STA 4241 Lecture, Week 4

September 14th, 2021

Overview of what we will cover

- Logistic regression continued
  - Stock market example
  - Outcomes with more than 2 categories
- Discriminant analysis
  - Linear discriminant analysis
  - Extending linear discriminant analysis to multiple covariates
  - Quadratic discriminant analysis
  - Maximum support classifier

Stock market data

- Let's apply these ideas to the stock market data from the ISLR package in R
- Our goal is to predict whether the stock market will go up or down given the previous 5 days returns and the day's volume
- The stock market is notoriously hard to predict and model
- How well will our classification approaches work?

Stock market data

- First, let's use all data through 2003 to train our model
- We will evaluate how well it predicts the market for 2004 and 2005
  - Test data
- If it works well, we can become rich!

Stock market data

- We obtain the following summary from the fit of a logistic regression model

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept)  0.31554    0.36991   0.853    0.394
Lag1        -0.04891    0.05416  -0.903    0.366
Lag2        -0.04284    0.05420  -0.790    0.429
Lag3         0.01040    0.05402   0.192    0.847
Lag4         0.02538    0.05407   0.469    0.639
Lag5        -0.01827    0.05347  -0.342    0.733
Volume      -0.25853    0.26921  -0.960    0.337
```

Stock market data

- The individual p-values are all fairly large
  - None of the predictors seem particularly predictive of whether the market will go up or down
- This doesn't mean that the covariates as a whole don't help to predict the outcome
- It's not clear how well the predictions will do from this output
  - Need to look at the test data set

Stock market data

- Our test data set error rate is 53%!
  - Really bad
- We literally could have flipped a coin and gotten 50%
- Roughly 56% of the days in 2004 and 2005 went up, therefore if we just naively guessed that every day would go up, our test set error rate would be 44%
- Not a good look for logistic regression

Stock market data

- What happened? How does our model do worse than random guessing?
  - Overfitting not a big concern with 6 covariates and hundreds of data points
- Only 49% of days went up in the period before 2004, while 56% went up in 2004 and 2005
- The market was fundamentally different in these two time periods
  - Our model fit on the pre-2004 data doesn't hold well in future time periods due to this shift
  - Problem with model extrapolation

Stock market data

- What if instead of using all pre-2004 data as our training data we use a random subset of the data for training
- I randomly assigned 750 data points to be training and 500 to be for testing
- This should avoid the extrapolation problem, and we can evaluate how our models do
- Hopefully now we can at least beat random guessing
  - But maybe not if the covariates aren't predictive of the outcome

Stock market data

- We obtain the following summary from the fit of a logistic regression model

