Part II: Web Content Mining
Chapter 3: Clustering

- Learning by Example and Clustering
- Hierarchical Agglomerative Clustering
- K-Means Clustering
- Probability-Based Clustering
- Collaborative Filtering

Learning by Example and Clustering

- Learning by Example
  - Given a set of objects each one labeled with a class (supervised learning)
  - The learning system builds a mapping between objects and classes
  - The mapping can be then used for classifying new (unlabeled) objects
- Clustering
  - Unsupervised learning (objects are not labeled)
  - Goal is finding common patterns, grouping similar objects or creating a hierarchy
- Web content learning
  - Objects are web documents and class labels are topics or user preferences
  - Supervised web learning builds a mapping between documents and topics
  - Clustering groups web documents or organizes them in hierarchies
- Web document clustering
  - Useful in web search for grouping search results into related sets
  - Can improve similarity search by focusing on relevant documents
  - Hierarchical clustering can be used to automatically create topic directories
  - Valuable technique for analyzing the Web

Types of Clustering

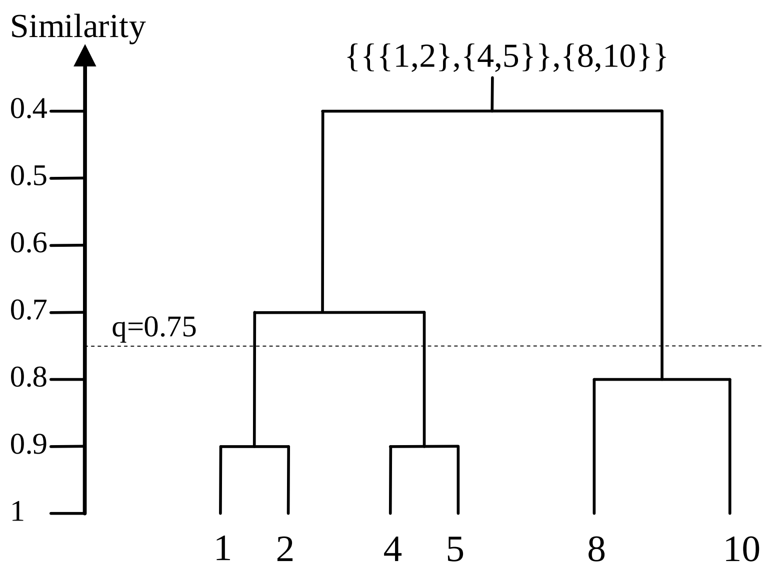
- *Model-based* (conceptual) vs. *partitioning*. Conceptual clustering creates models (explicit representations) of clusters, while partitioning enumerates the members of each cluster.
- *Deterministic* vs. *probabilistic*. Cluster membership may be defined as a boolean value (in deterministic clustering) or as a probability (in probabilistic clustering).
- *Hierarchical* vs. *flat*. Flat clustering splits the set of objects into subsets, while hierarchical clustering creates tree structures of clusters.
- *Incremental* vs. *batch*. Batch algorithms use the complete set of objects to create the clustering, while incremental algorithms take one object at a time and update the current clustering to accommodate it.

Representations for Clustering

- *Attribute-value (feature-value) representation*
  - A number of attributes are identified for the whole population and each object is represented by a set of attribute-value pairs.
  - If the order of attributes is fixed, a vector of values (data points) can be used instead. The *vector space model* is an example of this representation.
  - *Hierarchical agglomerative clustering* and *k-means* clustering use this representation.
- *Generative document modeling*
  - Considers documents as outcomes of random processes and tries to identify the parameters of these processes.
  - *Probabilistic model-based approach*, where each cluster is described by the probability distribution most likely to have generated the documents in it.
  - Documents are represented by terms (as in vector space), however they are considered as *elementary* (*atomic*) *random events*.
  - Does not use similarity measures or distances between documents or clusters
  - *Expectation maximization* (EM) uses this representation

Hierarchical Partitioning

- Produces a nested sequence of partitions of the set of data points,
- Can be displayed as a tree (called *dendrogram*) with a single cluster including all points at the root and singleton clusters (individual points) at the leaves.
- Example of hierarchical partitioning of set of numbers {1, 2, 4, 5, 8, 10}:



The similarity measure used in this example is computed as $(10-d)/10$, where d is the distance between data points or cluster centers.

