PAGE :
DATE : / /

Web Systems and Algorithms:

Unit 1:

- introduction
- basic element (content, data structure, algo)
- what application can be benefit from intelligence.
- eight fallacies (data is reliable, size of data doesn't matter, inference happens instantaneously, apply the same algo. everywhere.

Unit 2:

- searching
- indexing
- spider
- link analysis
- click analysis
- ranking
- precision, recall

Unit 3:

- recommendation engine
- collaborative (user, item)
- content based

Unit 4:

- clustering (Grouping the things together)
- K-means / K-medoids
- DBSCAN
- Hierarchical

PAGE :
DATE : / /

Unit 5:

- classification (placing thing where they belong)
- KNN
- Decision Tree
- Bayesian
- Back propagation

Unit 6:

- combining the classifiers
- don't tie your ship in a single rope
- don't tie your life in a single hope
- comparing multiple classifier on same data.
  Bagging and Bousing

Unit 7:

- semantic web
- motivation
- automation
- personalization
- IR
- Data Reuse (Linking of Data)
- RDF (Reuse, Distribution Framework)
- Web Ontology

Unit - 1

Intelligent Web

PAGE :
DATE : / /

1 Intelligent Web:

- implementing the art of intelligence in web application.

Eg: food ordering system

- machine ask → "What do you like to order today?"

- if every Wednesday you are ordering fish then in Wednesday;
  machine ask → "Would you like fish today?"

- human have learning capability then why don't machine.

- if the machine can't think, we can make them learn.

Eg: Social network offering fact checking, here machine can ensure whether your statements are consistent with your previous message.

Eg: Google Search Engine

- Here the machine take the advantage of interconnected documents to rank them.

Eg: Netflix (machine knows about the movie you enjoy)

- Amazon knows about your buying behaviour.

PAGE :
DATE : / /

Basic elements of Intelligent Applications:

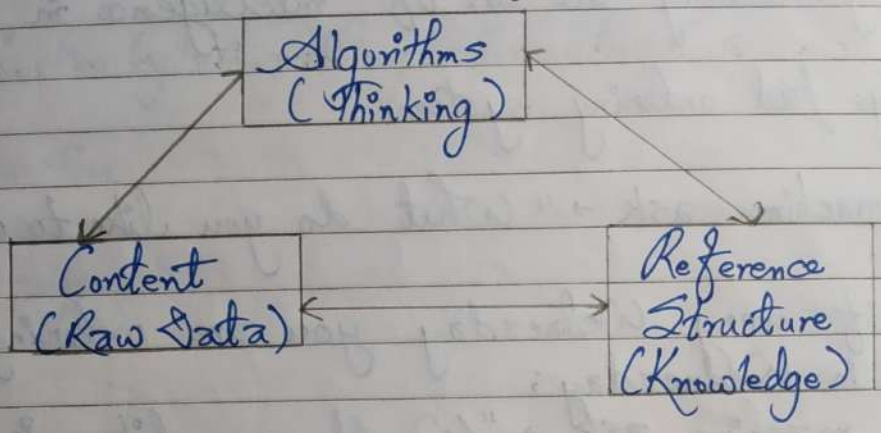


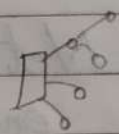
fig: Triangle of Intelligence.

Aggregated Content:

- dynamic rather than static
- large amount of data
- storage locations and origin could be geographically dispersed.

Reference Structure:

- provides semantic interpretation of the content.
- dictionaries, knowledge base, ontologies



Semantic web

Algorithm:

- modules that allows the application to extract the information.

- generalization, forecasting, prediction, improving interaction with users.

1. What applications can benefit from Intelligence?

- Social networking sites
- Job portal
- Aggregator (news)
- online games

1. How can I build intelligence in my own application?

1. Examine your functionality and your data:
   - does it deal with free text
   - does it deal with geographic data like map
   - is fraud detection an issue to your application
   - authentication
   - does your application make automated decision making
2. Get more data from the web:
   - Crawling
   - Parsing
   - Web Services API

PAGE :
DATE : / /

Eight Fallacies of Intelligent Application:

1. Your data is reliable:
   - data might contain outlier/noise.
   - missing values
   - data may change
   - different data types (numeric/non-numeric, discrete/continuous)
2. Inference happens instantaneously:
   - computing a solution takes time.
   - don't assume that an algorithm on all datasets run within the same response time.
3. The size of data doesn't matter:
   - response time
   - accuracy
4. Scalability of the solution isn't an issue:
   - parallel computing
   - Eg: not all clustering algorithm can run in parallel

5. Apply the same good library everywhere:

- when you are holding a hammer, everything looks like a nail.

Eg: Lucene Search Engine

6. The computation time is known:

- typically people expect that when we change the parameters of a problem, the problem can be solved consistently with respect to response time
- but it isn't true for all cases

Eg: distance between two locations.

7. Complicated Models are better

8. There are models without bias:

Eg: training and testing dataset preparation.

Unit-2

Searching (8 Hrs)

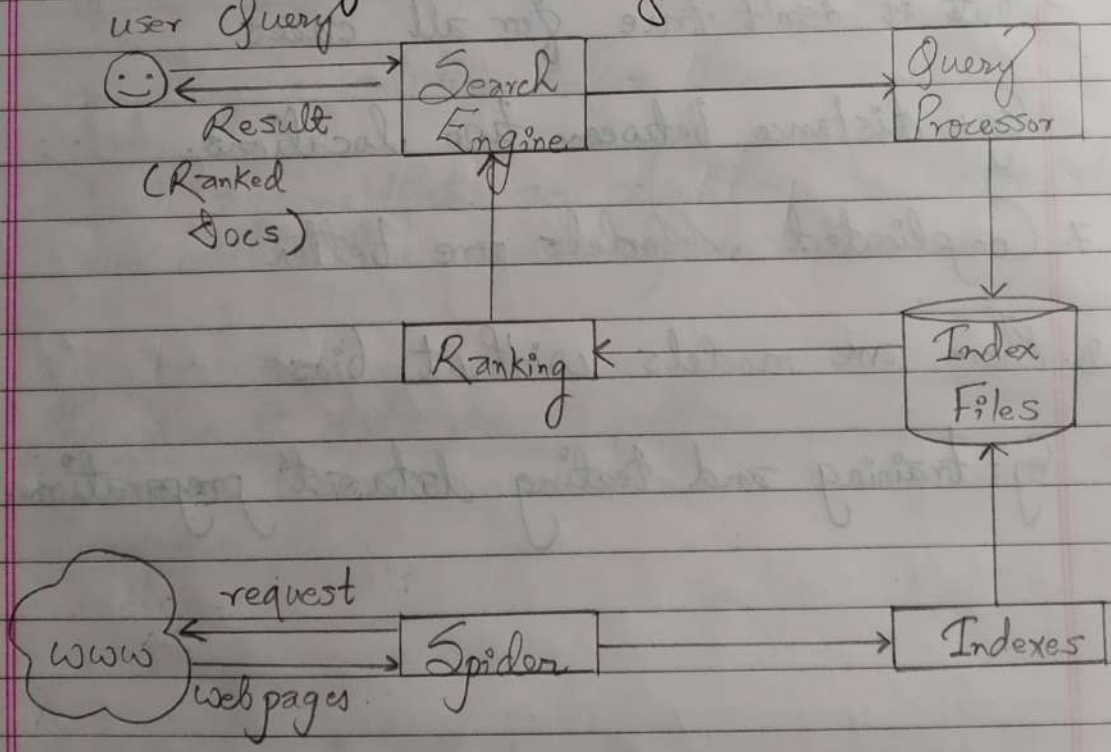
(8)

PAGE :
DATE : / /

1. Search Engine:

- a program that searches for document for specified keyword and returns a list of the documents where the keywords are found.
- a search engine works by sending out the spider to fetch the documents as many as
- indexer then reads these documents and create the index.

2. Architecture of Search Engine:



3. How does Search Engine work?

- Web Crawling
- Indexing
- Searching
- Ranking

PAGE :
DATE : / /

Web Crawling:

- is the process of gathering the pages from web.

Features of a Crawler:

(i) Robustness:

- Detecting the trap

(ii) Politeness:

- follows the restrictions to spider.

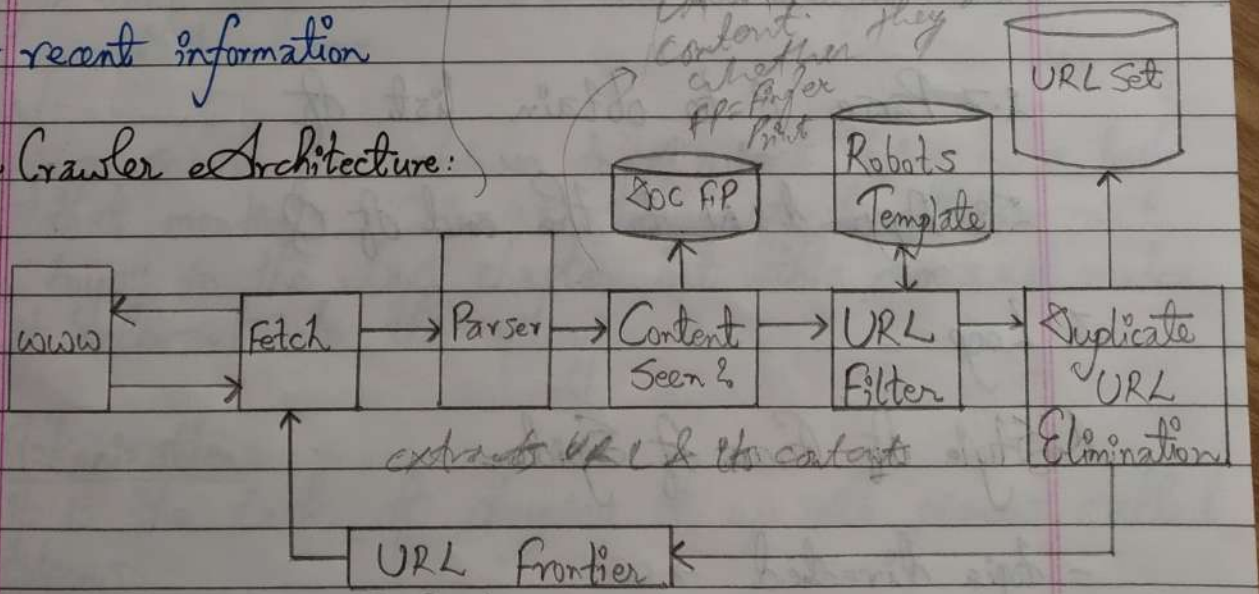
(iii) Quality:

- Distinguishes fake

(iv) Freshness:

- recent information

Crawler Architecture:



PAGE :
DATE : / /

1. Crawling Algorithm / Spidering Algorithm:

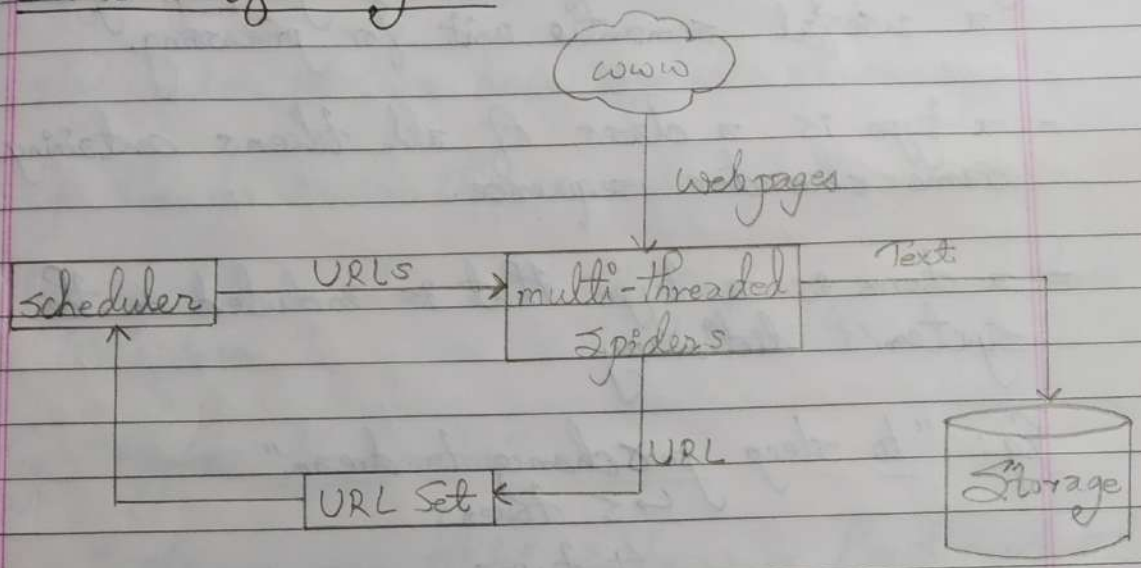
1. Initialize queue (Q) with initial set of known URLs.
2. Until Q is empty OR Page Limit exceeded OR time limit exceeded limit = 100 pages limit = 5
   - 2.1 Pop URL, L, from Q.
   - 2.2 If L is not to an HTML page (.gif, .pdf) then continue loop.
   - 2.3 If L is already visited then continue loop
   - 2.4 Download page P for L
   - 2.5 If cannot download P (eg. 404 Error, robots excluded, then continue loop.
   - 2.6 Index P
   - 2.7 Parse P to obtain list of
   - 2.8 Append N to the end of Q
3. Loop

2. Style of Focus of Spider

- topic directed → OS
- link directed

PAGE :
DATE : / /

Structure of a Spider:



Indexing:

- choosing a document unit
- determine what the document unit for indexing is?
- for very long documents, the issue of indexing granularity arises.
- eg a search for "chinese toys" might bring up a book that mentions "China" in the first chapter and "toys" in the last chapter but this does not make it relevant to the query.

Tokenization:

- it is the task of chopping it up into pieces called tokens.
- a token is an instance of characters in some

PAGE :
DATE : / /

particular document that are grouped together as a useful semantic unit for processing.

- a type is a class of all tokens containing the same character sequence.

- a term is a type that is included in the IR system's dictionary.

Eg: "to sleep perchance to dream"
↳ 5 tokens
4 types
3 terms ("to" as stop word)

- issues of tokenization are language specific.

Eg: chinese language has no space as word separation.

→ email id
→ URLs
→ IP Address
→ hyphen.

- but splitting on white spaces can cause bad retrieval result sometimes.

Eg: Mr. Tribhuvan Shrestha never goes to University.

PAGE :
DATE : / /

2. Dropping Common Terms:

- stop words are those having higher frequency.
- Eg: the, for, is, am, that, to, with, will etc.
- but a lot of time not indexing stop word does little harm

Eg: "Flights to London"

- "As we may think" ⇒ Vannevar Bush's Article.
- "Let it be" ⇒ Song of Beatles
- So instead of dropping the stop words, system has to cope with them

2. Normalization:

- There are cases, where tokens are not quite the same, but we still want to match them.

Eg: U.S.A, US, USA

window, windows, Window, WINDOW

- but sometimes it does more harm than good.

Eg: WHO → who

PAGE :
DATE : / /

2. Stemming and Lemmatization:

- reduce the size of vocabulary
- stemming is defined as heuristic process that chops off the words.
- Lemmatization refer to doing things properly with the use of vocabulary and morphological analysis of word.

Eg: am, are, is → be

car, cars, car's, cars' → car

2. Porter Algorithm:

- most common algorithm for stemming English terms with a common stem will usually have similar meanings.

Eg: connect, connected, connection, connecting.

- a consonant in a word is a letter other than A, E, I, O, U and other than Y preceded by a consonant.

Eg: "JOY" → the consonants are J and Y.

"SAZYGY". → the consonants are S, Z, G.

PAGE :
DATE : / /

- If a letter is not a consonant, it is a vowel.
- Let consonant be denoted by "c" and vowel by "v"
- any word has one of the following form:

  c... c
  c... v
  v... c
  v.... v

- rules in Porter Stemmer can is written as

  < condition > $S_1 \rightarrow S_2$

Eg: Rules	Example
1. SSES → SS	caresses → caress
2. IES → I	Ponies → Poni
3. SS → SS	caress → caress
4. S →	cars → car
5. (m>1) EMENT →	replacement → replac
excluding ement	but not cement → c
	m=2 (ement / cement)

- here "m" means the measure of word in the form of "VC"

  Eg: m=0 ⇒ TR, EE, TREE → no pair of VC
  m=1 ⇒ TROUBLE, OATS, TREES
  m=2 ⇒ TROUBLES, PRIVATE, OATEN

  2 pairs 2 pairs 2 pairs

PAGE :
DATE : / /

1. Building Inverted Index:

- also called postings.
- is a data structure storing a mapping from content

Eg: doc1 = new home sales
doc2 = home sales risk in July
doc3 = new home in July

home → doc1, doc2, doc3
July → doc2, doc3
new → doc1, doc3
risk → doc2
sales → doc1, doc2

2. Biword Index:

- index every consecutive pair of terms in the text as a phrase.

Eg: black panther attack
↓
black panther, panther attack

3. Positional Indexes:

- posting lists in a positional index in which each posting is a docID and a list of positions

Eg: Cat, 100 → no. of documents.
doc ID, no. of occurrence, list of positions

<1,6: <7,8,33,72,86,231>;
2,5: <1,17,74,222,255>;
4,2: <8,16>;
⋮
⋮
>

PAGE :
DATE : / /

2. Document Processing:

1. Lexical Analysis
2. Elimination of stop words
3. Stemming
4. Selection of Index
5. Thesaurus

Lexical Analysis of Text:

- basically space is involved as word separator, but however the following cases also have to consider followings:

1. **Digits:**

- Numbers are usually not good terms because without surrounding text, they are inherently vague
- Eg. query → "number of deaths between 1910 and 1984" might retrieve document which might contain the numbers only.
- but numbers like credit card number, special date might be index term.

DATE : / /

2. Hyphens:

- state-of-the-art, co-education, ... B-49

3. Punctuation Marks:

- normally punctuation marks are removed entirely in the process of lexical analysis while some punctuation marks are integral part of the word.

Eg: Dr., B.C.

4. Case of Letters:

- Bank and bank, WHO & who.

Thesaurus

- is a collection of words with its synonyms and related words.

Eg: Doctor → doc, medicine, surgeon, physician, anesthesists hospital.

1. Query Language:

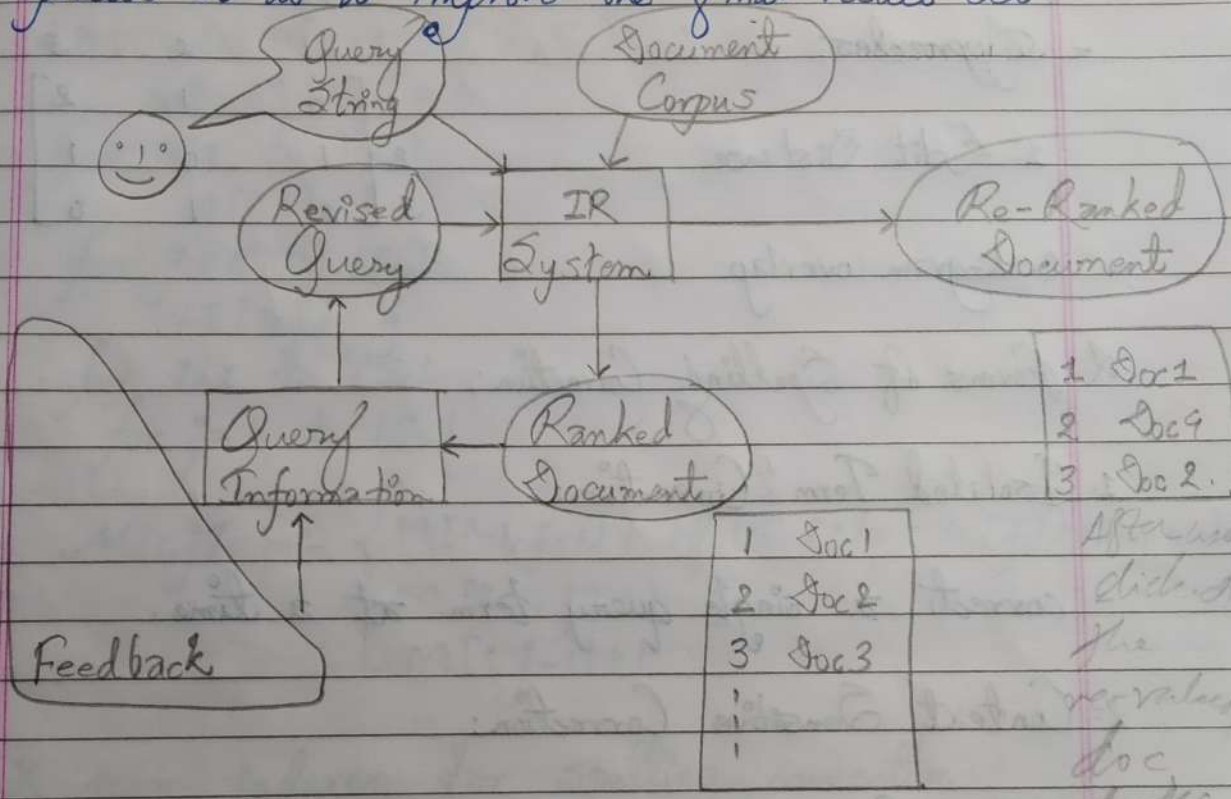
- a query is the formulation of a user information need.

