



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Text Mining

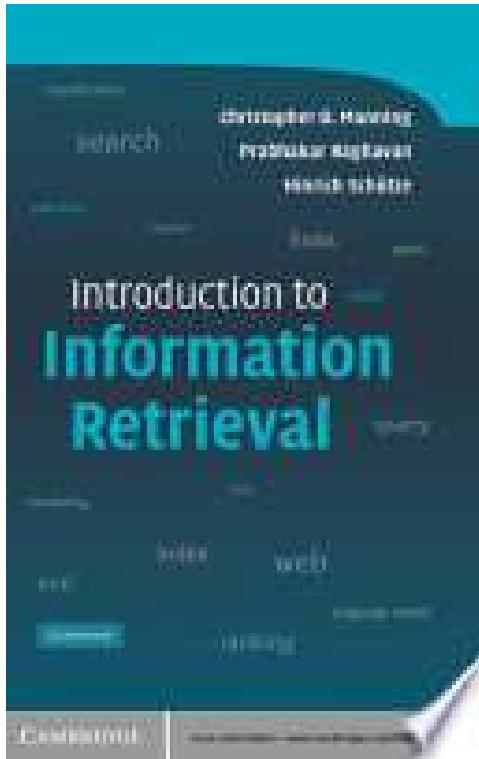
Prof.Aruna Malapati

Overview of the Feature Engineering Course

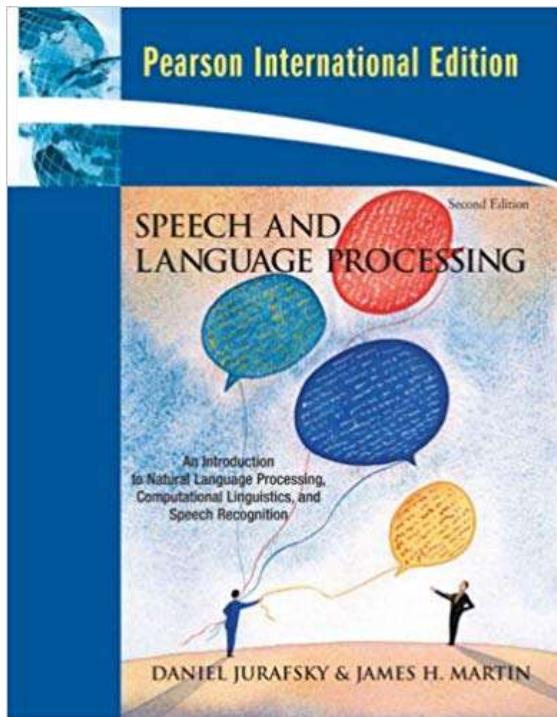
- Introduction to Text Mining
- Parts of speech Tagging
- Topic modelling using Latent Dirichlet Allocation
- Sentiment Analysis
- Recommender systems



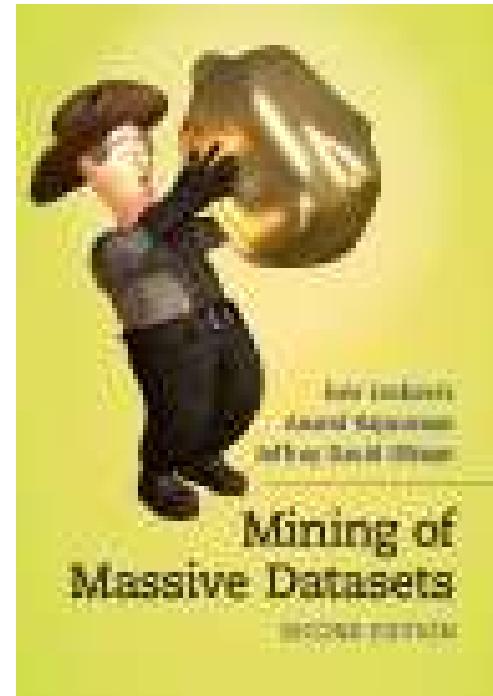
Books



Module-1



Module-2



Module-5

Evaluation

Evaluation Component	Marks	Type
Comprehensive Examination	40%	Closed
Quizzes (2)	24%	Open
2 Minor Projects (Evaluated twice)	24%	Open
Assignments/Exercises (2)	12%	Open



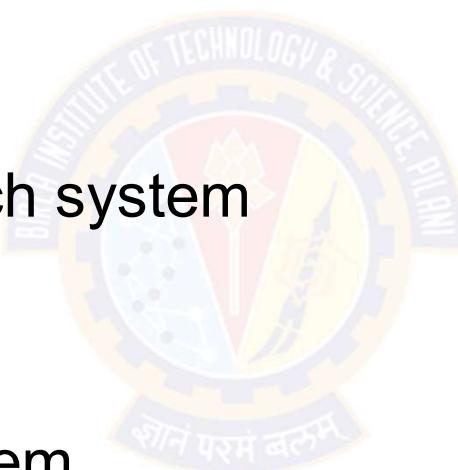
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Boolean Retrieval Model

