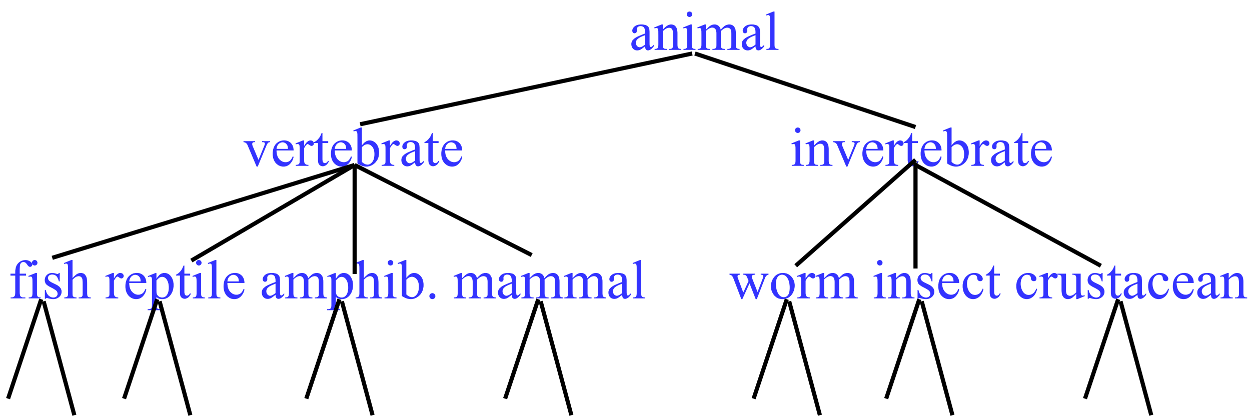
# Text Clustering

# Clustering

- Partition unlabeled examples into disjoint subsets of *clusters*, such that:
  - Examples within a cluster are very similar
  - Examples in different clusters are very different
- Discover new categories in an *unsupervised* manner (no sample category labels provided).

# Hierarchical Clustering

- Build a tree-based hierarchical taxonomy (*dendrogram*) from a set of unlabeled examples.




- Recursive application of a standard clustering algorithm can produce a hierarchical clustering.

# Hierarchical Agglomerative Clustering (HAC)

- Assumes a *similarity function* for determining the similarity of two instances.
- Starts with all instances in a separate cluster and then repeatedly joins the two clusters that are most similar until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

# HAC Algorithm

Start with all instances in their own cluster.
Until there is only one cluster:
    Among the current clusters, determine the two clusters, $c_i$ and $c_j$, that are most similar.
    Replace $c_i$ and $c_j$ with a single cluster $c_i \cup c_j$

# Cluster Similarity

- Assume a similarity function that determines the similarity of two instances: $sim(x,y)$.
  – Cosine similarity of document vectors.
- How to compute similarity of two clusters each possibly containing multiple instances?
  – Single Link: Similarity of two most similar members.
  – Complete Link: Similarity of two least similar members.
  – Group Average: Average similarity between members.

# Non-Hierarchical Clustering

- Typically must provide the number of desired clusters, *k*.
- Randomly choose *k* instances as *seeds*, one per cluster.
- Form initial clusters based on these seeds.
- Iterate, repeatedly reallocating instances to different clusters to improve the overall clustering.
- Stop when clustering converges or after a fixed number of iterations.

# K-Means

- Assumes instances are real-valued vectors.
- Clusters based on *centroids*, *center of gravity*, or mean of points in a cluster, $c$:

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

- Reassignment of instances to clusters is based on distance to the current cluster centroids.

# Distance Metrics

- Euclidian distance ($L_2$ norm):

$$L_2(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{m} (x_i - y_i)^2}$$

- $L_1$ norm:

$$L_1(\vec{x}, \vec{y}) = \sum_{i=1}^{m} |x_i - y_i|$$

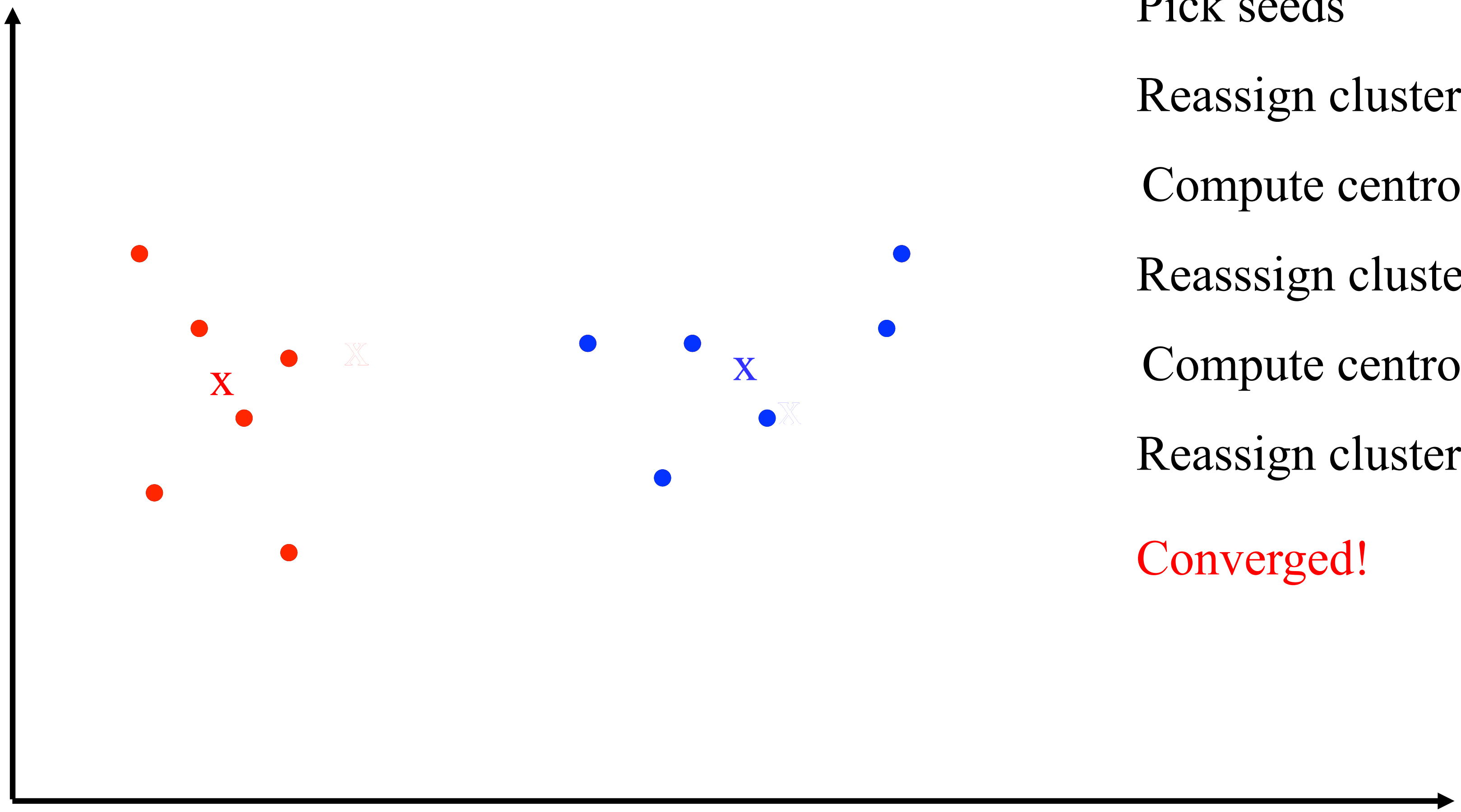
- Cosine Similarity (transform to a distance by subtracting from 1):

$$1 - \frac{\vec{x} \cdot \vec{y}}{|\vec{x}| \cdot |\vec{y}|}$$

# K-Means Algorithm

Let $d$ be the distance measure between instances.
Select $k$ random instances $\{s_1, s_2, \ldots s_k\}$ as seeds.
Until clustering converges or other stopping criterion:
    For each instance $x_i$:
        Assign $x_i$ to the cluster $c_j$ such that $d(x_i, s_j)$ is minimal.
    *(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

Reasssign clusters

Compute centroids

Reassign clusters

Converged!

# Text Clustering

- Applications:

  - During retrieval, add other documents in the same cluster as the initial retrieved documents to improve recall.
  - Clustering of results of retrieval to present more organized results to the user
  - Automated production of hierarchical taxonomies of documents for browsing purposes (à la Yahoo or Seznam).

yes but we talk just about document clustering…

# Text Clustering

the problem of clustering rows in this matrix is that of clustering documents,

whereas that of clustering columns in this matrix is that of clustering words/tokens.

In reality, the two problems are closely related, as good clusters of words may be leveraged in order to find good clusters of documents and vice-versa.

# Text Clustering

- HAC and K-Means can be applied to text in a straightforward way but
- document-term matrix needs to be normalized to prevent from influence of outlying features, so that the L2-norm ||Xi|| of each document is one unit.
- Then there is no difference between the use of the Euclidean distance, cosine similarity, or the dot product similarity, after such a normalization has been performed.
- L2 norm is a standard method to compute the length of a vector in Euclidean space. Given x = [x1x2 … xn]T, L2 norm of x is defined as the square root of the sum of the squares of the values in each dimension.

# Tools

- scikit-learn contains several text clustering tools
- R : tm package  can be used for preprocessing the documents
- R : stats package contains the kmeans and hclust functions by default
- Weka library also contains several Java implementations of clustering algorithms
- MATLAB has functions for k-means and hierarchical clustering. It also automatically computes the dendrogram from a data set.

# Conclusions

- Unsupervised learning induces categories from unlabeled data.
- There are a variety of approaches, including:
  - HAC
  - k-means
  - EM
- Semi-supervised learning uses both labeled and unlabeled data to improve results.