Approaches to Hierarchical Partitioning

- *Agglomerative*. Starts with the data points and at each step merges the two closest (most similar) points (or clusters at later steps) until a single cluster remains.
- *Divisible*. Starts with the original set of points and at each step splits a cluster until only individual points remain. To implement this approach we need to decide which cluster to split and how to perform the split.
- The agglomerative approach is more popular as it needs only the definition of a distance or similarity function on clusters/points.
- For data points in the Euclidean space the *Euclidean distance* is the best choice.
- For documents represented as TFIDF vectors the preferred measure is the *cosine similarity* defined as follows

$$sim(d_1, d_2) = \frac{d_1 \bullet d_2}{\|d_1\| \|d_2\|}$$

Agglomerative Hierarchical Clustering

There are several versions of this approach depending on how similarity on clusters $sim(S_1, S_2)$ is defined (S_1 and S_2 are sets of documents):

- Similarity between cluster *centroids*, i.e. $sim(S_1, S_2) = sim(c_1, c_2)$, where the centroid c of cluster S is $c = \frac{1}{|S|}\sum_{d \in S} d$
- *Maximum similarity* between documents from each cluster (*nearest neighbor clustering)*
$$sim(S_1, S_2) = \max_{d_1 \in S_1, d_2 \in S_2} sim(d_1, d_2)$$
- *Minimum similarity* between documents from each cluster (*farthest neighbor clustering)*
$$sim(S_1, S_2) = \min_{d_1 \in S_1, d_2 \in S_2} sim(d_1, d_2)$$
- *Average similarity* between documents from each cluster
$$sim(S_1, S_2) = \frac{1}{|S_1||S_2|} \sum_{d_1 \in S_1, d_2 \in S_2} sim(d_1, d_2)$$

Agglomerative Clustering Algorithm

S is the initial set of documents and G is the *clustering tree*. k and q are *control parameters* that stop merging when a desired number of clusters (k) is reached or when the similarity between the clusters to be merged becomes less than a specified threshold (q).

1. $G \leftarrow \{\{d\} \mid d \in S\}$ (initialize G with singleton clusters, each one containing a document from S)
2. If $|G| \leq k$ then exit (stop if the desired number of clusters is reached)
3. Find $S_i, S_j \in G$, such that $(i, j) = \arg\max_{(i,j)} sim(S_i, S_j)$ (find the two closest clusters)
4. If $sim(S_i, S_j) < q$ then exit (stop if the similarity of the closest clusters is less than q)
5. Remove S_i and S_j from G.
6. $G = G \bigcup \{S_i, S_j\}$ (merge S_i and S_j, and add the new cluster to the hierarchy)
7. Go to 2

For n documents both *time* and *space complexity* of the algorithm are $O(n^2)$.

Agglomerative Clustering Example 1

CCSU departments represented as TFIDF vectors with 671 terms
Nearest neighbor algorithm ($k = 0$)

