UNIVERSITÀ
DEGLI STUDI
DI MILANO

MEC ITALY

TEXT CLASSIFICATION AND CLUSTERING

ALFIO FERRARA, STEFANO MONTANELLI

Department of Computer Science

Via Comelico 39, 20135 Milano

{alfio.ferrara,stefano.montanelli}@unimi.it
http://islab.di.unimi.it/

Outline

- Text classification problem
- Flat clustering
- Hierarchical clustering
- Other clustering approches

TEXT CLASSIFICATION

TEXT CLASSIFICATION

Text classification is the problem of associating texts (i.e., documents) to classes (or labels denoting classes).

Classes may be:

- A partition of the document space (i.e., disjoint classes)
- Overlapping (i.e., a document may be classified in more than one class)

Methods may be:

- Based on a **training set** of pre-classified documents: **supervised**
- Based on documents only: **unsupervised**

APPLICATIONS

- Support to query answering and retrieval
- Spam detection
- Language detection
- Vertical search engines
- Sentiment detection
- User/product profiling
- ...

Clustering

Clustering is the problem of to group objects (i.e., documents but also terms) in **clusters** that are **coherent internally** but different from each other

Documents within a cluster should be as **similar** as possible and documents in a cluster should be as **dissimilar** as possible from documents in other clusters

Clustering is the most common form of **unsupervised learning** or **unsupervised classification**

CLUSTERING AND DISTANCES

Clustering is based on and depends from the choice of a **distance** measure between objects

Instead of using a distance it is possible to use a **similarity** measure

In case of **distance**, clustering is the problem of **minimizing** the distance among objects in a cluster. In case of similarity, clustering becomes a **maximization** problem

The assignment function to minimize (or maximize) is called **objective function**

Clustering approaches

Clustering structure

- **Flat clustering**: no explicit structure that relates clusters one to each other
- **Hierarchical clustering**: clusters are organized in a hierarchy

Types of assignment

- **Hard clustering**: each object is part of exactly one cluster usually with a boolean measure of assignment
- **Soft clustering**: the assignment of an object is a distribution over the clusters and each object has a fractional membership in several clusters

EVALUATION OF CLUSTERING

Given $\Omega = \{\omega_1, \omega_2, \ldots, \omega_K\}$ as the set of clusters and $\mathbb{C} = \{c_1, c_2, \ldots, c_J\}$ as the set of (expected) classes:

Purity

$$purity(\Omega, \mathbb{C}) = \frac{1}{N} \sum_k \max_j | w_k \cap c_j |$$

Normalized Mutual Information

$$NMI(\Omega, \mathbb{C}) = \frac{I(\Omega; \mathbb{C})}{(H(\Omega) + H(\mathbb{C}))/2}$$

where H denotes the entropy

ESTIMATE NMI

$$I(\Omega;\mathbb{C}) = \sum_k \sum_j P(\omega_k \cap c_j) \log \frac{P(\omega_k \cap c_j)}{P(\omega_k)P(c_j)} =$$

$$= \sum_k \sum_j \frac{|\omega_k \cap c_j|}{N} \log \frac{N \mid \omega_k \cap c_j|}{|\omega_k \mid\mid c_j|}$$

using MLE

$$H(\Omega) = -\sum_k P(\omega_k) \log P(\omega_k) = -\sum_k \frac{|\omega_k|}{N} \log \frac{|\omega_k|}{N}$$

RAND COEFFICIENT

See at clustering as a set of decisions over the N(N-1) / 2 pairs of objects in the dataset

	same cluster	**different cluster**
same class	TP	FN
different class	FP	TN

Rand coefficient

$$Rand = \frac{TP + TN}{TP + FP + FN + TN}$$

F-MEASURE

Precision

$$P = \frac{TP}{TP + FP}$$

Recall

$$R = \frac{TP}{TP + FN}$$

F-measure

$$F_\beta = \frac{(\beta^2 + 1)PR}{\beta^2 P + R}$$

K-MEANS

The objective of **K-means** is to minimize the average squared Euclidean distance of documents from their cluster centers

Cluster center:

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

Residual sum of squares

$$RSS = \sum_{k=1}^{K} \sum_{\vec{x} \in \omega_k} |\vec{x} - \vec{\mu}(\omega_k)|^2$$

