Clustering Algorithms

- Partitional algorithms
  - Usually start with a random (partial) partitioning
  - Refine it iteratively
    - *K* means clustering
    - Model based clustering
- Hierarchical algorithms
  - Bottom-up, agglomerative
  - Top-down, divisive

Partitioning Algorithms

- Partitioning method: Construct a partition of *n* documents into a set of *K* clusters
- Given: a set of documents and the number *K*
- Find: a partition of *K* clusters that optimizes the chosen partitioning criterion
  - Globally optimal: exhaustively enumerate all partitions
  - Effective heuristic methods: *K*-means and *K*-medoids algorithms

K-Means

- Assumes documents are real-valued vectors.
- Clusters based on *centroids* (aka the *center of gravity* or mean) of points in a cluster, *c*:

$$\vec{\mu}(c) = \frac{1}{|c|}\sum_{\vec{x}\in c}\vec{x}$$

- Reassignment of instances to clusters is based on distance to the current cluster centroids.
  - (Or one can equivalently phrase it in terms of similarities)

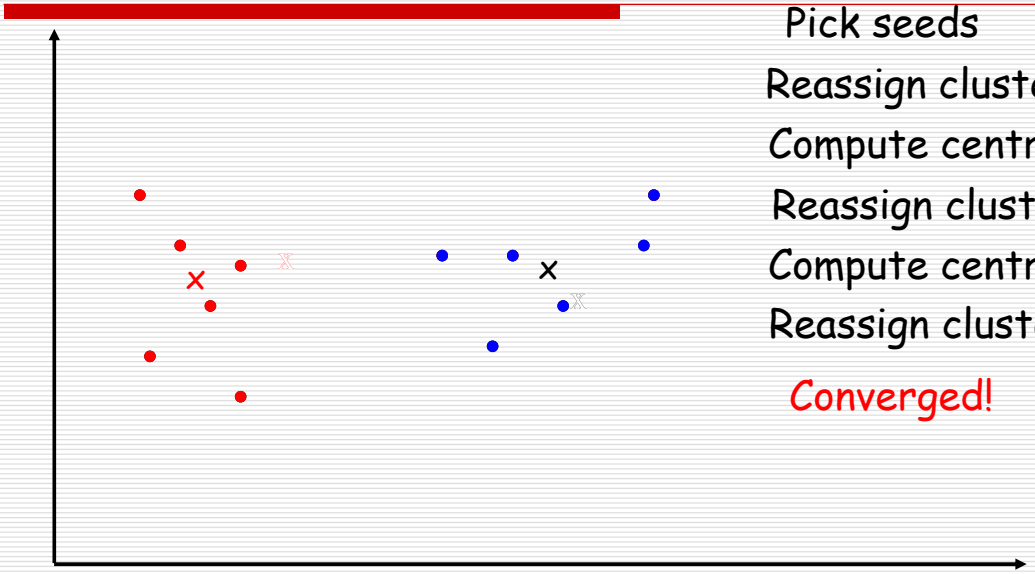
K-Means Algorithm

Select *K* random docs $\{s_1, s_2, \ldots s_K\}$ as seeds.

Until clustering converges or other stopping criterion:

1. For each doc d_i:
   Assign d_i to the cluster c_j such that $dist(x_i, s_j)$ is minimal
2. (*Update the seeds to the centroid of each cluster*)
   For each cluster c_j
   $s_j = \mu(c_j)$

K Means Example (K=2)



Pick seeds
Reassign clusters
Compute centroids
Reassign clusters
Compute centroids
Reassign clusters
Converged!

Termination conditions

- Several possibilities, e.g.,
  - A fixed number of iterations.
  - Based on Loss function (RSS)
  - Doc partition unchanged.
  - Centroid positions don't change.

Does this mean that the docs in a cluster are unchanged?

Convergence

- Why should the *K*-means algorithm ever reach a *fixed point*?
  - A state in which clusters don't change.
- *K*-means is a special case of a general procedure known as the *Expectation Maximization (EM) algorithm*.
  - EM is known to converge.
  - Number of iterations could be large.

Convergence of *K*-Means

- Define goodness measure of cluster *k* as sum of squared distances from cluster centroid:
  - $G_k = \Sigma_i (d_i - c_k)^2$ (sum over all d_i in cluster *k*)
- $G = \Sigma_k G_k$
- Reassignment monotonically decreases *G* since each vector is assigned to the closest centroid.