```
Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         -0.29074    0.38389  -0.757   0.4488
Lag1                 0.02066    0.06348   0.325   0.7449
Lag2                -0.08657    0.06413  -1.350   0.1770
Lag3                -0.04641    0.06734  -0.689   0.4907
Lag4                 0.03046    0.06348   0.480   0.6313
Lag5                 0.05341    0.06294   0.849   0.3961
Volume               0.04698    0.27927   0.168   0.8664
as.factor(Year)2002 -0.01101    0.24135  -0.046   0.9636
as.factor(Year)2003  0.39328    0.23628   1.665   0.0960 .
as.factor(Year)2004  0.52807    0.23903   2.209   0.0272 *
as.factor(Year)2005  0.35741    0.29893   1.196   0.2318
```

Stock market data

- It seems like maybe the year variable is important
  - Makes sense given the difference we observed in the two time periods
- Our prediction error is now 47%
- This is better than random guessing
- 53% of days went up in the test data so our approach doesn't do better than simply guessing that the market goes up every day

Baseline-category logit models

- There are extensions of the logistic model that allow for Y to have more than 2 categories
- Suppose our outcome has c categories
- First need to choose one as the baseline
    - Typically category c
    - Choice doesn't affect model fit

Baseline-category logit models

- Define $p_k = p_k(\boldsymbol{x}) = P(Y = k|\boldsymbol{X} = \boldsymbol{x})$
  - Conditional probability of being in category k
- The baseline-category logit model is defined as

$$\log\left(\frac{p_k}{p_c}\right) = \beta_{0k} + \sum_{j=1}^{p} \beta_{jk} X_{ij}, \quad k = 1, \dots c - 1$$

Baseline-category logit models

- Note this involves $c-1$ models
  - No model for the c^{th} level
- Different parameters for each logit
- Model compares each group to the baseline

Baseline-category logit models

- For groups $k = 1, \dots c-1$ the estimated probabilities are

$$\widehat{p}_k = \frac{e^{\widehat{\beta}_{0k} + \sum_{j=1}^{p} \widehat{\beta}_{jk} X_{ij}}}{1 + \sum_{h=1}^{c-1} e^{\widehat{\beta}_{0h} + \sum_{j=1}^{p} \widehat{\beta}_{jh} X_{ij}}}$$

- We know that the probabilities must sum to 1 over all the groups, therefore

$$\widehat{p}_c = \frac{1}{1 + \sum_{h=1}^{c-1} e^{\widehat{\beta}_{0h} + \sum_{j=1}^{p} \widehat{\beta}_{jh} X_{ij}}}$$

Baseline-category logit models

- Once we have the estimated probabilities, we can do classification
- Simply choose the class that has the highest estimated probability
- These models are not commonly used for classification purposes
- We will now discuss an approach that naturally handles multiple categories and is more widely used when the number of categories is more than 2

Linear discriminant analysis

- Linear discriminant analysis (also known as LDA) is an alternative approach to classification problems where we observe $(\boldsymbol{X}_i, Y_i)$
- Once nice feature of LDA is that the approach is the same regardless of whether there are 2 or more classes for the outcome Y
- In some cases, provides better estimates than logistic regression
  - Small sample sizes
  - Separation in the outcome classes
- Makes some additional assumptions that are not required of logistic regression

Linear discriminant analysis

- Suppose Y has K classes or categories
- LDA relies heavily on Bayes rule

$$\begin{aligned} P(Y = k|X = x) &= \frac{P(X = x|Y = k)P(Y = k)}{P(X = x)} \\ &= \frac{P(X = x|Y = k)P(Y = k)}{\sum_{l=1}^{K} P(X = x|Y = l)P(Y = l)} \end{aligned}$$

- Reversed the problem from needing to estimate $Y|X$ to needing to estimate $X|Y$

Linear discriminant analysis

- To use the same notation as the book, we will define

$$f_k(x) = P(X = x|Y = k)$$

which is the density function of X within class k

- Define $\pi_k = P(Y = k)$, the marginal probability of being in class k
- Bayes rule can now be written as

$$P(Y = k|X = x) = \frac{\pi_k f_k(x)}{\sum_{l=1}^{K} \pi_l f_l(x)}$$

Linear discriminant analysis

- We know from lecture 1 and the Bayes classifier that the best classifier is one that assigns a subject to the class where $P(Y = k|X = x)$ is the highest
- Now, we can simply estimate this probability
- Figure out which category maximizes this probability for each subject
  - Classify them to this group

Linear discriminant analysis

- Now the problem left for us is to estimate both π_k and $f_k(x)$ for all k
- You might be thinking that we've only complicated the problem
  - We now are estimating two probabilities instead of one
- Fortunately, estimating π_k is extremely easy
  - Sample proportion of subjects in class k, i.e. n_k/n
    - $n_k = \sum_{i=1}^{n} 1(Y_i = k)$
- Now we need to estimate the density of X

Linear discriminant analysis

- We will start by assuming X is one-dimensional
  - We will cover the extension to higher dimensions later in the slides
- LDA assumes that $X|Y = k \sim \mathcal{N}(\mu_k, \sigma^2)$
- In class k, X is normally distributed with mean μ_k
- Common variance σ^2 across groups
  - Will drop this assumption later

Linear discriminant analysis

- The normal density function (with parameters μ_k and σ^2) takes the form
$$f_k(x) = \frac{1}{\sqrt{2\pi\sigma^2}}\exp\left(-\frac{1}{2\sigma^2}(x-\mu_k)^2\right)$$
- Once we have an estimate of μ_k and σ^2, we have an estimate of $f_k(x)$
- Once we have an estimate of $f_k(x)$, we can find the class that maximizes $P(Y = k|X = x)$

Linear discriminant analysis

- First we estimate the mean as

$$\widehat{\mu}_k = \frac{1}{n_k} \sum_{i:Y_i=k} X_i$$

which is just the sample mean within class k