- Single word queries.
- boolean queries
- context queries

PAGE :
DATE : / /

Relevance Feedback and Query Expansion:

- The idea is to involve the user in the retrieval process so as to improve the final result set.



PAGE :
DATE : / /

Spelling Correction:

- correcting spelling errors in queries (S1) (S2)
  Eg: cat car
- Approaches:
  1. Edit Distance

	1	2	3
1	0	1	2
2	1	0	1
3	2	1	0

  2. k-gram overlap

Forms of Spelling Correction:

1. Isolated Term Correction:

- correct a single query term at a time.

2. Context Sensitive Correction:

- ~~neig~~ near the neighbour words.

Edit Distance:

- given two character strings S_1 and S_2, the edit distance between them is the minimum number of edit operations required to transform S_1 into S_2.
- Most commonly edit operations are:
  1. Inserting a character
  2. Deleting a character

3. Replacing a character

Algorithm:

```
EDIT DISTANCE (S1, S2)
    int M[i,j] = 0
for i=1 to |S1|
    M[i,0] = i
for j=1 to |S2|
    M[0,j] = j
for i=1 to |S1|
  for j=1 to |S2|
```

$$M[i,j] = \min \begin{cases} M[i-1, j-1] + \text{if } (S_1[i] == S_2[j]) \text{ then 0 else 1,} \\ M[i-1, j] + 1, \\ M[i, j-1] + 1 \end{cases}$$

K-gram indexes for spelling correction:

- a K-gram is a sequence of k characters

Eg: in term "castle" ⇒ "cas", "ast", "stl", "tle" are 3 gram

- use the k-gram index to retrieve vocabulary terms that have many k-grams in common with the query.

Eg: bo → aboard, about, boardroom, border
or → border, lord, morbid, sordid
rd → aboard, ardent, boardroom, border

Matching 2-gram in the query "bord"

PAGE :
DATE : / /

2. Weighting Mechanism for the term:

1. Term Frequency: (TF)

- how often a term is found in a document
- denoted by $tf_{t,d}$

2. IDF (Inverse Document Frequency)

- terms which appear very few in numbers
- may have higher probability of being relevant.

$$IDF_t = \log \frac{N}{df_t}$$

$df \propto \frac{1}{\text{priority}}$

Eg:

Terms	df_t	idf_t
Computer	1054	0.152
Monitor	508	0.470
Keyboard	475	0.500
Device	1247	0.08
Optical	1500	0

automatically comes in every doc & less priority

Here, N = total no. of document = 1500

3. IF-IDF

$$If\text{-}IDF_{t,d} = IF_{t,d} \times IDf_t$$

d1 → cat, cat, dog

$If_{cat,d1} = 2$

$If_{dog,d1} = 1$

$If_{tiger,d1} = 0$

PAGE :
DATE : / /

1. Cosine Similarity:

- finding the similarity between the document.

$$\text{cossim}(\vec{A}, \vec{B}) = \frac{\vec{A} \cdot \vec{B}}{|\vec{A}||\vec{B}|}$$

$\cos 0 = 1$

$\cos 90 = 0$

Eg: Doc1 = you say goodbye
Doc2 = hello goodbye hello goodbye hello
Doc3 = I say hello

query = hello goodbye

Find unique words: you, say, goodbye, hello, I
Alphabetical order: goodbye, hello, I, say, you

doc 1 <1, 0, 0, 1, 1>
doc 2 <2, 3, 0, 0, 0>
doc 3 <0, 1, 1, 1, 0>
query <1, 1, 0, 0, 0>

$$\text{cossim}(\text{query}, \text{doc1}) = \frac{1\times1 + 1\times0 + 0\times0 + 0\times1 + 0\times1}{\sqrt{1^2+1^2} \cdot \sqrt{1^2+1^2+1^2}} = \frac{1}{\sqrt{6}} = 0.41$$

$$\text{cossim}(\text{query}, \text{doc2}) = \frac{1\times2 + 1\times3}{\sqrt{2} \cdot \sqrt{2^2+3^2}} = \frac{5}{\sqrt{26}} = 0.98$$

$$\text{cossim}(\text{query}, \text{doc3}) = \frac{1\times1}{\sqrt{2} \cdot \sqrt{3}} = \frac{1}{\sqrt{6}} = 0.41$$

Ans: Doc2
Doc1
Doc3

PAGE :
DATE : / /

Consider the following documents:

Doc 1 = Information Retrieval System

Doc 2 = Information Storage

Doc 3 = Digital Speech Synthesis System

Doc 4 = Speech Filtering Speech Retrieval

Rank the above document for the Query "Speech System"

Solution: Unique words: Information, Retrieval, System, Storage, Digital, Speech, Synthesis, Filtering
8

Alphabetical order: Digital, Filtering, Information, Retrieval, Speech, Storage, Synthesis, System

doc 1 = < 0, 0, 1, 1, 0, 0, 0, 1 >
doc 2 = < 0, 0, 1, 0, 0, 1, 0, 0 >
doc 3 = < 1, 0, 0, 0, 1, 0, 1, 1 >
doc 4 = < 0, 1, 0, 1, 2, 0, 0, 0 >
query = < 0, 0, 0, 0, 1, 0, 0, 1 >

$$\text{cosim}(\text{query}, \text{doc 1}) = \frac{0\times0+0\times0+0\times1+0\times1+2\times0+0\times0+0\times0+0\times1}{\sqrt{1^2+1^2}\cdot\sqrt{1^2+1^2+1^2}}$$

$$= \frac{1}{\sqrt{2}\cdot\sqrt{3}}$$

$$= \frac{1}{\sqrt{6}}$$

PAGE :
DATE : / /

$$\text{cossim}(\text{query}, \text{doc2}) = \frac{0\times0+0\times0+0\times1+0\times0+1\times0+0\times1+0\times0+1\times0}{\sqrt{2}\cdot\sqrt{2}}$$

$$= \frac{0}{\sqrt{4}}$$

$$= 0$$

$$\text{cossim}(\text{query}, \text{doc3}) = \frac{0\times1+0\times0+0\times0+0\times0+1\times1+0\times0+0\times1+1\times1}{\sqrt{2}\cdot\sqrt{4}}$$

$$= \frac{2}{\sqrt{8}}$$

$$= \frac{2}{2\sqrt{2}}$$

$$= \frac{1}{\sqrt{2}}$$

$$\text{cossim}(\text{query}, \text{doc4}) = \frac{0\times0+0\times1+0\times0+0\times1+1\times2+0\times0+0\times0+1\times0}{\sqrt{2}\cdot\sqrt{2^2+1^2+2^2}}$$

$$= \frac{2}{\sqrt{2}\cdot\sqrt{9}} = \frac{2}{\sqrt{18}} = \frac{2}{2\sqrt{3}}$$

$$= \frac{1}{\sqrt{3}}$$

PAGE :
DATE : / /

1. Link Analysis:

- use of hyperlinks for ranking web search results.
- link analysis is one of many factors considered by web search engines in computing a score for a web page on any given query.
- two methods:
  1. Page Rank
  2. HITS (Hyperlink Induced Topic Search)

1. Page Rank:

- developed by Larry Page at SUN
- Link Analysis Algorithm.
- a hyperlink to a page counts as a vote of support
- a page that is linked to by many pages receives a high rank and if there is no links to a web page, there is no support for that page.
- a page rank of 0.5 means there is a 50% chance that a person clicking on a random link will be directed to the document with 0.5 page rank.

PAGE :
DATE : / /

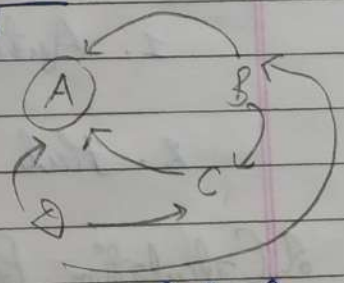
Algorithm:

- Assume a small universe of four web pages A, B, C and D
- the initial approximation of page rank would be evenly divided between four documents i.e. $1/4 = 0.25$
- if pages B, C and D each only link to A, they would confer 0.25 page rank to A.

i.e. $PR(A) = \frac{PR(B)}{1} + \frac{PR(C)}{1} + \frac{PR(D)}{1} = 0.75$

- Suppose that page B has link to page C as well as to page A, which D has links to all three pages

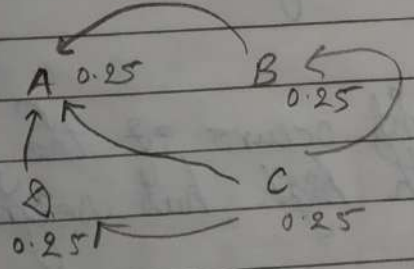
$$PR(A) = \frac{PR(B)}{2} + \frac{PR(C)}{1} + \frac{PR(D)}{3}$$



i.e. in general

$$PR(m) = \sum_{n \in B_m} \frac{PR(n)}{L(n)}$$

where, $B_m \rightarrow$ set of all pages link to m
$L(n) \rightarrow$ outgoing links from n.



PAGE :
DATE : / /

Authorities and Hubs:

- A page is called an authority for the query if it contains the valuable information on the subject
- eg. for a query car ⇒ "www.bmw.com"
- authoritative pages are truly relevant to the query.
- hubs contains useful links towards the authoritative pages.
- i.e. hubs point the search engine to the right decision.
- HITS algorithm is based on two parameter.
  1. Authority Weight
  2. Hub Weight

Calculation Process in HITS algorithm:

- authority weight and hub weight are calculated recursively.
- the higher authority weight occurs if the page is pointed to by pages with high hub weights.
- a higher hub weight occurs if the page points to many pages with high authority weights.

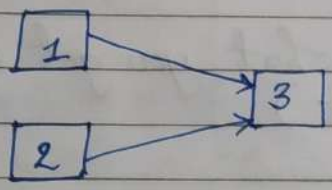
PAGE :
DATE : / /

- for a web page p,

$$h(p) = \sum_{p \to y} a(y)$$
$$a(p) = \sum_{y \to p} h(y)$$

} where $m \to n$ denotes the existence of hyperlink from m to n.

Eg:



- $a = A^T h$

- $h = Aa$

when A is adjacency matrix.

$A_{ij} = \begin{cases} 1, & \text{if there is a hyperlink from } i \text{ to } j \\ 0, & \text{otherwise} \end{cases}$

- in this case,

$$\begin{array}{c} \\ 1 \\ 2 \\ 3 \end{array} \begin{array}{c} \begin{array}{ccc} 1 & 2 & 3 \end{array} \\ \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \end{array}$$

$a = A^T h$

$$= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \quad \leftarrow \text{initialization of } h \text{ vector}$$

$$= \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix}$$

PAGE :
DATE : / /

$$h = Aa$$

$$= \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 2 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 0 \end{bmatrix}$$

2. Evaluation Criteria: (Is what you got what you want)

Precision:

- fraction of retrieved documents that are relevant.

$$\text{Precision} = \frac{\#(\text{relevant item retrieved})}{\#(\text{retrieved item})}$$

(# = no. of)

Recall:

- is the fraction of relevant documents that are retrieved.

$$\text{Recall} = \frac{\#(\text{relevant items retrieved})}{\#(\text{relevant items})}$$

F-Measure:

- harmonic mean of precision and recall.

$$\text{F-Measure} = \frac{2PR}{P+R}.$$

PAGE :
DATE : / /

Eg: An IR System returns 8 relevant documents and 10 non-relevant documents. There are 20 relevant documents in the collection. Calculate the precision, recall and F-measure.

