INFO251 – Applied Machine Learning

Lab 11

Emily Aiken

Announcements

- **PS6** due Monday April 18
- **PS7** released, due Monday May 2
- **Quiz 2** on Thursday, April 28
- It’s not too late to **participate**! ☺

Remaining Labs

- **Today**: Unsupervised learning
- **Next week (April 20)**: Quiz review
- **April 27**: Applied machine learning start-to-finish (guest lab from Esther Rolf)

Topics: Unsupervised Learning

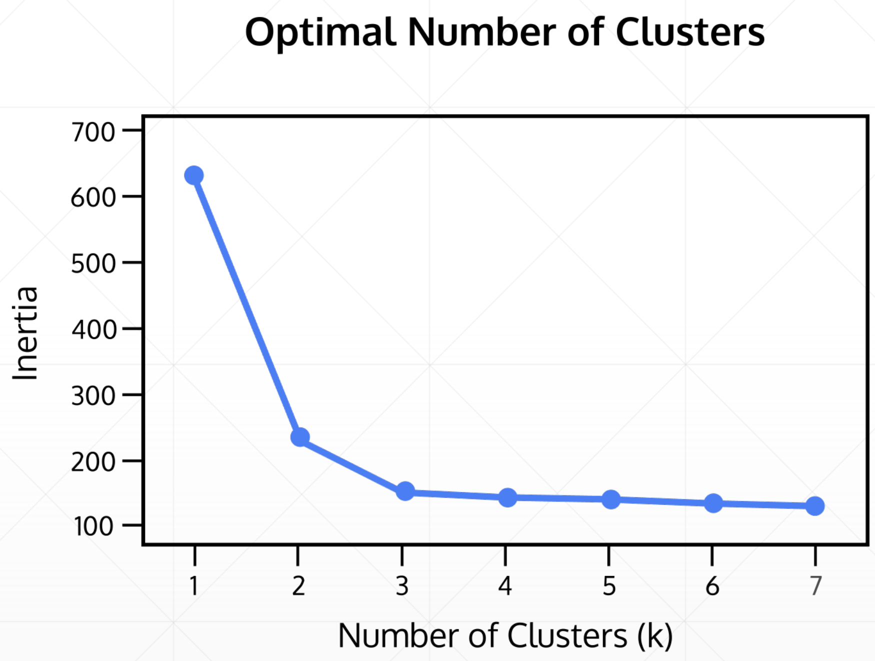
- K-Means clustering
- Other types of clustering
- Principal components analysis (PCA)

K-means clustering

- Fitting the k-means algorithm
  - Initialization: Guess some **cluster centers** at random
  - Repeat until converged:
    - All points are assigned to the closest cluster center
    - Cluster centers are redefined as the algorithmic mean of all points assigned to the cluster
- Guarantee: Cluster centers are the mean of the observations in each cluster, and each point is closer to its own cluster center than any other
  - Sensitive to random initialization

What is the “right” number of clusters?

- **Option 1:** Set number of clusters based on expert knowledge
- **Option 2:** Use the “elbow method”
  - **Inertia:** The average squared distance between an observation and its cluster center
    - Decreases monotonically with the number of clusters
  - Plot inertia as a function of the number of clusters, and look for where the drop in inertia begins to slow



Other clustering algorithms

- **Hierarchical agglomerative clustering (HAC):** Every observation starts in its own cluster, combine clusters recursively according to distance metric
- **Hierarchical divisive clustering (HDC):** All the observations start in one cluster, split clusters until every observation is separated
- **Density-based spatial clustering of applications with noise (DBSCAN):** Group together observations in high-density neighborhoods, mark low-density neighborhoods as outliers

Principal components analysis (PCA)

- **Goal:** Project data onto a lower n-dimensional subspace, such that each principal component explains the most variation possible and is perpendicular to all other principal components
- **Algorithm:**
  1. Standardize the data
  2. Calculate the covariance matrix of the data (m x m matrix, where m is the number of features in the dataset)
  3. Calculate the **eigenvectors** of the covariance matrix – these are the directions of the axes with the most variance
  4. The associated eigenvalues give the variance explained by each principal component

Uses for PCA

- Summarize high-dimensional data in a unidimensional vector for ranking or other unidimensional transformations
- Project high-dimensional data into a low-dimensional subspace (e.g. two dimensions) for visualization
- Dimensionality reduction for down-the-line supervised learning
  - Can help prevent overfitting
  - Reduces computational cost