||D1|| = sq.root(sum(d1.d1) for all d1)

**Lesson #9**

Absolute support, or, support count of X

Relative support: The fraction of transactions that contain X

Frequent Itemsets: An itemset X is frequent if X’s support is no less than a predefined minimum support threshold, called minsup.

An association rule has the form R1: X -> Y, where X and Y are itemsets. The rule says "whenever X occurs, we expect to find Y as well“.

Support of R1: s(R1) = support(X uinion Y) or probability that a transaction contains X and Y

Confidence of R1: c(R1) = support(X union Y) / support(X), or conditional probability that a transaction having X also contains Y.

The confidence of a rule tells us how reliable the rule is. The closer the value c(R1) comes to 1, the more reliable the rule R1 is.

A strong rule is a rule X -> Y with the following two properties:

1- support ≥ minimum support (minsup) and

2- confidence ≥ minimum confidence (minconf).

The Apriori Algorithm

Two-step process:

1. First, mine all frequent patterns. A frequent pattern is also called a frequent itemset or a large itemset.
2. Join step: Generate all possible candidate itemsets of size k+1 (we join only if the first k-1 items are identical.)
3. prune step: Remove those candidates in Ck+1 that cannot be frequent. (Using downward closure property)
4. Second, mine strong rules from frequent itemsets.
5. For each frequent itemset I, and for each proper nonempty subset X of I, Let Y = I - X then X -> Y is a strong rule if Confidence(X -> Y) ≥ minconf

The apriori property (downward closure property): A subset of a frequent itemset must also be a frequent itemset.

The lift of a rule measures how much more likely one itemset is purchased with another itemset compared to its typical rate of purchase.

(Negative correlation means X and Y occur together less frequently than would happen by chance.)

To see if a rule X->Y indicates that Y is purchased more frequently when X is purchased than otherwise, compare confidence(X->Y) to support(Y). This gives the Lift Formula: 𝑙𝑖𝑓𝑡 𝑋 → 𝑌 = 𝑐𝑜𝑛𝑓𝑖𝑑𝑒𝑛𝑐𝑒(𝑋→𝑌) / 𝑠𝑢𝑝𝑝𝑜𝑟𝑡(𝑌)

• lift > 1: positively correlated

• lift < 1: negatively correlated,

• lift = 1: independent (no correlation)

---------------------------------------------------------------------------------------------------------------------------

**Lecture #10**

Cluster Analysis = data segmentation

Clustering is a technique used for automatic identification of natural groupings of things

A good clustering method will produce high quality clusters

\* high intra-class similarity: cohesive within clusters

\* low inter-class similarity: distinctive between clusters

Clustering Types

**1- Bayesian (Decision Based, nonparametric)**

**2- Hierarchical (Divisive, agglomerative)**

**3- Partitional (Centroid or K-Means, Model Based, Graph Theoretic, Spectral)**

\* Hierarchical algorithms find successive clusters using previously established clusters. These algorithms can be either agglomerative (“bottom-up”) or divisive (“top-down”):

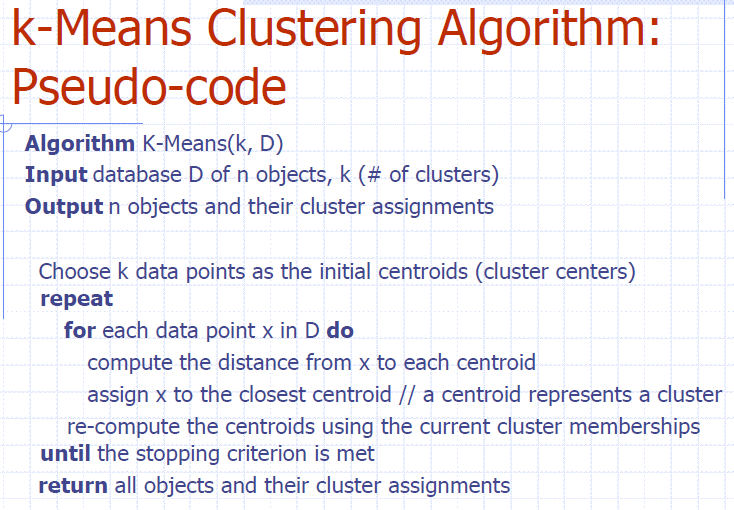
1- Agglomerative algorithms begin with each element as a separate cluster and merge them into successively larger clusters;

2- Divisive algorithms begin with the whole set and proceed to divide it into successively smaller clusters.

