**Week 1 lecture**

**Unstructured data**

Data without regular, decomposable internal structure

Examples: blogs, MP3 files, JPEG files

**Structured data**

Data which strictly conforms to a schema

Examples: ABN lookup, library catalogues

**Semi-Structured data**

Data which conforms in part to a schema

Examples: Wikipedia entries

**Supervised learning**

* Classification, predicting a discrete class
* Regression, predicting a numeric quantity

**Unsupervised learning**

* Reinforcement learning
* Recommender systems

Regular expression

\*: zero or more of the preceding element

?: zero or one of the preceding element

+: one or more of the preceding element

{n}: exactly n of the preceding element

{m,n}: between m and n (inclusive) of the preceding element

{n,}: n or more of the preceding element

{,m}: up to m of the preceding element

[0-9] = [[:digit:]] = \d

[a-zA-Z0-9\_] = [[:word:]] = \w

[\ \t\r\n\f] = [[:space:]] = \s

[^0-9] = \D

[^a-zA-Z0-9\_] = \W

[^\ \t\r\n\f] = \S

Tutorial 1

* Data: measurements (bit patterns for computers)
* Information: processed data; patterns that are satisfied for given data (sampling in the graph)
* Knowledge: information interpreted with respect to a user’s context to extend human understanding in a given area (where we have data)

Concrete tasks: mechanically processing data to an unambiguous solution; limited contribution to human understanding

Knowledge tasks: data is unreliable or the outcome is ill-defined (usually both); computers mediate between the user and the data, where context (for the user) is critical; enhance human understanding

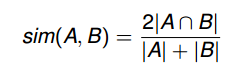
Week 2 lecture

**For similarity, in the vector space model (table), A and B represent the whether the term appear or not, except cosine similarity**

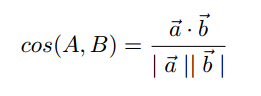
Jaccard Similarity:



Dice Similarity:

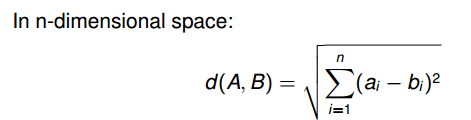


Cosine similarity;

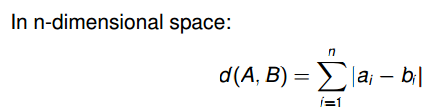


**For distance, using the real frequency to calculate the distance in vector space table**

Euclidean Distance:



Manhattan Distance:



**For similarity, the highest value would be the most similar**

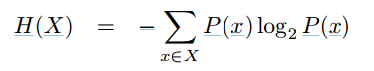
**For distance, the lowest value would be the most similar**

Tutorial 3

**Entropy** (Information Theory):

The information (**in bits**) required to predict an event is the distribution’s entropy or information value

* A high entropy value means x is unpredictable.
* A low entropy value means x is predictable.
* 0 ≤ Entropy ≤ log2(n), where n is number of outcomes



Week 3 lecture

**Neighbourhood Search:**

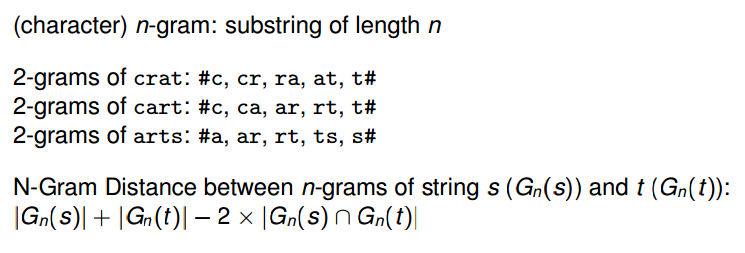
For a given string w of interest: Generate all variants of w that utilise at most k changes (**Insertions/Deletions/Replacements**) — neighbours, Check whether generated variants exist in dictionary