K-MEANS STEPS

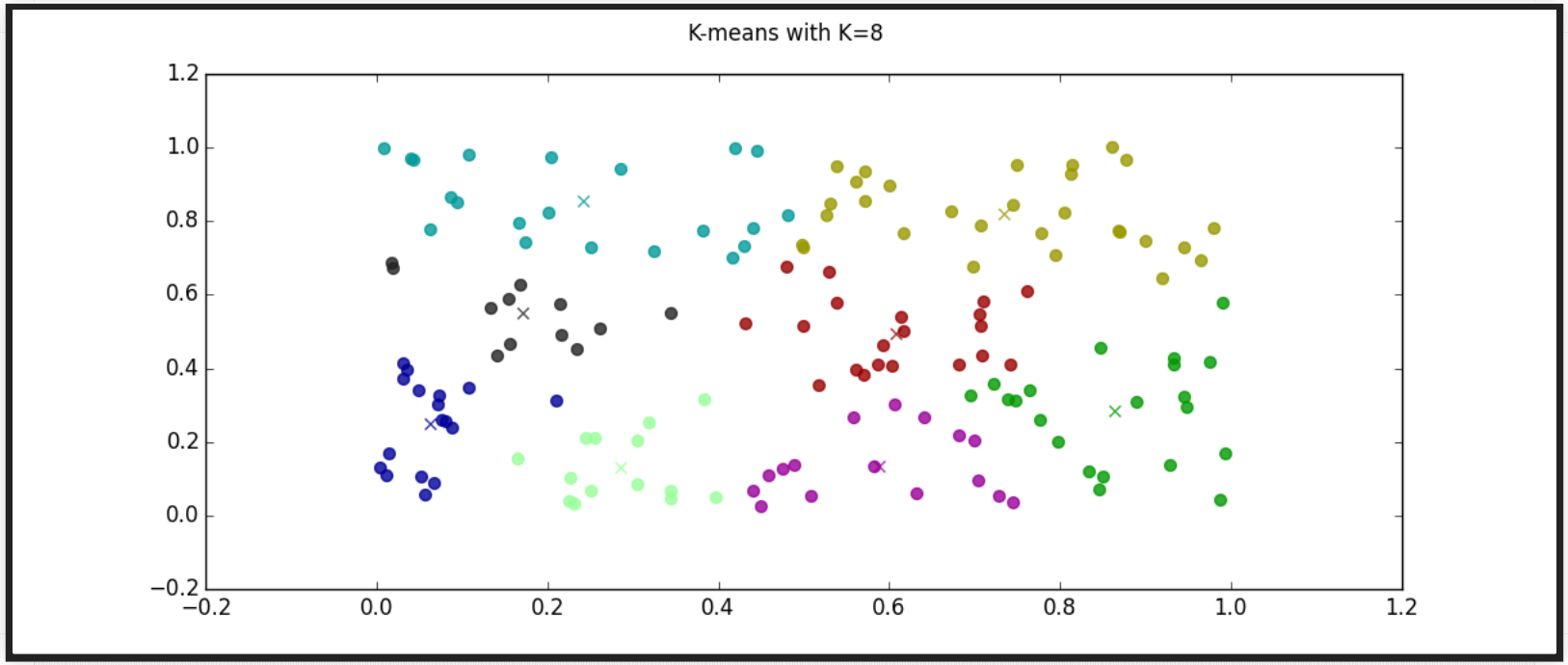
- Randomly select and initial set of **K** objects as centers, called **seeds**
- Each seed represent a cluster
- **Assign** documents to the cluster with the closest centroid
- **Recompute** centroids on the basis of the current objects in clusters
- Repeat the last two point until termination

Termination may be:

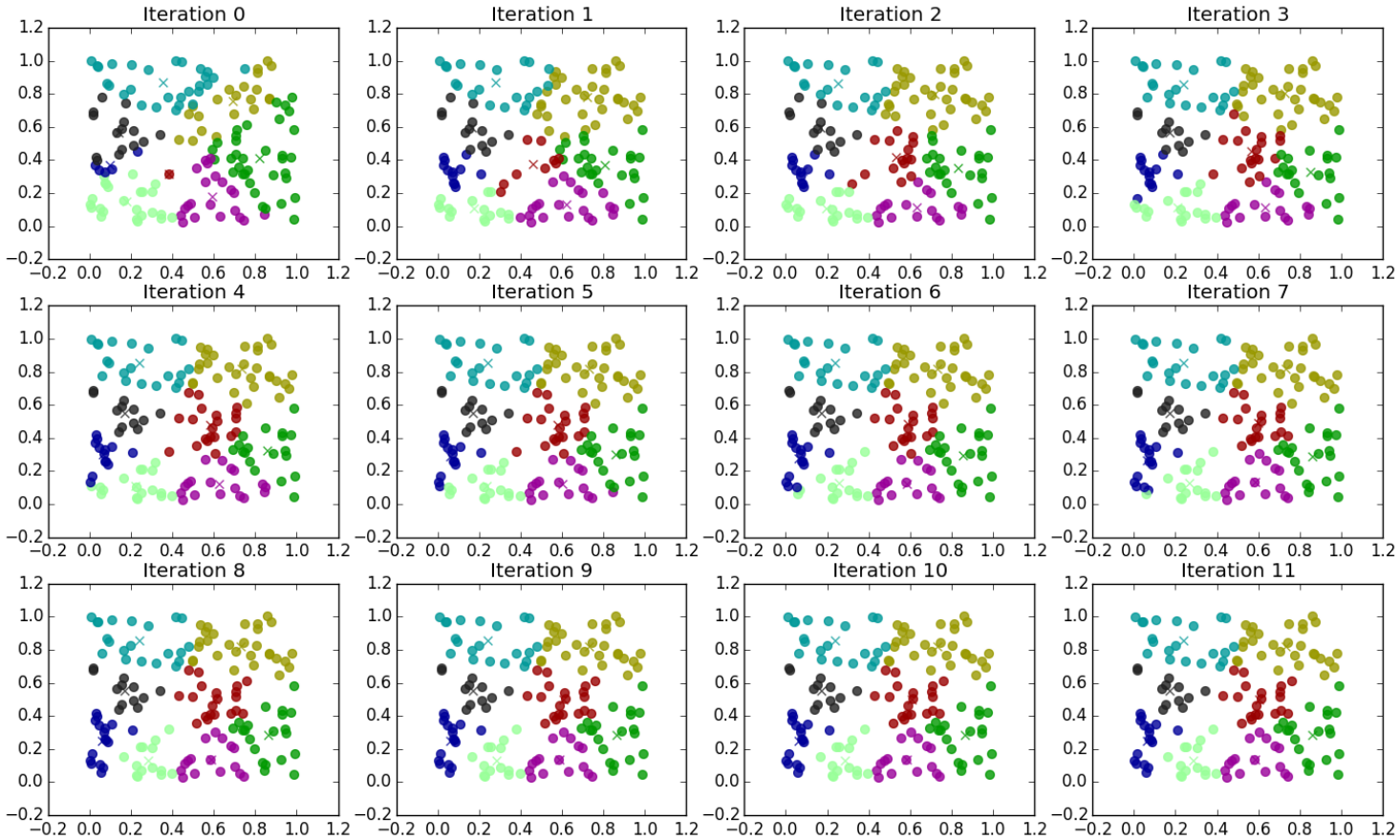
- Fixed number of iterations
- Assignment of documents to clusters does not change (i.e., centroids do not change)
- RSS falls below a threshold

