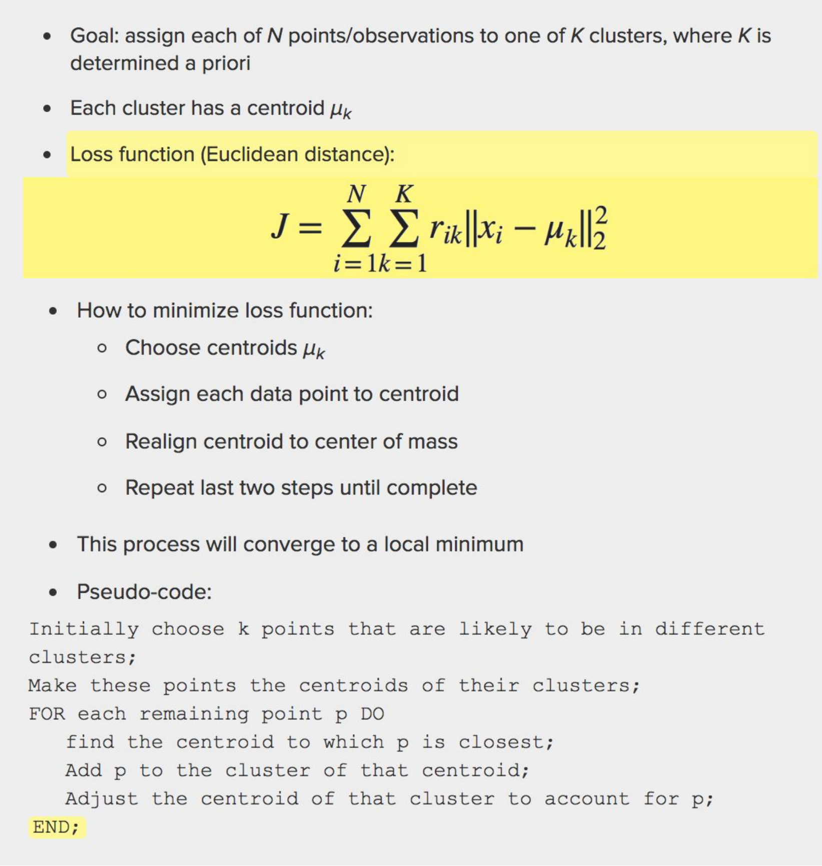
Week 9

Agenda

1. Wrap up supervised learning
2. K-means review
3. K-means notebook

K-means



- Goal: assign each of *N* points/observations to one of *K* clusters, where *K* is determined a priori
- Each cluster has a centroid μ_k
- Loss function (Euclidean distance):

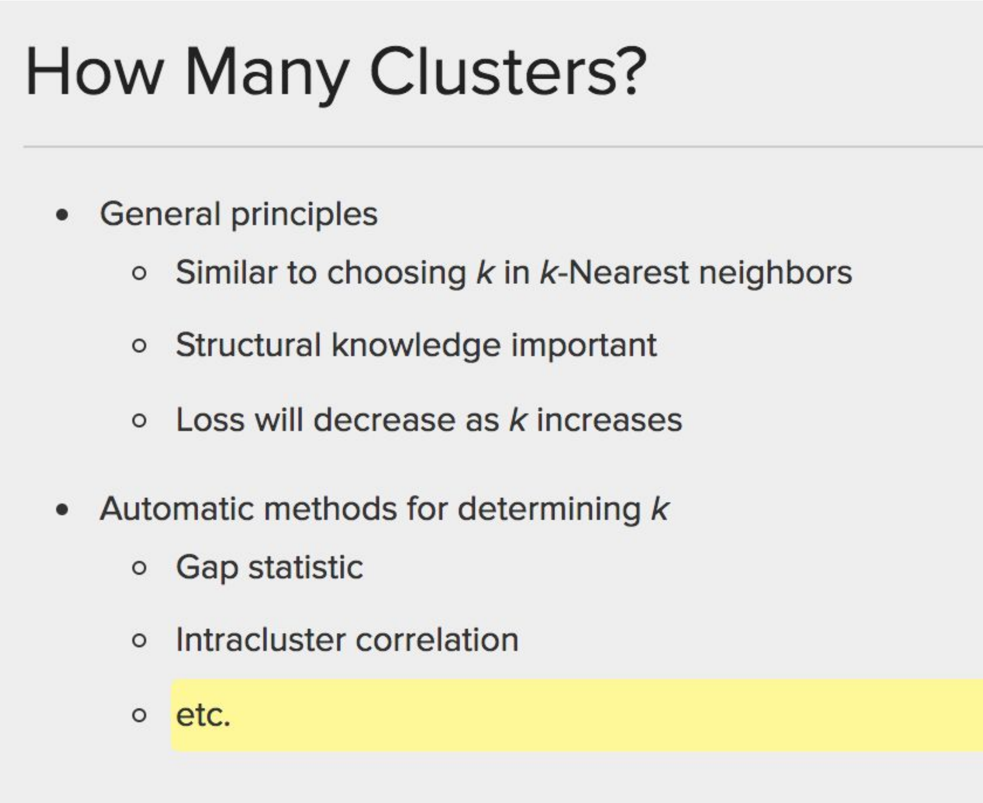
$$J = \sum_{i=1}^{N} \sum_{k=1}^{K} r_{ik} \|x_i - \mu_k\|_2^2$$