**Global Edit Distance:**

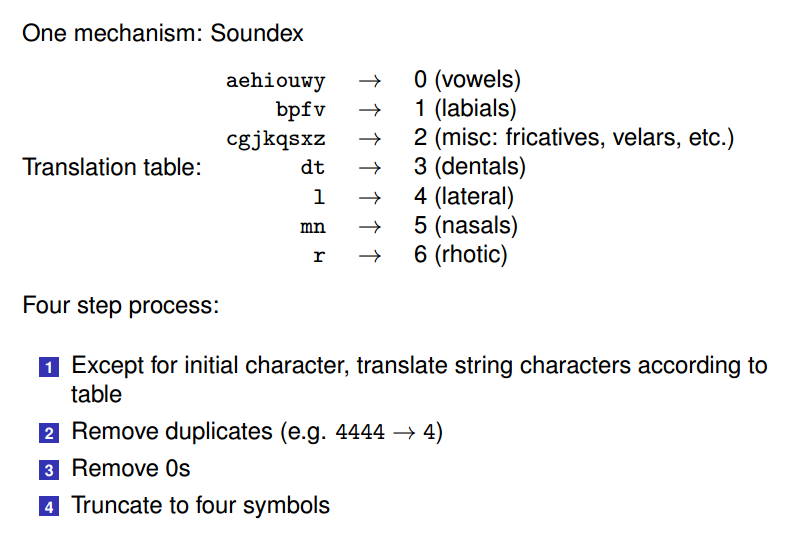
Video: https://www.youtube.com/watch?v=b6AGUjqIPsA

**Local Edit Distance:**

**N-Gram Distance:(no order when compare the two sets)**

****

**Soundex:**

****

Accuracy: correct/total

Precision: correct responses/ total attempted responses

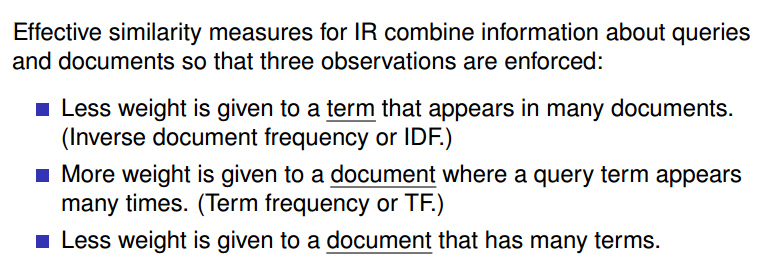
Recall:correct/wrong

Week 4 lecture

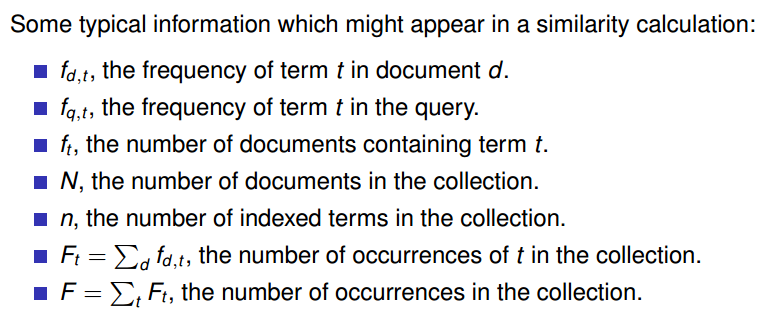
**Boolean querying**: There is no ordering; matching is yes/no

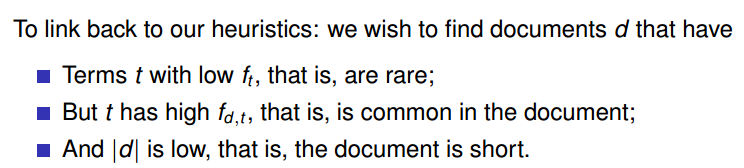
**Ranked retrieval:** a query is matched to a document by looking for evidence in the document that it is on the same topic as the query (or the same topic as an information need that the query might represent). The more similar or likely a document is, relative to the other documents in the collection, the higher its rank.