EXAMPLE

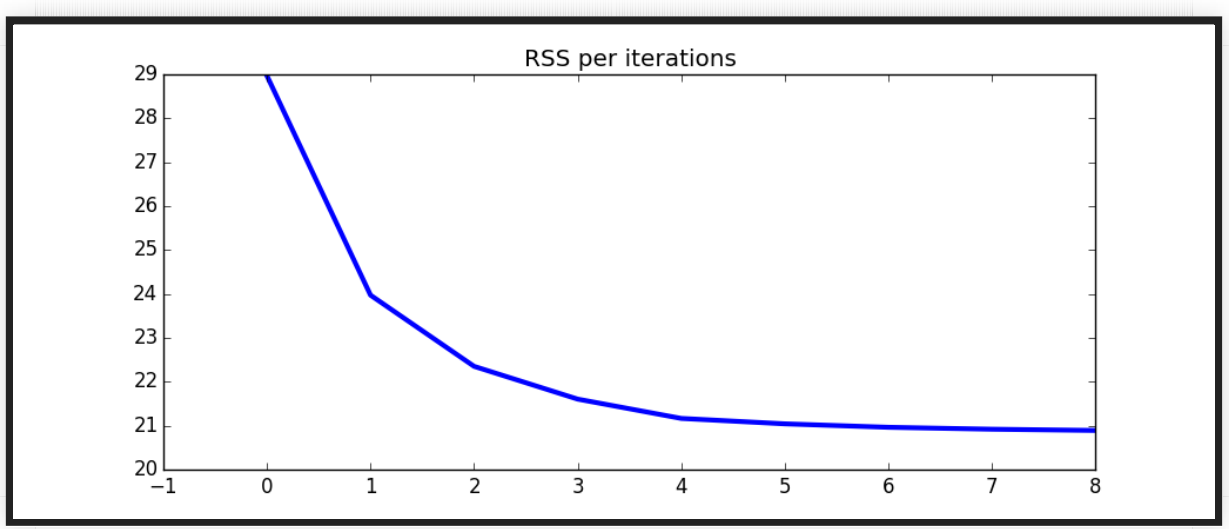
```
points, k, iterations = np.random.rand(150, 2), 8, 12
```



ITERATIONS


Iteration 0
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Iteration 6
Iteration 7
Iteration 8
Iteration 9
Iteration 10
Iteration 11

Rss



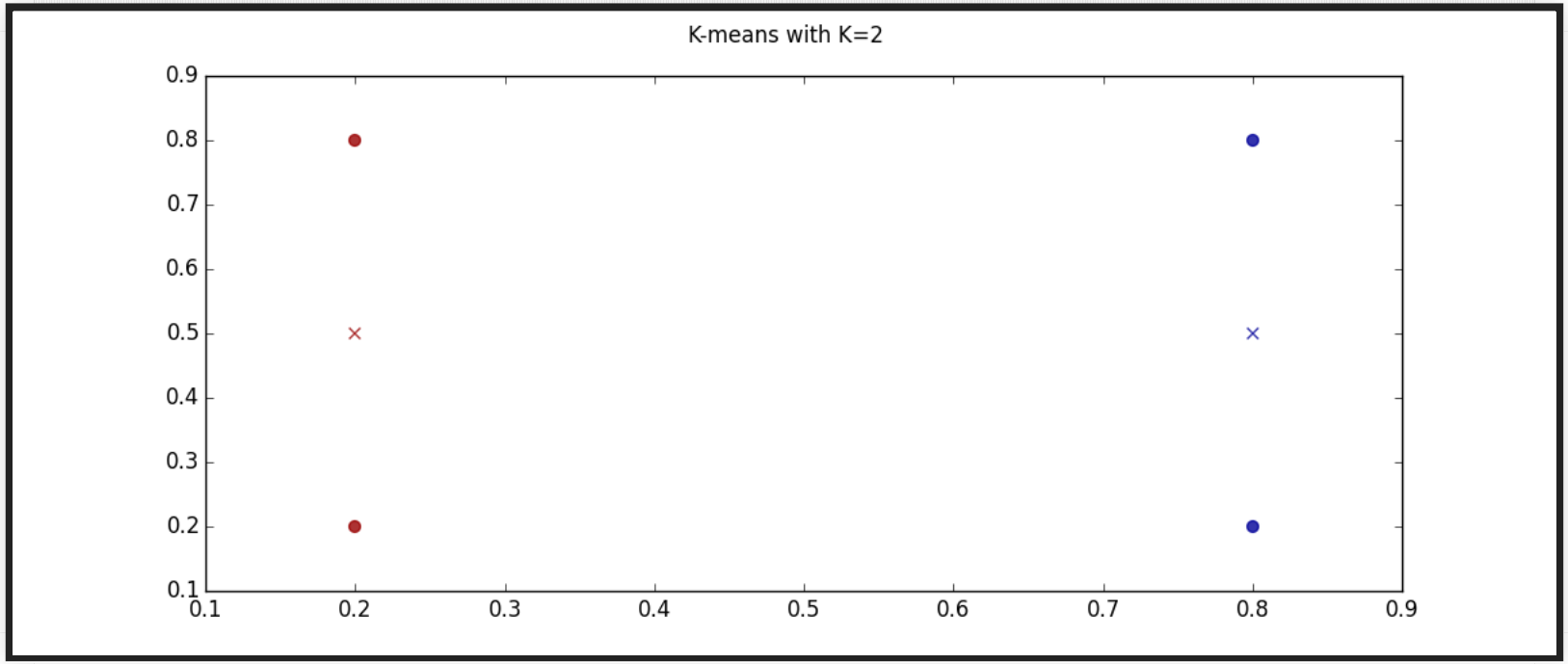
Note that RSS monotonically decreases also as K increases (reaching the minimum with K equal to the number of points)

ESTIMATING K

The best tradeoff between RSS and K can be estimated by:

$$K = argmin_k(min(RSS_k) + \lambda K)$$

PROBLEMS WITH K-MEANS



Model-based clustering

We can **generalize** k-means by interpreting the centroids as a model that generates the data

The idea is that we can pick a centroid and add some noise to generate a document

Cluster shape depends on the kind of distribution of noise

Im model-base clustering, we **assume** that data were generated by a model and we try to observe data in order to find the model

Clusters and **assignment** are then **latent** parameters that we want to estimate