- The variance is shared across all groups and is estimated as

$$\widehat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:Y_i=k} (X_i - \widehat{\mu}_k)^2$$

Linear discriminant analysis

- Now we have estimates of both π_k and $f_k(x)$
- We can estimate

$$
\begin{aligned}
\widehat{P}(Y = k | X = x) &= \frac{\widehat{\pi}_k \widehat{f_k}(x)}{\sum_{l=1}^{K} \widehat{\pi}_l \widehat{f_l}(x)} \\
&= \frac{\frac{\widehat{\pi}_k}{\sqrt{2\pi\widehat{\sigma}^2}} \exp\left(-\frac{1}{2\widehat{\sigma}^2}(x - \widehat{\mu}_k)^2 \right)}{\sum_{l=1}^{K} \frac{\widehat{\pi}_l}{\sqrt{2\pi\widehat{\sigma}^2}} \exp\left(-\frac{1}{2\widehat{\sigma}^2}(x - \widehat{\mu}_l)^2 \right)}
\end{aligned}
$$

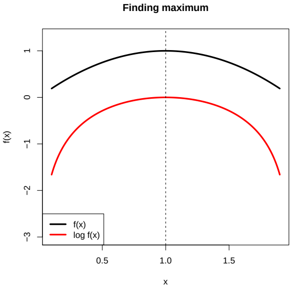
Linear discriminant analysis

- The last step is to maximize this quantity with respect to k
  - Find the class k with the highest probability
- Note that the denominator of this expression is the same for all classes
- Therefore we need only maximize the numerator of this expression

$$\frac{\widehat{\pi}_k}{\sqrt{2\pi\widehat{\sigma}^2}}\exp\left(-\frac{1}{2\widehat{\sigma}^2}(x-\widehat{\mu}_k)^2\right)$$

Linear discriminant analysis

- To simplify, we will maximize the log of this expression
- Because the logarithm is a monotone increasing function, maximizing the log of this expression with respect to k will give us the same maximizer
- Note in this class, unless otherwise stated, we are using the natural (base e) log



Linear discriminant analysis

- Our goal is to maximize

$$\log(\widehat{\pi}_k) - \log(\sqrt{2\pi\widehat{\sigma}^2}) - \frac{1}{2\widehat{\sigma}^2}(x - \widehat{\mu}_k)^2$$
$$= \log(\widehat{\pi}_k) - \log(\sqrt{2\pi\widehat{\sigma}^2}) - \frac{x^2}{2\widehat{\sigma}^2} + \frac{x\widehat{\mu}_k}{\widehat{\sigma}^2} - \frac{\widehat{\mu}_k^2}{2\widehat{\sigma}^2}$$

- But we can drop terms not involving $\widehat{\mu}_k$ or $\widehat{\pi}_k$ as they won't change the value of k that maximizes this expression, so we only need to maximize

$$\widehat{\delta}_k(x) = \log(\widehat{\pi}_k) + \frac{x\widehat{\mu}_k}{\widehat{\sigma}^2} - \frac{\widehat{\mu}_k^2}{2\widehat{\sigma}^2}$$

Linear discriminant analysis

- $\delta_k(x)$ is the discriminant function
- The procedure is called *linear* discriminant analysis because the discriminant function is linear in x
- In total the procedure has just a few steps
  1. Estimate $\widehat{\pi}_k$ and $\widehat{\mu}_k$
  2. Plug them in to obtain $\widehat{f_k}(x)$
  3. Calculate $\widehat{\delta_k}(x)$
  4. Classify a subject into the class k that maximizes $\widehat{\delta_k}(x)$

Extending to $p > 1$

- Extending to more than one covariate is relatively straightforward once we know the overall procedure
- Estimating $\widehat{\pi}_k$ remains unchanged
- The only difference is estimating $\widehat{f_k}(x)$
    - Need a multivariate density function
- Thankfully the normal distribution has an extension into higher dimensions called the multivariate normal density

Extending to $p > 1$

- The multivariate normal density with mean $\boldsymbol{\mu}_k$ and covariance $\boldsymbol{\Sigma}$ is given by

$$\frac{1}{(2\pi)^{p/2}|\boldsymbol{\Sigma}|^{1/2}}\exp\left(-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu}_k)^T\boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu}_k)\right)$$

- And we can estimate the mean and covariance as

$$\widehat{\boldsymbol{\mu}}_k = \frac{1}{n_k}\sum_{i:Y_i=k}\boldsymbol{X}_i$$

$$\widehat{\boldsymbol{\Sigma}} = \frac{1}{n-K}\sum_{k=1}^{K}\sum_{i:Y_i=k}(\boldsymbol{X}_i-\widehat{\boldsymbol{\mu}}_k)(\boldsymbol{X}_i-\widehat{\boldsymbol{\mu}}_k)^T$$

Extending to $p > 1$

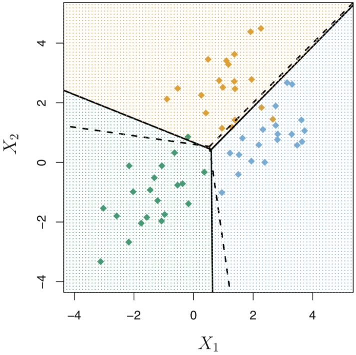
- If we perform the same algebra and maximization steps as before, replacing the univariate density with the multivariate one, we obtain a discriminant function of

$$\widehat{\delta}(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_k - \frac{1}{2} \widehat{\boldsymbol{\mu}}_k^T \boldsymbol{\Sigma}^{-1} \widehat{\boldsymbol{\mu}}_k + \log(\widehat{\pi}_k)$$

- This is again linear in $\boldsymbol{x}$
- All other steps for classification are the same as before

Extending to $p > 1$

- We can visualize the decision boundary given by LDA with two covariates
  - Solid line represents LDA decision boundary
  - Dotted line is the Bayes classifier



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Stock market data

- Let's apply this to the stock market data and see how it compares with our logistic regression estimates
- We will use the exact same training and testing data as before
  - 750 randomly chosen data points are training
  - 500 are to be used for testing
- We again want to include lags 1-5, volume, and year as predictors

Stock market data

- There is one major problem with this
- The year variable is almost certainly not normally distributed
- We included the year variable with dummy variables in the logistic regression model
- How do we use it here?
    - We could simply drop it
        - Not very satisfactory
    - Incorrectly assume it is normally distributed

Stock market data

- Dropping year as a predictor is unappealing because we saw it was the only variable associated with the outcome in our logistic regression model
- Let's fit the LDA model without year and see what happens
- We get a test set error rate of 51%
  - Higher than logistic regression, which was 47%
  - Not really better than random guessing either

Stock market data

- We obtain the following summary from the fit of LDA