**TF-IDF model:**

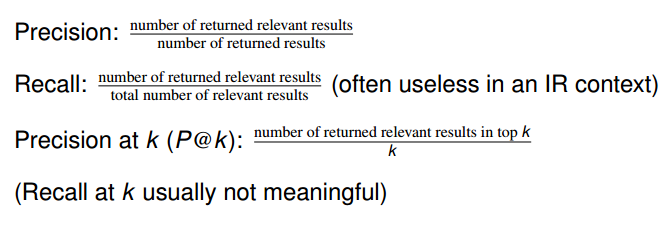
****

**Since many terms in a document would increase the happening probability of the query term**





**Evaluation Metrics in an IR Context**

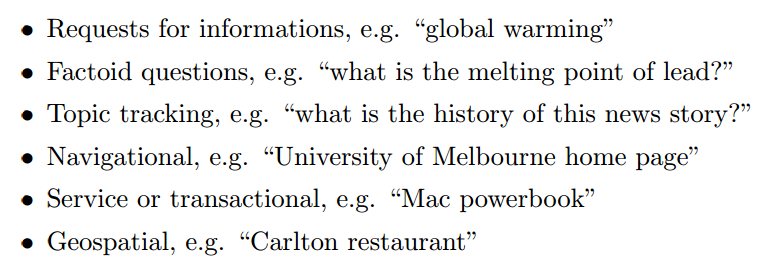
****

Tutorial 5

Data retrieval vs information retrieval: a relevant result in information retrieval depends on contextualising the data to the particular user. Data retrieval, there is a particular unit of data (bitstream) that we need to access in memory or on a hard drive, and there is generally no ambiguity.

* User has an information need
* User formulates a query
* IR engine retrieves a set of documents

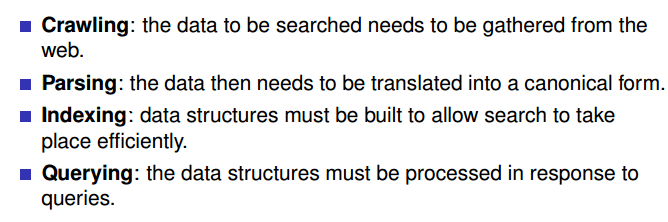
**A different type of information need:**

****

Any query can be construed as being navigational in nature (as the user is likely to click through to a relevant document),

Week 5 lecture

Web search --> four main technological components.

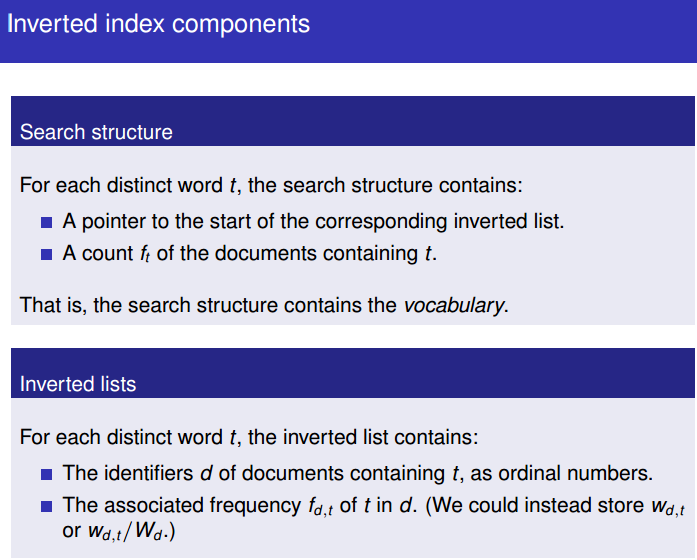


**Crawling:** Crawlers attempt to visit every page of interest and retrieve them for processing and indexing.

Page recognition (parsing): the words in the document are extracted, then added to a data structure that records which documents contain which words.

