Data Analysis Techniques

Main goal: Students are able to recognize what technique might be useful for a given problem
Secondary goals: Impress friends with fancy-sounding names. Know what other people are referring to when they use these names.

Rough division of areas but not used consistently:

- *Databases*: Store and manage data, execute prescribed (but ad-hoc) queries
- *Data mining:* Find patterns in large datasets
- *Machine learning*: Make inferences and predictions using large datasets

Basic database operations

1. Show `Temps` table
2. *Filtering*: city,state,temp where temp < 10 -- `VeryCold.csv`
3. *Sorting*: city,state sorted by latitude (hidden) -- `SouthNorth.csv`
4. *Aggregating*: overall average temperature, then average for each state (sorted warmest to coldest) -- `AvgTemp.csv`, `AvgByState.csv`
5. *Joining*: Show `Regions` table, combine `Temps` and `Region` matching on state -- `JoinedTables.csv`
6. *Composing operations*: Join two tables, average temperature for each region, filter regions with avg < 25, sort coldest to warmest, include warmest city in region (note two lines for Midatlantic due to tie for warmest city) -- `GrandFinale.csv`

Traditional data mining

1. Market basket data
2. Frequent itemsets and association rules
3. Examples seen in previous class

Machine learning concepts

- *Regression:* decide output value for an item based on set of input values
- *Classification:* decide category an item belongs to based on set of features
- *Regression versus classification:* regression input and output values are from an ordered domain, usually continuous; classification output values are from a set of unordered categories, input values may be ordered/continuous or not
- *Clustering:* create groups of similar items
- *Anomaly detection:* find items that don't conform to pattern
- *Supervised* (training data) versus *unsupervised* (no training data)
    - Regression and Classification usually supervised
    - Clustering usually unsupervised
    - Anomaly detection either

Regression

1. Explain simple linear regression, least squares measure
   Examples: SAT as function of GPA, test score as function of hours studied, sales as function of advertising dollars, body fat as a function of BMI (weight / height^2)
2. Correlation coefficient: complex formula on x,y values yields number between 1 (highly correlated) and -1 (highly reverse correlated); 0 is uncorrelated
3. Show temperature versus latitude -- `TempVsLat`, `TempVsLatRegression`
4. Based on outliers, speculate correlation with longitude; show temperature versus longitude -- `TempVsLong`, `TempVsLongRegression`
   Note error in longitude axis
5. Speculate perhaps Lat+Long would be best (multiple independent variables)
6. Underfitting, overfitting, limitations: `UnderOverFitting` graphic, `Anscombe's quartet`

Classification

K nearest neighbors (KNN)

1. Multidimensional feature space
   Customer example: gender, age, income, zipcode, profession
2. Distance metric
   Example: equality for gender, profession; difference for age, income; proximity for zipcode
3. Classification: assign to category, e.g., likelihood of buying in (high,medium,low)
   Find *k* closest items, assign item to most frequent category
4. Draw 2D representation
5. Temperature example: temperature categories, predict based only on latitude/longitude
   Show `TempsCat.csv` and `LatLongScatter.jpg`
6. Try two cities: Dallas, Texas (long 96.8, lat 32.8); Davenport, Iowa (long 90.6, lat 41.5)
   Truth: Dallas comfy, Davenport cold
7. Can also use for regression via average values
   Customer example: predict dollars spent
   Temperature example: predict temperature from latitude/longitude
   Show `LatLongScatterTemps.jpg` -- Dallas temp 34, Davenport temp 13

Decision tree classifier

- Multidimensional feature space, yes/no or partition questions over feature values
- Navigate to bottom of the tree, find category
- Customer example: gender split, then age partitions, then income, categories on leaves. Show classification of new customer.
- Temperature example: Show `CatNoTemps.csv`, want to predict category from other "features", speculate latitude as most discriminating, but what next? Show `CatNoTempsSorted.csv`
- Primary challenge is in building "good" tree from training data

Naive Bayes: probabilistic

- *Independence:* Given two features X and Y, the probability that X=x is independent of the probability that Y=y (e.g., possibly gender and age; *not* income and zipcode)
- *Conditional independence:* Given two features X and Y and a category c, if an item is in category c then the probability that X=x is independent of the probability that Y=y. More relaxed than full independence but in practice often the same. (This assumption is what makes the approach "naive".)
- Calculate from training data:
    a. Fraction (probability) of items in each category
    b. For each category, fraction (probability) of items in that category with X=x for each feature X and value x
- Given new item, for each category compute: probability of being in that category (a) times probability of being in that category given feature values (product of b's). Pick the category with the highest result.
- Example: Predict temperature category from region and coastal.
  Show `CategoryProbabilities.csv`, `ConditionalProbabilities.csv`
  Coastal city in Northeast, probabilities:
      warm: 0.1 * 1 * 0 = 0
      comfy: 0.27 * 0.8 * 0 = 0
      cool: 0.28 * 0.44 * 0.13 = 0.016
      cold: 0.25 * 0.29 * 0.15 = 0.011
      frigid: 0.09 * 0 * 0.2 = 0
  Non-coastal city in Southatlantic, probabilities:
      warm: 0.1 * 0 * 0.5 = 0
      comfy: 0.27 * 0.2 * 0.41 = 0.022
      cool: 0.28 * 0.56 * 0.13 = 0.020
      cold: 0.25 * 0.71 * 0 = 0
      frigid: 0.09 * 1 * 0 = 0

Underfitting and overfitting in classification

- Example: Classifying objects as chairs. *Underfitting*: Four legs and flat section; would also capture tables, elephants. *Overfitting*: four legs, 3.5 feet high, red cushion; would not capture most chairs
- Show `PresidentOverfitting.jpg`

Clustering

- Multidimensional feature space, distance metric
- Goal: Partition dataset into *k* groups such that items in groups are close to each other.
- *k-means*: Each partition has a mean value; for each item compute square of distance from mean. Goal is to minimize sum of those squares.
- Temperature example: Cluster cities into six groups based on latitude/longitude.
  Show `Points.jpg`, `Clusters.jpg`, `ClusterMeans.jpg`
- Note clusters need not be of similar sizes

Anomaly detection

- Find “outliers” either by examining data or using training set with normal/abnormal labels
- *Supervised version* = classification into two categories (normal,abnormal)
- *Unsupervised using regression*: distance from line, show `TempVsLatRegression`
- *Unsupervised using k nearest neighbors*: Item is an anomaly if more than *n*% of *k* nearest items are in a different category, show `LatLongScatter.jpg` but would need denser points
- *Unsupervised using clustering*: How much is clustering improved by removing item?