Clustering

- Clustering is an *unsupervised learning* approach: there is no target value (class label) to be predicted, the goal is finding common patterns or grouping similar documents.
- Motivation
  - Grouping search results
  - Creating topic hierarchies
  - Focusing similarity search
- Models/algorithms for clustering
  - Conceptual (model-based) vs. partitioning
  - Exclusive vs. overlapping
  - Deterministic vs. probabilistic
  - Hierarchical vs. flat
  - Incremental vs. batch learning
- Evaluating clustering quality: subjective approaches, objective functions.
- Major approaches
  - Hierarchical Agglomerative Clustering: partitioning, deterministic
  - K-means: flat, deterministic, partitioning or conceptual
  - Expectation Maximization (EM): flat, partitioning, probabilistic
  - Collaborative Filtering: clustering users using terms (preferences)

1 Hierarchical Agglomerative Clustering

- At each step merge the two closest (most similar) clusters.
- Distance/similarity function between instances (e.g. cosine similarity, Euclidean distance).
- Distance/similarity function between clusters (e.g. distance between centers, minimal distance, average distance).
- Criteria for stopping merging:
  - desired number of clusters;
  - distance between the closest clusters is above a threshold.
- Algorithms:
  - *Nearest neighbor (single-linkage)* agglomerative clustering: cluster distance = minimal distance between elements. Merging stops when distance > threshold. In fact, this is an algorithm for generating a *minimal spanning tree.*
  - *Farthest neighbor (complete-linkage)* agglomerative clustering: cluster distance = maximal distance between elements. Merging stops when distance > threshold. The algorithm computes the *complete* subgraph for every cluster.
- Visualization: dendrogram
- Problems: greedy algorithm (local minimum), once created a subtree cannot be restructured.

2 k-means

- Iterative distance-based clustering.
- Used by statisticians for decades.
- Similarly to Cluster/2 uses k seeds (predefined k), but is based on a distance measure:
  1. Select k instances (cluster centers) from the sample (usually at random).
  2. Assign instances to clusters according to their distance to the cluster centers.
  3. Find new cluster centers and go to step 2 until the process converges (i.e. the same instances are assigned to each cluster in two consecutive passes).
- The clustering depends greatly on the initial choice of cluster centers – the algorithm may fall in a local minimum.
- Example of bad chioce of cluster centers: four instances at the vertices of a rectangle, two initial cluster centers – midpoints of the long sides of the rectangle. This is a stable configuration, however not a good clustering.
- Solution to the local minimum problem: restart the algorithm with another set of cluster centers.
- Hierarchical k-means: apply $k = 2$ recursively to the resulting clusters.

3 Probabilty-based clustering

Why probabilities?

- Restricted amount of evidence implies probabilistic reasoning.
- From a probabilistic perspective, we want to find the most likely clusters given the data.
- An instance only has certain probability of belonging to a particular cluster.

4 Probabilty-based clustering – mixture models

- For a single attribute: three parameters - mean, standard deviation and sampling probability.
- Each cluster A is defined by a mean (μ_A) and a standard deviation (σ_A).
- Samples are taken from each cluster A with a specified probability of sampling $P(A)$.
- Finite mixture problem: given a dataset, find the mean, standard deviation and the probability of sampling for each cluster.
- If we know the classification of each instance, then:
  - mean (average), $\mu = \frac{1}{n}\sum_1^n x_i$;
  - standard deviation, $\sigma^2 = \frac{1}{n-1}\sum_1^n (x_i - \mu)^2$;
  - probability of sampling for class A, $P(A)$ = proportion of instances in it.
- If we know the three parameters, the probability that an instance x belongs to cluster A is:

  $P(A|x) = \frac{P(x|A)P(A)}{P(x)}$,

  where $P(x|A)$ is the density function for A, $f(x; \mu_A, \sigma_A) = \frac{1}{\sqrt{2}\pi\sigma_A}\, e^{\frac{-(x-\mu_A)^2}{2\sigma_A^2}}$.

  $P(x)$ is not necessary as we calculate the numerators for all clusters and normalize them by dividing by their sum.

  $\Rightarrow$ In fact, this is exactly the Naive Bayes approach.
- For more attributes: naive Bayes assumption – independence between attributes. The joint probabilities of an instance are calculated as a product of the probabilities of all attributes.

5 EM (expectation maximization)

- Similarly to k-means, first select the cluster parameters (μ_A, σ_A and $P(A)$) or guess the classes of the instances, then iterate.
- Adjustment needed: we know cluster probabilities, not actual clusters for each instance. So, we use these probabilities as weights.
- For cluster A:

  $\mu_A = \frac{\sum_1^n w_i x_i}{\sum_1^n w_i}$, where w_i is the probability that x_i belongs to cluster A;

  $\sigma_A^2 = \frac{\sum_1^n w_i (x_i - \mu)^2}{\sum_1^n w_i}$.
- When to stop iteration - maximizing overall likelihood that the data come form the dataset with the given parameters ("goodness" of clustering):

  $Log\text{-}likelihood = \sum_i log(\ \sum_A P(A) P(x_i|A)\ )$

  Stop when the difference between two successive iteration becomes negligible (i.e. there is no improvement of clustering quality).

6 Evaluating quality of clustering

- Distance (similarity) based functions
  - Sum of squared error

$$J = \sum_{A} \sum_{x \in A} ||x - \mu_A||^2$$

  - Optimal clustering minilizes J: *minimal variance* clustering.
- Probability (entropy) based functions
  - Probability of instance $P(x_i) = \sum_A P(A)P(x_i|A)$
  - Probability of sample $x_1, ..., x_n$:

$$\Pi_i^n(\sum_A P(A)P(x_i|A))$$

  - Log-likelihood:

$$\sum_i^n \log(\sum_A P(A)P(x_i|A))$$

- Evaluate clusters with respect to classes using preclassified instances (known classes)
  - Error: proportion of instances with different class and cluster labels.
  - Precision, Recall (n_i instances in class i, n_j instances in cluster j, n_{ij} members of class i in cluster j):

$$R(i,j) = \frac{n_{ij}}{n_i}, \qquad P(i,j) = \frac{n_{ij}}{n_j}$$

  - F–measure

$$F(i,j) = \frac{2R(i,j)P(i,j)}{R(i,j) + P(i,j)}, \qquad F = \sum_i \frac{n_i}{n} \max_j F(i,j)$$

7 Collaborative Filtering (Recommender Systems)

Matrix representation

- Persons (rows)
- Items (columns)
- $M(i, j) = 1$ if person i likes item j; 0 otherwise.

Task: predicting missing values in rows

Clustering approach

- Cluster persons using items as features (e.g. k-means)
- Use the values for the items in each cluster (e.g. centroids)

EM-like approach (symmetric w.r.t. persons and items)

1. Assign random cluster labels to persons and items
2. Take a person and an item at random:
   - Compute the probabilty that the person belongs to the person clusters
   - Compute the probabilty that the item belongs to the item clusters
   - Compute the probabilty that the person likes the item
3. Esimate the maximul likelihood values to the above probabilities
4. If parameter estimation is satisfactory terminate, else go to 2.

8 Using hyperlink structure to compute similarity

Estimate similarity between d_1 and d_2 using:

- Length of shortest path between d_1 and d_2
- Number of common ancestors of d_1 and d_2
- Number of common successors of d_1 and d_2
- Vector-space (TFIDF) similarity between d_1 and d_2