Cluster similarity cut off parameter $q = 0$

Average Intracluster Similarity = **0.4257**

```
1 [0.0224143]
   2 [0.0308927]
      3 [0.0368782]
         4 [0.0556825]
            5 [0.129523]
               Art
               Theatre
            Geography
         6 [0.0858613]
            7 [0.148599]
               Chemistry
               Music
            8 [0.23571]
               Computer
               Political
      9 [0.0937594]
         10 [0.176625]
            Communication
            Economics
         Justice
   11 [0.0554991]
      12 [0.0662345]
         13 [0.0864619]
            14 [0.177997]
               History
               Philosophy
            15 [0.186299]
               English
               Languages
         16 [0.122659]
            Anthropology
            Sociology
      17 [0.0952722]
         18 [0.163493]
            19 [0.245171]
               Biology
               Math
            Psychology
         Physics
```

```
1 []
   2 [0.0554991]
      3 [0.0662345]
         4 [0.0864619]
            5 [0.177997]
               History
               Philosophy
            6 [0.186299]
               English
               Languages
         7 [0.122659]
            Anthropology
            Sociology
      8 [0.0952722]
         9 [0.163493]
            10 [0.245171]
               Biology
               Math
            Psychology
         Physics
   11 [0.0556825]
      12 [0.129523]
         Art
         Theatre
      Geography
   13 [0.0858613]
      14 [0.148599]
         Chemistry
         Music
      15 [0.23571]
         Computer
         Political
   16 [0.0937594]
      17 [0.176625]
         Communication
         Economics
      Justice
```

Cluster similarity cut off parameter $q = 0.04$

Average Intracluster Similarity = **0.4516**

Agglomerative Clustering Example 2

CCSU departments represented as TFIDF vectors with 671 terms

Using two different similarity functions (k = 0, q = 0)

Farthest neighbor

$$sim(S_1, S_2) = \max_{d_1 \in S_1, d_2 \in S_2} sim(d_1, d_2)$$

Average Intracluster Similarity = **0.304475**

```
1 [0.098857]
  2 [0.108415]
    3 [0.126011]
      4 [0.129523]
        5 [0.142059]
          6 [0.148069]
            7 [0.148331]
              8 [0.148599]
                9 [0.169039]
                  10 [0.17462]
                    11 [0.176625]
                      12 [0.201999]
                        13 [0.202129]
                          14 [0.223392]
                            15 [0.226308]
                              16 [0.23571]
                                Computer
                                Political
                              Economics
                            Chemistry
                          Anthropology
                        17 [0.245171]
                          Biology
                          Math
                      Communication
                    Physics
                  Psychology
                Music
              18 [0.177997]
                History
                Philosophy
            19 [0.186299]
              English
              Languages
          Art
        Theatre
      Sociology
    Geography
  Justice
```

```
1 [0.138338]
  2 [0.175903]
    3 [0.237572]
      4 [0.342219]
        5 [0.57103]
          Art
          Psychology
        6 [0.588313]
          Communication
          Economics
      7 [0.39463]
        8 [0.617855]
          Computer
          Political
        9 [0.622585]
          Biology
          Math
    10 [0.292074]
      11 [0.519653]
        Justice
        Theatre
      12 [0.541863]
        Geography
        Physics
  13 [0.209028]
    14 [0.323349]
      15 [0.56133]
        Anthropology
        Sociology
      16 [0.5743]
        Chemistry
        Music
    17 [0.357257]
      18 [0.588999]
        History
        Philosophy
      19 [0.59315]
        English
        Languages
```

Intracluster similarity

$$sim(S) = \frac{1}{|S|^2} \sum_{d_i, d_j \in S} sim(d_i, d_j)$$

Average Intracluster Similarity = **0.434181**

K-means Clustering

- Number of clusters (k) is known in advance
- Clusters are represented by the centroid $c = \frac{1}{|S|}\sum_{d \in S} d$ of the documents that belong to that cluster
- Cluster membership is determined by the most similar cluster centroid

1. Select k documents from S to be used as cluster centroids. This is usually done at random.
2. Assign documents to clusters according to their similarity to the cluster centroids, i.e. for each document find the most similar centroid and assign that document to the corresponding cluster.
3. For each cluster recompute the cluster centroid using the newly computed cluster members.
4. Go to step 2 until the process converges, i.e. the same documents are assigned to each cluster in two consecutive iterations or the cluster centroids remain the same.

K-means Clustering Discussion

- In step 2 documents are moved between clusters in order to maximize the intracluster similarity.
- The clustering maximizes the *criterion function* (a measure for evaluating *clustering quality*).
- In distance-based k-means clustering the criterion function is the *sum of squared errors* (based on Euclidean distance and means).
- For k-means clustering of documents a function based on centroids and similarity is used:

$$J = \sum_{i=1}^{k} \sum_{d_j \in D_i} sim(c_i, d_j)$$

- Clustering that *maximizes* this function is called *minimum variance clustering*
- K-means algorithm produces minimum variance clustering, but does not guarantee that it always finds the global maximum of the criterion function.
- After each iteration the value of *J* increases, but it may converge to a local maximum.
- The result greatly depends on the initial choice of cluster centroids.

