Lecture 3: KNN Classifier

Beidi Qiang

SIUE

Recall: The Classification Setting

Assume that there is some relationship between the label Y and features $\mathbf{x}$. Say

$$Y = f(\mathbf{x}) + \epsilon$$

We seek to estimate f on the basis of training observations $\{(x_1, y_1), \cdots, (x_n, y_n)\}$, where $y_1, \cdots y_n$ are qualitative (categorical). Most common approach for quantifying the accuracy of our estimate $\hat{f}$ is the error rate, defined as

$$\frac{1}{n} \sum I(y_i \neq \hat{y}_i)$$

- $\hat{y}_i$ is the predicted class label for the ith example using $\hat{f}$, and y_i is the true label.
- $I(y_i \neq \hat{y}_i)$ is an indicator variable that equals 1 if $y_i \neq \hat{y}_i$ and if 0 $y_i = \hat{y}_i$.

The Bayes Classifier

A good classifier is one for which the test error rate is smallest. It is possible to show that the test error rate is minimized, on average, by a very simple classifier called Bayes Classifier. Bayes classifier simply assign a test observation with features x_0 to the class j for which $Pr(Y = j|X = x_0)$ is largest.

- $Pr(Y = j|X = x_0)$ is a conditional probability: it is the probability conditional that $Y = j$, given the feature vector x_0 .
- The Bayes classifier produces the lowest possible test error rate, called the Bayes error rate, which equals to

$$1 - E\left(max_j Pr(Y = j|X)\right).$$

- Bayes error rate is always greater than 0 in any real world problem, because the classes overlap in the population. It is analogous to the irreducible error.

K-Nearest Neighbors

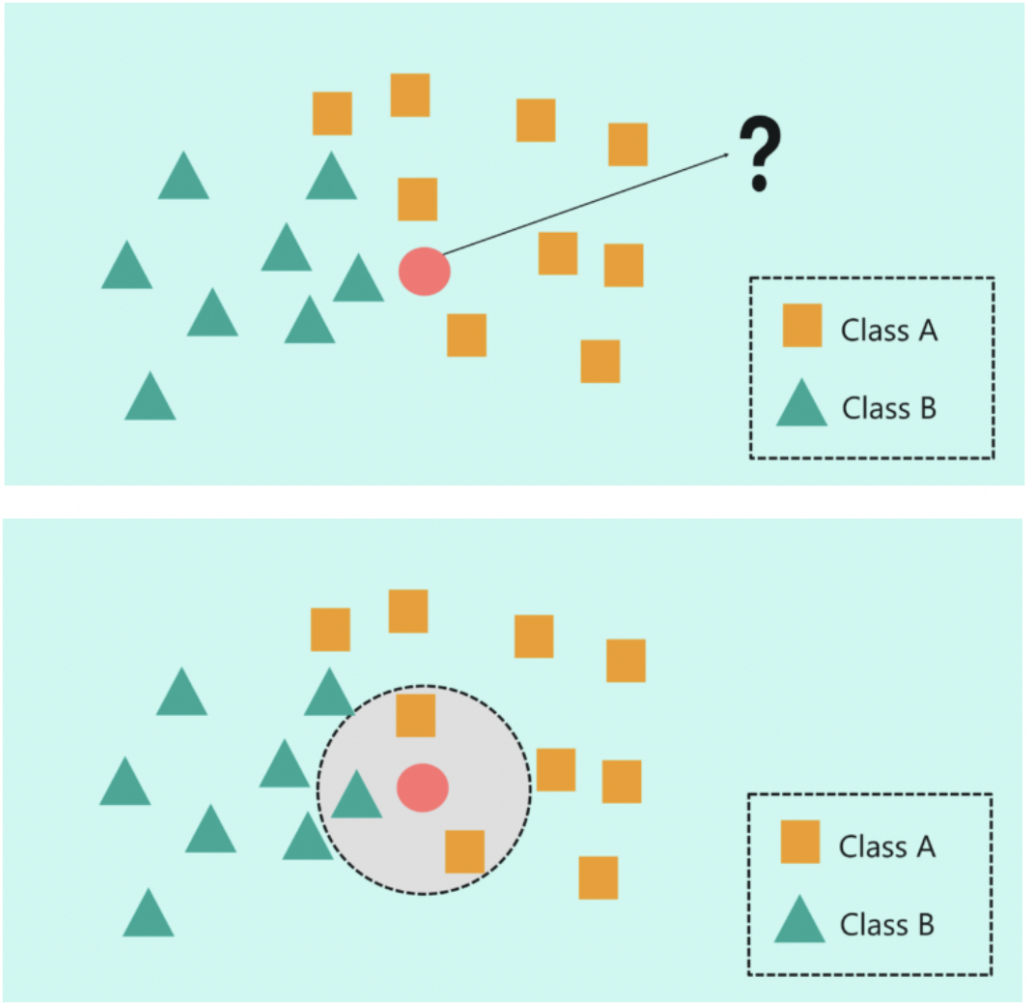
In theory we would always like to predict qualitative labels Y using the Bayes classifier. But for real data, we do not know the conditional distribution of Y given X.