Prof.Aruna Malapati

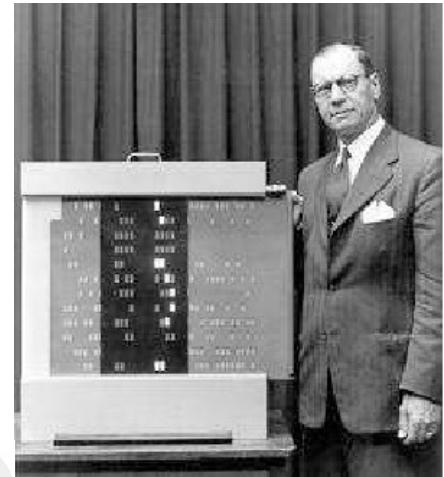
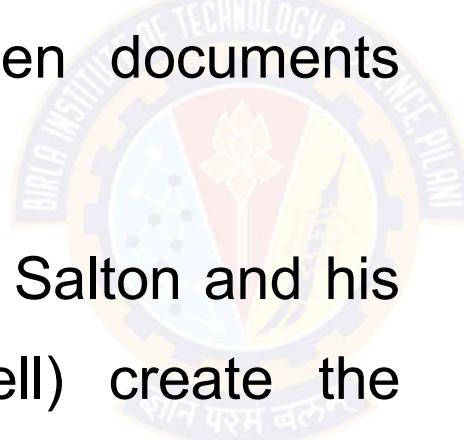
Learning objectives

- History of web search
- Jargons
- Architecture of a search system
- Bag of words model
- Boolean retrieval System



A Brief History of Web Search

- 1957: Hans-Peter Luhn (IBM) uses words as indexing units for documents – Measure similarity between documents by word overlap
- 1960s and 1970s: Gerard Salton and his students (Harvard, Cornell) create the SMART system – Vector space model – Relevance feedback



A Brief History of Web Search

- 1991: Tim Berners-Lee “invents” the World Wide Web
- First Web search engines:
 - Archie: Query file names by regular expressions
 - Architext/Excite: Full text search, simple ranking (1993)
- Until 1998, web search meant information retrieval
- 1998: Google was founded – Exploits link structure using the PageRank algorithm



Jargons

➤ Corpus



Examples

- ✓ Medline / Pubmed document collection
- ✓ Tweets
- ✓ Face books posts
- ✓ Customer Reviews about a product



➤ Information Need



Examples

- ✓ What is the capital of India?
- ✓ Will the Finance Ministry reduce personal taxes?
- ✓ What is the currency in India?

Jargons (Contd..)