K-means Clustering Example (data)

CCSU Departments data with 6 TFIDF attributes

	history	science	research	offers	students	hall
Anthropology	0	0.537	0.477	0	0.673	0.177
Art	0	0	0	0.961	0.195	0.196
Biology	0	0.347	0.924	0	0.111	0.112
Chemistry	0	0.975	0	0	0.155	0.158
Communication	0	0	0	0.780	0.626	0
Computer	0	0.989	0	0	0.130	0.067
Justice	0	0	0	0	1	0
Economics	0	0	1	0	0	0
English	0	0	0	0.980	0	0.199
Geography	0	0.849	0	0	0.528	0
History	0.991	0	0	0.135	0	0
Math	0	0.616	0.549	0.490	0.198	0.201
Languages	0	0	0	0.928	0	0.373
Music	0.970	0	0	0	0.170	0.172
Philosophy	0.741	0	0	0.658	0	0.136
Physics	0	0	0.894	0	0.315	0.318
Political	0	0.933	0.348	0	0.062	0.063
Psychology	0	0	0.852	0.387	0.313	0.162
Sociology	0	0	0.639	0.570	0.459	0.237
Theatre	0	0	0	0	0.967	0.254

K-means Clustering Example (results)

Clustering of CCSU Departments data with 6 TFIDF attributes (k = 2)

Bad choice of initial cluster centroids

Iteration	Cluster A	Cluster B	Criterion function
1	{Computer, Political}	{Anthropology, Art, Biology, Chemistry, Communication, Justice, Economics, English, Geography, History, Math, Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	1.93554 (A) + 4.54975 (B) = 6.48529
2	{Chemistry, Computer, Geography, Political}	{Anthropology, Art, Biology, Communication, Justice, Economics, English, History, Math, Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	3.82736 (A) + 10.073 (B) = 13.9003
3	{Anthropology, Chemistry, Computer, Geography, Political}	{Art, Biology, Communication, Justice, Economics, English, History, Math, Languages, Music, Philosophy, Physics, Psychology, Sociology, Theatre}	4.60125 (A) + 9.51446 (B) = 14.1157

Good choice of initial cluster centroids

Iteration	Cluster A	Cluster B	Criterion function
1	{Anthropology, Biology, Economics, Math, Physics, Political, Psychology}	{Art, Chemistry, Communication, Computer, Justice, English, Geography, History, Languages, Music, Philosophy, Sociology, Theatre}	5.04527 (A) + 5.99025 (B) = 11.0355
2	{Anthropology, Biology, Computer, Economics, Math, Physics, Political, Psychology, Sociology}	{Art, Chemistry, Communication, Justice, English, Geography, History, Languages, Music, Philosophy, Theatre}	7.23827 (A) + 6.70864 (B) = 13.9469
3	{Anthropology, Biology, Chemistry, Computer, Economics, Geography, Math, Physics, Political, Psychology, Sociology}	{Art, Communication, Justice, English, History, Languages, Music, Philosophy, Theatre}	8.53381 (A) + 6.12743 (B) = 14.6612

K-means Clustering Extensions

- By applying the algorithm recursively to the obtained clusters k-means can be easily extended to produce *hierarchical clustering*.
- Other *content-based similarities* can be used, such as *Jaccard* similarity and *document resemblance*.
- *Link-based similarity* may be used too:
  - The length of shortest path between the documents in the web graph
  - The number of common ancestors of the documents (pages with links to both).
  - The number of common successors of the documents (pages that are pointed by links in both).
- *Combined similarity* may be computed as the maximum of the content similarity (cosine, Jaccard or resemblance) plus a weighted sum of the three link-based similarities.