8 → R] 8+10
10 → NR]
R = 20
Precision = 8/18
Recall = 8/20

Recall → 100% retrieved.

F-measure = $\frac{P+R}{2}$ 100%

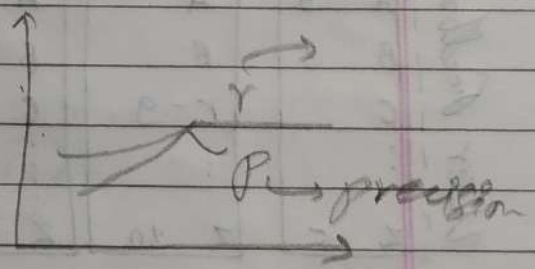
750

In case of Ranked Documents:
# of relevant documents = 6

Rank	docID	Relevance	Precision	Recall
1	100	✓	1/1	1/6
2	103	✓	2/2	2/6
3	104	x	2/3	2/6
4	106	✓	3/4	3/6
5	105	x	3/5	3/6

Assignment:

Click Analysis



- Scope and Applications
- Interest Mining
- Disambiguity bat → mammal
  → cricket bat
- Rating (Relevance Feedback)

Unit-3

32

PAGE :
DATE : / /

1. Recommender System:

- a specific type of information filtering system, techniques that attempts to recommend information, items that are likely to be interest to the user.

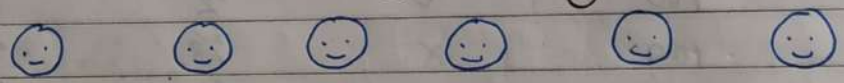
2. Approaches:

1. Collaborative Filtering
2. Content Based Recommendation

1. Collaborative Filtering:

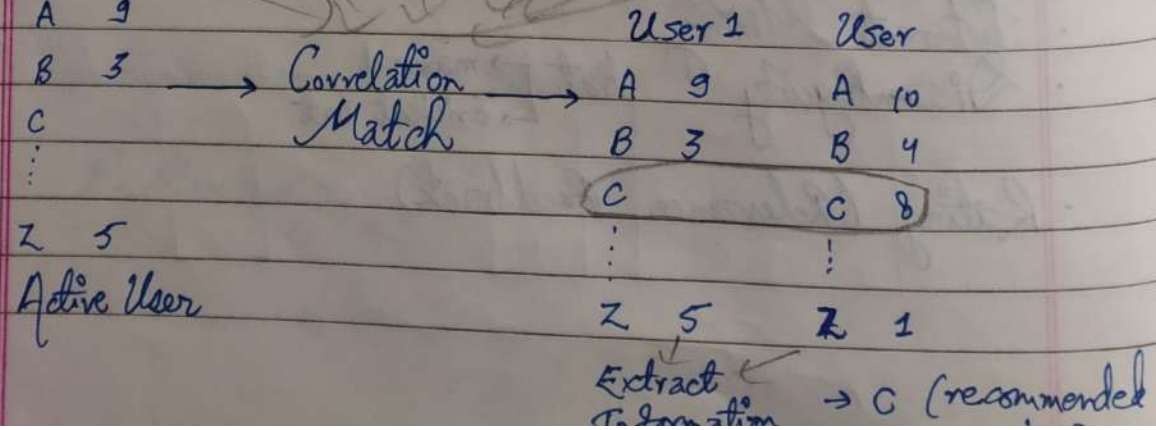
- Tell me your friends tell I will tell you who you are.

Collaborative Filtering Working Mechanism:



User feedback

user 1	user 2	user 3	user 4	user 5	user 6
A 9	A	A	A	A 6	A 10
B 3	B	B	B	B 4	B 4
C	C 9	C	C 8	C	C 8
⋮	⋮	⋮	⋮	⋮	⋮
Z 5	Z 10	Z 7	Z	Z	Z 1



- Typically, Pearson correlation coefficient is used between active user and other user.

$$C_{a,u} = \frac{covar(r_a, r_u)}{\sigma_{r_a} \sigma_{r_u}}$$

where,

$$covar(r_a, r_u) = \frac{\sum_{i=1}^{m} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{m}$$

Covariance

$$r_x = \frac{\sum_{i=1}^{m} r_{x,i}}{m}$$

$$\sigma_{r_a} = \sqrt{\frac{\sum_{i=1}^{m} (r_{x,i} - \bar{r}_x)^2}{m}}$$

$m \rightarrow$ no. of commonly related items.

Significance Weighting:

$$W_{a,u} = S_{a,u} C_{a,u} \text{ where, } S_{a,u} = \begin{cases} 1 & \text{if } m > 50 \\ 0 & \text{otherwise} \end{cases}$$

Rating Prediction:

$$P_{a,i} = r_a + \frac{\sum_{u=1}^{n} W_{a,u} (r_{u,i} - \bar{r}_u)}{\sum_{u=1}^{n} W_{a,u}}$$

Limitations:

- Cold Start
- First Rater
- Popularity Bias (unique choice)

PAGE :
DATE : / /

1. Consider the following rating matrix:

Users \ Items	I_1	I_2	I_3	I_4
U_1	4	?	5	5
U_2	4	2	1	X
U_3	3	X	2	4
U_4	4	4	X	X
U_5	2	1	3	5

Q. Calculate the prediction value of item I_2 for User U_4

Here,

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot \omega_{a,u}}{\sum_{u \in U} |\omega_{a,u}|}$$

where,

$$\omega_{a,u} = \frac{\sum_{i \in m} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i=m} (r_{a,i} - \bar{r}_a)^2} \cdot \sqrt{\sum_{i \in m} (r_{u,i} - \bar{r}_u)^2}}$$

items that is rated by both users.

$i \in m = (1,3)$ for initial case

$\omega_{a,u}$: active user, this user

U = other users who have ranked I_2.

m commonly rated by a and u.

$\omega_{a,u}$ = similarity between two users a and v.

We have, $r_1 = (4+5+5)/3 = 4.67$

$r_2 = (4+2+1)/3 = 2.33$

$r_3 = (3+2+4)/3 = 3$

$r_4 = (4+4)/2 = 4$

$r_5 = (2+1+3+5)/4 = 2.75$

PAGE :
DATE : / /

Calculate co-rated items:

(users) (items)

$\omega_{1,2}$	1, 3
$\omega_{1,3}$	1, 3, 4 (∵ no need to compute as there is no
$\omega_{1,4}$	1 rating on Item I_2)
$\omega_{1,5}$	1, 3, 4

$i = 1, 3$ (items)

$$\omega_{1,2} = \frac{(r_{1,1} - \bar{r}_1)(r_{2,1} - \bar{r}_2) + (r_{1,3} - \bar{r}_1)(r_{2,3} - \bar{r}_2)}{\sqrt{(r_{1,1} - \bar{r}_1)^2 + (r_{1,3} - \bar{r}_1)^2} \cdot \sqrt{(r_{2,1} - \bar{r}_2)^2 + (r_{2,3} - \bar{r}_2)^2}}$$

$\omega_{2,4}$

Here,

$$\bar{r}_1 = \frac{4+5}{2} = 4.5 \qquad \bar{r}_2 = \frac{4+1}{2} = 2.5$$

$$\omega_{1,2} = \frac{(4-4.5)(4-2.5) + (5-4.5)(1-2.5)}{\sqrt{(4-4.5)^2 + (5-4.5)^2} \cdot \sqrt{(4-2.5)^2 + (1-2.5)^2}}$$

$$= \frac{-1.5}{0.707 \times 2.1213}$$

$$= -1.0002.$$

$$\omega_{1,3} = \frac{(r_{1,1} - \bar{r}_1) \cdot (r_{3,1} - \bar{r}_3) + (r_{1,3} - \bar{r}_1) \cdot (r_{3,3} - \bar{r}_3) + (r_{1,4} - \bar{r}_1)(r_{3,4} - \bar{r}_3)}{\sqrt{(r_{1,1} - \bar{r}_1)^2 + (r_{1,3} - \bar{r}_1)^2 + (r_{1,4} - \bar{r}_1)^2} \cdot \sqrt{(r_{3,1} - \bar{r}_3)^2 + (r_{3,3} - \bar{r}_3)^2 + (r_{3,4} - \bar{r}_3)^2}}$$

Here, $\bar{r}_1 = \frac{4+5+5}{3} = 4.67 \qquad \bar{r}_3 = \frac{3+2+4}{3} = 3.$

$$\omega_{1,3} = \frac{(4-4.67)(3-3) + (5-4.67) \cdot (2-3) + (5-4.67) \cdot (4-3)}{\sqrt{(4-4.67)^2 + (5-4.67)^2 + (5-4.67)^2} \cdot \sqrt{(3-3)^2 + (2-3)^2 + (4-3)^2}}$$

$$= 0$$

$$\omega_{1,4} = \frac{(r_{1,1} - \bar{r}_1) \cdot (r_{4,1} - \bar{r}_4)}{\sqrt{(r_{1,1} - \bar{r}_1)^2} \cdot \sqrt{(r_{4,1} - \bar{r}_4)^2}}$$

Here,

$\bar{r}_1 = 4$ $\quad$ $\bar{r}_4 = 4$

$$\omega_{1,4} = \frac{(4-4)\cdot(4-4)}{\sqrt{(4-4)^2} \cdot \sqrt{(4-4)^2}}$$

$$= 0$$

$$\omega_{1,5} = \frac{(r_{1,1} - \bar{r}_1)(r_{5,1} - \bar{r}_5) + (r_{1,3} - \bar{r}_1)\cdot(r_{5,3} - \bar{r}_5) + (r_{1,4} - \bar{r}_1)(r_{5,4} - \bar{r}_5)}{\sqrt{(r_{1,1} - \bar{r}_1)^2 + (r_{1,3} - \bar{r}_1)^2 + (r_{1,4} - \bar{r}_1)^2} \cdot \sqrt{(r_{5,1} - \bar{r}_5)^2 + (r_{5,3} - \bar{r}_5)^2 + (r_{5,4} - \bar{r}_5)^2}}$$

Here,

$\bar{r}_1 = \frac{4+5+5}{3} = 4.67$ $\quad$ $\bar{r}_5 = \frac{2+3+5}{3} = 3.33$

$$\omega_{15} = \frac{(4-4.67)\cdot(2-3.33) + (5-4.67)\cdot(3-3.33) + (5-4.67)(5-3.33)}{\sqrt{(4-4.67)^2 + (5-4.67)^2 + (5-4.67)^2} \cdot \sqrt{(2-3.33)^2 + (3-3.33)^2 + (5-3.33)^2}}$$

$$= \frac{1.333}{0.8167 \times 2.1603}$$

$$= 0.7555$$

Now, $P_{a,i} = \bar{r}_a + \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_u) \cdot \omega_{a,u}}{\sum_{u \in U} |\omega_{a,u}|}$ $\quad$ other users $\rightarrow u \in U = \{2,3,4,5\}$

$$P_{1,2} = \bar{r}_1 + \frac{(r_{2,2} - \bar{r}_2)\cdot\omega_{1,2} + (r_{3,2} - \bar{r}_3)\cdot\omega_{1,3} + (r_{4,2} - \bar{r}_4)\cdot\omega_{1,4} + (r_{5,2} - \bar{r}_5)\cdot\omega_{1,5}}{|\omega_{1,2}| + |\omega_{1,3}| + |\omega_{1,4}| + |\omega_{1,5}|}$$

simplified

Here, $\bar{r}_2 = \frac{4+1}{2} = 2.5$ $\quad$ $\bar{r}_3$ = Not required since U_3 has not rated I_2.