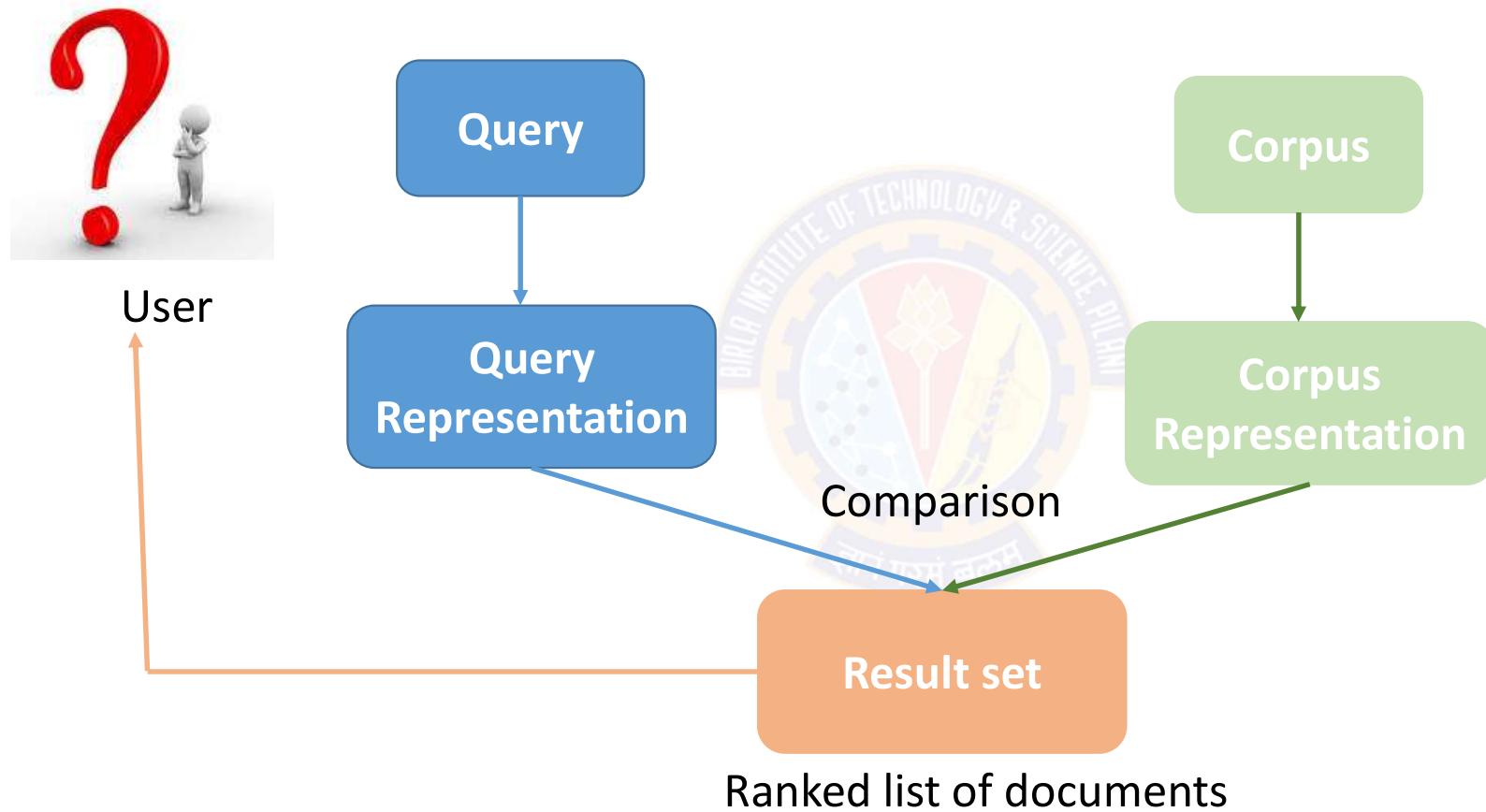
➤Query



Examples

- ✓ Medline / Pubmed document collection

Information Retrieval Systems (IRS)



Bag of Words representation

- A very popular and basic representation of documents is the bag of words model.
- Each document is represented by a **bag (= multiset)** of terms from a predefined vocabulary.

The Jackal was eyeing
at the grapes

He was as cunning as
a Jackal

These Grapes are too
sweet but the poor
Jackal could not have it.

1	2	0		1			1	1			0								
a	a	at		C			h	w			ja								
s				u			e	a			c								
				n				s			k								
				n							al								
				i															
				n															
				g															



Term Incidence Matrix

Two Forms of Term incidence Matrix

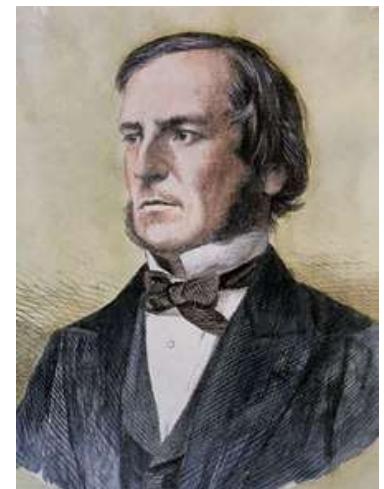
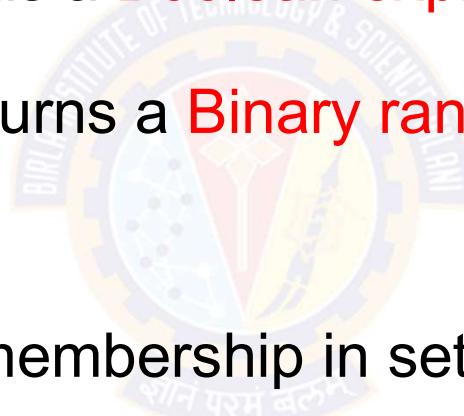
	T1	T2	T3	T4	T5	T6
D1	1	0	0	1	1	0
D2	0	1	0	1	1	0
D3	1	0	1	0	1	1
D4	1	0	1	0	1	1



	T1	T2	T3	T4	T5	T6
D1	6	0	0	2	1	0
D2	0	8	0	5	3	0
D3	2	0	6	0	5	2
D4	5	0	2	0	6	7

Boolean Retrieval Model

- Documents are represented as a **vector of the indexed terms**.
- Query is represented as a **Boolean expressions over index terms**.
- The search system returns a **Binary ranking function**, i.e. 0/1-valued.
- Retrieval is based on membership in sets



Boolean Operators

- The following are the three operators used in Boolean Retrieval model.

- AND (Conjunction or \wedge)
- OR (Disjunction or \vee)
- NOT (Negation or \neg)

\wedge	0	1
0	0	0
1	0	1

\vee	0	1
0	0	1
1	1	1

\neg	
0	1
1	0

Example

<i>doc</i>	<i>t₁</i>	<i>t₂</i>	<i>t₃</i>	<i>t₄</i>	<i>t₅</i>	<i>t₆</i>	<i>t₇</i>	<i>t₈</i>	<i>t₉</i>	<i>t₁₀</i>	<i>t₁₁</i>
<i>D₁</i>	0	0	1	0	1	1	0	0	0	1	0
<i>D₂</i>	1	1	0	1	0	0	0	0	1	0	1
<i>D₃</i>	1	1	0	0	0	1	0	0	0	1	1
<i>D₄</i>	0	0	0	0	0	1	0	0	1	0	1