THE PROBLEM

- $\Theta = \{\overrightarrow{\mu_1}, \overrightarrow{\mu_2}, \ldots, \overrightarrow{\mu_K}\}$ is the set of model parameters (the centroids to be found for k-means)
- $L(D \mid \Theta)$ is the log-likelihood of having the data D generated by Θ (this quantity has to be maximized: **objective function**)

$$\Theta = argmax_{\Theta} L(D \mid \Theta) = argmax_{\Theta} \log \prod_{n=1}^{N} P(d_n \mid \Theta) = argmax_{\Theta} \sum_{n=1}^{N} \log P(d_n \mid \Theta)$$

Note that the assignment probability $P(d_n \mid \omega_k; \Theta)$ that can be computed having Θ implies that a document can be assigned to different clusters with different probabilities: **soft clustering**

AFFINITY PROPAGATION CLUSTERING

Input: a similarity matrix among documents

e.g., $s(i,k) = -||\vec{i} - \vec{k}||^2$

Input: special values for $s(k,k)$, where larger values are more likely to be chosen as exemplars for clusters (preferences)

Documents exchange messages that can be combined to decide which points are exemplars and for other points which examplar it belongs to

Messages

Responsibility

$r(i, k)$ sent from i to k denotes how well k is an exemplar for i

Availability

$a(i, k)$ sent from candidate exemplar k to i denotes how appropriate is for i to choose k as exemplar

Procedure

Initially set availabilities to 0 and then recompute iteratively responsibilities and availabilities

Responsibility

$$r(i,k) = s(i,k) - max_{k' s.t. k' \neq k}\{a(i,k') + s(i,k')\}$$

Availability

$$a(i,k) = min\left\{0, r(k,k) + \sum_{i' s.t. i' \notin \{i,k\}} max\{0, r(i',k)\}\right\}$$

Self availability

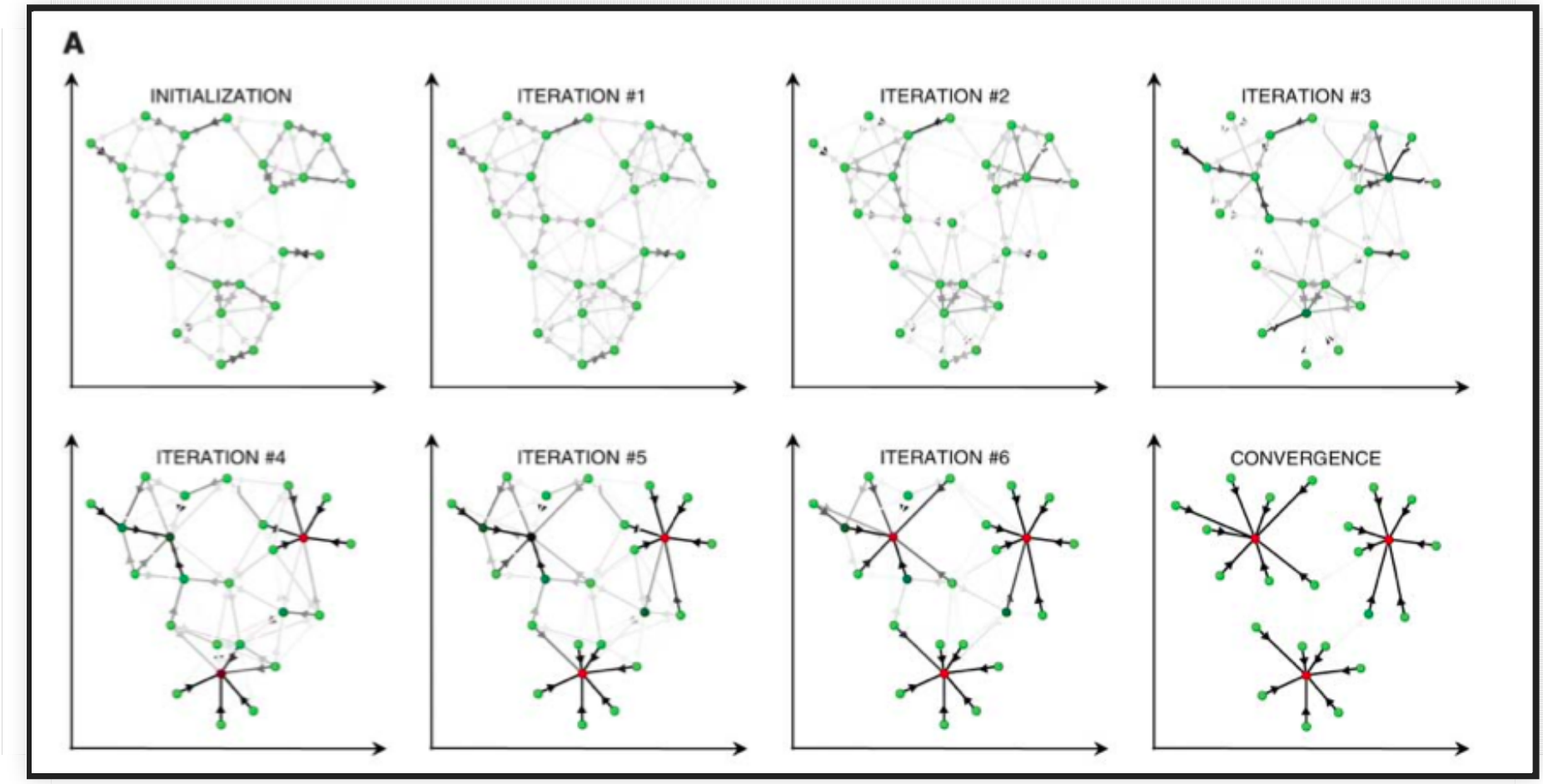
$$a(k,k) = \sum_{i' s.t. i' \neq k} max\{0, r(i',k)\}$$

Use a dumping factor λ. Each message is set to λ times its values from the previous iteration plus $1 - \lambda$ times its prescribed value

CHOICE OF CLUSTERS

At any point, for a point **i** the value of **k** that maximize **a(i, k) + r(i, k)** either identifies **i** as an exemplar if **i = k** or identifies the data point **k** that is the exemplar for **i**

EXAMPLE



Frey, Brendan J., and Delbert Dueck. "Clustering by passing messages between data points." science 315.5814 (2007): 972-976.