$\bar{r}_4 = 4$

$\bar{r}_1 = 4.67 = \frac{4+5+5}{3}$

PAGE :
DATE : / /

$$P_{1,2} = 4.67 + \frac{(2-2.5)\cdot(-1.0002) + (4-4)\cdot 0 + (1-3.33)\cdot 0.7555}{|-1.0002| + |0| + |0.7555|}$$

$$= 4.67 + \left(\frac{-1.2602}{1.7557}\right)$$

$$= 3.9522$$

Content Based Recommendation System:

- This approach analyze a set of documents previously liked by a user, and build a model of user, and build a model of interest based on the features of the objects rated or liked by the user.
- Worked on three steps:

1. Content Analyzer
2. Profile Learner
3. Filtering Component

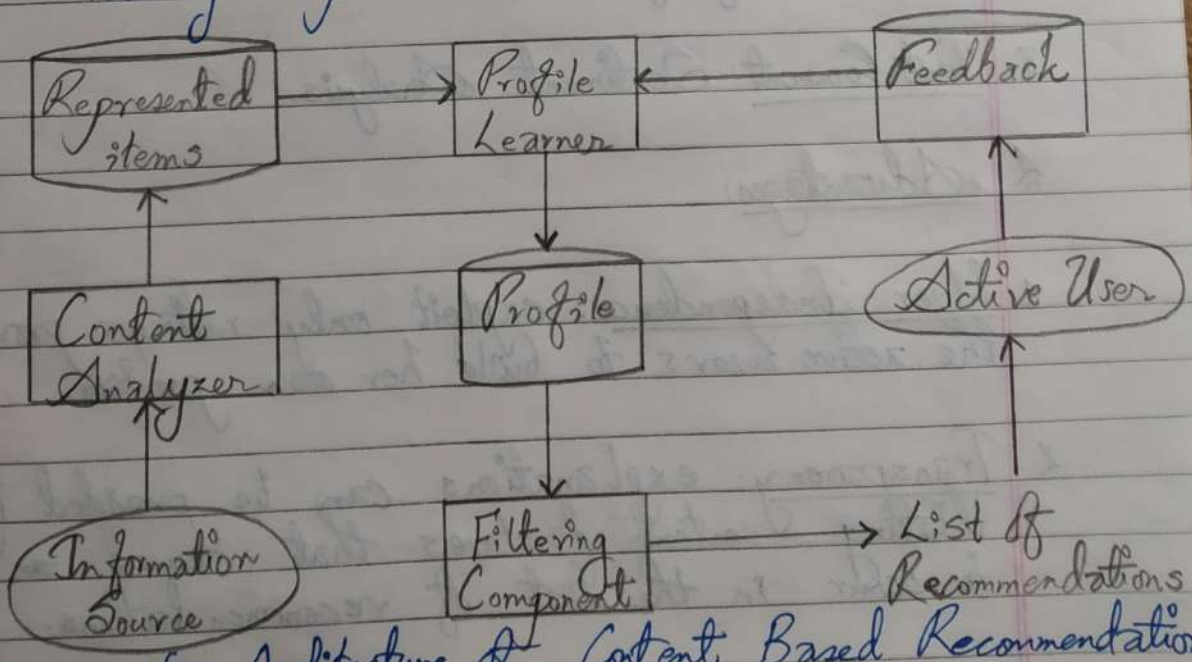


fig: Architecture of Content Based Recommendation

Content Analyzer: preprocessing to extract relevant information.

Profile Learner: collects data of user preferences and tries to generalize the data.

- like/dislike past
- +ve/-ve feedback

Filtering Component: suggest relevant items.

2. Approaches to get explicit relevance feedback.

1. Like/Dislike: items are classified as relevant or not relevant by adopting a simple binary rating scale. (1-5)

2. Ratings: a discrete numeric scale is usually adopted to judge items.

3. Text Comment: Sentiment Analysis.

2. Advantages:

1. User Independence: exploit only ratings provided by the active users to build her own profile.

2. Transparency: explanations can be provided by explicitly listing content features that caused an item to occur in the list of recommendations.

PAGE :
DATE : / /

3. New Item: capable of recommending items not yet rated by user.

Limitations:

1. Limited Content Analysis:

- Domain Knowledge is often needed. eg. while movie recommendations, the system needs to know about director, actor etc.

2. Over Specialization:

- Sometimes called serendipity.

3. New User:

- enough ratings have to be collected.

upto here

fill

fill

Pera

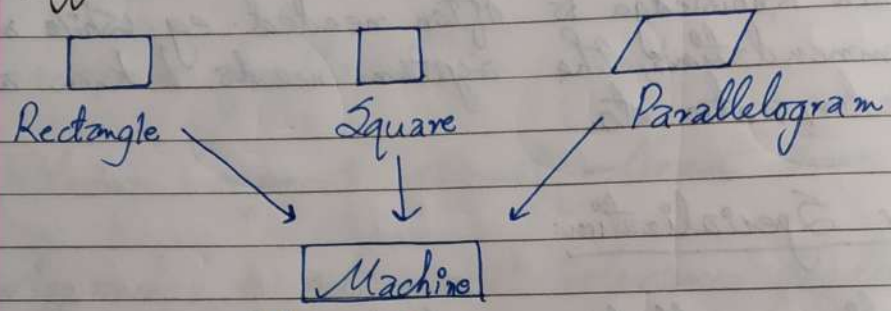
Unit: 4

PAGE :
DATE : / /

Clustering:

- grouping the things together.
- unsupervised learning.

Approach:



They are all bounded with 4 st. lines so they are rectangular.

1. Algorithm:

- K-means
- K-medoid
- Hierarchical Clustering → Agglomerative / Divisive
- DBSCAN
- ROCK → link based clustering algorithm.

PAGE :
DATE : / /

1. K-means Algorithm:

Input: Datasize (D), no. of clusters (K).

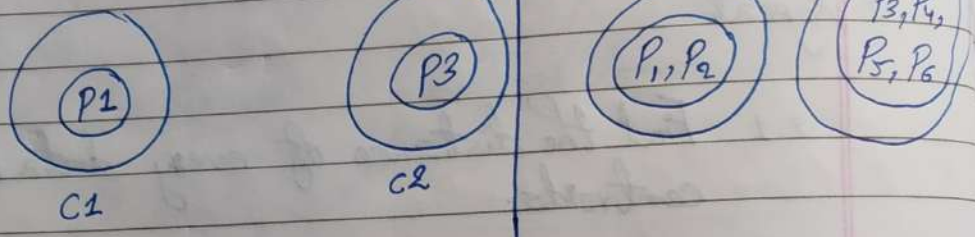
Output: K-clusters

1. Select the k sample randomly as initial centroid.
2. Repeat
   - 2.1 Find the distance of every data objects with the centroids.
   - 2.2. Assign the object to the closest cluster.
   - 2.3. Update the centroid
3. Until
   - 3.1 No change in centroid
   - 3.2 No change in object of clusters.

PAGE :
DATE : / /

Eg: $P_1(x_1, y_1)$
$P_2(x_2, y_2)$
$P_3(x_3, y_3)$
$P_4(x_4, y_4)$
$P_5(x_5, y_5)$
$P_6(x_6, y_6)$

$k=2$



1st Iteration:

$P_1 \begin{cases} c_1 \\ c_2 \end{cases} \begin{cases} q - d_{11} \\ \rightarrow d_{21} \end{cases} \min \; P_1 \in C_1$

$P_2 \begin{cases} c_1 \\ c_2 \end{cases} \begin{cases} d_{12} \\ d_{22} \end{cases} \min$

P_3 ⋮

P_4

P_5

P_6

IInd Iteration:

$P_1 \begin{cases} c_1 & -\frac{P_1 \cdot x + P_2 \cdot x}{2} \\ c_2 & \frac{P_1 \cdot y + P_2 \cdot y}{2} \end{cases}$

⋮

$P_6 \begin{cases} c_1 \\ c_2 \end{cases}$

PAGE :
DATE : / /

Q. Apply K-means (K=2) to make clusters from following documents:

Doc 1 = go Longhorns go

Doc 2 = go Texas

Doc 3 = Texas Longhorns

Doc 4 = Longhorns Longhorns

- Assume Doc 1 and Doc 3 as initial centroid and perform the computation upto 2-iterations.

Solution: Unique words: go, Longhorns, Texas:

Sorting in Ascending Order we get: go, Longhorns, Texas

Now, let's construct the vector as:

doc 1 = <2, 1, 0>

doc 2 = <1, 0, 1>

doc 3 = <0, 1, 1>

doc 4 = <0, 2, 0>

Since doc 1 and doc 3 are considered as initial centroid so, we classify them to cluster 1 and cluster 2 respectively and calculate the cosine distance to include the documents into the probable clusters.

We have vector as:

doc 1 = <1, 0, 0>
doc 2 = <1, 0, 1>
doc 3 = <0, 1, 1>
doc 4 = <0, 1, 0>

1st Iteration:

doc1 — c_1 – doc1 = 1
doc1 — c_2 – doc3 = 0

doc2 — c_1 – doc1 = 0.75
doc2 — c_2 – doc3 = 0.5

doc3 — c_1 – doc1 = 0
doc3 — c_2 – doc3 = 1

doc4 — c_1 – doc1
doc4 — c_2 – doc3

C_1 = Doc1 and C_2 = Doc3.