Query: t1 AND t2 AND NOT t4

Pros and Con's of Boolean Retrieval Model

- + Simple query paradigm, easy to understand
- A binary ranking function returns a set of results, i.e. it is unordered
- Doc-term matrix is **too sparse**
- Controlling the result size is difficult
- Similarity queries are not supported

Westlaw - Online legal research service for US law

- Includes more than 40,000 databases of case law, state and federal statutes, administrative codes, law journals, newspapers ...
- Offers search by:
 - “Terms and Connectors” – Boolean Search
 - “Natural Language” – Free text querying (added in 1992)
- Boolean search includes the Boolean operators plus some proximity operators
 - space = OR • /s, /p, /k = matches in the same sentence, paragraph or within k-words respectively
 - & = AND • ! = a trailing wildcard query
- Example: “trade secret” /s disclos! /s prevent /s employ!
 disab! /p access! /s (work-site work-place) (employment /3 place)



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Information Retrieval Pipeline

Prof. Aruna Malapati

Learning objectives

- Information Retrieval pipeline
- Inverted index construction



Information Retrieval Pipeline

Documents collected from various sources



Ram and Shyam are childhood friends.

⋮

Token stream

Tokenizer

Ram

Shyam

childhood

friends

Linguistic modules

DE pluralization

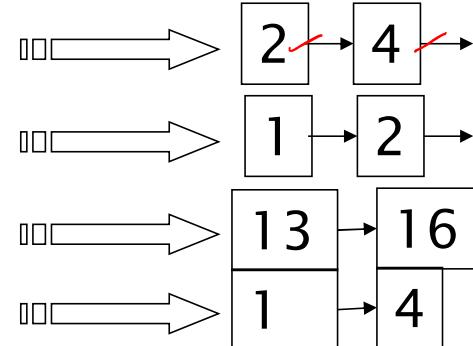
Case folding

ram shyam childhood friend

Modified tokens/
Stream of normalized
tokens

Indexer

childhood