HIERARCHICAL CLUSTERING

Hierarchical clustering produces a **hierarchy** of clusters

Does not require to **specify the number of clusters** (but a criterion of optimal cluster selection)

Most hierarchical clustering algorithms are **deterministic**

Complexity is at least **quadratic** in the number of documents

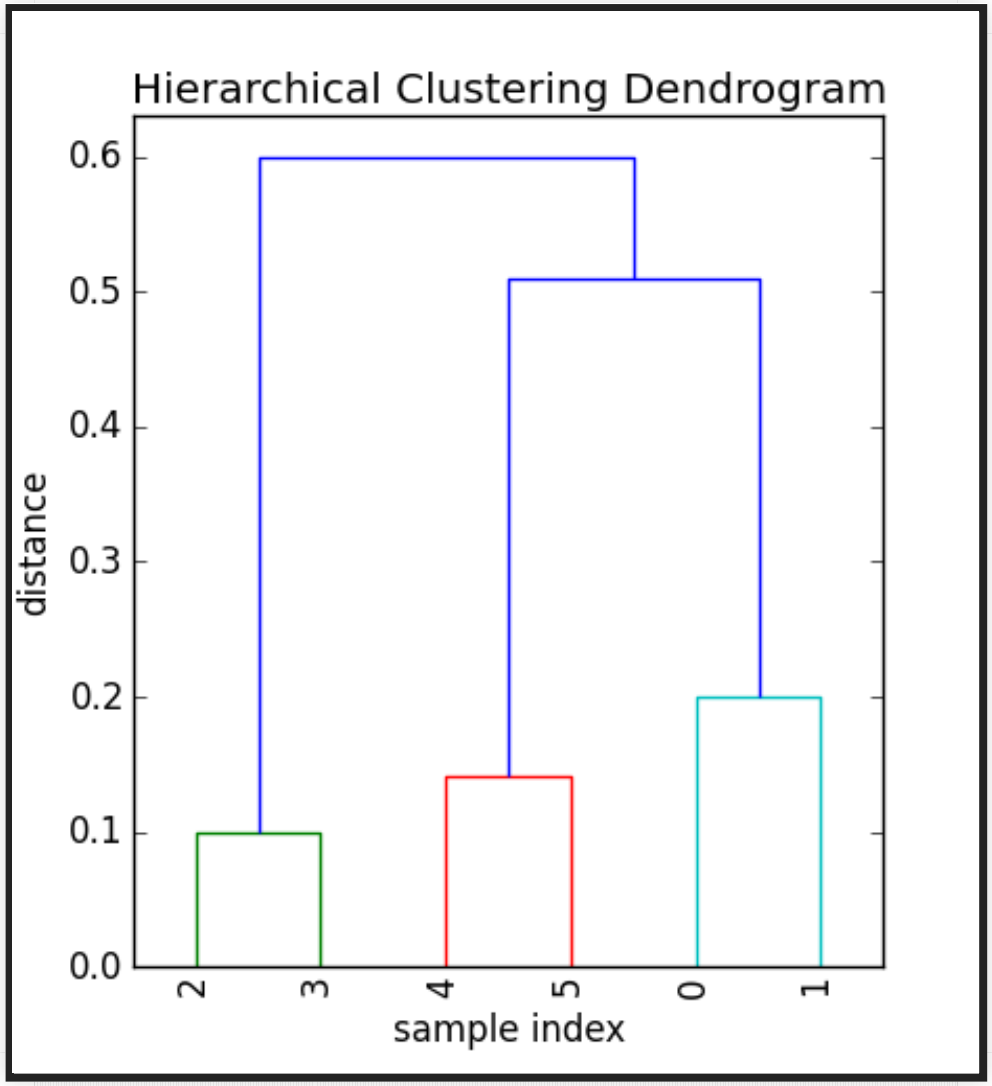
TYPES OF HIERARCHICAL CLUSTERING

Bottom-up clustering or **agglomerative** clustering treat each document as a singleton cluster and successively merge clusters until a single cluster containing all the documents has created

Top-down clustering or **divisive** clustering starts from a single cluster containing all documents and recursively splits clusters until singleton clusters are formed

DENDOGRAM

```
points = np.array([
[0.8, 0.7],
[0.8, 0.9],
[0.2, 0.8],
[0.2, 0.7],
[0.8, 0.1],
[0.7, 0.2],
])
```