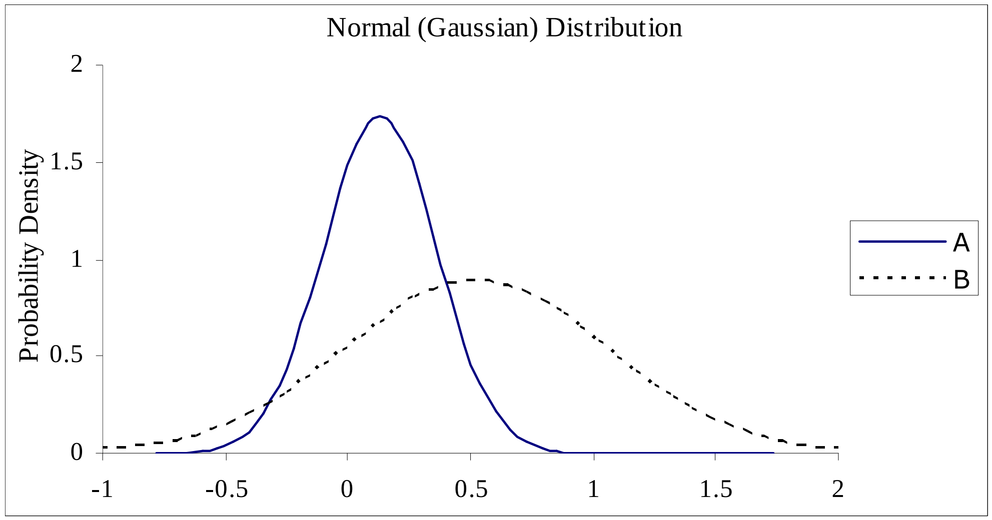
Probability-Based Clustering

- Document is a *random event* that occurs according to different *probability distributions* depending on which cluster the document belongs to
- Parameters involved:
  - The document class labels (may be known or unknown)
  - The parameters of the probability distribution for each cluster
  - The way terms are used in the document representation
- Generally there are three ways a term can be used in this model:
  - As a *binary variable* taking values 0/1 depending on whether or not the term occurs in the document. Documents are binary vectors following *multivariate binary distribution*.
  - As a *natural number* indicating the number of occurrences (frequency) of the term in the document. Documents are *vectors of* natural numbers following *multinominal distribution*.
  - A normally distributed *continuous variable* taking TFIDF values. Documents are TFIDF vectors following *multivariate normal distribution*.

Two Class Mixture Model

Values of "offers" taken from the CCSU departments collection (data table on Slide 13)

A	0
B	0.961
A	0
A	0
B	0.780
A	0
B	0
A	0
B	0.980
A	0
B	0.135
A	0.490
B	0.928
B	0
B	0.658
A	0
A	0
A	0.387
A	0.570
B	0



Class	Mean	Standard deviation	Probability of sampling
A	$\mu_A = 0.132$	$\sigma_A = 0.229$	$P(A) = 0.55$
B	$\mu_B = 0.494$	$\sigma_B = 0.449$	$P(B) = 0.45$

Finite Mixture Problem

Given the labeled data, for each class *C* compute:

- *mean* $\mu_C = \frac{1}{|C|}\sum_{x \in C} x$
- *standard deviation* $\sigma_C = \sqrt{\frac{1}{|C|}\sum_{x \in C}(x - \mu_C)^2}$
- *probability of sampling* $P(C)$

Generative document model $\langle \mu_C, \sigma_C, P(C) \rangle$

Finite Mixture Problem (example)