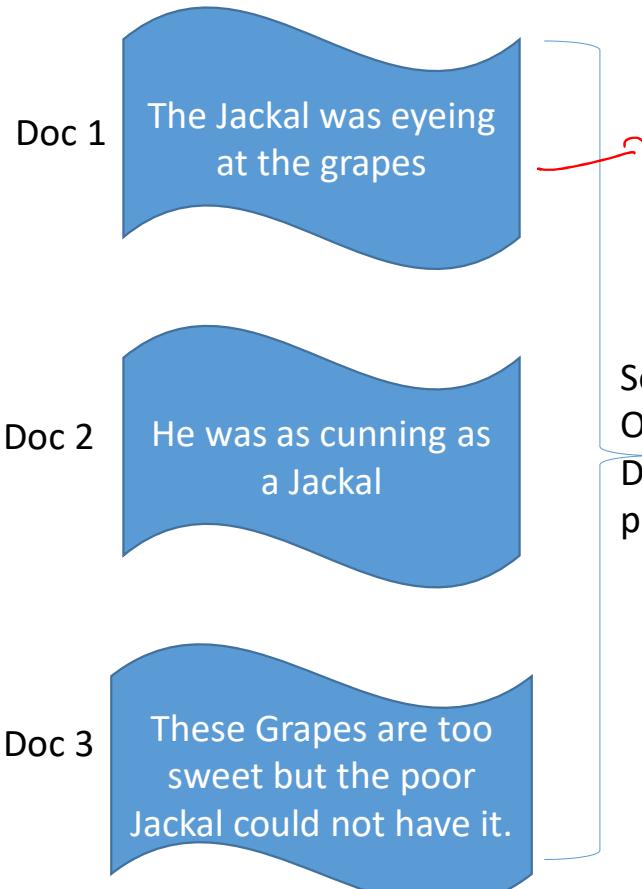
Inverted index

friend

ram

shyam

Steps during indexing



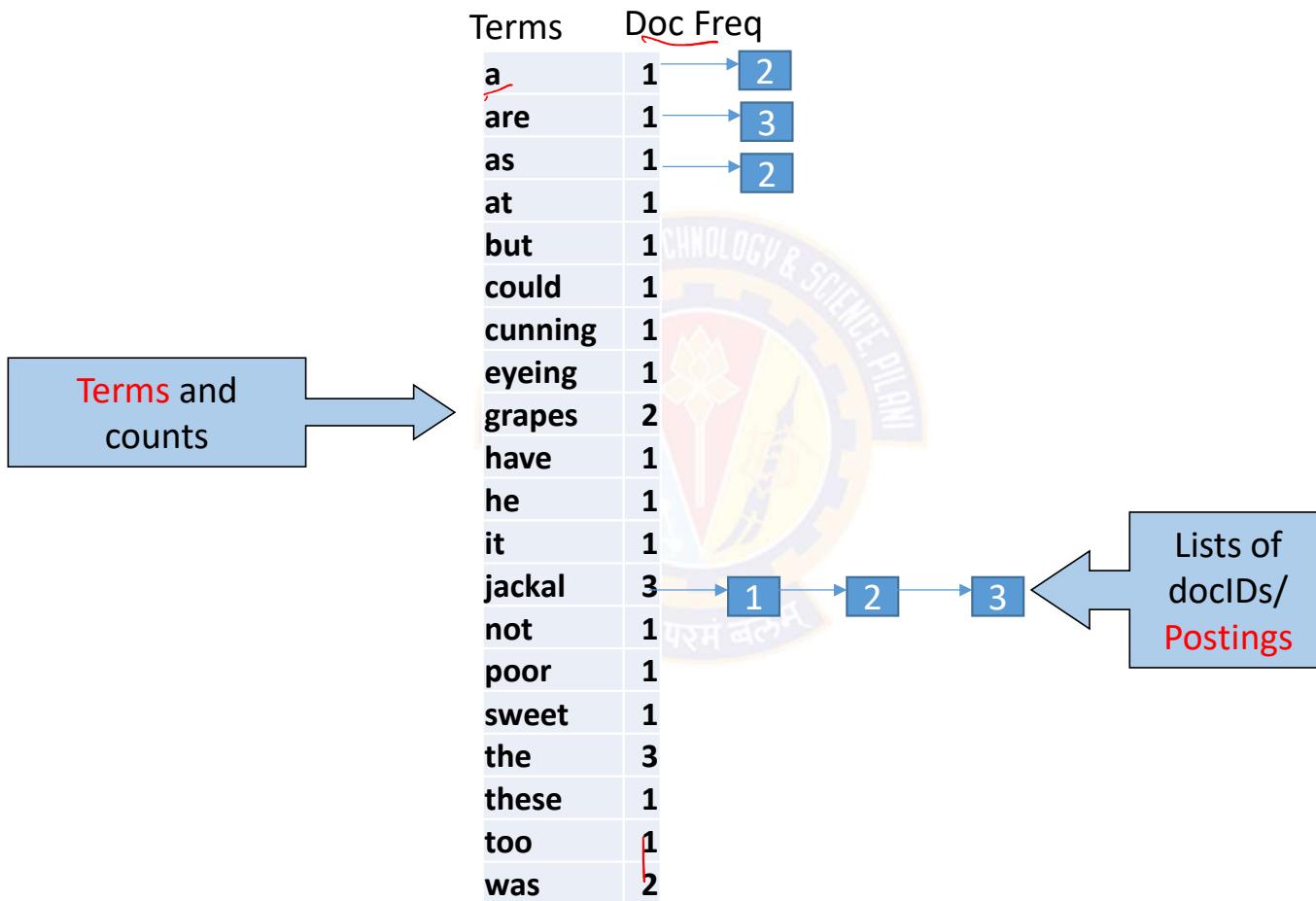
The index table shows the tokens and their document IDs, with annotations for sorting:

Token	Document ID
the	1
Jackal	1
Was	1
eyeing	1
at	1
the	1
grapes	1
he	2
was	2
as	2
cunnin	
g	2
as	2
a	2
jackal	2
these	3
grapes	3
are	3
too	3
sweet	3
.	
.	
.	

Annotations on the right side of the table indicate the sorting steps:

- "Sort by terms" is written vertically next to the first four columns of the table.
- "Sort by Doc ids" is written vertically next to the last two columns of the table.
- A red bracket groups the first four columns, and another red bracket groups the last two columns.
- A red arrow points from the bracket under "Sort by terms" to the first column of the table.

Inverted Index



Posting list implementations

- Arrays vs Linked list
- Factors that influence the decision
 - Is the corpus fixed?
 - Can we fit the entire posting list in main memory?





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Merge Algorithm

Prof.Aruna Malapati

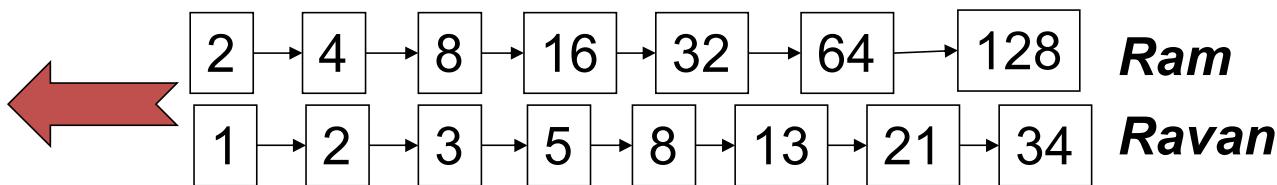
Learning objectives

- Answering Queries using merge algorithm



Query processing: AND

- Consider processing the query:
 - **Ram AND Ravan**
 - Locate **Ram** in the Dictionary;
 - Retrieve its postings.
 - Locate **Ravan** in the Dictionary;
 - Retrieve its postings.
 - “Merge” the two postings:



Intersecting two postings lists (a “merge” algorithm)