\* Partitional algorithms: Given a set of n objects, a partitioning method constructs k partitions of the data, where each partition represents a cluster and k <= n.

SSE (sum of squared errors)





**Stopping Criteria**

1. The membership assignment does not change.
2. After each reassignment, E is computed and if E falls below a predefined threshold.
3. If the decrease in E, between two consecutive iterations, falls below a predefined threshold.
4. Run for a predetermined number of iterations (e.g., run 20 iterations).

**Manhattan (city block) distance**



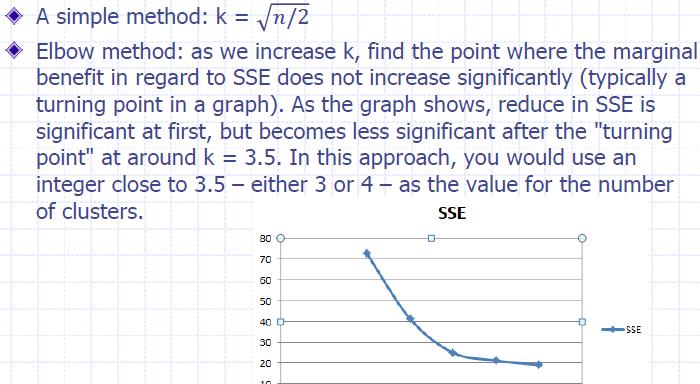
**Euclidean distance**



**Minkowski distance**



**Determine the number of clusters**



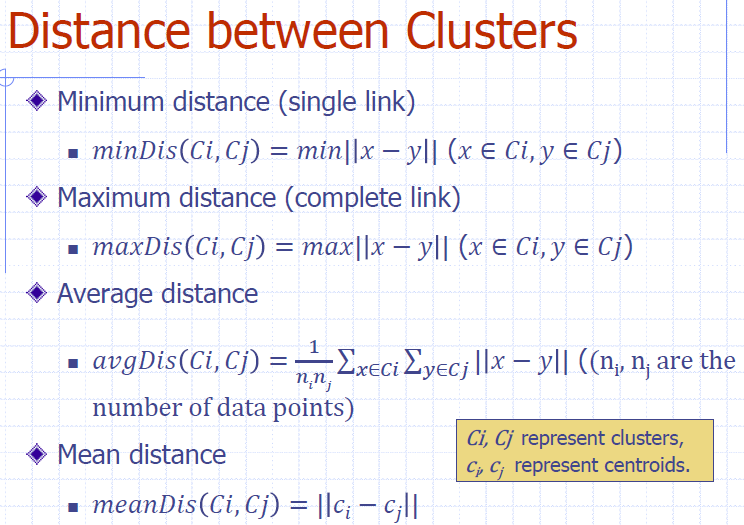
**k-means adv.**

1. Efficient K-Means is considered a linear algorithm.
2. Simple

Dis-adv.

1. Initial random selection of centroids affects the results
2. Applicable only when the mean of objects can be defined. Use the k-modes method for categorical data
3. Sensitive to outliers
4. Not suitable to discover clusters with arbitrary shapes

K-Medoids: instead of mean, use medians (center) of each cluster. not affected by extreme values.