Now, cosine(doc1, c1) = $\frac{1\times1+0\times0+0\times0}{\sqrt{1^2}\cdot\sqrt{1^2}}$

= 1

cosine(doc2, c1) = $\frac{1+0+0}{\sqrt{1^2+0^2+1^2}\cdot\sqrt{1^2+1^2+0^2}}$

= 0.63

cosine(doc3, c1) = $\frac{0+1+0}{\sqrt{0^2+1^2+1^2}\cdot\sqrt{1^2+1^2+0^2}}$

= 0.31

PAGE :
DATE : / /

$$\text{cosine}(doc4, c1) = \frac{0+2+0}{\sqrt{0^2+2^2+0^2} \cdot \sqrt{2^2+1^2+0^2}}$$

$$= 0.44$$

$$\text{cosine}(doc1, c2) = \frac{0+1+0}{\sqrt{2^2+1^2+0^2} \cdot \sqrt{0^2+1^2+1^2}}$$

$$= 0.31$$

$$\text{cosine}(doc2, c2) = \frac{0+0+1}{\sqrt{1^2+0^2+1^2} \cdot \sqrt{0^2+1^2+1^2}}$$

$$= 0.5$$

$$=$$

$$\text{cosine}(doc3, c2) = 1$$

$$\text{cosine}(doc4, c2) = \frac{0+2+0}{\sqrt{0^2+2^2+0^2} \cdot \sqrt{0^2+1^2+1^2}} = 0.70$$

$c_1 = doc1, doc2.$
$c_2 = doc3, doc4.$

Iteration 2:

$$c_1 = \left(\frac{2+1}{2}, \frac{1+0}{2}, \frac{0+1}{2}\right) = (1.5, 0.5, 0.5)$$

$$c_2 = \left(\frac{0+0}{2}, \frac{1+2}{2}, \frac{1+0}{2}\right) = (0, 1.5, 0.5)$$

$$\text{cosine}(doc1, c1) = \frac{3+0.5+0}{\sqrt{2^2+1^2+0^2} \cdot \sqrt{(1.5)^2+(0.5)^2+(0.5)^2}}$$

$$= 0.94$$

$$\text{cosine}(doc2, c1) = \frac{1.5+0+0.5}{\sqrt{1^2+0^2+1^2} \cdot \sqrt{(1.5)^2+(0.5)^2+(0.5)^2}}$$

$$\text{cosine}(doc3, c1) = \frac{0+0.5+0.5}{\sqrt{0^2+1^2+1^2} \cdot \sqrt{(1.5)^2+(0.5)^2+(0.5)^2}}$$

$$= 0.42$$

$$\text{cosine}(doc4, c1) = \frac{0+1+0}{\sqrt{0^2+2^2+0^2} \cdot \sqrt{(1.5)^2+(0.5)^2+(0.5)^2}}$$

$$= 0.30$$

$$\text{cosine}(doc1, c2) = \frac{0+1.5+0}{\sqrt{2^2+1^2+0^2} \cdot \sqrt{0^2+(1.5)^2+(0.5)^2}}$$

$$= 0.42$$

$$\text{cosine}(doc2, c2) = \frac{0+0+0.5}{\sqrt{1^2+0^2+1^2} \cdot \sqrt{0^2+(1.5)^2+(0.5)^2}}$$

$$= 0.44$$

$$\text{cosine}(doc3, c2) = \frac{0+1.5+0.5}{\sqrt{0^2+1^2+1^2} \cdot \sqrt{(0)^2+(1.5)^2+(0.5)^2}}$$

$$= 0.89$$

$$\text{cosine}(doc4, c2) = \frac{0+3+0}{\sqrt{0^2+2^2+0^2} \cdot \sqrt{0^2+(1.5)^2+(0.5)^2}}$$

$$= 0.94$$

Hence, $C1 = doc1, doc2$ and $C2 = doc3, doc4$.

PAGE :
DATE : / /

Lab No. 2

News Aggregator:

1. Design a spider to extract the news (nepali) from web. seed url → ekantipur, onlinekhabar, ratopati.
2. Implement the parser and filter the topics.
3. Use k-means algorithm to make the clusters of the news (k=5).

PAGE :
DATE : / /

Lab No. 2

News Aggregator:

1. Design a spider to extract the news(nepali) from web.
   seed url → ekantipur, onlinekhabar, ratopati.
2. Implement the parser and filter the topics.
3. Use k-means algorithm to make the clusters of the news (k=5).
4. K-medoids method:

- pick actual object to represent clusters instead of mean values.
- each remaining object is clustered with the representative object (medoid) to which is the most similar.
- the algorithm minimizes the sum of dissimilarities between each object and its corresponding reference point.

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - O_i|$$

Idea:
- initial representation are chosen randomly

for each representative object O_i

→ for each non representative object R, swap O and R

→ choose the configuration with the lowest cost

Eg:

$O_1(2,6)$, $O_2(3,4)$, $O_3(3,8)$, $O_4(4,7)$, $O_5(6,2)$, $O_6(6,4)$, $O_7(7,3)$, $O_8(7,4)$, $O_9(8,5)$, $O_{10}(7,6)$

$K=2$

$O_2(3,4)$ $O_8(7,4)$

C_1 C_2

$C_1 \rightarrow \{O_1, O_2, O_3, O_4\}$

$C_2 \rightarrow \{O_5, O_6, O_7, O_8, O_9, O_{10}\}$

$Err(O_2, O_8) = \underbrace{(O_2-O_1)+(O_2-O_2)+(O_2-O_3)+(O_2-O_4)}_{C_1} + \underbrace{(O_8-O_5)+(O_8-O_6)+(O_8-O_7)+(O_8-O_9)+(O_8-O_{10})}_{C_2}$

$= (3+4+4)+(3+1+1+2+2)$

$= 20$

$Err(O_2, O_7) = (2+4+4)+(2+2+1+3+3) = 22.$

∴ It is bad idea to swap O_8 and O_7.

PAGE :
DATE : / /

2. Hierarchical Clustering:

- Start with one cluster, individual item in its own cluster and iteratively merge clusters until all the items belongs to one cluster.
- Dendograms are pictorically used to represent hierarchical clustering.

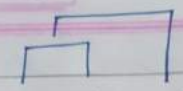
2. Approaches:

1. Single Linkage: distance between the closest members of two clusters.
2. Complete Linkage: distance between the numbers that are farthest apart.
3. Average Linkage: involves looking at the distance between all pairs and average of these distances.

Eg:

	P_1	P_2	P_3	P_4	P_5
P_1	0				
P_2	9	0			
P_3	6	7	0		
P_4	3	5	9	0	
P_5	11	10	2	8	0

↑ min



update the matrix

	P_1	P_2	P_3, P_5	P_4
P_1	0			
P_2	9	0		
P_3, P_5	?3	?7	0	
P_4	6	5	9	0

$(P_3, P_5) \rightarrow 11$

(P_3, P_1), (P_5, P_1)

min $(6, 11)$

$= 6$

$(P_3, P_5) \rightarrow P_4$

(P_3, P_4) (P_5, P_4)

6 is the updated value

(P_3, P_1), (P_5, P_1) → single (min) = 3

complete (max) = 11

average = 7

1. Consider the following dataset:

P_1 (0.40, 0.53)
P_2 (0.22, 0.38)
P_3 (0.35, 0.32)
P_4 (0.26, 0.19)
P_5 (0.08, 0.41)
P_6 (0.45, 0.30)

Apply the hierarchical clustering approach to cluster these points using complete linkage approach. Use Euclidean distance to find the similarity.

PAGE :
DATE : / /

$d = \sqrt{(y_2 - y_1)^2 + (x_2 - x_1)^2}$

	P_1	P_2	P_3	P_4	P_5	P_6
P_1	0					
P_2	0.23	0				
P_3	0.22	0.14	0			
P_4	0.37	0.19	0.16	0		
P_5	0.34	0.14	0.28	0.63	0	
P_6	(0.06)	0.24	0.10	0.22	0.39	0

Merge P_1, P_6.

	P_1, P_6	P_2	P_3	P_4	P_5
P_1, P_6	0				
P_2	0.24	0			
P_3	0.22	(0.14)	0		
P_4	0.37	0.19	0.16	0	
P_5	0.39	(0.14)	0.28	0.63	0

$d(P_1, P_2) = 0.23$ $\quad d(P_6, P_2) = 0.24$ choosing maximum
$d(P_1, P_3) = 0.22$ $\quad d(P_6, P_3) = 0.10$
$d(P_1, P_4) = 0.37$ $\quad d(P_6, P_4) = 0.22$
$d(P_1, P_5) = 0.34$ $\quad d(P_6, P_5) = 0.39$

Merge P_2, P_3, P_5

	P_1, P_6	P_2, P_3, P_5	P_4
P_1, P_6	0		
P_2, P_3, P_5	0.39	0	
P_4	(0.37)	0.63	0

$d(P_1, P_2) = 0.23$ $\quad d(P_1, P_3) = 0.22$ $\quad d(P_1, P_5) = 0.34$ } max
$d(P_6, P_2) = 0.24$ $\quad d(P_6, P_3) = 0.10$ $\quad d(P_6, P_5) = 0.39$
$d(P_2, P_4) = 0.19$ $\quad d(P_3, P_4) = 0.16$ $\quad d(P_5, P_4) = 0.63$

Merge P_1, P_6, P_4

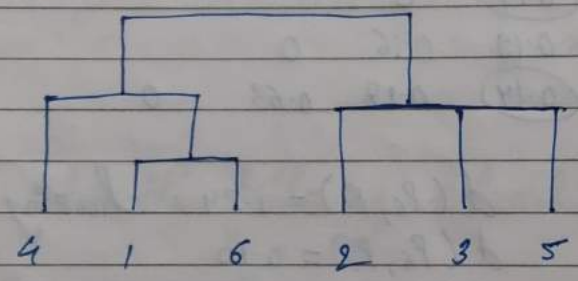
	P_1, P_6, P_4	P_2, P_3, P_5
P_1, P_6, P_4	0	
P_2, P_3, P_5	0.63	0

$d(P_1, P_2) = 0.23$ $\quad d(P_1, P_3) = 0.22$ $\quad d(P_1, P_5) = 0.34$

$d(P_6, P_2) = 0.24$ $\quad d(P_6, P_3) = 0.10$ $\quad d(P_6, P_5) = 0.39$

$d(P_4, P_2) = 0.19$ $\quad d(P_4, P_3) = 0.16$ $\quad d(P_4, P_5) = 0.63$

Complete Linkage Dendrogram.

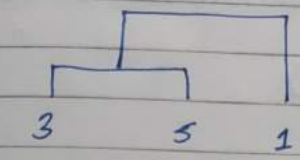


Previous Example:

	P1	P2	P3	P4	P5
P1	0				
P2	9	0			
P3	3	7	0		
P4	6	5	9	0	
P5	11	10	(2)	8	0

↑ min

	P_1	P_2	P_3, P_5	P_4
P_1	0			
P_2	9	0		
P_3, P_5	③	7	0	
P_4	6	5	8	0



(P_3, P_1) and (P_5, P_1) single (min) = 5
complex (max) = 11
average = 7.