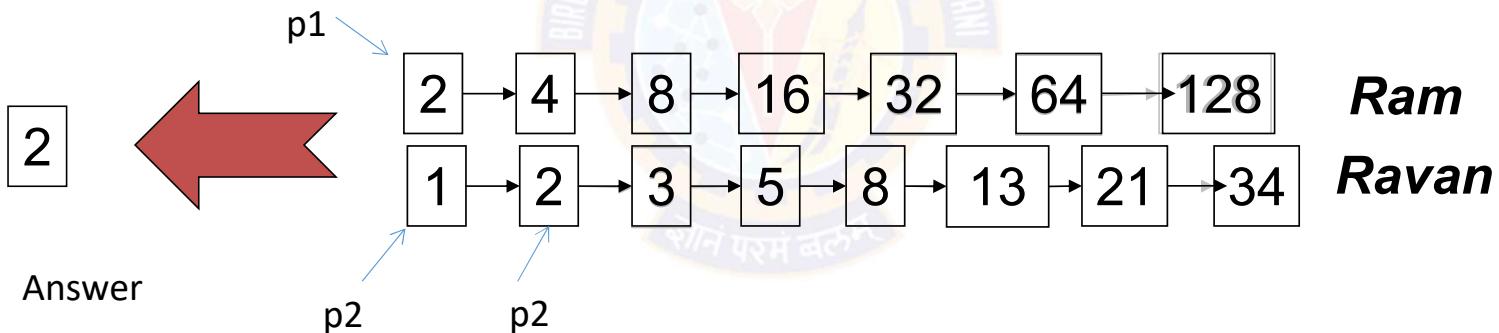
INTERSECT(p_1, p_2)

```
1  answer  $\leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then ADD(answer,  $\text{docID}(p_1)$ )
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7      else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8          then  $p_1 \leftarrow \text{next}(p_1)$ 
9          else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return answer
```

Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1

P2: pointer to current location in list2



```
INTERSECT( $p_1, p_2$ )
1  $answer \leftarrow \langle \rangle$ 
2 while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3 do if  $docID(p_1) = docID(p_2)$ 
4     then ADD( $answer, docID(p_1)$ )
5          $p_1 \leftarrow next(p_1)$ 
6          $p_2 \leftarrow next(p_2)$ 
7     else if  $docID(p_1) < docID(p_2)$ 
8         then  $p_1 \leftarrow next(p_1)$ 
9     else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

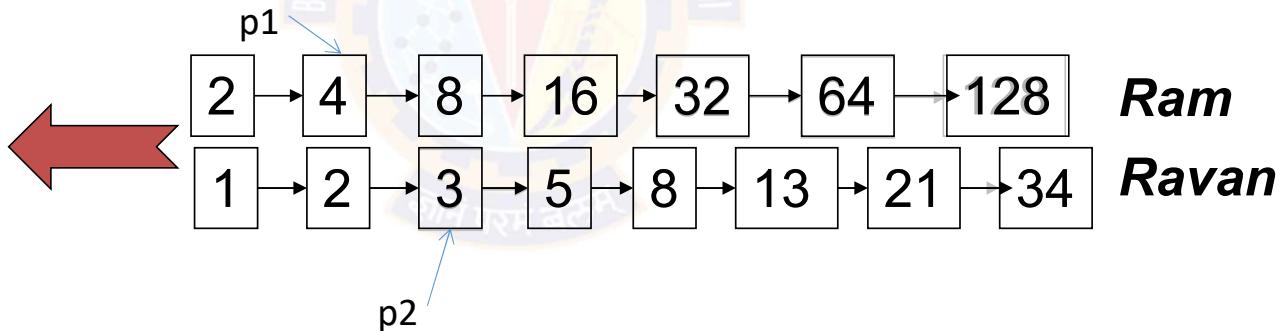
Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1

P2: pointer to current location in list2

2

Answer



INTERSECT(p_1, p_2)

```
1  answer ← ⟨ ⟩
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $\text{docID}(p_1) = \text{docID}(p_2)$ 
4      then ADD(answer, docID( $p_1$ ))
5           $p_1 \leftarrow \text{next}(p_1)$ 
6           $p_2 \leftarrow \text{next}(p_2)$ 
7  else if  $\text{docID}(p_1) < \text{docID}(p_2)$ 
8      then  $p_1 \leftarrow \text{next}(p_1)$ 
9  else  $p_2 \leftarrow \text{next}(p_2)$ 
10 return answer
```

Intersecting two postings lists (a “merge” algorithm)

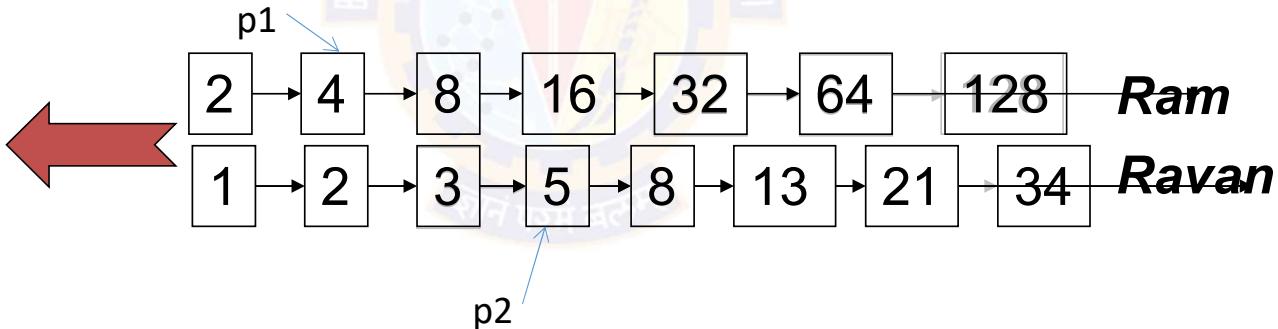
P1: pointer to current location in list1

P2: pointer to current location in list2