* Tokenisation: The aim of parsing is to reduce a web page, or a query, to a sequence of tokens. If the tokenisation is successful, the tokens in a query will match those of the web page, allowing query evaluation to proceed without any form of approximate matching.
* Canonicalization:  Any indexing process that relies on fact extraction may need information in a canonical form. "Authoritative," "standard," or "official."  For example, two formulas such as 9 + x and x + 9 are said to be equivalent because they mean the same thing, but the second one is in canonical form because it is written in the usual way, with the highest power of x first.
* Stemming: **Stemming is the process of stripping away affixes.**
* Zoning: Web documents can usually be segmented into discrete zones such as title, anchor text, headings, and so on. Parsers also consider issues such as font size, to determine which text is most prominent on the page and thus generate further zones. Web search engines typically calculate weights for each of these zones, and compute similarities for documents by combining these results on the fly.

Indexing:



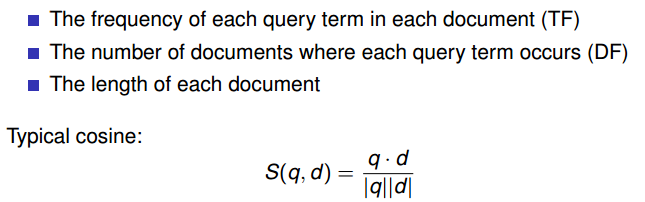
An inverted index allows for fast querying because:

(1) The terms in the query correspond to the search structure

(2) The index only indicates documents where the term is present

Ranked Querying:

Using cosine similarity measures:

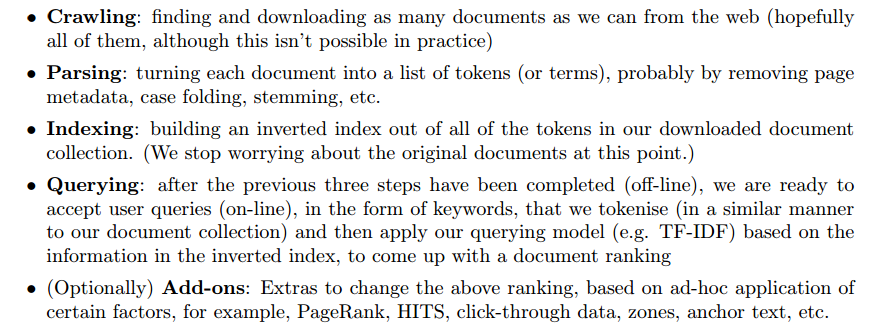


Using an inverted index:

Inverted index for Boolean querying:

<https://nlp.stanford.edu/IR-book/html/htmledition/the-extended-boolean-model-versus-ranked-retrieval-1.html>

Tutorial 6



**What is “tokenisation”?**

Tokenisation is the act of turning the raw text of a document into tokens that we will use to compare against the (similarly tokenised) query, for example, by removing meta-data and punctuation, folding case, canonicalising for dialect differences, and so on.

**What is “phrase querying”?**

Phrase querying is where we treat the query as a phrase, which is to say, that we care about the order of terms in the query— in particular, we wish to return documents where the terms occur one after each other, in the order that they occur in the query

**Three main strategies for phrase query evaluation:**

Process queries as bag-of-words, so that the terms can occur anywhere in matching documents, then post-process to eliminate false matches.

Add word positions to the index entries,

Use some form of phrase index or word-pair index(no need to use the inverted index)

**What is “link analysis”?**

Link analysis is another method for approximating this: basically people tend to provide hypertext links on their webpages to useful documents. If a document has many (good) incoming links, it is more likely to be relevant than a document with few (good) incoming links.

