DATA 8

Fall 2016

Lecture 35, November 18

Comparing Two Samples

Slides created by Ani Adhikari and John DeNero

Announcements

- Project 3 checkpoint Tuesday 11/22, final deadline Tuesday 11/29
- Homework will be assigned today:
  - Early submission: Wed 11/23 (usual schedule)
  - “Regular” submission: Monday 11/28 after the break
- GSI/Tutor office hours locations from now on:
  - Mondays Etcheverry 3106
  - Wednesdays Etcheverry 3108
  - Other days: no change

Finding the *k* Nearest Neighbors

To find the *k* nearest neighbors of a point:

- Find the distance between the point and each point in the training set
- Augment the training data table with a column containing all the distances
- Sort the augmented table in increasing order of the distances
- Take the top *k* rows of the sorted table

(Demo)

The Classifier

To classify a point:

- Find its *k* nearest neighbors
- Take a majority vote of the *k* nearest neighbors to see which of the two classes appears more often
- Assign the point the class that wins the majority vote

(Demo)

Assessing Accuracy

- Separate the data at random into a training set and a test set
- Use the training set to classify each point in the test set
- Find the fraction of points for which the classification is correct

(Demo)

A Much Simpler Classifier

- Take just one categorical attribute
- Compare:
  - Its distribution among Class 0 individuals
  - Its distribution among Class 1 individuals
- If the distributions are different, maybe you can use that to create another classifier

(Demo)

Relation between Attribute and Class

Categorical variables; how to decide about “relation”?

- **Null:**
  - In the population, the attribute and class are not related.
- **Alternative:**
  - In the population, the attribute and class are related

(Demo)

Permutation Test

- For whether two samples are drawn randomly from the same underlying distribution
  - the distribution of the attribute don’t depend on the class
- If the null is true, all rearrangements of the attribute values among the two classes are equally likely
- So compute the observed test statistic
  - Then shuffle the attribute values and recompute the statistic; **repeat**; compare with observed statistic