Lecture 09
Linear Classification Models and Support Vector Machines

17 February 2016

Taylor B. Arnold
Yale Statistics
STAT 365/665

Yale

Notes:

- Reminder that I will have office hours immediately after today's lecture two buildings down, at 24 Hillhouse
- Problem Set 2 due on Friday; make sure to test your code with the train.csv, test.csv and results.csv given on the website (aim for a high correlation, not the exact same results)
- Problem 3 will be posted tonight, and due next Friday

Today

- Another look at linear models for classification
- An introduction to support vector machines
- Data examples of SVMs

Linear Classification Models

On the problem sets I have had you encode a categorical variable y as ± 1, and then run linear regression where we pretend that the y values are continuous. For example, we might assume the following linear model:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \epsilon$$

And use ordinary least squares to estimate the unknown β coefficients.

Linear Classification Models, cont.

Once we have estimates $\widehat{\beta}$, we can convert these to class predictions by determining the sign of the fitted values:

$$\widehat{y}_i = \text{sign}(\widehat{\beta}_0 + \widehat{\beta}_1 x_{i,1} + \widehat{\beta}_2 x_{i,2})$$

And use ordinary least squares to estimate the unknown β coefficients.

Linear Classification Models, cont.

Now, one interesting thing about this predictor, is that we can understand the set:

$$\left\{(x_1, x_2) \quad \text{s.t.} \quad \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 = 0\right\}$$

As being a line in two dimensional space. Furthermore, the set:

$$\left\{(x_1, x_2) \quad \text{s.t.} \quad \widehat{\beta}_0 + \widehat{\beta}_1 x_1 + \widehat{\beta}_2 x_2 > 0\right\}$$

Is one of the two half space created by the previous line. The set where the linear predictor is negative is simply the other half space.

Aside: Why does this line have three parameters rather than the usual two?

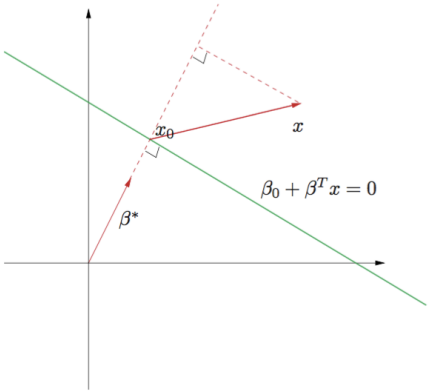
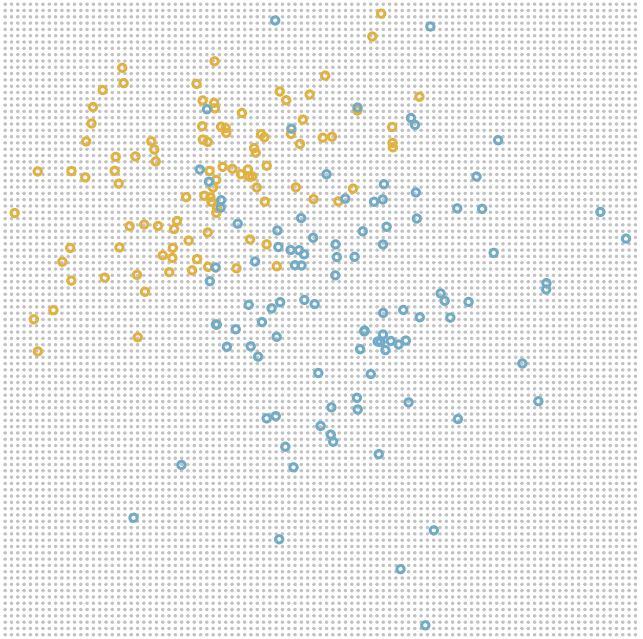
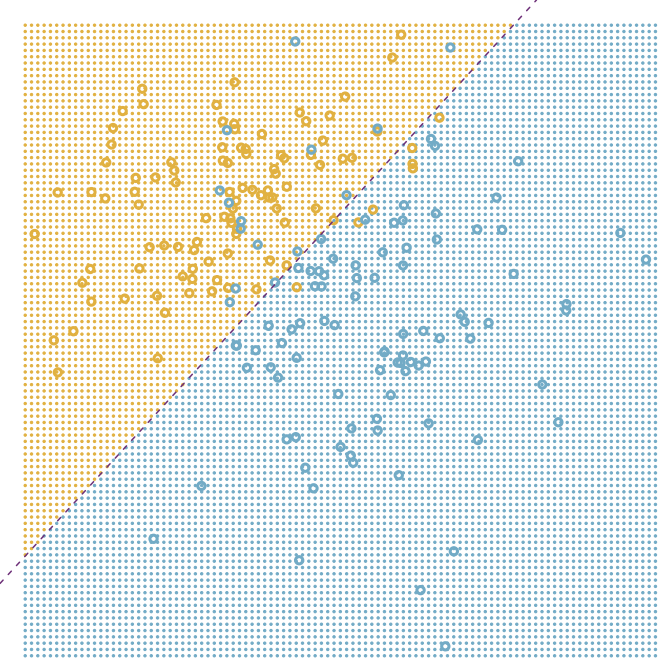


