OVERVIEW Study Notes for my CS506 FINAL

1 Distance and Similarity

Distance

- A way to measure how different two data points are
- **Good distance measures:**
  - Always positive (distance can't be negative)
  - Symmetric (distance from A to B = distance from B to A)
  - Follows the triangle shortcut rule (going directly from A to C is never longer than going A→B→C)

Types of Distance Measures

1. **Straight-Line Distance (Euclidean)**
   - Like measuring with a ruler, DIRECT DISTANCE
   - Works well for "normal" numerical data
1. **Manhattan**
   - Like walking in a grid city (only horizontal/vertical moves)
   - Less sensitive to outliers than Euclidean
- If a "distance" breaks the triangle rule, it's not a true distance function

2 Clustering

K-Means (The Classic Method)

- **Goal:** Group data into **k** clusters
- **How it works:**
  a. Pick **k** random centers
  b. Assign each point to the nearest center
  c. Move centers to the average of their points
  d. Repeat until centers stop moving
- **Problems:**
  - Gets stuck in bad groupings if centers start poorly
  - Works best on round, evenly sized clusters

K-Means++

- Chooses first center randomly
- Next centers are picked from points far from existing centers
- Helps avoid terrible initial groupings

Hierarchical Clustering

- **Bottom-Up Approach:**
  - Start with every point as its own cluster
  - Repeatedly merge the two closest clusters
  - Stop when everything is in one big cluster
- **How to Measure "Close":**
  - **Single Link:** Distance between closest points in clusters
  - **Complete Link:** Distance between farthest points
  - **Average Link:** Average distance between all points
  - **Result:** A tree (dendrogram) showing how clusters merged

DBSCAN (Density-Based Clustering)

- Finds clusters based on crowded areas
- **Two Settings:**
  - How close points need to be to be neighbors
  - Minimum neighbors to form a dense area
- **Types of Points:**
  - **Core Points:** Have enough neighbors to start a cluster
  - **Border Points:** In a cluster but not dense enough to hold it together
  - **Noise Points:** Don't belong anywhere
- **For** Odd-shaped clusters and noisy data

Gaussian Mixture Models (GMM)

- Assumes data comes from several overlapping bell curves
- **Soft Clustering:** Points can belong partially to multiple clusters
- **How it Works:**
  a. Guess some bell curves
  b. Assign points probabilistically to each curve
  c. Adjust curves to fit better
  d. Repeat until curves stabilize

3 SVD (Simplifying Data)

What It Does

- Breaks data into simpler, more important parts
- Like finding the main directions where data varies most

Key Concepts

1. **Rank:** Number of truly independent directions in data
    - A flat line has rank 1 (all points along one direction)
    - A filled square has rank 2 (needs two directions to describe it)
1. **Principal Components:**
    - First component points where data spreads most
    - Next components capture remaining spread, perpendicular to previous ones
1. **Using SVD:**
    - **D:** Keep only important components, discard weak ones
    - **C:** Represent data with fewer numbers

4 Classification

K-Nearest Neighbors (KNN)

- **Simple Rule:** A point is whatever its closest neighbors are
- **Choosing k:**
    - Small k (like 1): Follows every twist in data (risks overfitting)
    - Large k: Smoothes out quirks (may miss details)
- **Critical Step:** Make sure all features are on similar scales!

Decision Trees

How They Work:

- Ask yes/no questions to split data (eg, "Is age > 30?")
- Keep splitting until groups are pure enough

Measuring Split Quality:

- **GINI Impurity:** How mixed a group is (0 = all same, 05 = evenly split)
- Better splits lower impurity in child groups

Weakness: Can grow too complex and memorize data (overfitting)

Naive Bayes

- **Assumption:** Features affect result independently (often not true, but works surprisingly well)
- **Fast and Simple:** Good for quick baseline models

Support Vector Machines (SVM)

- **Goal:** Find the widest possible "street" between classes
- **Kernel Trick:** Can twist data into higher dimensions to make separation easier
- **RBF Kernel:** Controls how flexible the boundary is (small γ = smooth, large γ = wiggly)

5 Regression

Linear Regression

- **Fits a Straight Line** to predict numbers
- **Assumptions:**
  - Relationship is roughly linear
  - Errors are normally scattered around the line

Logistic Regression

- Predicts probabilities (like chance of being in class 1)
- **Sigmoid Function:** Squashes predictions into 0-1 range
- **Decision Boundary:** Where probability = 50%

Regression Trees

- **Like Decision Trees for Numbers:**
- Splits data based on feature values
- Predicts the average in each final bucket
- **Handles Non-Linear Data Well**