A	0
B	0.961
A	0
A	0
B	0.780
A	0
B	0
A	0
B	0.980
A	0
B	0.135
A	0.490
B	0.928
B	0
B	0.658
A	0
A	0
A	0.387
A	0.570
B	0

$$\mu_A = \frac{1}{11}(0+0+0+0+0+0+0.49+0+0+0.387+0.57) = 0.132$$

$$\mu_B = \frac{1}{9}(0.961+0.780+0+0.980+0.135+0.928+0+0.658+0) = 0.494$$

$$\sigma_A = 0.229$$

$$\sigma_B = 0.449$$

$$P(A) = \frac{11}{20} = 0.55 \qquad P(B) = \frac{9}{20} = 0.45$$

Classification Problem

- Given $\langle \mu_C, \sigma_C, P(C) \rangle$ for class A and B
- Compute $P(A|x)$ and $P(B|x)$
- Use

$$P(C\,|\,x) = \frac{P(x\,|\,C)\,P(C)}{P(x)}$$, if x is a discrete variable

$$P(C\,|\,x) \approx \frac{f_C(x)\,P(C)}{P(x)},$$ if x is a continuous variable

Probability density function $f_C(x) = \frac{1}{\sqrt{2\pi}\sigma_C} e^{\frac{-(x-\mu_C)^2}{2\sigma_C{}^2}}$

Classification Problem (example)

- Given the value of the "offers" attribute (0.78) find the class label of the of the "Communication" document (fifth row in the data table).
- Compute likelihoods

$$P(A\,|\,0.78) \approx f_A(0.78)\,P(A) = 0.032 \times 0.55 = 0.018$$

$$P(B\,|\,0.78) \approx f_B(0.78)\,P(B) = 0.725 \times 0.45 = 0.326$$

- Normalize and get probabilities

$$P(A\,|\,0.78) = \frac{0.018}{0.018 + 0.326} = 0.05$$

$$P(B\,|\,0.78) = \frac{0.326}{0.018 + 0.326} = 0.95$$

- $P(B\,|\,0.78) > P(A\,|\,0.78)$ $\Rightarrow$ The class of "Communication" is B

Classification Problem (multiple attributes)

- Independence assumption (*Naïve Bayes assumption*)

$$P(x_1, x_2, ..., x_n \mid C) = \prod_{i=1}^{n} P(x_i \mid C)$$

- *Naïve Bayes* classification algorithm

$$P(C \mid (x_1, x_2, ..., x_n)) \approx \prod_{i=1}^{n} f_C^i(x_i) \frac{P(C)}{P((x_1, x_2, ..., x_n))}$$

- Find the class of "Theatre" (0, 0, 0, 0, 0.967, 0.254) (data table on Slide 13)

$$P(A \mid (0,0,0,0,0.976,0.254) \approx \frac{f_A^1(0)\, f_A^2(0)\, f_A^3(0)\, f_A^4(0)\, f_A^5(0.976)\, f_A^6(0.254) P(A)}{P((0,0,0,0,0.976,0254))}$$

$$P(A \mid (0,0,0,0,0.976,0.254) \approx 7.979 \times 0.5 \times 0.423 \times 1.478 \times 0.007 \times 1.978 \times 0.55 = 0.019$$

$$P(B \mid (0,0,0,0,0.976,0.254) \approx 0.705 \times 7.979 \times 7.979 \times 0.486 \times 0.698 \times 1.604 \times 0.45 = 10.99$$

$$P(A \mid (0,0,0,0,0.976,0.254) = \frac{0.019}{0.019 + 10.99} = 0.002 \qquad P(B \mid (0,0,0,0,0.976,0.254) = \frac{10.99}{0.019 + 10.99} = 0.998$$

Clustering Problem

- Given the set of values and a predefined number of clusters (e.g. $k = 2$)
  - Assign a label (*A* or *B*) to each value, or
  - Find the cluster parameters $\langle \mu_A, \sigma_A, P(A) \rangle$ and $\langle \mu_B, \sigma_B, P(B) \rangle$
- *Expectation Maximization* (EM) is a popular algorithm used for this purpose
- EM is an *optimization approach*, which given some initial approximation of the cluster parameters *iteratively* performs two steps:
  - The "expectation" step computes the "expected" values of the cluster probabilities.
  - The "maximization" step computes the distribution parameters and their likelihood given the data.
- EM iterates until the parameters being optimized reach a fixpoint or the *log-likelihood function*, which measures the quality of clustering, reaches its (local) maximum.