FIGURE 4.15. *The linear algebra of a hyperplane (affine set).*

Linear Classification Models, cont.

Let's look at this using a small two dimensional example.





Logistic regression

There is more than one way to estimate the best separating line that gives a linear partition of the parameter space into two classes. One commonly used example is logistic regression.

For this discussion, I'll re-parameterize the response variable y to be 0 and 1.

Logistic regression, cont.

Consider a model where we assume that for each y_i there exists an unknown p_i such that:

$$\mathbb{P}[y_i = 0] = 1 - p_i \tag{1}$$
$$\mathbb{P}[y_i = 1] = p_i \tag{2}$$

In statistical terminology, we would say that y_i is a random variable with a Bernoulli distribution with parameter p_i.

Logistic regression, cont.

Now, we want to somehow specify a relationship between a set of predictor variables x_i and the value p_i. A common selection is to use the logit function:

$$p_i = \frac{1}{1 + e^{-(x_i^t \beta + \beta_0)}}$$

This has some deeper motivations if we look at the theory of exponential families or describe the quantity in terms of the log-odds ratio.

For us, just notice that if $x_i\beta + \beta_0$ is zero, we get a p_i of 0.5. When the linear quantity goes to positive infinity, the probabilities go to 1, and likewise when limiting to negative infinity, the probabilities go to zero.

Logistic regression, cont.

How do we use this formulation to actually predict the $\widehat{\beta}$ and $\widehat{\beta}_0$? The standard approach is to use maximum likelihood estimation. In short, we maximize the probability of observing the data y_i conditioned on the estimated parameters and the x_i's.

Logistic regression, cont.

How do we use this formulation to actually predict the $\widehat{\beta}$ and $\widehat{\beta}_0$? The standard approach is to use maximum likelihood estimation. In short, we maximize the probability of observing the data y_i conditioned on the estimated parameters and the x_i's.

Computational this can be done by a modified form of iteratively re-weighted least squares. Conceptually, we iteratively fit models that look like this:

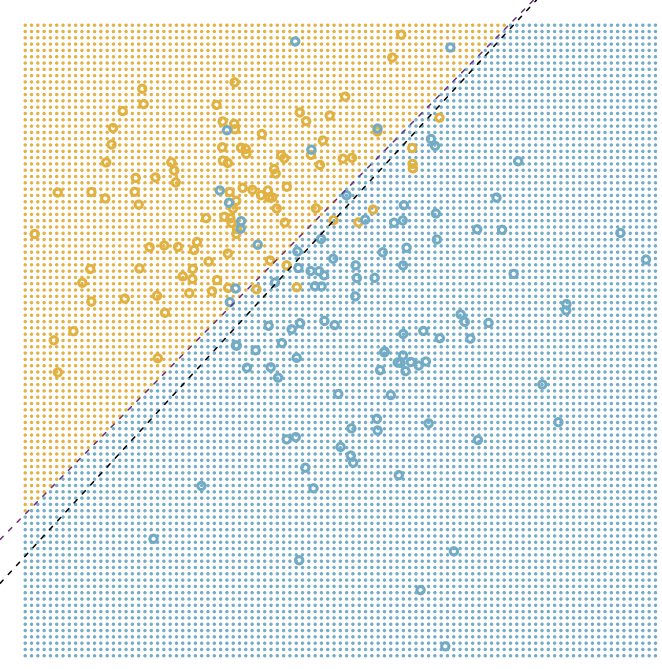
$$\widehat{\gamma} = (X^t W X)^{-1} X^t W y$$