```
Prior probabilities of groups:
     Down        Up
0.4906667 0.5093333

Group means:
           Lag1        Lag2         Lag3         Lag4        Lag5   Volume
Down 0.01789402  0.03535598  0.042214674 -0.009557065 -0.02494022 1.462135
Up   0.05055236 -0.06130105 -0.005104712  0.039586387  0.06011780 1.488195

Coefficients of linear discriminants:
              LD1
Lag1    0.1448677
Lag2   -0.4907217
Lag3   -0.2610266
Lag4    0.2910141
Lag5    0.4352779
Volume  1.2857296
```

- The coefficients give the linear combination of the covariates that is used in the decision rule
  - Higher values of this linear combination make it more likely the model predicts the market to go up

Stock market data

- Alternatively, we can simply include dummy variables for year into the LDA algorithm
- This breaks the multivariate normality assumption on $X|Y = k$
- Still might perform well with respect to classification
- There has been some research done to support the fact that LDA is at least mildly robust to misspecification of this assumption
- Note that the main idea behind LDA applies to non-normal distributions as well
    - Most software implementations assume normality

Stock market data

- We fit LDA with dummy variables for year

```
Prior probabilities of groups:
     Down        Up
0.4906667 0.5093333

Group means:
           Lag1        Lag2         Lag3         Lag4        Lag5   Volume
Down 0.01789402  0.03535598  0.042214674 -0.009557065 -0.02494022 1.462135
Up   0.05055236 -0.06130105 -0.005104712  0.039586387  0.06011780 1.488195
     as.factor(Year)2002 as.factor(Year)2003 as.factor(Year)2004
Down           0.2201087           0.1956522           0.1739130
Up             0.1701571           0.2225131           0.2277487
     as.factor(Year)2005
Down           0.1820652
Up             0.2068063