**PageRank:** set up a network where each page has “credits” associated with it. Then we have an iterative algorithm where pages transfer their credits to the pages to which they have outgoing links, and receive credits from pages with links pointing to them. After a number of iterations, the pages with a large number of credits tend to be better than documents with few credits, and we might use this to alter our document ranking somewhat

Week 6 lecture

**Data mining**: information from data

Needed: programs that detect patterns and regularities in the data

Strong patterns → good predictions

**Supervised learning:** Teach the computer how to do something (by example), then let it use its new-found knowledge to do it

Classification: predicting a discrete class

Regression: predicting a numeric quantity

**Unsupervised learning:** Let the computer learn how to do something without given answers

The input to a machine learning system consists of: instances, attributes and concepts

Nominal attributes: Values are distinct symbols (e.g. {sunny, overcast, rainy})

Ordinal attributes: An explicit order is imposed on the values (e.g. {hot, mild, cool} where hot > mild > cool)

Continuous attributes (numeric): Suppose your table in the database has a column which stores the temperature of the day or say a furnace. The values for that column come from a continuous domain of temperature values. If the table has a column named gender. Then that is discrete in the sense that only two or maybe three values comprise its domain.

Tutorial 7

Data mining: extracting implicit, previously unknown, potentially useful information from data

Machine learning: algorithms for acquiring structural descriptions from examples (special case of above?)

Knowledge task: the information/descriptions we produce are unknown and useful to humans

k-means clustering : <https://www.youtube.com/watch?v=rjm4slbER_M>

Week 7 lecture

Under-fitting: model not expressive enough to capture patterns in the data.

Over-fitting: model too complicated; capture noise in the data.

Appropriate-fitting model captures essential patterns in the data.

Bias and Variance in Evaluation:

* The (training) bias of a classifier is the average distance between the expected value and the estimated value, Bias is large if the learning method produces classifiers that are consistently wrong.
* The (test) variance of a classifier is the standard deviation between the estimated value and the average estimated value, Variance measures how inconsistent the decisions are, not whether they are correct or incorrect.

Aim to minimise classifier bias and variance

Tutorial 8

**Overfitting** is when the classifier fails to generalise — it builds a model which describes the training data very well, but doesn’t describe the test data well.

Confusion matrix :

"TP of C1" is all C1 instances that are classified as C1.

"TN of C1" is all non-C1 instances that are not classified as C1.

"FP of C1" is all non-C1 instances that are classified as C1.

"FN of C1" is all C1 instances that are not classified as C1.

Holdout: Train a classifier over a fixed training dataset, and evaluate it over a fixed held-out test dataset

Random Subsampling: Perform holdout over multiple iterations, randomly selecting the training and test data (maintaining a fixed size for each dataset) on each iteration

Let us assume we have N data points for which we know the labels.

Leave-One-Out: We choose each data point as test case and the rest as training data (no partitions comparing to cross validation) ( no sampling bias)( training N times)

M-fold Cross-Validation: We partition the data into M (approximately) equal size partitions. We choose each partition for testing and the remaining M-1 partitions for training. We need to train the system only M times unlike Leave-One-Out which requires training N times. (There can be a bias in evaluating the system due to sampling, how data is distributed among the M partitions)

Week 8 lecture

Small margin separating planes:

* are more fragile to noise
* may over-fit the data

Large margin separating planes:

* are more robust to noise
* From statistical learning theory: large margin planes generalises better to unseen data

