Lecture 10: Classification examples

Reading: Chapter 4

STATS 202: Data mining and analysis

Sergio Bacallado
October 14, 2013

Example. Predicting `default`

Used LDA to predict credit card default in a dataset of 10K people.

Predicted "yes" if $P(\texttt{default} = \text{yes}|X) > 0.5$.

		True default status		
		No	Yes	Total
Predicted default status	No	$9,644$	252	$9,896$
	Yes	23	81	104
	Total	$9,667$	333	$10,000$

- The error rate among people who do **not** default (false positive rate) is very low.
- However, the rate of false negatives is 76%.
- It is possible that false negatives are a bigger source of concern!
- One possible solution: Change the threshold.

Example. Predicting `default`

Changing the threshold to 0.2 makes it easier to classify to "yes".

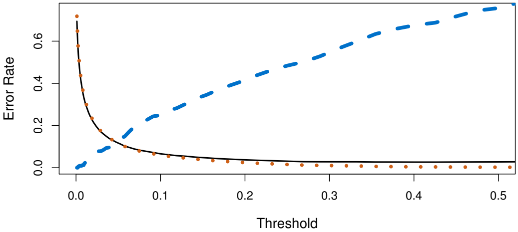
Predicted "yes" if $P(\texttt{default} = \text{yes}|X) > 0.2$.

		True default status		
		No	Yes	Total
Predicted default status	No	9,432	138	9,570
	Yes	235	195	430
	Total	9,667	333	10,000

Note that the rate of false positives became higher! That is the price to pay for fewer false negatives.

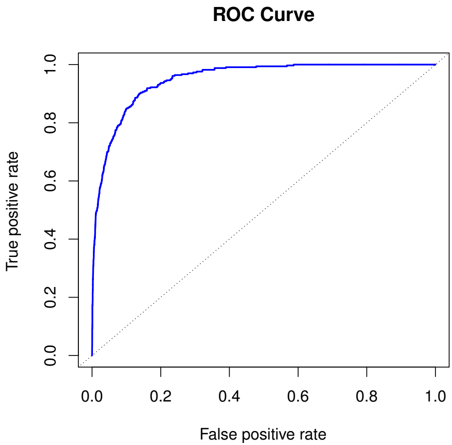
Example. Predicting `default`

Let's visualize the dependence of the error on the threshold:



- – – – False negative rate (error for defaulting customers)
- · · · · · False positive rate (error for non-defaulting customers)
- —— 0-1 loss or total error rate.

Example. The ROC curve



- Displays the performance of the method for any choice of threshold.
- The area under the curve (AUC) measures the quality of the classifier:
  - 0.5 is the AUC for a random classifier
  - The closer AUC is to 1, the better.

Thinking about the loss function is important

Most of the **regression** methods we've studied aim to minimize the RSS, while **classification** methods aim to minimize the 0-1 loss.

In classification, we often care about certain kinds of error more than others; i.e. the natural loss function is not the 0-1 loss.

Even if we use a method which minimizes a certain kind of training error, we can *tune* it to optimize our true loss function.

- e.g. Find the threshold that brings the False negative rate below an acceptable level.

In the Kaggle competition, what is our loss function?

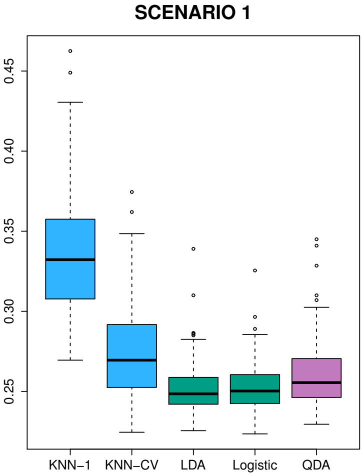
Comparing classification methods through simulation

1. Simulate data from several different known distributions with 2 predictors and a binary response variable.

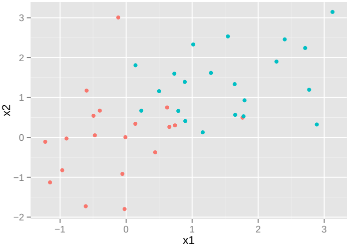
2. Compare the test error (0-1 loss) for the following methods:

   - KNN-1
   - KNN-CV ("optimal" KNN)
   - Logistic regression
   - Linear discriminant analysis (LDA)
   - Quadratic discriminant analysis (QDA)

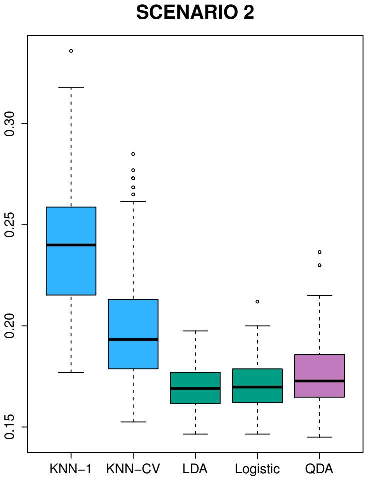
Scenario 1



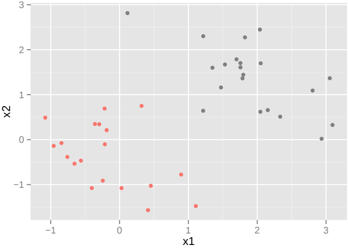
- X_1, X_2 standard normal.
- No correlation in either class.