Coefficients of linear discriminants:
                            LD1
Lag1                 0.07952303
Lag2                -0.33229770
Lag3                -0.17793279
Lag4                 0.11649313
Lag5                 0.20515731
Volume               0.18362613
as.factor(Year)2002 -0.03975370
as.factor(Year)2003  1.52807535
as.factor(Year)2004  2.04923481
as.factor(Year)2005  1.39009922
```

Stock market data

- Again seems like year plays an important role
  - Large coefficients in linear discriminants
- The test set prediction error is 47%
  - Same as logistic regression
- In fact, for this data set, the predictions for logistic regression and LDA are identical
  - Same for each subject

Linear discriminant analysis

- Advantages of LDA over logistic regression
  1. Works for multiple classes
  2. Works better when the classes are well separated
  3. Works better in small sample sizes

- Cons of LDA
  1. Assumption of normality
  2. Assuming a shared variance for each class

Quadratic discriminant analysis

- We will now work on alleviating the assumption of a shared variance across classes
  - Called quadratic discriminant analysis
- LDA assumed that $X|Y = k \sim \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma})$
- QDA assumes that $X|Y = k \sim \text{MVN}(\boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k)$
- We will see that this leads to greater flexibility in the classifier

Quadratic discriminant analysis

- QDA proceeds in the same manner as LDA, simply replacing the previous multivariate normal density with the new one that has unique variances for each group
- Estimating $\widehat{\pi}_k$ is the same as before
- Estimating $\widehat{\mu}_k$ is the same as before
- We now use the following estimate of the group specific variance

$$\widehat{\boldsymbol{\Sigma}}_k = \frac{1}{n_k - K} \sum_{i:Y_i=k} (\boldsymbol{X}_i - \widehat{\boldsymbol{\mu}}_k)(\boldsymbol{X}_i - \widehat{\boldsymbol{\mu}}_k)^T$$

Quadratic discriminant analysis

- This may seem like a small change, but it greatly increases the number of unknown parameters
- Suppose we have K classes and p covariates
- LDA has $Kp + \frac{p(p+1)}{2}$ unknown parameters
  - Kp corresponds to estimating the means, $\boldsymbol{\mu}_k$ for $k = 1, \ldots K$.
  - $\frac{p(p+1)}{2}$ is the number of unique elements in $\boldsymbol{\Sigma}$

Quadratic discriminant analysis

- QDA has $Kp + \frac{Kp(p+1)}{2}$ unknown parameters
  - Massive increase over LDA when p is large
- This means that QDA requires a bigger sample size to implement
  - Need more data in each class to estimate class-specific covariances
- There is an inherent bias-variance trade-off between LDA and QDA
  - LDA has fewer parameters and is more efficient, but is more susceptible to bias
  - QDA has more parameters and is more flexible, but this comes with increased variance

Quadratic discriminant analysis

- It still may not be exactly clear in what manner QDA can improve on LDA
- To gain intuition for this, we can derive the discriminant function for QDA and compare with the discriminant function for LDA
- Using Bayes rule, we can see that

$$P(Y = k|X = x) = \frac{\frac{\pi_k}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_k|^{1/2}} \exp\Bigg(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)\Bigg)}{\sum_{l=1}^{K} \frac{\pi_l}{(2\pi)^{p/2}|\boldsymbol{\Sigma}_l|^{1/2}} \exp\Bigg(-\frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_l)^T \boldsymbol{\Sigma}_l^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_l)\Bigg)}$$

Quadratic discriminant analysis

- Again we want to find the class k that maximizes this probability
- We can ignore the denominator since it is shared by all classes, and focus on maximizing the numerator
- Again we will take the log of the numerator