Many approaches attempt to estimate the conditional distribution, and then classify a given observation to the class with highest estimated probability. One such method is the K-nearest neighbors (KNN) classifier.

- Take the K Nearest Neighbor of a new example from the training data.
- Estimates the conditional probability for class j as the fraction of points in neighbors whose label equals to j.
- Classifies the new example to the class with the largest estimated conditional probability, or equivalently the class, where you counted the most neighbors.

KNN Example

Consider the following example:



Distance measure

KNN uses distance measures to check the distance between a new example and its neighbors. Let's use $x_0 = (x_{01}, \cdots, x_{0m})$ to denote the m dimensional feature space of the new example. Commonly used distance measures are:

- Minkowski distance (Lp norm): a metric intended for real-valued feature vector.

$$D = \left(\sum_{i=1}^{m} |x_{0i} - x_{1i}|^p \right)^{1/p}$$

  - Euclidean Distance (L2 norm): $p = 2$
  - Manhattan distance (L1 norm): $p = 1$

- Hamming Distance: a metric for comparing two binary data strings. It is the number of bit positions in which the two bits are different. This is used mostly when you one-hot encode your data.
  - Suppose we have two strings "10100" and "11001". The hamming distance here will be 3.
- Gower distance: a metric that measures the dissimilarity with mixed feature vector (both quantitative and categorical variables).

Normalization/Standardization

When calculating distances, we have to take into account the units used. If the scale of features is very different, normalization or standardization is required. This is because the distance calculation done in KNN uses feature values. When the one feature values are large than other, that feature will dominate the distance hence the outcome of the KNN.

- Normalization: recalculation of feature to values between 0 (the minimum) and 1 (the maximum) with the formula:

$$x_{i,normalized} = (x_i - x_{min})/(x_{max} - x_{min})$$

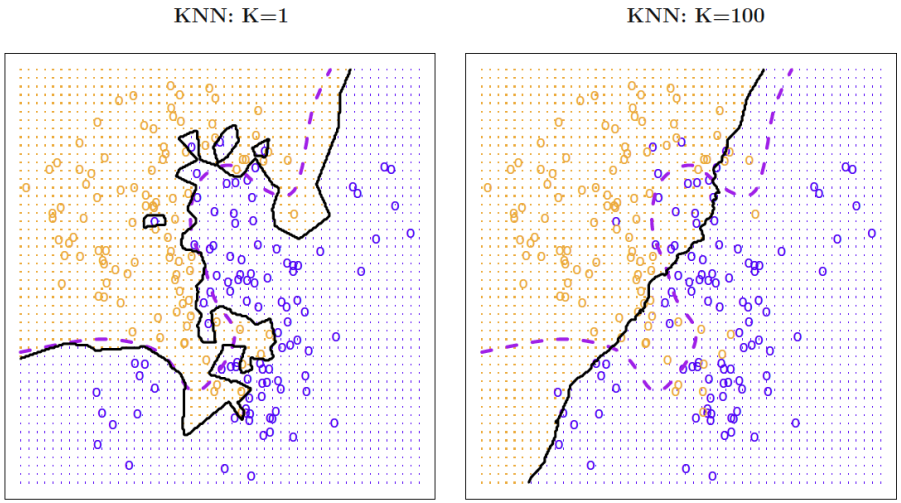
- Standardization: we determine the Z-scores of the data.

$$x_{i,standardized} = (x_i - mean(x))/sd(x)$$

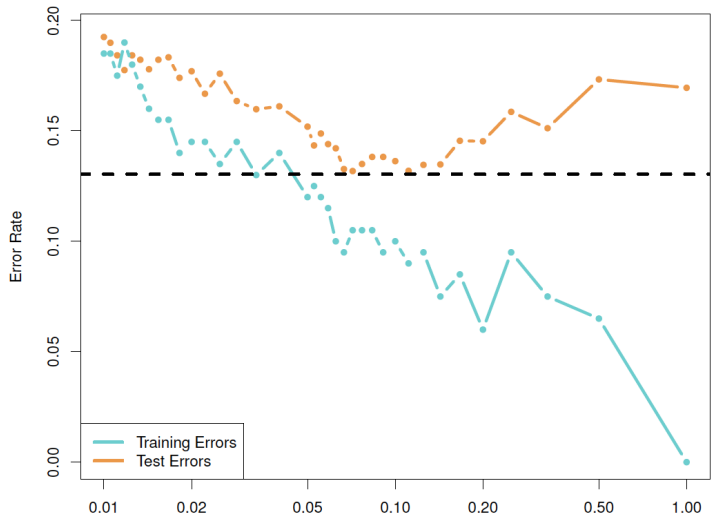
  Standardization is preferable to normalization if the criteria are unbounded (e.g., ranging from 0 to ∞) and/or there are outliers in either direction

The choice of K

The choice of K has a drastic effect on the KNN classifier.



The following plot gives the KNN test/training errors v.s. $1/K$.



Application of KNN in R

Let's apply the KNN algorithm step by step to an example. The steps are as follows:

1. Reading and describing the data
2. Preparing the data: Normalize (or standardize) the data and splitting the data into a training set and a test set
3. Training the model on the data
4. Evaluating the model
5. Improving the model.