Merge P_1, P_3, P_5

	P_1, P_3, P_5	P_2	P_4
P_1, P_3, P_5	0		
P_2	7	0	0
P_4	6	5	

$d((P_1, (P_3, P_5)), P_2) = \min((P_1, P_2), ((P_3, P_5), P_2))$

$= \min(9, 7) = 7.$

$d(P_1, (P_3, P_5), P_4) = \min(d(P_1, P_4), d((P_3, P_5), P_4))$
$= \min(6, 8) = 6$

Merge P_2, P_4.

	$(P_1,(P_3,P_5))$	(P_2,P_4)
$(P_1,(P_3,P_5))$	0	
(P_2,P_4)	6	0

$d((P_1,(P_3,P_5)),(P_2,P_4)) = \min\{d((P_1,(P_3,P_5)),P_2), d((P_1,(P_3,P_5)),P_4)\}$

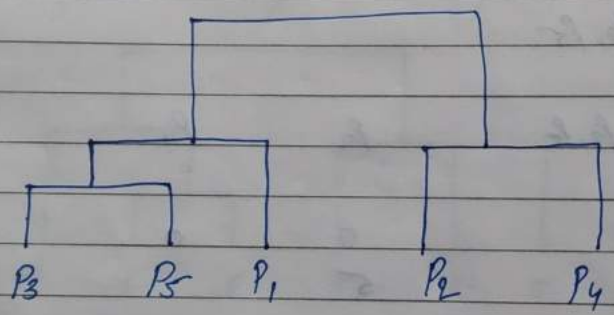
$= \min\{7,6\}$

$= 6$

Merge: $(P_1,(P_3,P_5))$ and (P_2,P_4)

Unit-

The single linkage dendrogram is:

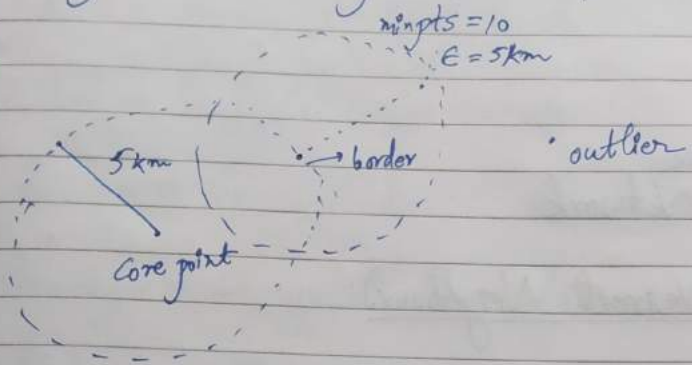


DBSCAN:

- Density Based Spatial Clustering Application.
- Based on two parameters
- 1. ϵ-neighborhood → threshold for determining the range.
- 2. Min pts → minimum number of points to form a cluster.

PAGE :
DATE : / /

- core points, border points, outlier/noise.



Unit: 1-5

Classification

- placing the things where they belong.
- based on two

1. Model Construction (Training Phase)
2. Model Usage (Testing Phase)

Ex

Paradigm:

Training Data Set:

Name	Post	Years	Pension
A	Prof.	1	Yes
B	Asst. Prof.	2	No
C	Asst. Prof.	5	Yes
D	Prof	5	Yes
E	Asst. Prof	6	Yes

Testing Data (x, Prof, 3)

→ [Machine] → YES

⇓ what it learns

If POST == "Prof" OR Years >= 5

THEN Pension = Yes.

Algorithm:

- KNN
- Bayesian
- Decision Tree
- Neural Network

KNN (K-Nearest Neighbour):

Algorithm:

1. For testing the object x
2. Compute the feature vector of x = d
3. for each $\langle y, c(y) \rangle \in \emptyset$

   $S_y = \text{cossim}(d, y)$
4. Sort S_y in decreasing order.
5. Let N be the first k.
6. Return the majority class

PAGE :
DATE : / /

Eg:

Food = turkey stuffing = d_1

Food = buffalo wings = d_2

Beverage = Cream Soda. = d_3

Beverage = Orange Soda. = d_4

Solution: turkey, stuffing, buffalo, wings, cream, soda, orange

$d_1 \langle 1,1,0,0,0,0,0 \rangle$ — S_1

$d_2 \langle 0,0,1,1,0,0,0 \rangle$ — S_2 — $d \langle 1,0,0,0,0,1,0 \rangle$

$d_3 \langle 0,0,0,0,1,1,0 \rangle$ — S_3

$d_4 \langle 0,0,0,0,0,1,1 \rangle$ — S_4

Classify the document, $\langle$turkey, soda$\rangle$ with $k=3$

$S_1 = \frac{1}{\sqrt{2}\cdot\sqrt{2}} = \frac{1}{2} = 0.5$

$S_2 = 0$

$S_3 = \frac{1}{\sqrt{2}\cdot\sqrt{2}} = \frac{1}{2} = 0.5$

$S_4 = \frac{1}{\sqrt{2}\cdot\frac{1}{\sqrt{2}}} = 0.5$

Sort

$0.5 \rightarrow S_1 (d_1) \rightarrow B$

$0.5 \rightarrow S_3 (d_3) \rightarrow F$ } food.

$0.5 \rightarrow S_4 (d_4) \rightarrow F$ }

$0 \rightarrow S_2 (d_2) \rightarrow$

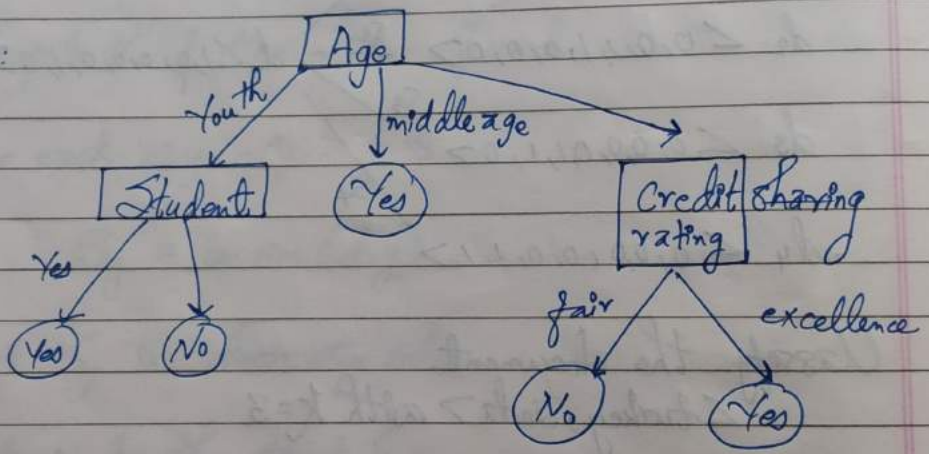
58

PAGE :
DATE : / /

1. Decision Trees:

- is a flowchart like structure where each internal node represent the attributes and leaf represent the label of the class.
- the values on arc represent the possible outcomes of an attribute
- each internal node is represented by rectangle.
- each leaf is represented by oval.

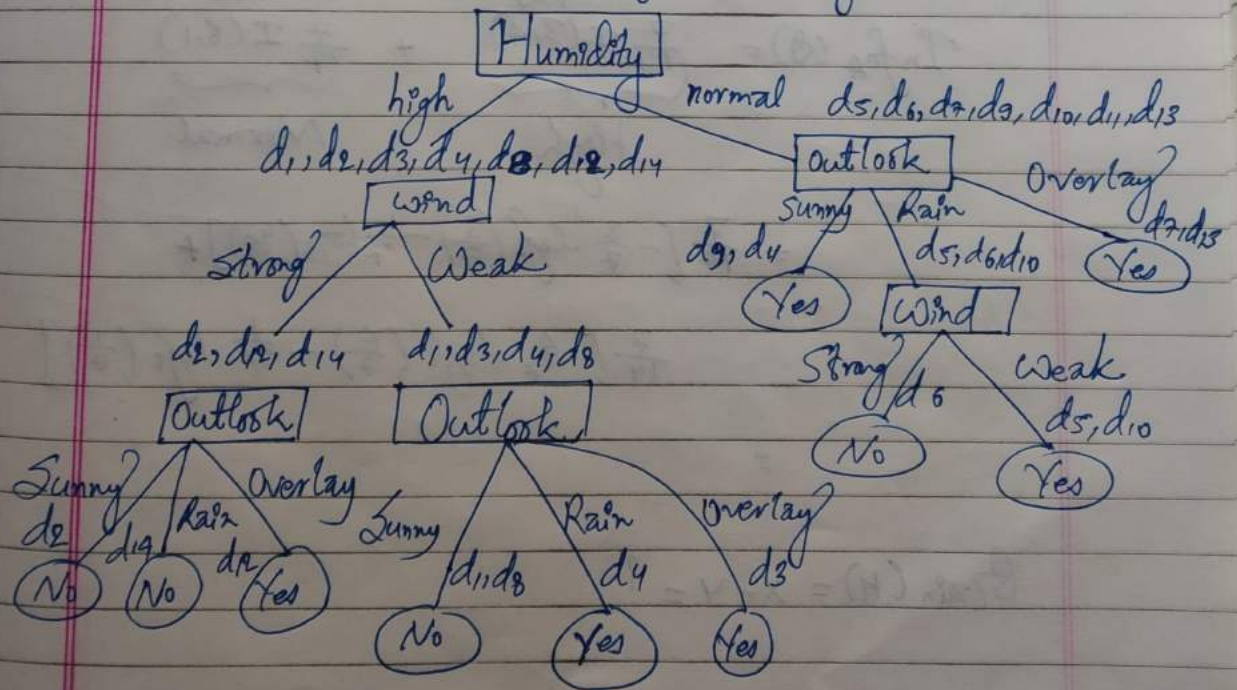
Eg:



PAGE :
DATE : / /

Training DataSet:

Day	Outlook	Temperature	Humidity	Wind	Play Tennis (Class)
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Strong	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cool	Normal	Weak	Yes
D10	Rain	Mild	Normal	Weak	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No.



