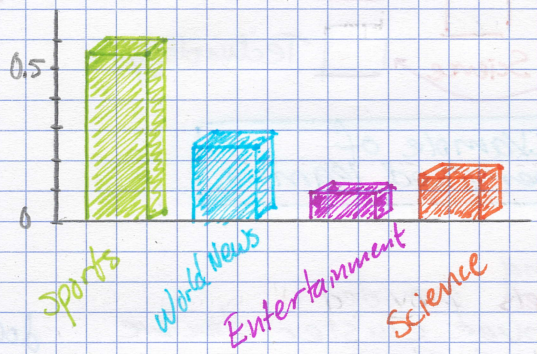
CLUSTERING WITH K-MEANS

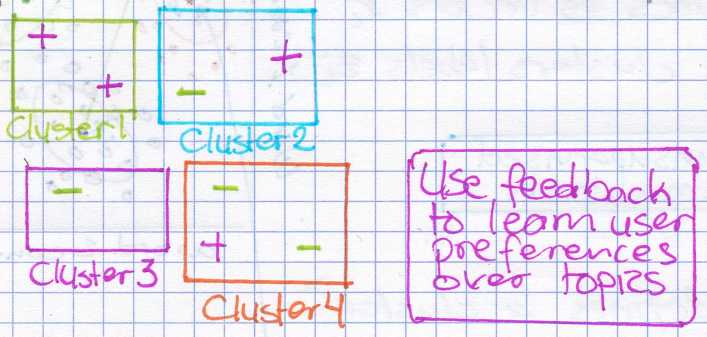
THE GOAL OF CLUSTERING

Goal: Structure documents by topic
Discover groups (clusters) of related articles

Why might clustering be useful



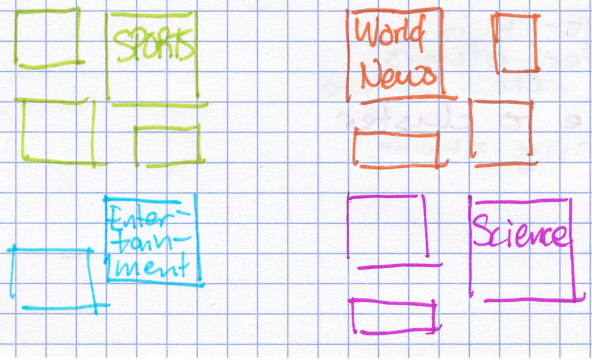
Learn user preferences
Set of clustered documents read by user



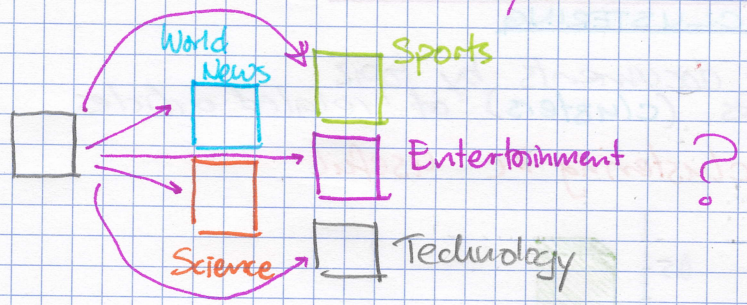
An unsupervised task

Clustering: An unsupervised task

What if some of the labels are known?
Training set of labeled docs



Multiclass classification problem

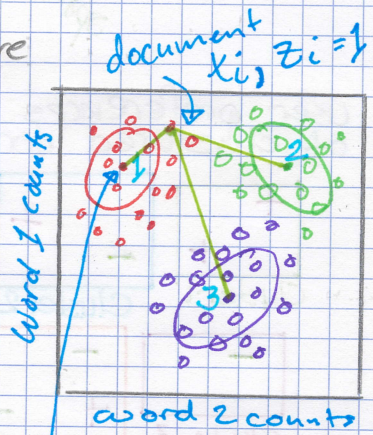


Example of supervised learning

Clustering
No labels provided
... uncover cluster structure from input alone

Input: docs as vectors x_i
Output: clusters labels z_i

An unsupervised learning task



What defines a cluster?

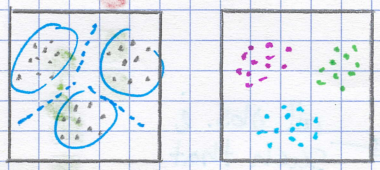
Cluster defined by
center & shape/spread

Assign observation x_i (doc) to cluster k (topic label) if
- Score under cluster k is higher than under others
- For simplicity, often define score as distance to cluster center (ignoring shape)

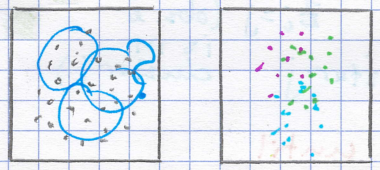
Center "centroid"

