Linear Classifiers

kNN, LDA, Naive Bayes

Ramesh Srinivasan

October 10, 2024

Bayes Classifier

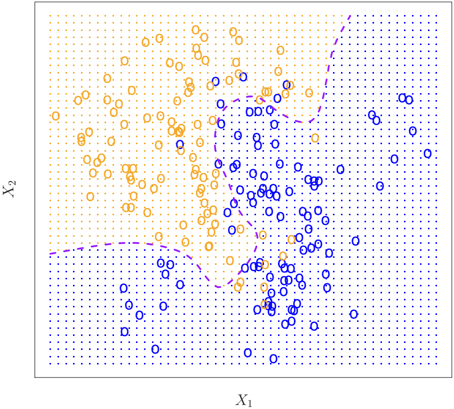
- Suppose we fit a classifier model $\hat{f}(x)$ to some **training** data $Tr = \{x_i, y_i\}$ of size N. Our metric of performance is of course the performance of test data that we have set aside.
- a very simple classifier that assigns each observation to the most likely class, given its predictor values will minimize the test error rate **on average**. In other words, we should simply assign a test observation with predictor vector x_i to the class j for which

$$p(Y = j|X = x_i)$$

  is largest.
- This simple classifier is called a Bayes classifier.
- In a 2-class (0/1) classifier, if

$$p(Y = 1|X = x_i) > 0.5$$

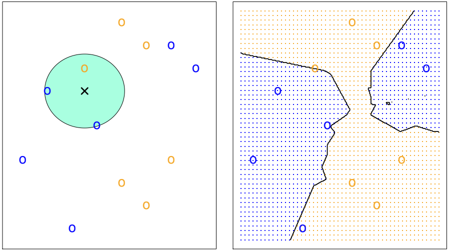
  the sample x_i is of class 1.



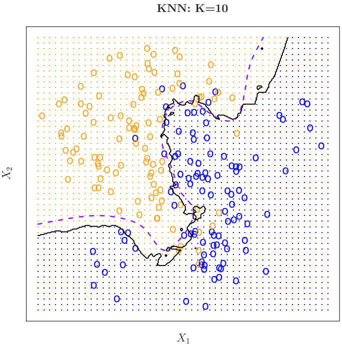
A simulated data set consisting of 100 observations in each of two groups, indicated in blue and in orange. The purple dashed line represents the Bayes decision boundary. The orange background grid indicates the region in which a test observation will be assigned to the orange class, and the blue background grid indicates the region in which a test observation will be assigned to the blue class.

K-nearest neighbor classifier

- Since we dont know the probability distribution that generated the real data, we cant ever know the Bayes classifer.
- K-NN classifiers attempt to estimate the conditional distribution of Y given X, so that we can classify an observation x_i to the class with the highest estimated probability.
- the KNN classifier identifies K points in the training data closest to x_i. The conditional probability for each class j is just the fraction of the K points where $Y = j$.
- KNN computes the probability for each j and then chooses the maximum probability.



The KNN approach, using K = 3, is illustrated in a simple situation with six blue observations and six orange observations. Left: a test observation at which a predicted class label is desired is shown as a black cross. The three closest points to the test observation are identified, and it is predicted that the test observation belongs to the most commonly-occurring class, in this case blue. Right: The KNN decision boundary for this example is shown in black. The blue grid indicates the region in which a test observation will be assigned to the blue class, and the orange grid indicates the region in which it will be assigned to the orange class.



The black curve indicates the KNN decision boundary on the data from Figure 2.13, using K = 10. The Bayes decision boundary is shown as a purple dashed line. The KNN and Bayes decision boundaries are very similar.

Logistic Regression

- Logistic Regression models $p(Y = 1|X = x_i)$ using the logistic function.
- It is in fact a model of the conditional distribution of Y given X.

$$p(Y = 1|X = x) = p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

$$\frac{p(Y = 1|X = x)}{p(Y = 0|X = x} = \frac{p(x)}{1 - p(x)}$$

$$ln(\frac{p(x)}{1 - p(x)}) = \beta_0 + \beta_1 x$$

- The expressions get more complicated for Multiple Logistic Regression (more than one predictor) or Multinomial Logistic Regression (more than 2 classes) but the concept remains the same.