Questions:

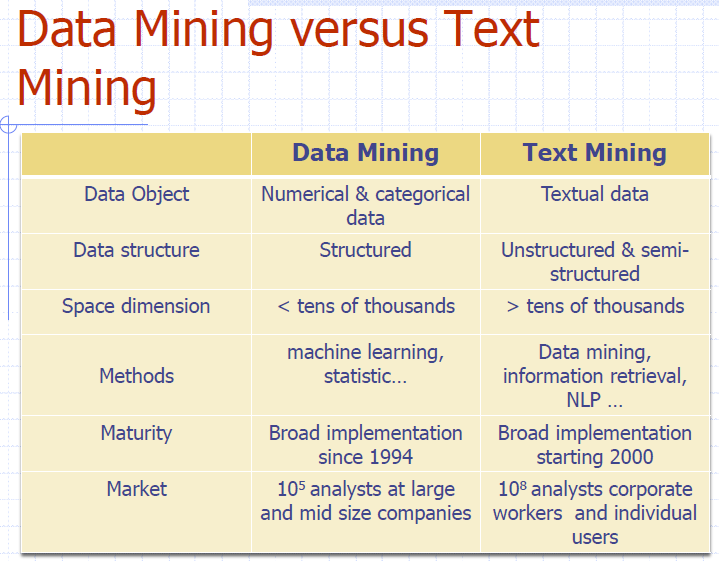
1. What is k-modes
2. A good clustering method will produce high quality clusters ???
   1. \* high intra-class similarity: cohesive within clusters
   2. \* low inter-class similarity: distinctive between clusters
3. #compute within-cluster sum of squares for cluster 1. Is it included?

**Lecture 11**

Text Mining is the process of deriving meaningful information from the natural language text.

Data Mining versus Text Mining

1. Both seek for novel and useful patterns
2. Difference is the nature of the data:
3. Data Mining works on structured data stored in databases
4. Text Mining works on unstructured data in Word documents, PDF files, XML files, etc
5. Text mining – first, impose structure to the data, then mine the structured data



Text Mining Process – three steps

1. Establish the Corpus of Text: Gather documents, clean, prepare for analysis
2. Collect all relevant unstructured data
3. Digitize, standardize the collection (e.g., all in ASCII text files)
4. Place the collection in a common place (e.g., in a flat file, or in a directory as separate files)
5. Structure using Document Term Matrix (DTM): Select a bag of words, compute frequencies of occurrence
   1. A document term matrix is a matrix with documents as the rows, terms as the columns, and count of the frequency of words as the cells of the matrix.
6. Mine DTM for Patterns: Apply data mining tools like classification and cluster analysis

* Corpora are R objects held fully in memory.

Pre-processing transformations such as:

Converting the text to lower case

Removing numbers and punctuation

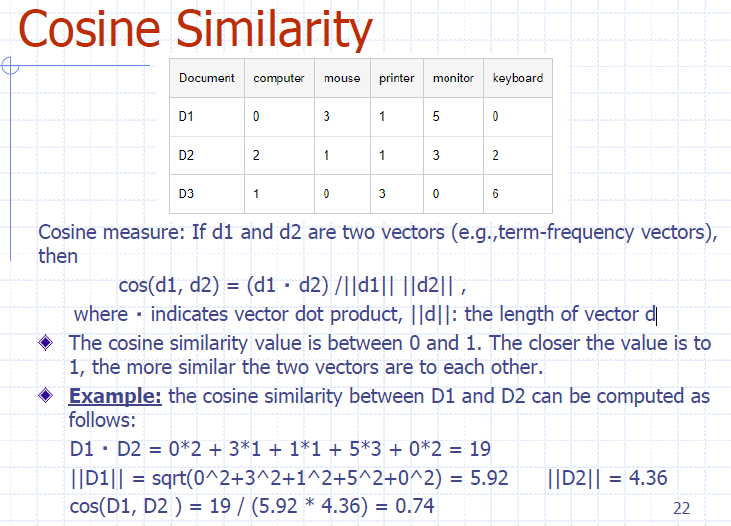
Removing stop words

Stop words examples:„ the, and, a, an, is, of, that

Stemming and identifying synonyms

Stemming: reducing words to their root form, such as ‘fishes’ -> ‘fish’, ‘played’ -> ‘play’…

Cosine distance (cosine similarity)



||D1|| = sq.root(sum(d1^2) for all d1)

Term frequency-inverse document frequency (tf-idf): TF-IDF is a weighting mechanism that calculates the importance of each term for each document by increasing the importance based on the term frequency while decreasing the importance based on the document frequency.

Normalization maps term frequency to a number between 0 and 1.