**For linearly separable data**: a max-margin solution is guaranteed to exist

**For non- linearly** **separable data**: a solution does not exist

Tutorial 9

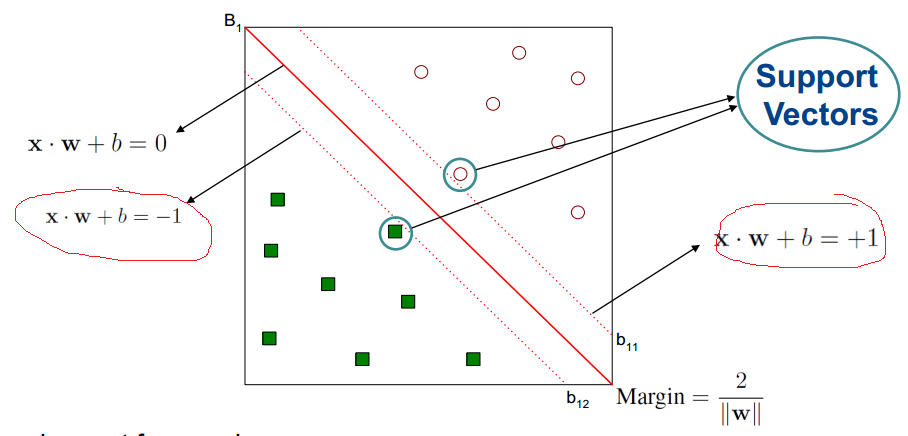
**Support vector machines** attempt to partition the training data based on the best line (hyperplane) that divide the positive instances of the class that we’re looking for from the negative instances

**linearly separable**, when the points in our k-dimensional vector space corresponding to positive instances can indeed be separated from the negative instances by a line

“best” line: we choose a pair of parallel lines, one for the positive instances, and one for the negative instances. The pair that we choose is the one that has the greatest perpendicular distance between these parallel lines. This is called the “margin”.

These two lines are called the “support vectors”, if the instance is on the +1 side then positive

If the instance is on the -1 side then negatives



**Soft margins**: A small number of points are allowed to be on the “wrong” side of the line, if we get a (much) better set of support vectors that way (i.e. with a (much) larger margin). If we accept that we probably aren’t going to classify every single test instance correctly anyway, we can produce a classifier that hopefully generalises better to unseen data.

Kernel function:

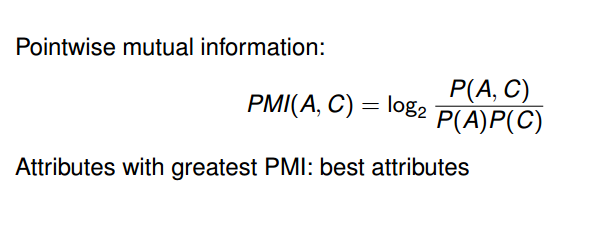
A kernel function implicitly maps data to a high-dimensional space. For Support Vector Machines, sometimes the data isn’t linearly separable, but after applying some kind of function — where some useful properties of the data remain (for example, monotonicity of the inner product) — the data becomes linearly separable.

SVM is binary classifier, and multi classifier is:

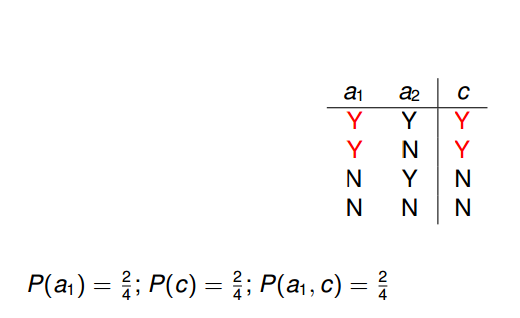
One-versus-all: build M classifiers for M classes. Choose class with largest margin for test data

One-versus-one: one classifier per pair of classes (M (M-1)/2 classifiers in total), choose class selected by most classifiers