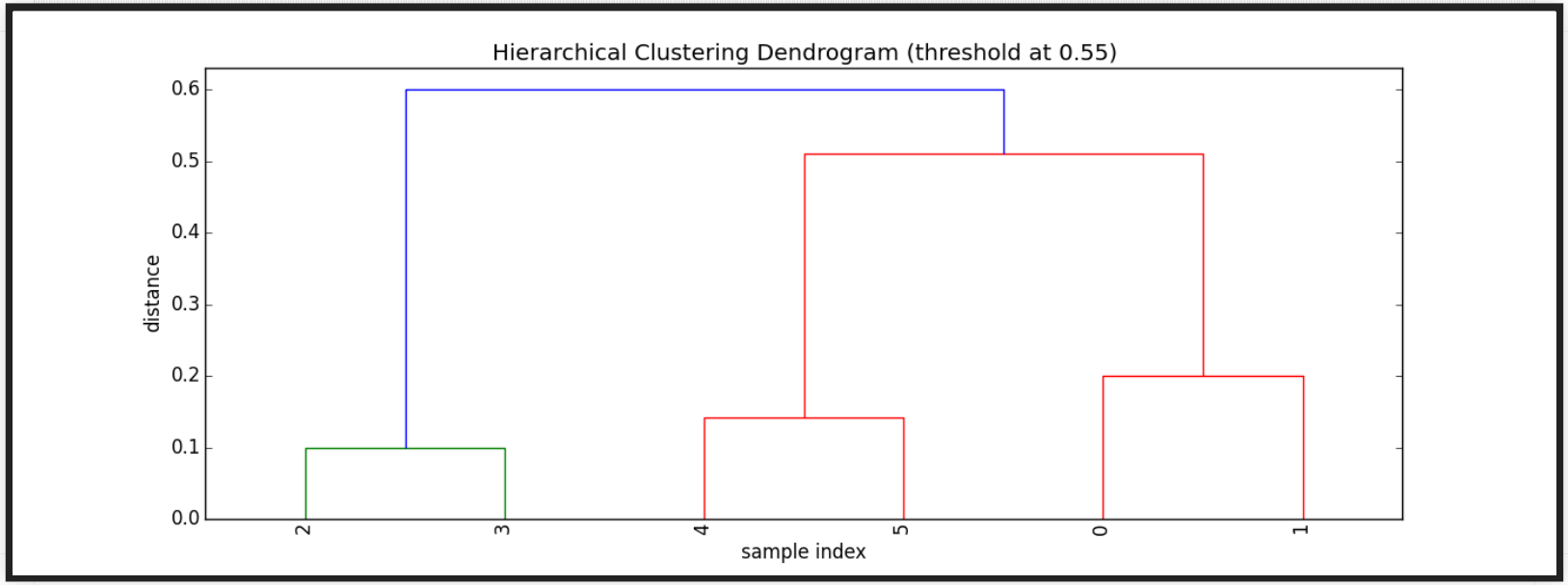
How to select clusters

- Cut the dendogram at a specific level of similarity
- Cut the dendogram where the gap between two successive combination similarities is largest
- Apply

$$K = argmin_{K'}[RSS(K') + \lambda K']$$

  where K' is the cut threshold resulting in K clusters
- Pre-define K and cut the dendogram accordingly

EXAMPLE


Hierarchical Clustering Dendrogram (threshold at 0.55)
distance
0.0
0.1
0.2
0.3
0.4
0.5
0.6
2
3
4
5
0
1
sample index

NAIVE IMPLEMENTATION

- Compute **N x N** similarity matrix **S**
- Execute **N - 1** steps in which the most similar clusters are merged and the corresponding rows and columns are updated with the similarity of the new merged cluster with all the other clusters
- Store history of subsequent aggregations

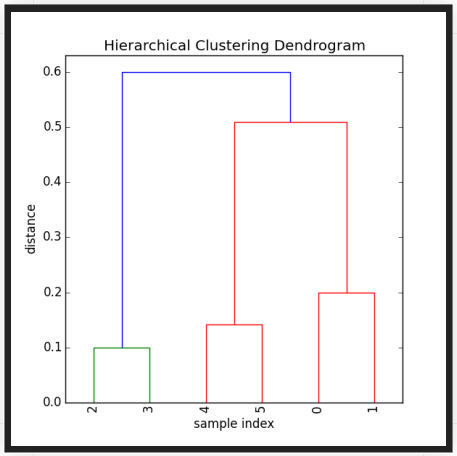
STRATEGIES FOR CLUSTER SIMILARITY

There are different stategies for computing the similarity between clusters for the merging step:

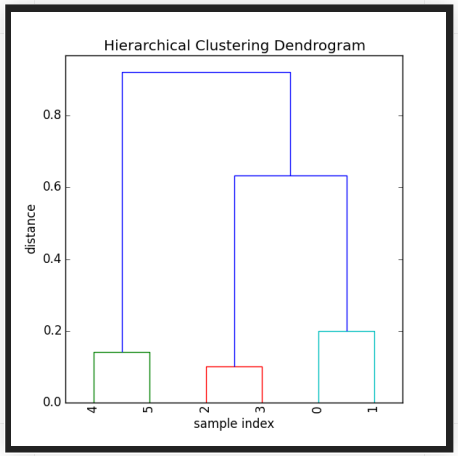
- **Single link**: $d(u, v) = min(u[i], v[j])$
- **Complete link**: $d(u, v) = max(u[i], v[j])$
- **Average link**: $d(u, v) = \sum_{ij} \frac{d(u[i], v[j])}{|u||v|}$
- **Weighted link**: $d(u, v) = (d(s, v) + d(t, v))/2$, where u was formed with cluster s and t and v is a remaining cluster in the forest.
- **Centroid link**: $d(u, v) = || \vec{u} - \vec{v} ||^2$
- **Median link**: same as centroid, but the new centroid of a cluster is the average of the two centroids
- **Ward link**: $d(u, v) = \sqrt{\frac{|v|+|s|}{|v|+|s|+|t|} d(v, s)^2 + \frac{|v|+|t|}{|v|+|s|+|t|} d(v, t)^2 + \frac{|v|}{|v|+|s|+|t|} d(s, t)^2}$