$$L = \sum_{i=1}^{n} \log \sum_{C} P(x_i \mid C)\, P(C)$$

Expectation Maximization (EM)

- Given
  - set of values $x_1, x_2, ..., x_n$ of a normally distributed random variable
  - parameter *k* (predefined number of clusters)
  - set of initial cluster parameters $\mu_C, \sigma_C, P(C)$ (usually selected at random)
- Iterate through the following steps:
  - for each x_i and for each cluster *C* compute the probability that x_i belongs to *C*, $w_i = P(C \mid x_i) \approx f_C(x_i)P(C)$. Normalize w_i across all clusters.
  - $P(C)$ is the sum of the weights w_i for cluster *C* (from the previous step)
  - Compute *weighted* mean and standard deviation

$$\mu_C = \frac{\sum_{i=1}^{n} w_i\, x_i}{\sum_{i=1}^{n} w_i} \qquad \sigma_C^2 = \frac{\sum_{i=1}^{n} w_i\,(x_i - \mu_C)^2}{\sum_{i=1}^{n} w_i}$$

- Stop when the process converges, or the log-likelihood criterion function reaches it maximum $L = \sum_{i=1}^{n} \log \sum_{C} P(x_i \mid C)\, P(C) \approx \sum_{i=1}^{n} \log \sum_{C} w_i$ (w_i before normalization)

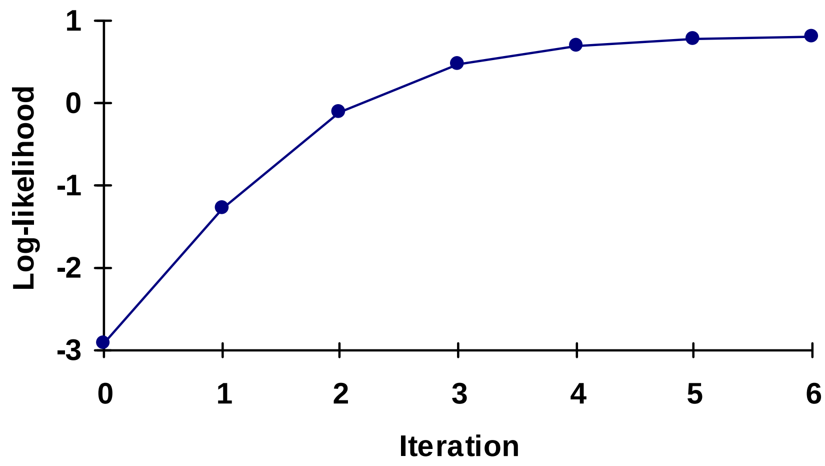
Expectation Maximization (example 1)

EM iterations with one attribute ("students" from data table on Slide 13)

Iteration		0		1		2		3		4		5		6	
Data		w_i		w_i		w_i		w_i		w_i		w_i		w_i	
i	x_i	A	B	A	B	A	B	A	B	A	B	A	B	A	B
1	0.67	**1**	0	**0.99**	0.01	**1**	0	**1**	0	**1**	0	**1**	0	**1**	0
2	0.19	**1**	0	0.4	**0.6**	0.35	**0.65**	0.29	**0.71**	0.26	**0.74**	0.23	**0.77**	0.21	**0.79**
3	0.11	0	**1**	0.41	**0.59**	0.29	**0.71**	0.19	**0.81**	0.13	**0.87**	0.1	**0.9**	0.09	**0.91**
4	0.15	0	**1**	0.39	**0.61**	0.31	**0.69**	0.23	**0.77**	0.18	**0.82**	0.15	**0.85**	0.13	**0.87**
5	0.63	**1**	0	**0.99**	0.01	**1**	0	**1**	0	**1**	0	**1**	0	**1**	0
...	…	…	…	…	…	…	…	…	…	…	…	…	…	…	…
$\sum w_i$		13	7	12.2	7.8	11.1	8.9	10.2	9.8	9.6	10.4	9.2	10.8	9.1	10.9
Cluster Prob.		0.65	0.35	0.61	0.39	0.56	0.44	0.51	0.49	0.48	0.52	0.46	0.54	0.45	0.55
λ		0.35	0.19	0.40	0.14	0.44	0.11	0.49	0.10	0.52	0.09	0.54	0.09	0.55	0.09
σ		0.35	0.14	0.34	0.12	0.33	0.10	0.32	0.09	0.31	0.09	0.30	0.09	0.29	0.09
Log-likelihood		-2.92201		-1.29017		-0.099039		0.47888		0.697056		0.769124		0.792324	