Generative Modeling Approach

- Model the distribution of the predictors X separately in each of the response classes (i.e. for each possible value of Y), i.e., $p(X = x_i | Y = k)$
- Make use of Bayes theorem:

$$p(Y = k | X = x_i) = \frac{p(X = x_i | Y = k) p(Y = k)}{\sum_k (p(X = x_i | Y = k) p(Y = k)}$$

- The estimate $p(Y = k)$ is the simply fraction of the training data in each class.
- Limiting case of 2 classes (0/1), and a "balanced" training set with equal numbers of each class:

$$p(Y = 1 | X = x_i) = \frac{p(X = x_i | Y = 1)}{p(X = x_i | Y = 1) + p(X = x_i | Y = 0)}$$

- The challenge is to model the probability distribution.

Decision Criterion

- The objective in our modeling is to select the class such that $p(Y = k|X = x_i)$ is maximized.
- If we think about a two class (1/0) problem we can simply evaluate the odds ratio:

$$\frac{p(Y = 1|X = x_i)}{p(Y = 0|X = x_i)} = \frac{p(X = x_i|Y = 1)p(Y = 1)}{p(X = x_i|Y = 0)p(Y = 0)}$$

  if this ratio is larger than 1 choose class 1, otherwise choose class 0.
- In general if we have K classes, we can use class K as the baseline and do the same evaluation. If the ratio is below 1 choose class K, or else choose the highest ratio.

Linear Discriminant Analysis - univariate (p=1)

- Linear Discriminant Analysis (LDA) makes the assumption that the forms of $p(x)$ are the normal distribution,

$$f_k(x) = \frac{1}{\sqrt{2\pi}\sigma_k} e^{-\frac{(x-\mu_k)^2}{2\sigma_k^2}}$$

where μ_k and σ_k^2 are the mean and variance for each class k.

- A further simplifying assumption is that the variance is shared across classes, so we can drop the k subscript for σ^2
- Again lets consider the two class case. If we plug this into our decision criterion for observation x_i:

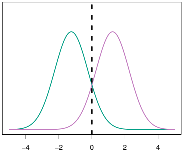
$$\frac{p(Y=1|X=x_i)}{p(Y=0|X=x_i)} = \frac{(e^{-\frac{(x-\mu_1)^2}{2\sigma^2}}\Delta x)p(Y=1)}{(e^{-\frac{(x-\mu_0)^2}{2\sigma^2}}\Delta x)p(Y=0)}$$

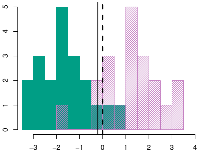
Linear Discriminant Analysis - univariate (p=1)

- Lets again make the simplifying assumption that our training set is balanced between Y = 1/0 and compute the log:

$$ln(\frac{p(Y=1|X=x_i)}{p(Y=0|X=x_i)}) = -\frac{(x-\mu_1)^2}{2\sigma^2} + \frac{(x-\mu_0)^2}{2\sigma^2}$$

- If this log odds is greater than 0, I should pick class 1 and if its less than 0 I should pick class 0. That boils down to, if the observation is farther to the mean of class 0, I should pick class 1 and vice versa.
- The optimal decision boundary is the midpoint between the two distributions.





Linear Discriminant Analysis - univariate (p=1)

- The linear discriminant analysis (LDA) method approximates the linear Bayes classifier by estimating:

$$\hat{\mu}_k = \frac{1}{n_k} \sum_{i:y_i=k} x_i$$

$$\hat{\sigma}^2 = \frac{1}{n-K} \sum_{k=1}^{K} \sum_{i:y_i=k} (x_i - \hat{\mu}_k)^2$$

$$\hat{p}_{Y=k} = \frac{n_k}{n}$$

where n is the total number of observations and n_k is the number of observations in class k.

Linear Discriminant Analysis - multivariate

- When we have p predictors, we now have to fit the distributions with a multivariate normal.

$$f(\boldsymbol{x}) = \frac{1}{(2\pi)^{p/2}\det(\boldsymbol{\Sigma})^{\frac{1}{2}}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}-\boldsymbol{\mu})}$$

- Please note my notation, where I use BOLD for vectors and matrices.
- Here $\boldsymbol{x}$ is a vector containing one observation of the predictors, $\boldsymbol{\mu}$ is a vector of the means, and $\boldsymbol{\Sigma}$ is the covariance matrix.
- The Bayes classifier assigns the observation to the class k for which

$$\delta_k(\boldsymbol{x}) = \boldsymbol{x}^T \boldsymbol{\Sigma}^{-1} \boldsymbol{\mu_k} - \frac{1}{2}\boldsymbol{\mu_k}^T \boldsymbol{\Sigma} \boldsymbol{\mu_k} + ln(\hat{p}_{Y=k})$$

