Naïve Bayes

In the anemia classi\_er with two features (for example, the hemoglobin

concentration level and the age), the na\_ve hypothesis implies that

hemoglobin concentration and age are conditionally independent.

In the spam classi\_er, the na\_ve hypothesis implies that the probability of

occurrence of a certain word is not related to the occurrence of the other

words in the email.

Under conditional independence PXi jXi+1;:::;Xd ;Y = PXi jY and hence for the k-class

Problem

Why do we need another method, when we have logistic regression?

There are several reasons:

• When the classes are well-separated, the parameter estimates for the

logistic regression model are surprisingly unstable. Linear discriminant

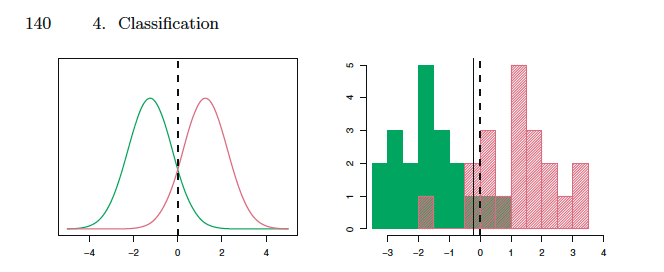
analysis does not suffer from this problem.

• If n is small and the distribution of the predictors X is approximately

normal in each of the classes, the linear discriminant model is again

more stable than the logistic regression model.

• As mentioned in Section 4.3.5, linear discriminant analysis is popular

when we have more than two response classes.

notice for two classes y =1 and 0 , the sigma sq =1 is same, something that we will assume in LDA that for both the classesthe sigma sq is same

**LDA**

Class-specific performance is also important in medicine and biology,

where the terms sensitivity and specificity characterize the performance of

sensitivity

specificity a classifier or screening test. In this case the sensitivity is the percentage of

true defaulters that are identified, a low 24.3% in this case. The specificity

is the percentage of non-defaulters that are correctly identified, here (1 −

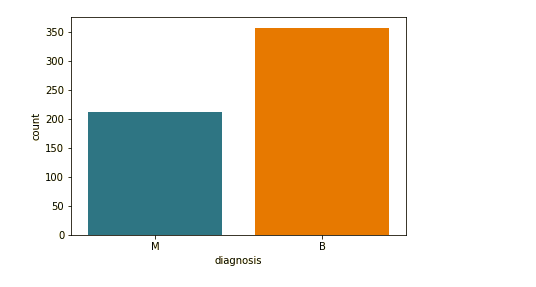
23/9, 667)Å~ 100 = 99.8%.

Why does LDA do such a poor job of classifying the customers who default?

In other words, why does it have such a low sensitivity? As we have

seen, LDA is trying to approximate the Bayes classifier, which has the lowest

total error rate out of all classifiers

We can start by showing this 

Counter intuitive – SVM data reduction

So first you decrease the dimensionality and then you choose higher dimensions of your features

Data dimensionality for support Victor machine. It seems to be counterintuitive because first we reduce the number of dimensions for the features and then we have to increase the number of dimensions to use our kernel tricks.

Model Validation-

Hyperparameter optimization –

**Accuracy**

– check for train accuracy and test accuracy (with the validation set I guess no the overall testing set but the testing set used in cross validation)

**F-measure**