$$\log \pi_k - \frac{p}{2}\log(2\pi) - \frac{1}{2}\log|\boldsymbol{\Sigma}_k| - \frac{1}{2}(\boldsymbol{x} - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma}_k^{-1}(\boldsymbol{x} - \boldsymbol{\mu}_k)$$

Quadratic discriminant analysis

- We can ignore any terms that do not involve $\boldsymbol{\mu}_k, \pi_k$ or $\boldsymbol{\Sigma}_k$
- This leaves us with the following discriminant function

$$\delta_k(x) = \log \pi_k - \frac{1}{2} \log |\boldsymbol{\Sigma}_k| - \frac{1}{2} \boldsymbol{x}^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{x} + \boldsymbol{x}^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k - \frac{1}{2} \boldsymbol{\mu}_k^T \boldsymbol{\Sigma}_k^{-1} \boldsymbol{\mu}_k$$

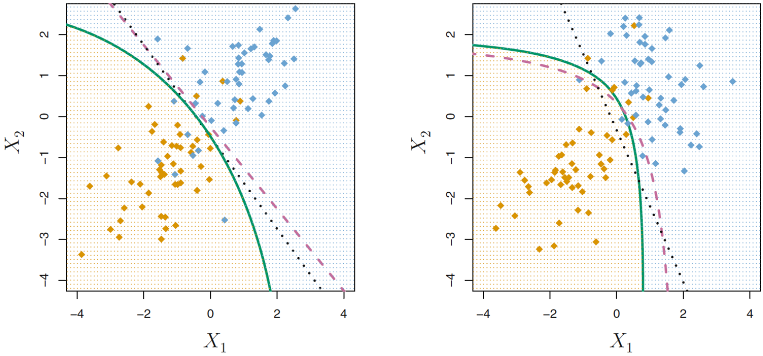
- Note this is a quadratic function of $\boldsymbol{x}$
  - Hence the name, quadratic discriminant analysis
- We will classify a subject based on the value of k that maximizes this discriminant function

Quadratic discriminant analysis

- One way to think of QDA is that it is a parametric approach that lies somewhere between the linear parametric approaches of logistic regression or LDA and the fully nonparametric approaches, such as KNN
- If the truth is nonlinear, QDA will perform much better than LDA
- If the truth is linear, QDA will have worse predictions than LDA
- We can see this visually on two separate examples

Quadratic discriminant analysis

- The dashed line is the truth, the dotted line is the LDA estimate, and the green line is from QDA



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Stock market data revisited

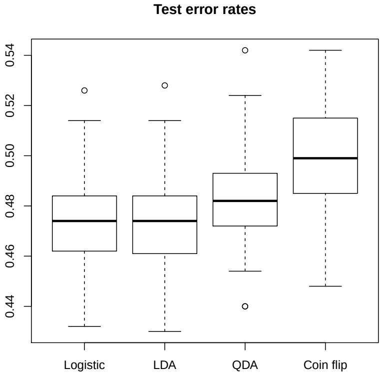
- If we use the same testing set as before we obtain an error rate of 47.6%
- This is close to, but slightly worse, than LDA or logistic regression
- In the textbook, they use the 2005 data as the testing set
  - Find that QDA does the best
- We should be careful not to read too much into the results from one small to moderate sized testing data set

Stock market data revisited

- There is sampling variability when the test data set is small
  - A different test data set might lead to slightly different results
- For instance, we know that random guessing (flipping a coin) for classification should lead to a testing error rate of 50%
- In any one data set, however, it can vary around this number
  - Magnitude of this variability is a function of the test set size

Stock market data revisited

- I randomly drew 500 days to be testing days
- I repeated this process 100 times keeping track of the testing error rates for each estimator on each test set



Stock market data revisited

- The coin flip estimator has an error rate of 50% on average as expected
    - As low as 45% on some data sets!
- LDA and logistic regression seem to perform the best
    - Very similar results between the two
- The additional flexibility provided by QDA is only hurting us by adding extra variability into predictions

Support vector machines

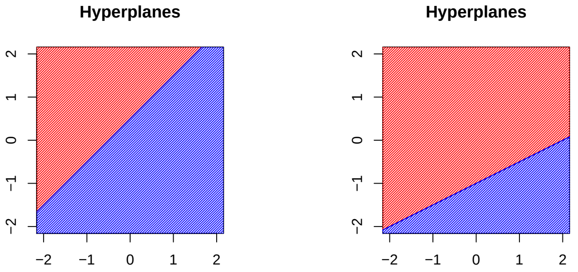
- Now we will work towards understanding a popular machine learning algorithm used for classification called support vector machines (SVM)
- SVMs are a widely used technique for classification
  - Been shown to work well in many settings
  - Very flexible
- We will start with a simple classifier and slowly extend it to the SVM

Maximal margin classifier

- We begin with the simplest form of SVM, called the maximal margin classifier
- In practice, this algorithm is not very useful, but is useful for illustrating the ideas behind SVMs in a more straightforward manner
  - Easy to visualize
- Before discussing the maximal margin classifier, we need to understand hyperplanes

Maximal margin classifier

- In a $p-$dimensional space, a hyperplane is a flat subspace of dimension $p-1$ that splits the space into two