```
INTERSECT( $p_1, p_2$ )
1  $answer \leftarrow \langle \rangle$ 
2 while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3 do if  $docID(p_1) = docID(p_2)$ 
4     then ADD( $answer, docID(p_1)$ )
5          $p_1 \leftarrow next(p_1)$ 
6          $p_2 \leftarrow next(p_2)$ 
7     else if  $docID(p_1) < docID(p_2)$ 
8         then  $p_1 \leftarrow next(p_1)$ 
9     else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

2

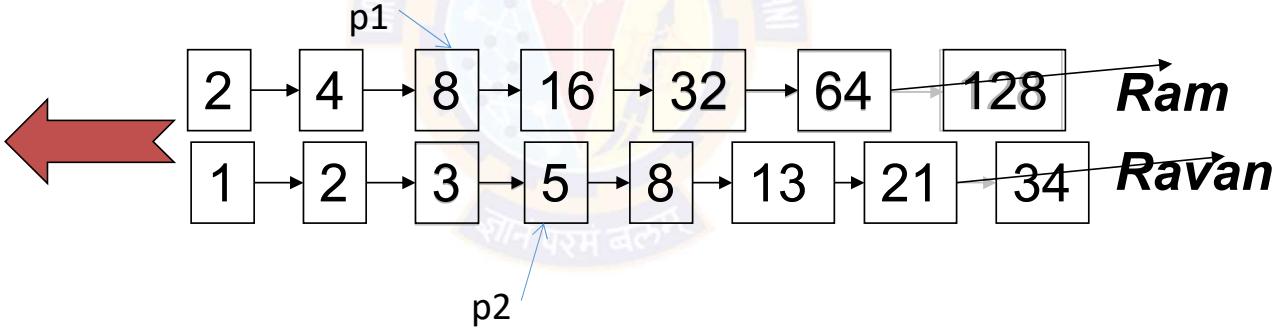
Answer



P1: pointer to current location in list1
P2: pointer to current location in list2

2

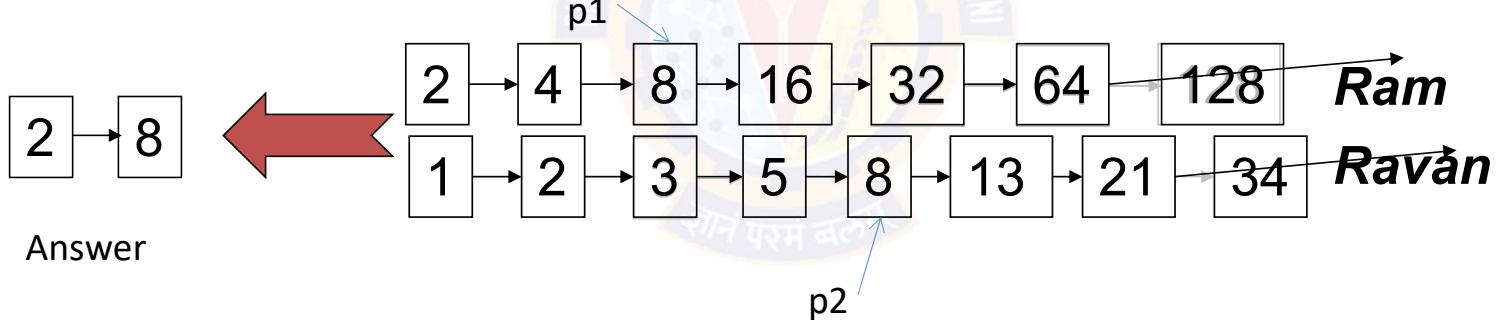
Answer



```
INTERSECT( $p_1, p_2$ )
1  $answer \leftarrow \langle \rangle$ 
2 while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3 do if  $docID(p_1) = docID(p_2)$ 
4     then ADD( $answer, docID(p_1)$ )
5          $p_1 \leftarrow next(p_1)$ 
6          $p_2 \leftarrow next(p_2)$ 
7     else if  $docID(p_1) < docID(p_2)$ 
8         then  $p_1 \leftarrow next(p_1)$ 
9     else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

Intersecting two postings lists (a “merge” algorithm)

P1: pointer to current location in list1
P2: pointer to current location in list2



Postings sorted by DocIds.