- How to minimize loss function:
  - Choose centroids μ_k
  - Assign each data point to centroid
  - Realign centroid to center of mass
  - Repeat last two steps until complete
- This process will converge to a local minimum
- Pseudo-code:

```
Initially choose k points that are likely to be in different
clusters;
Make these points the centroids of their clusters;
FOR each remaining point p DO
   find the centroid to which p is closest;
   Add p to the cluster of that centroid;
   Adjust the centroid of that cluster to account for p;
END;
```

1. Describe the training algorithm.
2. What is the training complexity? Prediction?
3. Can K-means be trained online?
4. What might you use K-means for?

K-means

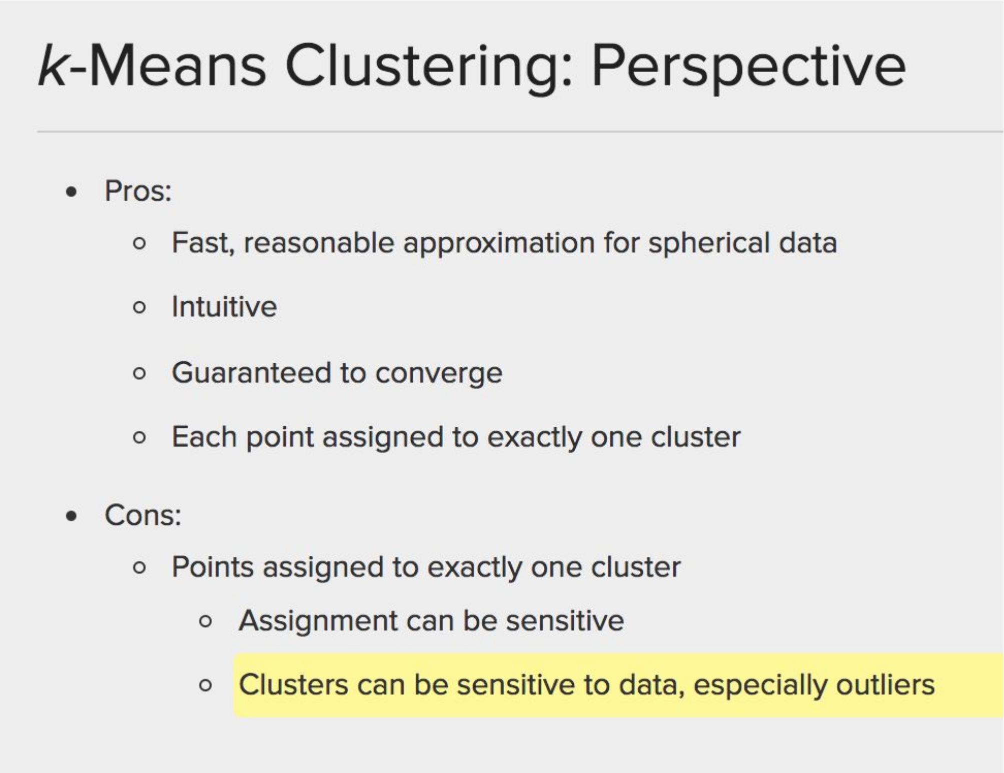


How Many Clusters?

- General principles
  - Similar to choosing *k* in *k*-Nearest neighbors
  - Structural knowledge important
  - Loss will decrease as *k* increases
- Automatic methods for determining *k*
  - Gap statistic
  - Intracluster correlation
  - etc.

1. What might make for a ‘good’ number of clusters?

K-means



k-Means Clustering: Perspective

- Pros:
  - Fast, reasonable approximation for spherical data
  - Intuitive
  - Guaranteed to converge
  - Each point assigned to exactly one cluster
- Cons:
  - Points assigned to exactly one cluster
    - Assignment can be sensitive
    - Clusters can be sensitive to data, especially outliers

1. Review.

Final Thoughts?