Testing Data → <Sunny, Hot, Normal, Weak> ⇒ { Yes }

Attribute Selection Algorithm (ID3):

Select the information with the highest information gain.

$$Gain(A) = Info(\vartheta) - Info_A(\vartheta) \text{ where,}$$

$$Info(\vartheta) = -\sum_{i=1}^{m} P_i \log_2(P_i)$$

$$Info_A(\vartheta) = \sum_{i=1}^{v} \frac{|\vartheta_i|}{|\vartheta|} * I(\vartheta_i)$$

Eg: A = Humidity

$$Info(\vartheta) = \underbrace{-\frac{9}{14}\log_2\left(\frac{9}{14}\right)}_{\text{Yes}} \underbrace{-\frac{5}{14}\log_2\left(\frac{5}{14}\right)}_{\text{No}}$$

$$Info_A(\vartheta) = \underbrace{\frac{7}{14} I(3,4)}_{\text{High}} + \underbrace{\frac{7}{14} I(6,1)}_{\text{Normal}}$$

$$= \frac{7}{14}\left[-\frac{3}{7}\log_2\left(\frac{3}{7}\right) - \frac{4}{7}\log_2\left(\frac{4}{7}\right)\right] +$$

$$\frac{7}{14}\left[-\frac{6}{7}\log_2\left(\frac{6}{7}\right) - \frac{1}{7}\log_2\left(\frac{1}{7}\right)\right]$$

$$=$$

$Gain(H) = X - Y = ?$

PAGE :
DATE : / /

2. Rocchio Algorithm:

Training Phase:

- For each class C_i, calculate

$$P_i = P_i + d \text{ where } d \in C_i$$

Testing Document:

- Let y be the featured vector of document to be tested.

- For each class find cosine similarity

$$S_i = \cos\sin(P_i, y)$$

return C_i with maximum S_i

Eg: Food = turkey stuffing d_1

Food = buffalo wings d_2

Beverage = cream soda d_3

Beverage = Orange soda d_4

Testing → turkey soda d

PAGE :
DATE : / /

turkey, stuffing, buffalo, wings, cream, soda, orange

$d_1 \langle 1,1,0,0,0,0,0 \rangle$] C_1 (food, $P_1 = d_1 + d_2 = \langle 1,1,1,1,0,0,0 \rangle$
$d_2 \langle 0,0,1,1,0,0,0 \rangle$]

$d_3 \langle 0,0,0,0,1,1,0 \rangle$] C_2 (Beverage, $P_2 = d_3 + d_4 = \langle 0,0,0,0,1,2,1 \rangle$
$d_4 \langle 0,0,0,0,0,1,1 \rangle$]

$d \langle 1,0,0,0,0,1,0 \rangle$

$S_1 = \cos\text{sim}(d, P_1) = \frac{1}{\sqrt{4} \cdot \sqrt{2}} = \frac{1}{\sqrt{8}} = \frac{1}{2\sqrt{2}}$

$S_2 = \cos\text{sim}(d, P_2) = \frac{2}{\sqrt{6} \cdot \sqrt{2}} = \frac{2}{\sqrt{12}} = \frac{2}{2\sqrt{3}} = \frac{1}{\sqrt{3}}$

PAGE :
DATE : / /

Classification using Backpropagation:

1. Initialize the input and weights.
2. Calculate the output

   2.1 For each input layer

   $$O_j^0 = I_j$$

   2.2 For each hidden layer and output layer

   $$I_j = \sum_k O_k w_{kj}^0$$

   $$O_j = \frac{1}{e^{-I_j^0}}$$

3. Calculate the error:

   3.1 For output layer (T → Target)

   $$Err_j = O_j^0 (1 - O_j^0)(T - O_j^0)$$

   3.2 For Hidden layer

   $$Err_j^0 = O_j^0 (1 - O_j^0) \sum_k Err_k w_{jk}^0$$

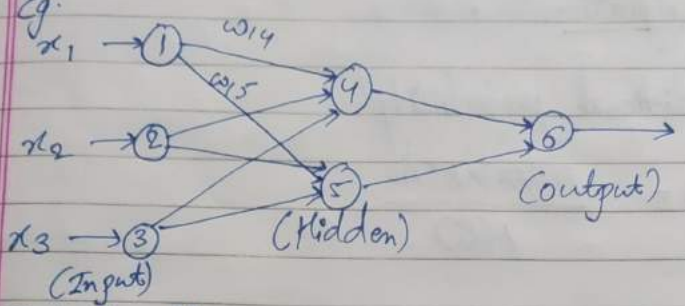
4. Update the weight

   $$\Delta w_{ij}^0 (new) = w_{ij}^0 (old) + l \cdot Err_j^0$$

   (l ↓ learning rate.)

PAGE :
DATE : / /

Eg:



Initialization:

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}
0	1	0	0.2	0.3	0.8	0.1	0.2	0.3	0.4	0.5

Input: $O_1 = x_1 = 0$; $O_2 = x_2 = 1$; $O_3 = x_3 = 0$

Hidden Layer:

For O_4

$I_4 = O_1 w_{14} + O_2 w_{24} + O_3 w_{34}$

$O_4 = \frac{1}{1+e^{-I_4}}$

For O_5

$O_5 = O_1 w_{15} + O_2 w_{25} + O_3 w_{35}$

$O_5 = \frac{1}{1+e^{-I_5}}$

For O_6

$I_6 = O_4 w_{46} + O_5 w_{56}$

$O_6 = \frac{1}{1+e^{-I_6}}$

Calculate Error:

$Err_6 = O_6(1-O_6)(1-O_6)$

If $I_6\ Err_6 \approx 0$ Terminate

$Err_4 = O_4(1-O_4)\, Err_6 \cdot w_{46}$

$Err_5 = O_5(1-O_5)\, Err_6\, w_{56}$

Update the weight

$w_{14}(new) = w_{14}(old) + 0.9 * Err_4$

$w_5(new) = w_{15}(old) + 0.9 * Err_5$

$w_{46}(new) = w_{46}(old) + 0.9 * Err_6$

2. Bayesian Classification:

- based on conditional probability.

$$P(A|B) = \frac{P(B|A) * P(A)}{P(B)}$$

Eg:

Age	Income	Credit Rating	Buys computer
20-30	>10K	Fair	No
20-30	>20k	Exellence	Yes
30-40	>10K	Fair	Yes
30-40	>20k	Exellence	Yes
20-30	>10k	Exellence	Yes

(20-30, >10k, Fair) ⇒ ?

Are your result credible?

Predictive Result \ Actual Result	Yes	No
Yes	TP (True Positive)	FP (False Positive)
No	FN (False Negative)	TN (True Negative)

$$\text{Precision} = \frac{TP}{TP + FP}$$

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

$X = (\underbrace{20-30}_{\text{Age}}, \underbrace{>10k}_{\text{Income}}, \underbrace{\text{Fair}}_{\text{C.R.}}) = ?$

PAGE :
DATE : / /

$P(Yes/X) = P(X/Yes) * P(Yes)$

$= \frac{2}{4} * \frac{2}{4} * \frac{1}{4} * \frac{4}{5}$

$P(No/X) = P(X/No) * P(No)$

Lab:

Q. Compute the entropy value at each level in previous training set using ID3.

Unit-6

Combining Classifier

- significantly reduce the risk of using a classifier that will be unadequate on unseen data.

McNemar's Test:

- comparing two classifier on the same data.

	Algorithm A failed	Algorithm A succeeded
Algorithm B failed	N_{ff}	N_{sf} ↳ success/fail.
Algorithm B succeeded	N_{fs}	N_{ss}

$$z = \frac{(|N_{sf} - N_{fs}| - 1)}{\sqrt{N_{sf} + N_{fs}}}$$ ↳ testing set

- when $z=0$, the two algorithms are said to show similar performance

Cochran's Q Test:

- Comparing more than two classifiers
- Test the hypothesis, that there is no difference

$H_0 : P_1 = P_2 = \dots = P_L$ ↳ no. of classifier

- $Q = (L-1) \frac{L \sum_{i=1}^{L} G_i^2 - T^2}{LT - \sum_{j=1}^{N_{ts}} (L_j)^2}$

degree of freedom

where, $T = \sum_{i=1}^{L} G_i$

PAGE :
DATE : / /

G_{ts} → no. of object out of N_{ts}, correctly classified by D_i, $i = 1, 2, \ldots, L$

L_j → is the no. of classifier out of L that correctly classified object.

Ensemble Method:

- uses a combination of models to increase accuracy
- methods:

1. Bagging
2. Boosting

1. Bagging:

- averaging the predictions over a collection of classifiers.

Training:

- a classifier model M_i is learned from each training set D_i (with replacement)

Testing:

- classify the unknown sample X
- each classifier M_i returns its class prediction.
- return the majority class.

PAGE :
DATE : / /

Boosting:

- is an iterative technique which adjust the weight of an observation based on the last classification.

Steps:

- Draw a random subset of training samples d_1 without replacement from the training set S to train a weak learner C_1.

- Draw a second random training subset d_2 without replacement from the training set and add 50% of the samples that were previously falsely classified to weak learner C_2.

- Find the training samples d_3 in the training set S on which C_1 and C_2 disagree to train to a third weak learner C_3.

Unit-7

Semantic Web

PAGE
DATE : / /

Semantic Web:

- provides a common framework that allows data to be shared with understanding of its semantic meaning.
- XML

RDF (Resource Description Framework)

- provides the technology for expressing the meaning of terms and concepts in a form that machine can process.

Eg: <Album>

<Singer> Narayan Gopal </Singer>

<Price> — </Price>

<Edition> — </Edition>

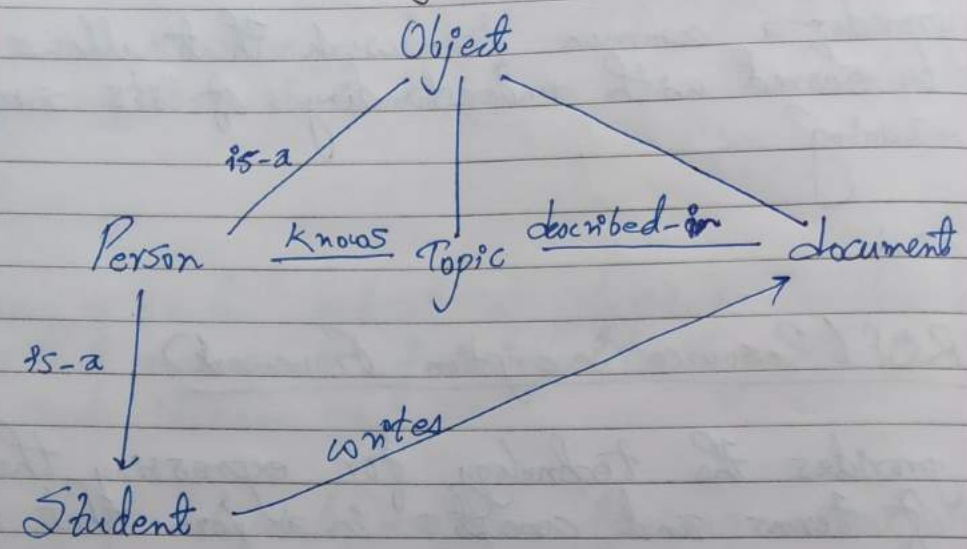
</Album>

Ontologies:

- defines the relation between concepts and specifies logical rules for reasoning about them.

PAGE :
DATE : / /

Sample of Ontology:



Reasoning Over Ontologies:

- inferencing capabilities

Eg: X is author of Y ⇒ Y is written by X.