For some diagonal matrix of weights W. This is a second order method and converges quite rapidly.

Logistic regression, cont.

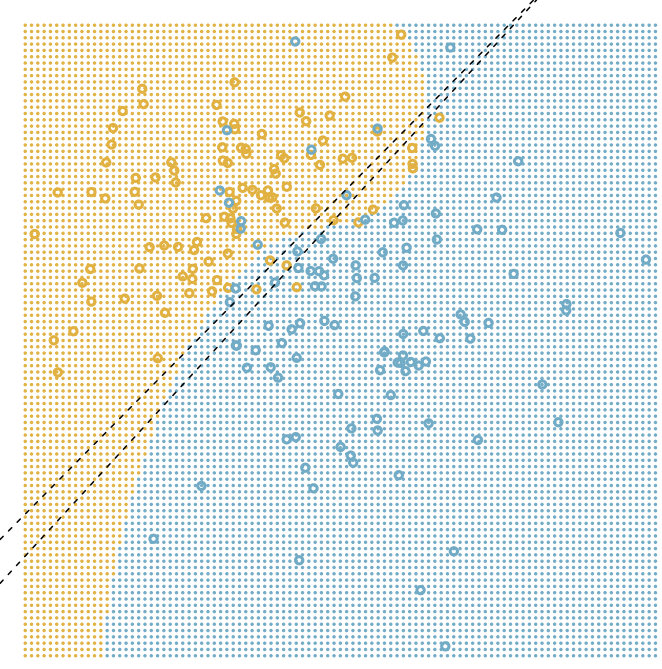
Notice that once again, we have a hyperplane defined by the set $x_i^t\beta + \beta_0$ that gives a linear separation between the two classes we wish to categorize.

How does this compare to the plane produced by linear regression?



Logistic regression, cont.

As with linear regression, we can do basis expansion to get non-linear classification boundaries. So, for example, here we could treat x_1^2 and x_1^3 as the new third and fourth dimensions of the predictor matrix. We learn a linear separating plane in this higher dimensional space. However, when we project these predictions back down into the original space, the effect is to give a non-linear boundary between the classes.



Support vector machines

Support vector machines are powerful machine learning algorithms that also construct separating planes for classification. They are conceptually fairly simple, but the underlying mathematics for learning them from data can become a bit tricky.

Support vector machines, cont.

The linear, hard-margin classification case can be described quite succinctly: Pick two parallel hyperplanes that separate the two classes such that the distance between the hyperplanes is maximal; the maximum-margin hyperplane is the midpoint of these two separating planes.

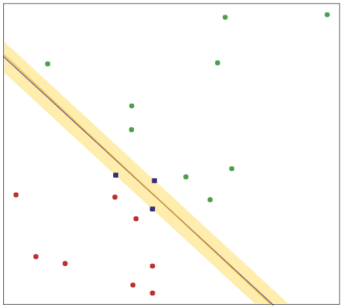


FIGURE 4.16. *The same data as in Figure 4.14. The shaded region delineates the maximum margin separating the two classes. There are three support points indicated, which lie on the boundary of the margin, and the optimal separating hyperplane (blue line) bisects the slab. Included in the figure is the boundary found using logistic regression (red line), which is very close to the optimal separating hyperplane (see Section 12.3.3).*