EXAMPLE

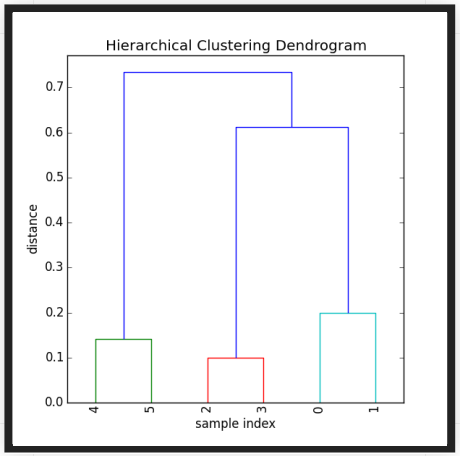
single



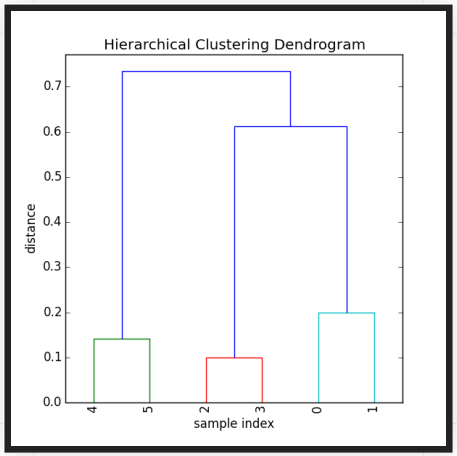
complete



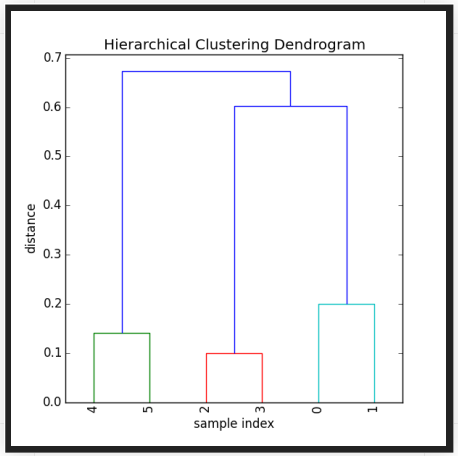
average



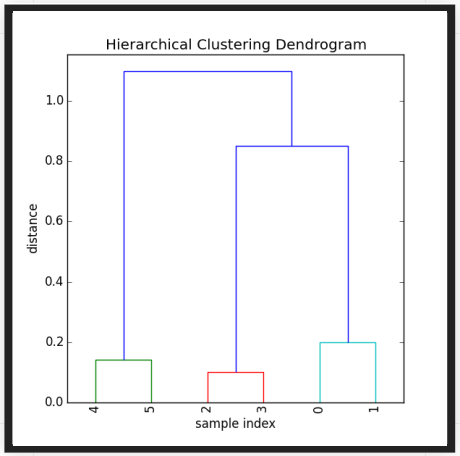
weigthed



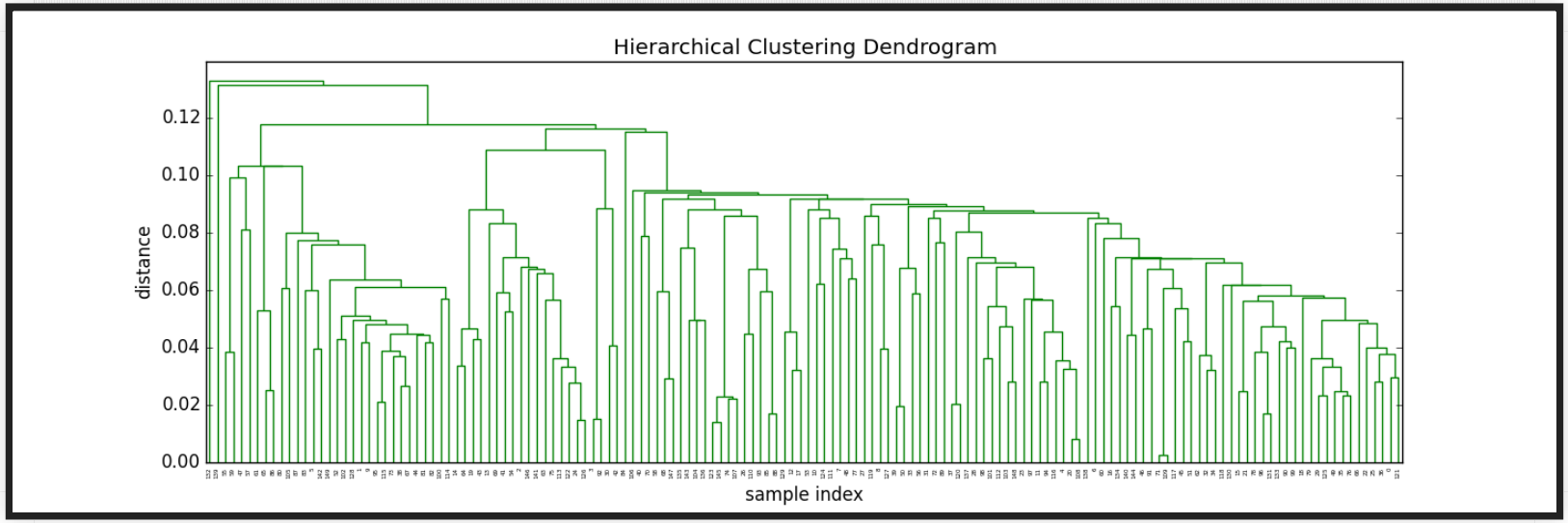
median

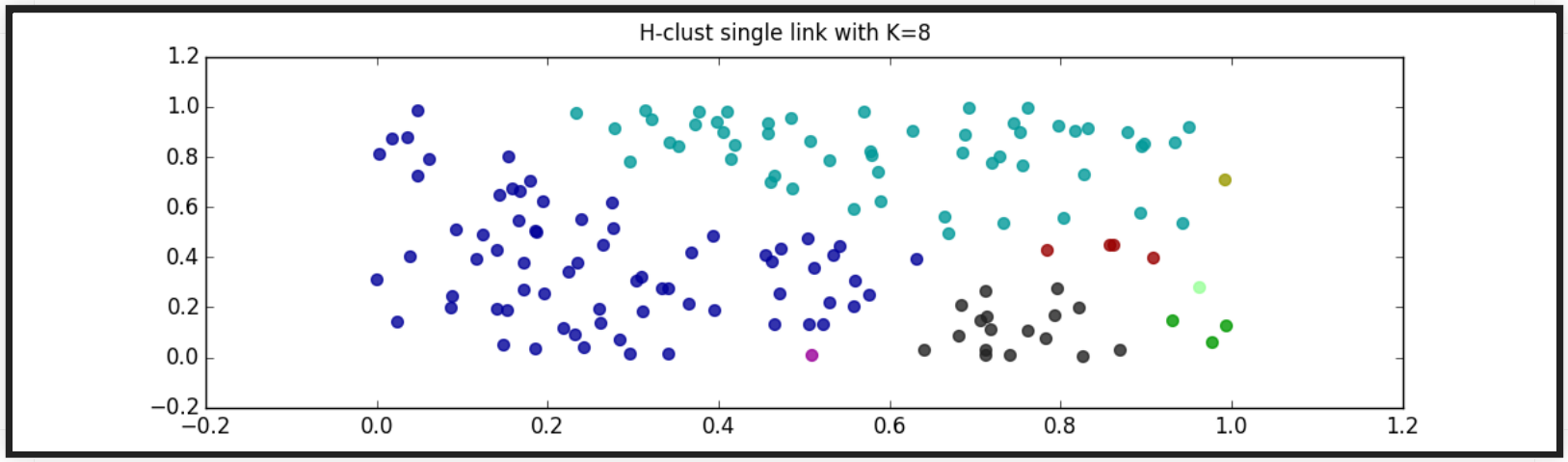


ward

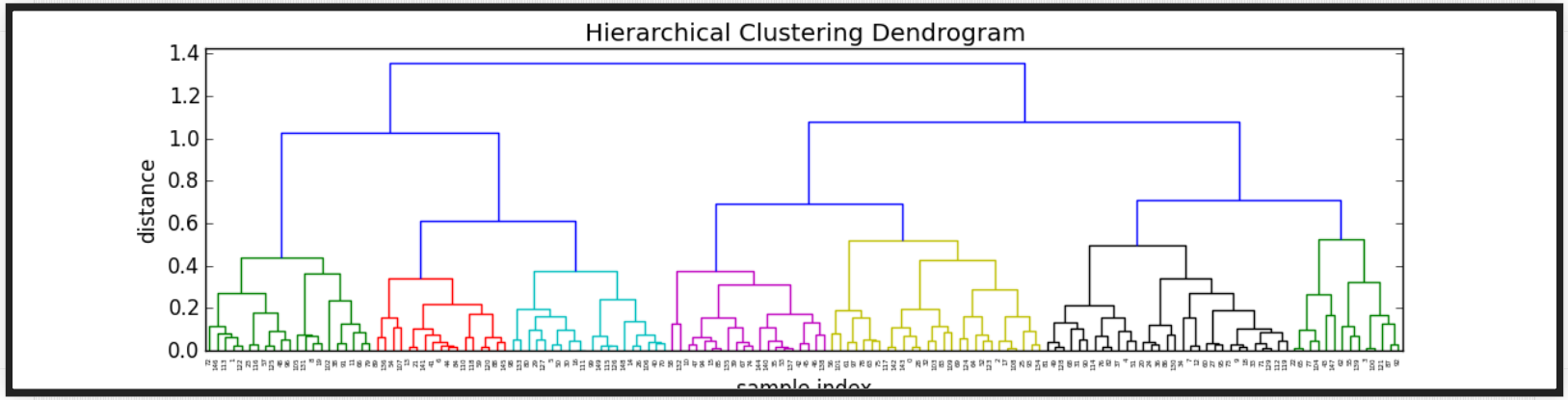


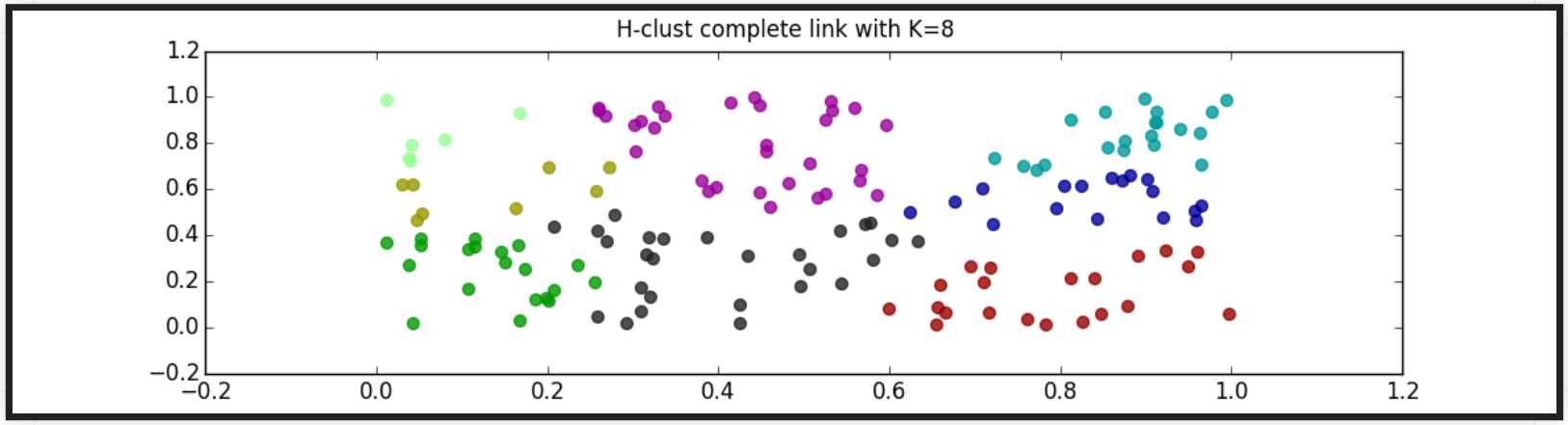
EXAMPLE (SINGLE LINK)





EXAMPLE (COMPLETE LINK)





TOP-DOWN CLUSTERING

Create a **single** cluster grouping all the documents

Apply a **flat hard clustering** solution (e.g., k-means) for splitting the cluster in K partitions

Recursively apply the partition step until all the clusters contain only one document

Keep track of the parent of each cluster obtained by partition

CLUSTER LABELING

A general problem for any clustering approach is to generate a good set of labels for cluster description

- **Differential labeling** : select cluster labels by comparing the distribution of terms in one cluster with that of other clusters. To this end, **specificity** measures, **mutual information**, and/or χ^2 test may be used
- **Internal labeling** : uses information internal to each cluster for labeling (e.g., title of document which is closest to the cluster centroid, list of terms with high weights in the centroid of cluster)

RECAP

Topic	Chapters
Clustering	16, 17