- In two dimensions, a hyperplane is simply a line



Maximal margin classifier

- In three dimensional space, a hyperplane is a plane
- In higher dimensions, it is difficult to visualize, but we can think of a generalization of a flat subspace that cuts the space into two parts
- Hyperplanes are simple mathematically, and defined by

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p = 0$$

- If a point $(x_1, \ldots, x_p)$ satisfies this equation, then it lies on the hyperplane

Maximal margin classifier

- Of more use to us is how the hyperplane splits the space into two types of points

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p > 0$$

or

$$\beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p < 0$$

- For any point, we can determine which side of the hyperplane it lies on
  - Effectively splits the $p-$dimensional space in half

Maximal margin classifier

- Now suppose that we are still interested in classification
- We observe n data points $(\boldsymbol{X}_i, Y_i)$ for $i = 1, \ldots, n$
- Assume Y_i is binary for now and that $Y_i \in \{-1, 1\}$
  - No longer takes values 0,1
  - Conceptually no different than before
  - Defining classes in this way will help with some of the mathematical formulation later

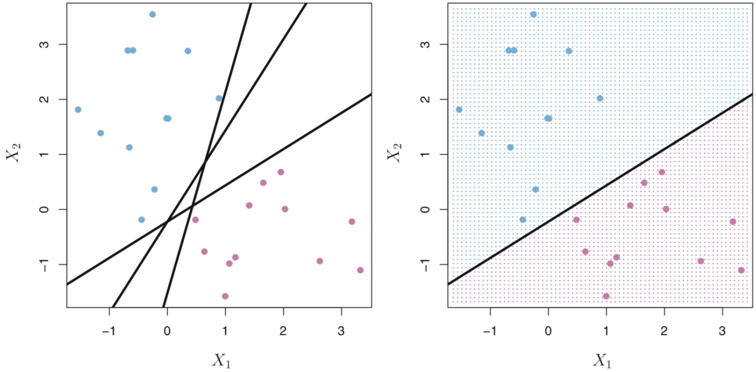
Maximal margin classifier

- As is usually the case, we are interested in building a model using training data with the goal of predicting a test data point
- The maximal margin classifier is based on the idea of a separating hyperplane
- We want to find a hyperplane that perfectly separates our training data into the two classes
  - On one side of the hyperplane, all $Y_i = 1$
  - $Y_i = -1$ on the other side

Maximal margin classifier

- Suppose for now that this is possible
  - There exists such a hyperplane
  - We will drop this assumption later

- Below is an example where the classes are perfectly separable



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Maximal margin classifier

- Base classification on whether a point is above or below the hyperplane
- If $Y_i = 1$, we have that

$$\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} > 0$$

- If $Y_i = -1$, we have that

$$\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} < 0$$

- Alternatively can write that

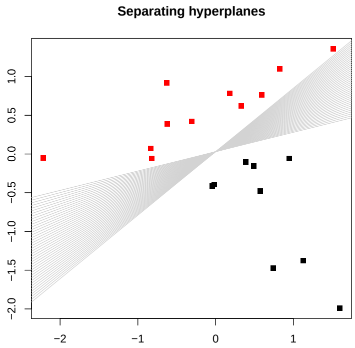
$$Y_i(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}) > 0$$

Maximal margin classifier

- This looks similar to the logistic regression or LDA classifiers
- Linear classifier in the covariates
- We will see that it differs in how the coefficients are estimated
- Further, we will see that it is very easy to imbed nonlinearity into this model in a different way than we have seen previously

Maximal margin classifier

- Is this hyperplane unique?
- There are actually infinitely many hyperplanes that separate the data when the data are separated

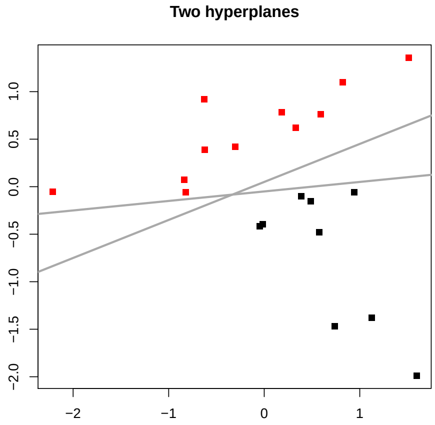


Maximal margin classifier

- We need to choose one hyperplane to use as our classifier
- All of these perfectly separate the training data
  - No difference on the training data
- We can choose the one that we think is most likely to work on the testing data
- To do this, we will find the hyperplane that is farthest from the training data

Maximal margin classifier

- Here is the previous example with two possible hyperplanes
- Which do we think is best?

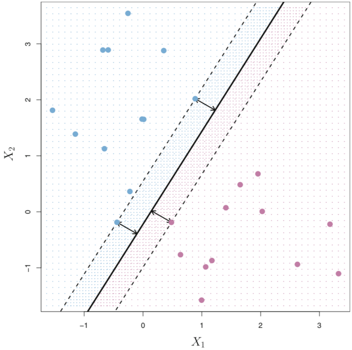


Maximal margin classifier

- For any given hyperplane, we can calculate the distance between the training points and hyperplane
- We can find the minimum distance among all training data points
  - This is called the margin
- The maximal margin classifier is the separating hyperplane that has the largest margin