As we just saw, a hyperplane can be represented as the set:

$$\left\{x \in \mathbb{R} \quad \text{s.t.} \quad \beta_0 + \beta^t x = 0\right\}$$

For a given β_0 and $\beta^t \in \mathbb{R}^p$. And the 'side' of the hyperplane that a point x is on can be determined by:

$$\text{sign}\left(\beta_0 + \beta^t x\right)$$

Notice that we want $\beta_0 + \beta^t x_i$ to be positive if y_i is positive and negative if y_i is negative. We can then compactly write the necessary and sufficient condition for a hyperplane correctly separating the input points:

$$y_i(x_i^t\beta + \beta_0) > 0, \quad i = 1, \ldots, n.$$

Notice that we want $\beta_0 + \beta^t x_i$ to be positive if y_i is positive and negative if y_i is negative. We can then compactly write the necessary and sufficient condition for a hyperplane correctly separating the input points:

$$y_i(x_i^t\beta + \beta_0) > 0, \quad i = 1, \ldots, n.$$

Assuming we have such a separating hyperplane, the minimal value of the left hand side gives a measurement of the distance of the closest point to the separating plane. In order to make this distance consistent, we only consider for the moment $||\beta||_2 = 1$.

Now, we have said that a support vector machine minimizes the margin of a separating hyperplane. This can be written as:

$$\begin{aligned} \max_{||\beta||_2=1} \quad & M \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > M, \quad i = 1, \ldots, n. \end{aligned}$$

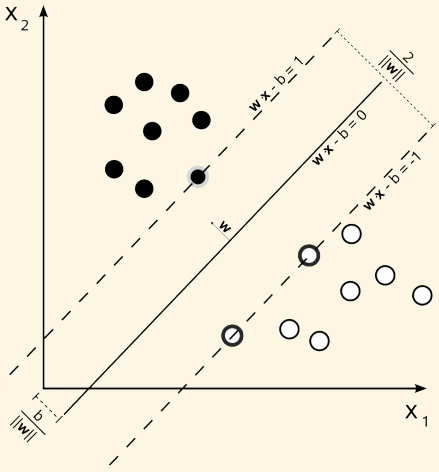
The quantity M is called the margin.

It will be easier going forward to fix the value of M to be 1 and then minimize the size of β:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1, \quad i = 1, \ldots, n. \end{aligned}$$

Where the factor of $1/2$ and squared norm are added for later notational convenience.

This defines a margin around the linear decision plane of width $\frac{1}{||\beta||}$


x2
x1
w x - b = 1
w x - b = 0
w x - b = -1
w
2/||w||
|-b-1|/||w||
|-b|/||w||

Soft margin and Cost Function

What happens, as in our original example, when there is not such separating hyperplane? We introduce slack variables that produce a so-called soft margin: the optimization algorithm has a certain amount of leeway in allowing some points to be on the wrong side of the classification.

Incorporating into our current specification, we add a ξ_i for each observation and rewrite our optimization problem as:

$$\begin{aligned} \min \quad & \frac{1}{2}||\beta||_2^2 \\ \text{s.t.} \quad & y_i(x_i^t\beta + \beta_0) > 1 - \xi_i, \quad i = 1, \ldots, n \\ \xi_i > 0, & \sum_i \xi_i \leq \text{Constant}. \end{aligned}$$

So ξ_i will be non-zero for mis-classified points, and the amount of misclassification allowed is controlled by the constant in the model.