Week 9 lecture

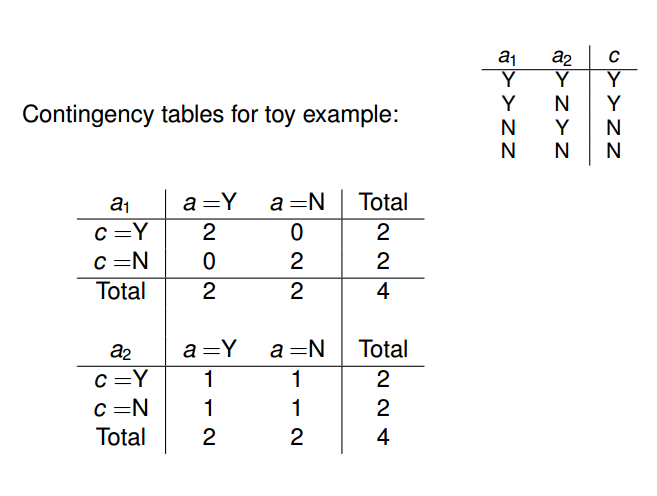


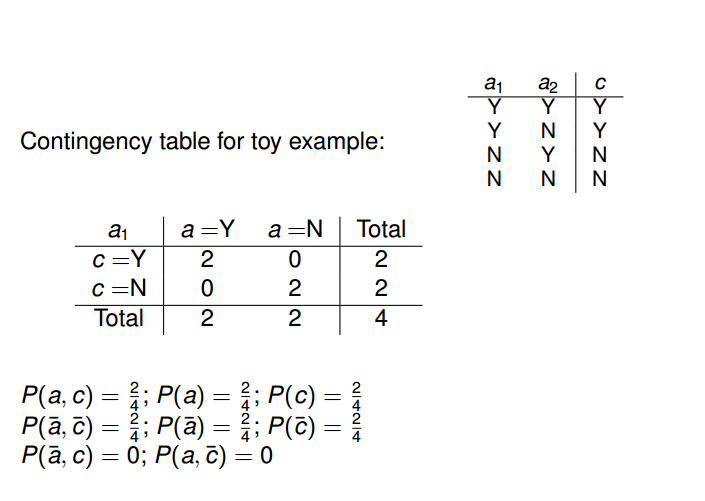
Single attribute only!!!!

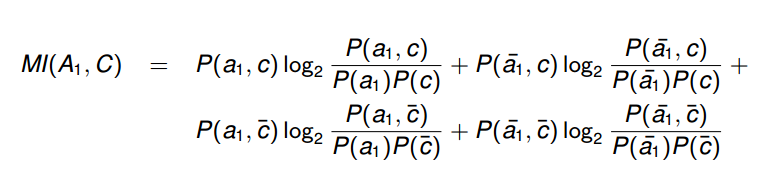


P (a1, c) means the a1 and c happens at the same time(Y, Y at the same time)

**Mutual Information**







Nominal attributes: Treat as multiple binary attributes,( sunny=Y, overcast=N, rainy=N)

Ordinal attributes

Tutorial 10

**ZeroR** is the simplest classification method which relies on the target and ignores all predictors. ZeroR classifier simply predicts the majority category (class). Although there is no predictability power in ZeroR, it is useful for determining a baseline performance as a benchmark for other classification methods.

**OneR**, short for "One Rule", is a simple, yet accurate, classification algorithm that generates one rule for each predictor in the data, then selects the rule with the smallest total error as its "one rule".  To create a rule for a predictor, we construct a frequency table for each predictor against the target. It has been shown that OneR produces rules only slightly less accurate than state-of-the-art classification algorithms while producing rules that are simple for humans to interpret

Week 10 lecture

Random Forest:

* Train multiple decision trees on random subsets of samples
* Decision via majority voting

Each tree is built on a random subset of records of the data: Tree bagging

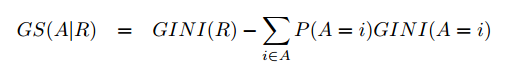
Each tree is built on a random subset of features of the data: Random subspace

Tutorial 11

Information Gain:



Gini Index:





Gain ratio:



Split information:



Week 11 lecture

Itemset: A collection of one or more items, Example: {Milk, Bread, Diaper}

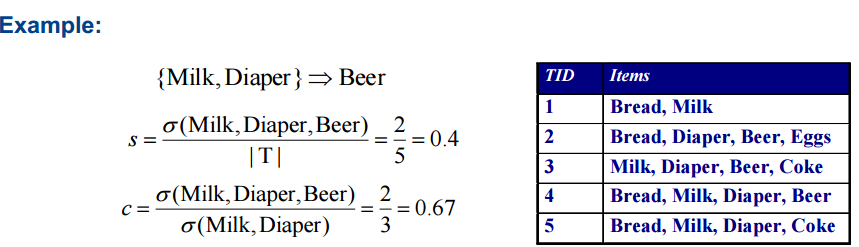
Support count: Frequency of occurrence of an itemset, e.g. ({Milk, Bread, Diaper}) = 2

Support(s): Fraction of transactions that contain an itemset, E.g. ({Milk, Bread, Diaper}) = 2/5

Frequent Itemset: An itemset whose support is greater than or equal to a minsup threshold

Confidence(c): Measures how often items in A appear in transactions that contain B

Frequent Itemset has a support greater than a given minsup support threshold.



(computing the support and confidence, there is no order limited in the itemset)

Association Rule:

An implication expression of the form A -> B, where A and B are itemsets

A: antecedent B: consequent

**The goal of association rule mining is to find all rules having – support ≥ minsup threshold – confidence ≥ minconf threshold**

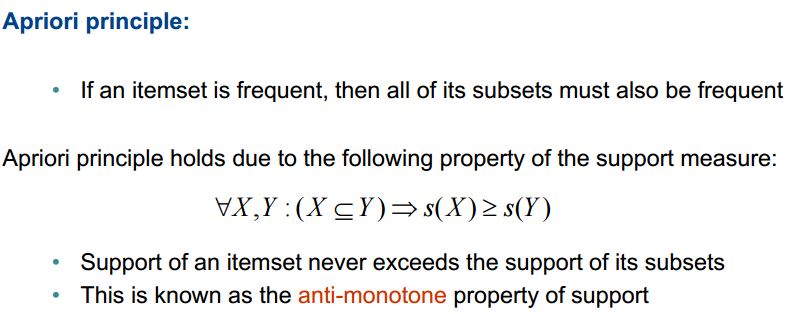
Association rule of Two-step approach:

Step 1: Frequent Itemset Generation Generate all itemsets whose support minsup

Step 2: Rule Generation Generate high confidence rules from each frequent itemset, where each rule is a binary partitioning of a frequent itemset

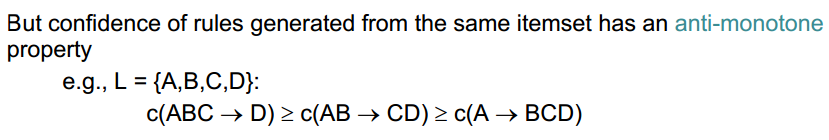
Frequent Itemset Generation Strategies:

* Reduce the number of candidates (M)
* Reduce the number of transactions (N)
* Reduce the number of comparisons (NM)

****

Rule Generation:

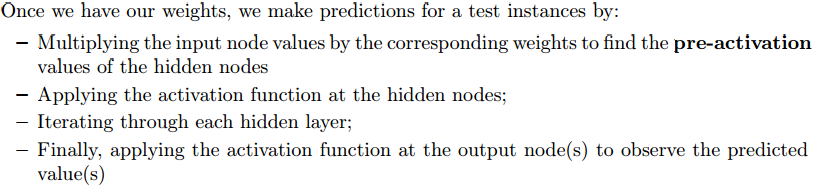
Given a frequent itemset L, If |L| = k, then there are 2k – 2 candidate association rules (ignoring L ->empty and empty-> L)



Association Rules are only applicable to nominal attributes

**What occurs in the training phase of a neural network? What occurs in the test phase?**

In the training phase, we attempt to determine the weights on the branches in the network, so as to minimise the calculated error function between our predicted values at the output node(s), and the actual (true) values, across the training data.

****