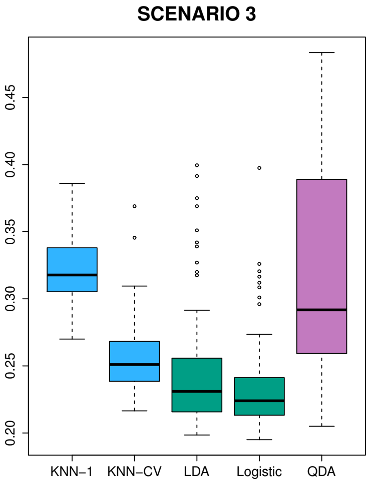
Scenario 2



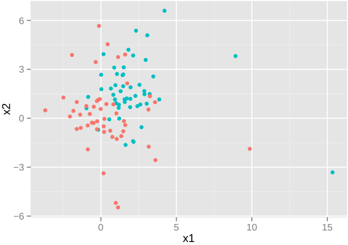
- X_1, X_2 standard normal.
- Correlation is -0.5 in both classes.



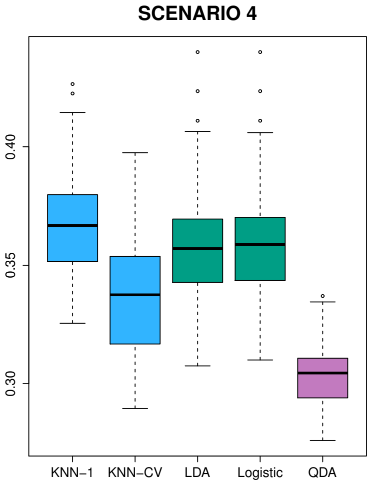
Scenario 3



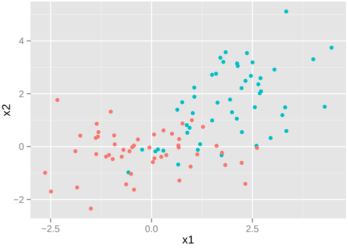
- X_1, X_2 Student t random variables.
- No correlation in either class.



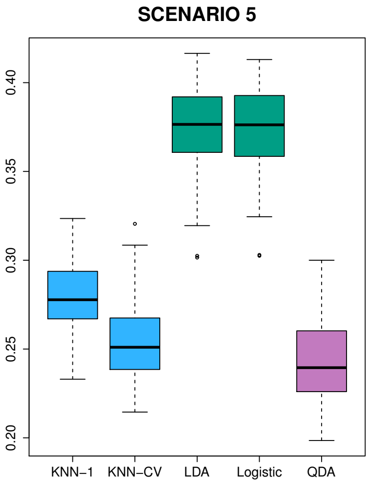
Scenario 4



- X_1, X_2 standard normal.
- First class has correlation 0.5, second class has correlation -0.5.



Scenario 5

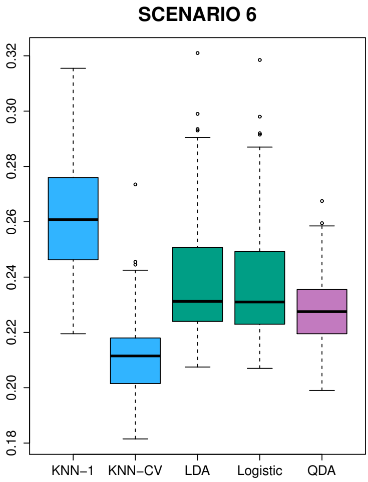


- X_1, X_2 uncorrelated, standard normal.
- Response Y was sampled from:

$$P(Y = 1|X) = \frac{e^{\beta_0+\beta_1(X_1^2)+\beta_2(X_2^2)+\beta_3(X_1X_2)}}{1 + e^{\beta_0+\beta_1(X_1^2)+\beta_2(X_2^2)+\beta_3(X_1X_2)}}.$$

- The true decision boundary is quadratic.

Scenario 6



- X_1, X_2 uncorrelated, standard normal.
- Response Y was sampled from:

$$P(Y=1|X) = \frac{e^{f_{\text{nonlinear}}(X_1,X_2)}}{1+e^{f_{\text{nonlinear}}(X_1,X_2)}}.$$

- The true decision boundary is very rough.

Next time: Cross-validation

Problem: Choose a supervised method that minimizes the test error. In addition, *tune* the parameters of each method:

- k in k-nearest neighbors.
- The number of variables to include in forward or backward selection.
- The order of a polynomial in polynomial regression.

Cross-validation is one way to approximate the test error:

- Divide the data into two parts.
- Train each model with one part.
- Compute the error on the other.