + \lambda \text{Error}(w,b) \\ \text{such that } y_n(w^\top x_n + b) \geq 1,\ n = 1, \ldots, N \end{cases}$$

where λ is an intensity parameter.

So just how should we compute the error for a given decision boundary?

Support Vector Classifier: Soft Margin

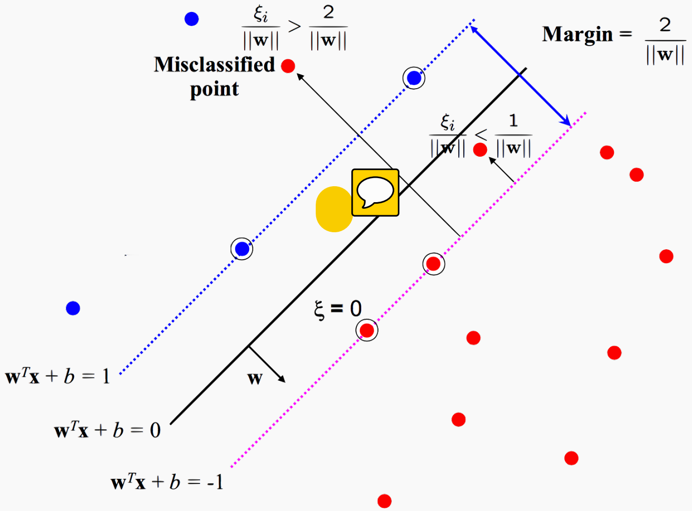
We want to express the error as a function of distance to the decision boundary.

Recall that the support vectors have distance $1/\|w\|$ to the decision boundary. We want to penalize two types of 'errors'

- **(margin violation)** points that are on the correct side of the boundary but is inside the margin. They have distance $\xi/\|w\|$, where $0 < \xi < 1$.
- **(misclassification)** points that are on the wrong side of the boundary. They have distance $\xi/\|w\|$, where $\xi > 1$.

Specifying a nonnegative quantity for ξ_n is equivalent to quantifying the error on the point x_n.

Support Vector Classifier: Soft Margin


$\frac{\xi_i}{||\mathbf{w}||} > \frac{2}{||\mathbf{w}||}$
Misclassified point
Margin = $\frac{2}{||\mathbf{w}||}$
$\frac{\xi_i}{||\mathbf{w}||} < \frac{1}{||\mathbf{w}||}$
ξ = 0
$\mathbf{w}^T\mathbf{x} + b = 1$
$\mathbf{w}$
$\mathbf{w}^T\mathbf{x} + b = 0$
$\mathbf{w}^T\mathbf{x} + b = -1$

Support Vector Classifier: Soft Margin

Formally, we incorporate error terms ξ_n's into our optimization problem by:

$$\begin{cases} \min\limits_{\xi_n \in \mathbb{R}^+, w, b} \|w\|^2 + \lambda \sum\limits_{n=1}^{N} \xi_n \\ \text{such that } y_n(w^\top x_n + b) \geq 1 - \xi_n, \; n = 1, \dots, N \end{cases}$$

The solution to this problem is called ***soft margin support vector classification*** or simply ***support vector classification***.

Tuning SVC

Choosing different values for λ in

$$\begin{cases} \min\limits_{\xi_n \in \mathbb{R}^+, w, b} \|w\|^2 + \lambda \sum\limits_{n=1}^{N} \xi_n \\ \text{such that } y_n(w^\top x_n + b) \geq 1 - \xi_n, \; n = 1, \ldots, N \end{cases}$$

will give us different classifiers.

In general,

- small λ penalizes errors less and hence the classifier will have a large margin -> wider pipe
- large λ penalizes errors more and hence the classifier will accept narrow margins to improve classification
- setting $\lambda = \infty$ produces the hard margin solution

Example

[Compare different classifiers]
[Investigate variance]