Expectation Maximization (results 1)

- Initial clustering (iteration 0):
  - **A** = {Anthropology, **Art**, Communication, Justice, **English**, Geography, **History**, **Math**, **Languages**, **Philosophy**, Physics, **Political**, Theatre}
  - **B** = {Biology, Chemistry, Computer, Economics, Music, **Psychology**, **Sociology**}
- Final clustering (iteration 6):
  - **A** = {Anthropology, Communication, Justice, Geography, Physics, **Psychology**, **Sociology**, Theatre}
  - **B** = {**Art**, Biology, Chemistry, Computer, Economics, **English**, **History**, **Math**, Music, **Languages**, **Philosophy**, **Political**}
- Log-likelihood graph (a threshold of 0.03 stops the algorithm at iteration 6)



Expectation Maximization (example 2)

Two runs (random choice of initial distributions) of the *multivariate* EM algorithm (6 attributes)

Document	k-means labels	Run 1 (Log-likelihood = 0.1334)		Run 2 (Log-likelihood = 4.8131)	
		w_i (cluster A)	w_i (cluster B)	w_i (cluster A)	w_i (cluster B)
Anthropology	A	1	0	0.99999	0.00001
Art	B	0	1	**0.9066**	0.0934
Biology	A	0.99995	0.00005	1	0
Chemistry	A	**1**	0	1	0
Communication	B	0	1	**0.96278**	0.03722
Computer	A	1	0	1	0
Justice	B	0.0118	0.9882	**0.98363**	0.01637
Economics	A	0.70988	0.29012	0.99999	0.00001
English	B	0	1	**0.81042**	0.18958
Geography	A	1	0	0.99999	0.00001
History	B	0.01348	0.98652	0	1
Math	A	1	0	0.99999	0.00001
Languages	B	0	1	**0.71241**	0.28759
Music	B	0.01381	0.98619	0	1
Philosophy	B	0	1	0	1
Physics	A	0.06692	**0.93308**	0.99999	0.00001
Political	A	1	0	1	0
Psychology	A	0.0368	**0.9632**	0.99999	0.00001
Sociology	A	0.00016	**0.99984**	0.99982	0.00018
Theatre	B	0.0023	0.9977	**0.98818**	0.01182

Collaborative Filtering

- Basic relations used in the description
  - "document contains term" (content-based document retrieval, clustering and classification)
  - "web user likes web page" or "person likes item" (*collaborative filtering*, *recommender systems*)
- Assume that we have m persons and n items (e.g. books, songs, movies, web pages etc.)
  - matrix $M(m \times n)$, where each row is a person, each column is an item
  - if person i likes item j then $M(i, j) = 1$, otherwise $M(i, j) = 0$
- Many cells in the matrix are empty, i.e. we don't know whether or not a person likes an item.
- The task of a collaborative filtering system is to predict the missing values by using the rest of the information in the matrix.

Collaborative Filtering (clustering approach)

- A straightforward approach to solve this problem is *clustering*
  - items are used as attributes to represent persons as vectors
  - person vectors are clustered (e.g. k-means or EM)
  - the missing values are taken from the cluster representation, where the person belongs
- Problems
  - highly sparse data (still can be handles by probabilistic approaches)
  - persons often appear in multiple clusters (people usually have multiple interests)
  - uses only the similarity between persons

Collaborative Filtering (EM-like algorithm)

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 maximal likelihood values of the above probabilities
4. If the parameter estimation is satisfactory terminate, else go to 2