Convergence of K-Means

- Recomputation monotonically decreases each G_k since (m_k is number of members in cluster k):
  - $\Sigma (d_i - a)^2$ reaches minimum for:
  - $\Sigma -2(d_i - a) = 0$
  - $\Sigma d_i = \Sigma a$
  - $m_K a = \Sigma d_i$
  - $a = (1/ m_k) \Sigma d_i = c_k$
- K-means typically converges quickly

Time Complexity

- Computing distance between two docs is $O(m)$ where m is the dimensionality of the vectors.
- Reassigning clusters: $O(Kn)$ distance computations, or $O(Knm)$.
- Computing centroids: Each doc gets added once to some centroid: $O(nm)$.
- Assume these two steps are each done once for I iterations: $O(IKnm)$.

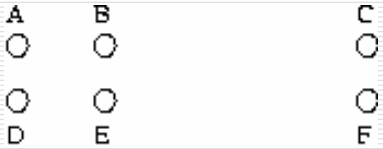
Time Complexity

- So, k-means is linear in all relevant factors (iterations, number of clusters, number of documents, and dimensionality of the space)
- But M>100.000 !!!
- Docs are sparse but centroids tend to be dense -> distance computation is time consuming
- Effective heuristics can be defined for making centroid-doc distance computation as efficient as doc-doc distance computation
- K-medoids is a variant of k-means that compute medoids (the docs closest to the centroid) instead of centroids as cluster centers.

Seed Choice

- Results can vary based on random seed selection.
- Some seeds can result in poor convergence rate, or convergence to sub-optimal clusterings.
  - Select good seeds using a heuristic (e.g., doc least similar to any existing mean)
  - Try out multiple starting points
  - Initialize with the results of another method.

Example showing sensitivity to seeds



In the above, if you start with B and E as centroids you converge to {A,B,C} and {D,E,F}
If you start with D and F you converge to {A,B,D,E} {C,F}

How Many Clusters?

- Number of clusters *K* is given
  - Partition *n* docs into predetermined number of clusters
- Finding the "right" number of clusters is part of the problem
  - Given docs, partition into an "appropriate" number of subsets.
  - E.g., for query results - ideal value of *K* not known up front - though UI may impose limits.

K not specified in advance

- Say, the results of a query.
- Solve an optimization problem
  - $MIN_K\ L(K) + \lambda\ q(K)$ which penalizes having lots of clusters
  - application dependent, e.g., compressed summary of search results list.
- Tradeoff between having more clusters (better focus within each cluster) and having too many clusters

Model-based Clustering

- A different way of posing the clustering problem is to formalize the clustering as a parameterized model Θ and then search the parameters that maximize the likelihood of the data

$$MAX\ L(D|\Theta)$$

- Where $L(D|\Theta) = \log \prod_n P(d_n|\Theta) = \sum_n \log P(d_n)$
- This can be done by an EM (Expectation Maximization procedure)
- K-means can be seen as an instance of EM when the model is a mixture of multivariate Gaussians

Model-based Clustering

- Instead of a Gaussian mixture, we focus on mixture of multivariate binomials, i.e. $P(d|\omega_k,\Theta) = \prod_m P(X_m=I(w_m \in d)|\omega_k)$

- The mixture model $P(d|\Theta) = \sum_k \gamma_k \prod_m P(X_m=I(w_m \in d)|\omega_k)$

- N.B. K-means performs an hard assignment while the binomial EM clustering performs a soft assignment

EM

- Maximization Step: computes the conditional parameters $q_{mk} = P(X_m = 1 \mid \omega_k)$ and the priors γ_k

$$q_{mk} = \frac{\sum_{n=1}^{N} r_{nk} I(\omega_m \in d_n)}{\sum_{n=1}^{N} r_{nk}} \qquad \gamma_k = \frac{\sum_{n=1}^{N} r_{nk}}{N}$$

- Expectation Step: computes the soft assignment of documents to clusters given the current parameters

$$r'_{nk} = \gamma_k \left(\prod_{w_m \in d_n} q_{mk}\right)\left(\prod_{w_m \notin d_n} (1 - q_{mk})\right)$$

$$r_{nk} = \frac{r'_{nk}}{\sum_k r'_{nk}}$$

Considerations

- Finding good seeds is even more critical for EM than for k-means (EM is prone to get stuck in local optima)
- Therefore (as in k-means) an initial assignment is often computed by another algorithm
- If the model of the data is correct EM algorithm finds the correct structure
- Hardly a document collection can be considered generated by a simple mixture model
- At least, model based clustering allows for analysis and adaptations