  - Greatest distance between hyperplane and closest training point

Maximal margin classifier

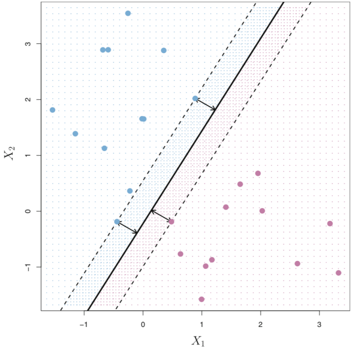
- Imagine placing two parallel hyperplanes that border the data
  - Chosen to maximize distance between these planes
- The maximal margin classifier is the plane halfway between these two planes



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Maximal margin classifier

- In this case, there are 3 points that are equally close to the hyperplane
- These three points are called support vectors



James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.

Maximal margin classifier

- If the support vectors were to move, then the maximal margin hyperplane would move as well
- The maximal margin hyperplane does not depend in any way on the other points
- If the other points were to move, it would not change the hyperplane unless they moved inside the boundary defined by the hyperplane and margin
- We will see this reliance on only a small set of points has implications for estimating SVMs later

Maximal margin classifier

- Now we need to find the maximal margin classifier
- It is easy to visualize in 2d, but this is not possible in higher dimensions
- The maximal margin classifier is the solution to an optimization problem
- We want to maximize the margin, while ensuring that the classes are separated

Maximal margin classifier

- Mathematically this optimization problem takes the following form

$$\underset{\beta_0,\beta_1,\ldots,\beta_p,M}{\text{Maximize}}\ M$$

$$\text{subject to } \sum_{j=1}^{p} \beta_j^2 = 1$$

$$Y_i(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}) > M \ \forall \ i = 1, \ldots, n$$

- Let's discuss each of these lines separately

Maximal margin classifier

- The constraint that $\sum_{j=1}^{p} \beta_j^2 = 1$ is not really a restriction, because if

$$\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip} = 0$$

then also

$$k(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}) = 0$$

so for the purposes of classification or defining the hyperplane, this constraint does not matter

Maximal margin classifier

- The main reason for this constraint is that if it is true, then the distance from a point $(X_{i1}, \ldots, X_{ip})$ to the hyperplane is given by

$$Y_i(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip})$$

- This then helps us understand the third line, which states

$$Y_i(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip}) > M \ \forall \ i = 1, \ldots, n$$

  This means that the distance from all points to the hyperplane must be at least M

Maximal margin classifier

- M is the margin, and is the quantity we want to maximize
- Note in this algorithm that

$$Y_i(\beta_0 + \beta_1 X_{i1} + \cdots + \beta_p X_{ip})$$

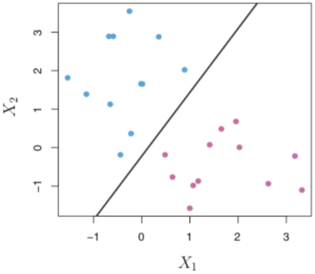
  is greater than M for all i
- That means that no data points can lie on the wrong side of the hyperplane
  - Perfectly separated data
  - Lying on the wrong side would lead to a negative value of this expression

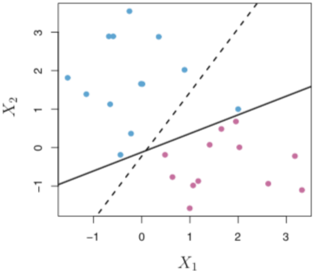
Maximal margin classifier

- What happens if we can't separate the data with a hyperplane?
- This optimization won't have a solution for $M > 0$
- In nearly all applications, we won't be able to separate the data completely
- We need to relax this optimization algorithm in a way that finds a hyperplane that still separates the data reasonably well, but allows for some observations to lie on the wrong side of the hyperplane

Maximal margin classifier

- Another reason to relax this optimization is that the maximal margin classifier is highly sensitive to individual data points
- This highlights that the classifier might be overfit to the data





James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). An introduction to statistical learning. New York: springer.