Hope for unsupervised learning, and some challenge cases

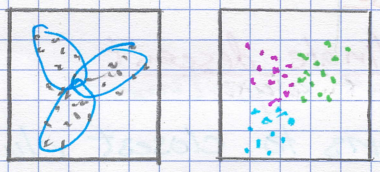
Easy



Impossible



In between



The k-means algorithm

k-means: A clustering algorithm

k-means

Assume
- Score = distance to cluster center (smaller better)

k-means algorithm

0. Initialize cluster centers $\mu_1, \mu_2, \ldots, \mu_K$
1. Assign observations to closest cluster center

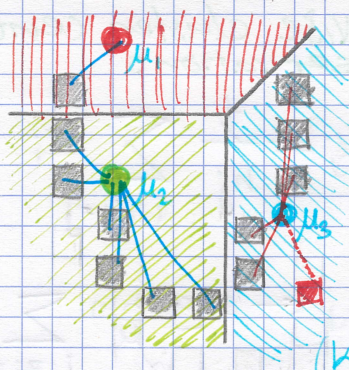
$$z_i \leftarrow \arg\min_j \|\mu_j - x_i\|_2^2$$

z_i: Inferred label for obs i, whereas supervised learning has given label y_i

j: j^{th} cluster center (varying)

i: i^{th} obs (fixed)

return index j of the cluster whose center is closest to obs x_i (whereas $\min$ returning minimum value of $\|\cdot\|_2^2$)



Voronoi tesselation (for visualization only... you don't need to compute this!)

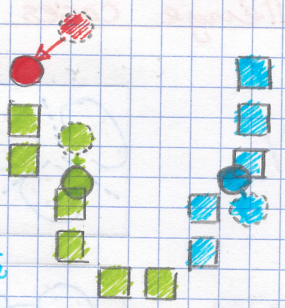
($K = 3$ clusters)

2. Revise cluster centers as mean of assigned observations

$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=j} x_i$$

$i:z_i=j$ ← all obs. i such that $z_i=j$ (obs. i is in cluster j)

n_j ← # of obs. in cluster j



3. Repeat 1+2 until convergence

K-means as coordinate descent

A coordinate descent algorithm

1. Assign observations to closest cluster center

$$z_i \leftarrow \arg\min_j \|\mu_j - x_i\|_2^2$$

2. Revise cluster centers as mean of assigned observations

$$\mu_j = \frac{1}{n_j} \sum_{i:z_i=j} x_i$$

equivalent to

$$\mu_j \leftarrow \arg\min_{\mu} \sum_{i:z_i=j} \|\mu - x_i\|_2^2$$

Alternating minimization
1. (z given μ) and 2. (μ given z)
= coordinate descent

Convergence of k-means
Converges to:

- ~~Global optimum~~
- Local optimum
- ~~neither~~

Smart initialization via k-means++

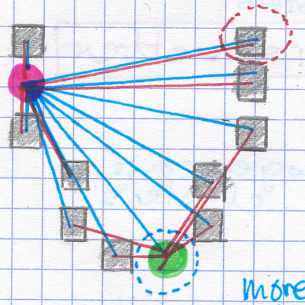
k-means++ overview

Initialization of k-means algorithm is critical to quality of local optima found

Smart initialization:

1. Choose first cluster center uniformly at random from data points
2. For each obs x, compute distance $d(x)$ to nearest cluster center
3. Choose new cluster center from amongst data points, with probability of x being chosen proportional to $d(x)^2$
4. Repeat Steps 2 and 3 until K centers have been chosen

k-means++ visualized



more likely to select a datapoint as a cluster center if that datapoint is far away ($dist^2$ increases this effect)

k-means++ pros/cons

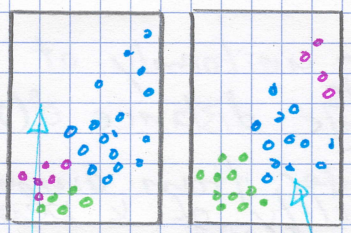
Computationally costly relative to random initialization, but the subsequent k-means often converges more rapidly

Tends to improve quality of local optimum and lower runtime

Assessing the quality and choosing the number of clusters

k-means objective

{Measure of quality of given clustering



k-means is trying to minimize the sum of squared distances:

$$\sum_{j=1}^{k} \sum_{i:z_i=j} \|\mu_j - x_i\|_2^2$$

Sum of squared distances in cluster j

Cluster heterogeneity

$$\min_{\{z_i\},\{\mu_j\}}$$

Sum over all clusters

Lower is better

more heterogeneous (more dissimilar objects within cluster)

more homogeneous tighter clusters

What happens as k increases?

Can refine clusters more and more to the data
→ overfitting!

Extreme case of k-N: (N = # of observations)

- can set each cluster center equal to datapoint
- heterogeneity = 0 (all distances to cluster centers are 0)

Lowest possible cluster heterogeneity decreases with increasing k

How to choose k?