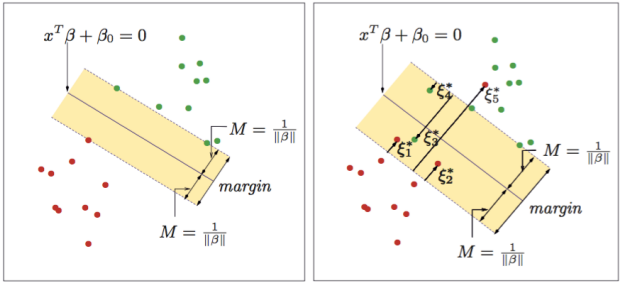
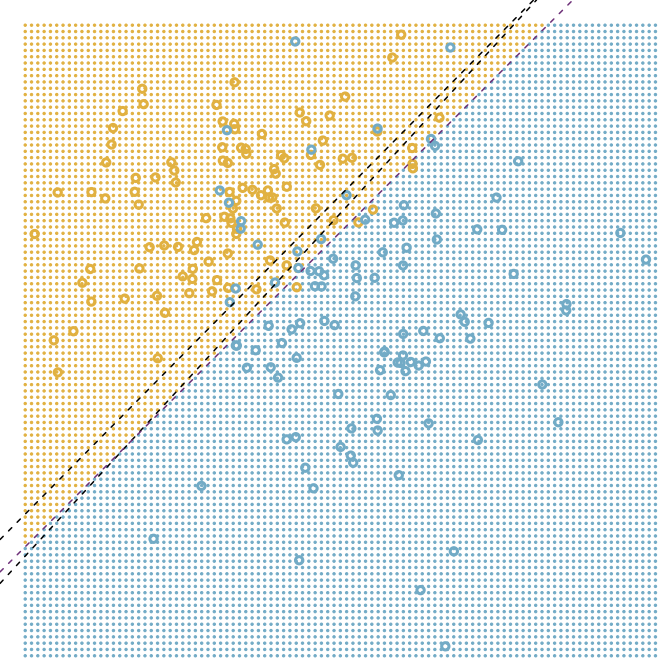


FIGURE 12.1. *Support vector classifiers. The left panel shows the separable case. The decision boundary is the solid line, while broken lines bound the shaded maximal margin of width $2M = 2/\|\beta\|$. The right panel shows the nonseparable (overlap) case. The points labeled ξ_j^* are on the wrong side of their margin by an amount $\xi_j^* = M\xi_j$; points on the correct side have $\xi_j^* = 0$. The margin is maximized subject to a total budget $\sum \xi_i \leq$ constant. Hence $\sum \xi_j^*$ is the total distance of points on the wrong side of their margin.*

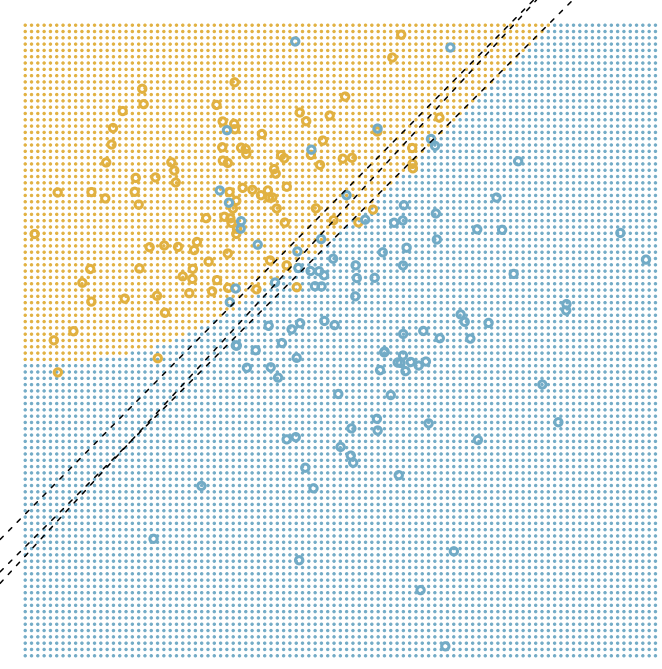
Soft margin and Cost Function, cont.

So now, how does this boundary compare to the example we were working with for linear and logistic regression?



Non-linear SVM

Much like linear and logistic regression, there is a way to do SVM in a higher dimensional space via basis expansion in order to capture non-linear effects.



Next for SVMs

So we have only scratched the surface of some of the interesting complexity of support vector machines. Next time I'll delve into the actual computational aspects of optimizing the SVM equations. This is actually quite important to understand as it gives us more intuition for why they are so predictive in higher dimensional spaces.