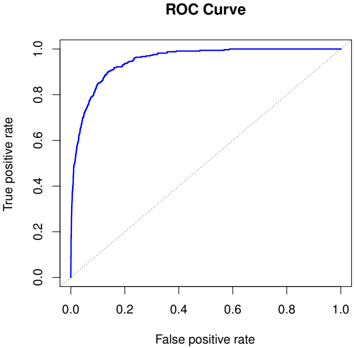
  is maximum
- The most important thing for you to pay attention here is to note that the estimator depends on a covariance matrix that is invertible.
- In practice this is a hard criteria to meet unless you only have a few weakly correlated variables.

Decision Rule and Reciever Operating Characteristic (ROC)

- In a two-class problem (1/0) The Bayes classifier uses a Decision Rule to choose class 1 if

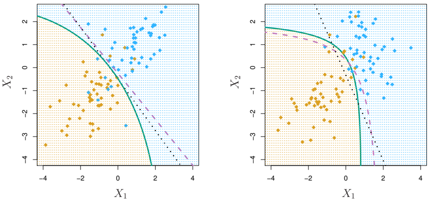
$$p(Y = 1|\boldsymbol{x} = \boldsymbol{x_i}) > 0.5$$

- In practice your threshold depends on your utility function. The Decision Rule can be adapted to accept more false positives by lowering the threshold, or are you willing to increase the chance of false negatives by raising the threshold.
- LDA is scikit-learn will return prediction probabilities to allow you to make such decisions.
- The ROC curve allows you to visualize this trade off, and the AUC provides a metric of classifier performance, independent of threshold choice.



QDA vs LDA

- QDA allows for a quadratic decision boundary, but at a high cost in terms of parameter estimates as QDA estimates a separate covariance matrix for each class.
- LDA is a much less flexible classifier than QDA, and so has substantially lower variance. This can potentially lead to improved prediction performance. But there is a trade-off: if LDA's assumption that the K classes share a common covariance matrix is badly off, then LDA can suffer from high bias.



Naive Bayes Classifier

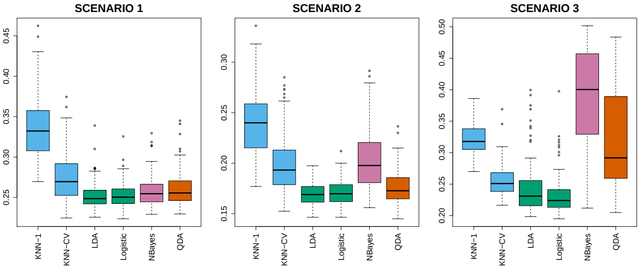
- Within the kth class, the p predictors are independent.

$$f_k \boldsymbol{x} = f_{k1}(x_1) \times f_{k2}(x_2) \times .... \times f_k p(x_p)$$

- We make a simplifying assumption that there is no covariance between predictors.
- If we make the normal assumption, we can compute μ_{kp} and σ_{kp} for each predictor in each class.
- We could instead make a non-parametric estimator like a kernel density estimate (i.e., smoothed histogram).
- If we have qualitative predictors, we can just make a histogram to estimate the probability distribution.

Simulations

- Scenario 1: 20 observations of uncorrelated random variables with a different mean in each class.
- Scenario 2: Same as (1) with predictors having a correlation of -0.5
- Scenario 3: Generated from a t-distribution (more extreme values) strong negative correlation.



Summary

- LDA assumes that the log-odds is a limear function of the data, while QDA assumes its quadratic. You need a **lot** of data and evidence that the covariance among predictors strongly depends on class to want to explore QDA.
- Logistic Regression is a linear model of the odds but used a different optimization procedure based on maximizing Likelihood of the logistic model. Simply put, if the data is close to normal, LDA will perform better, otherwise Logistic will do better.
- The ability to use empirical distributions means it tends to do better if the sample size is small. However, correlation among predictors renders Naive Bayes to be unusable violating the independence assumption.
- kNN can handle complex scenarios well but simply does not scale with the number of predictors (your homework).