```
INTERSECT( $p_1, p_2$ )
1  $answer \leftarrow \langle \rangle$ 
2 while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3 do if  $docID(p_1) = docID(p_2)$ 
4   then ADD( $answer, docID(p_1)$ )
5      $p_1 \leftarrow next(p_1)$ 
6      $p_2 \leftarrow next(p_2)$ 
7   else if  $docID(p_1) < docID(p_2)$ 
8     then  $p_1 \leftarrow next(p_1)$ 
9   else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

More query processing

- Brutus OR Caesar
- NOT Brutus
- Brutus AND NOT Caesar
- Brutus OR NOT Caesar





Thank You!

In our next session: Query Optimization





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Query Optimization

Prof. Aruna Malapati

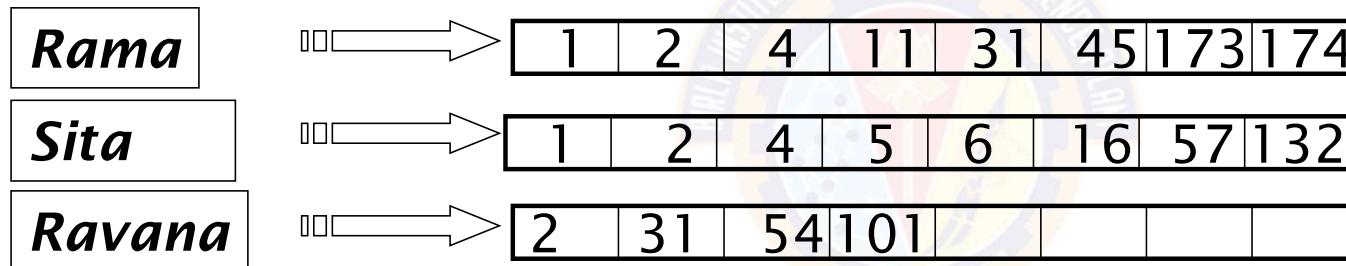
Learning objectives

- Apply query optimization for any type of query



Query Optimization

- Consider a query that is an *and* of t terms.
- For each t terms get the postings list, then AND them together.



QUERY: Rama AND Sita AND Ravana

Rama AND (Sita AND Ravana)
(Rama AND Sita) AND Ravana
(Rama AND Ravana) AND Sita

Query Optimization

- Process in the order of increasing document frequency.
- Intersect the two smallest postings list
- All intermediate results will be no bigger than the smallest postings list, so we are likely to minimize the work.

Rama	⇒	1 2 4 11 31 45 173 174
Sita	⇒	1 2 4 5 6 16 57 132
Ravana	⇒	2 31 54 101

QUERY: Rama AND Sita AND Ravana

Execute the query as (Rama AND Sita) AND Ravana

This is why the doc
freq is stored



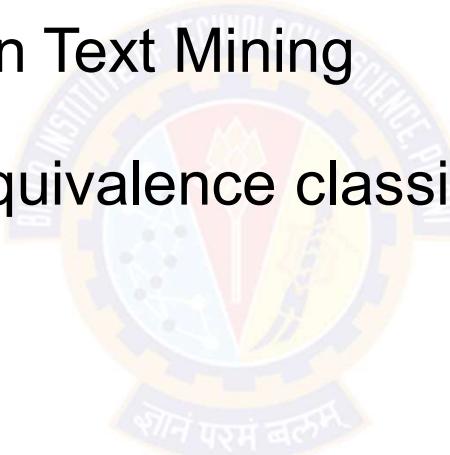
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Tolerant Retrieval using Normalization

Prof.Aruna Malapati

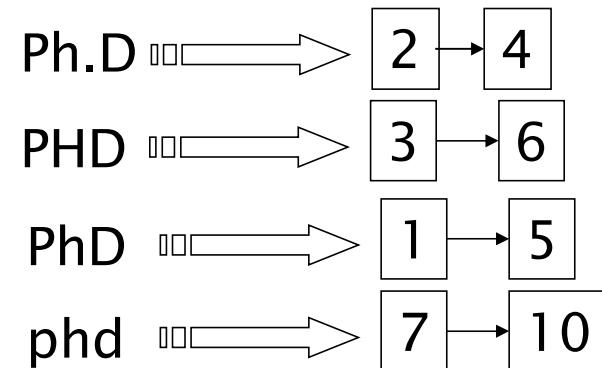
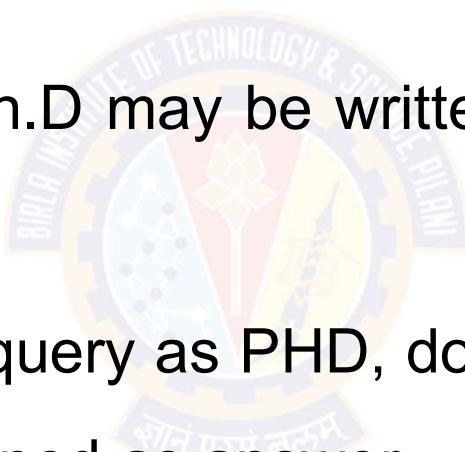
Learning objectives

- Explain Tolerant Retrieval
- Define Normalization in Text Mining
- Normalization using equivalence classing and query expansion



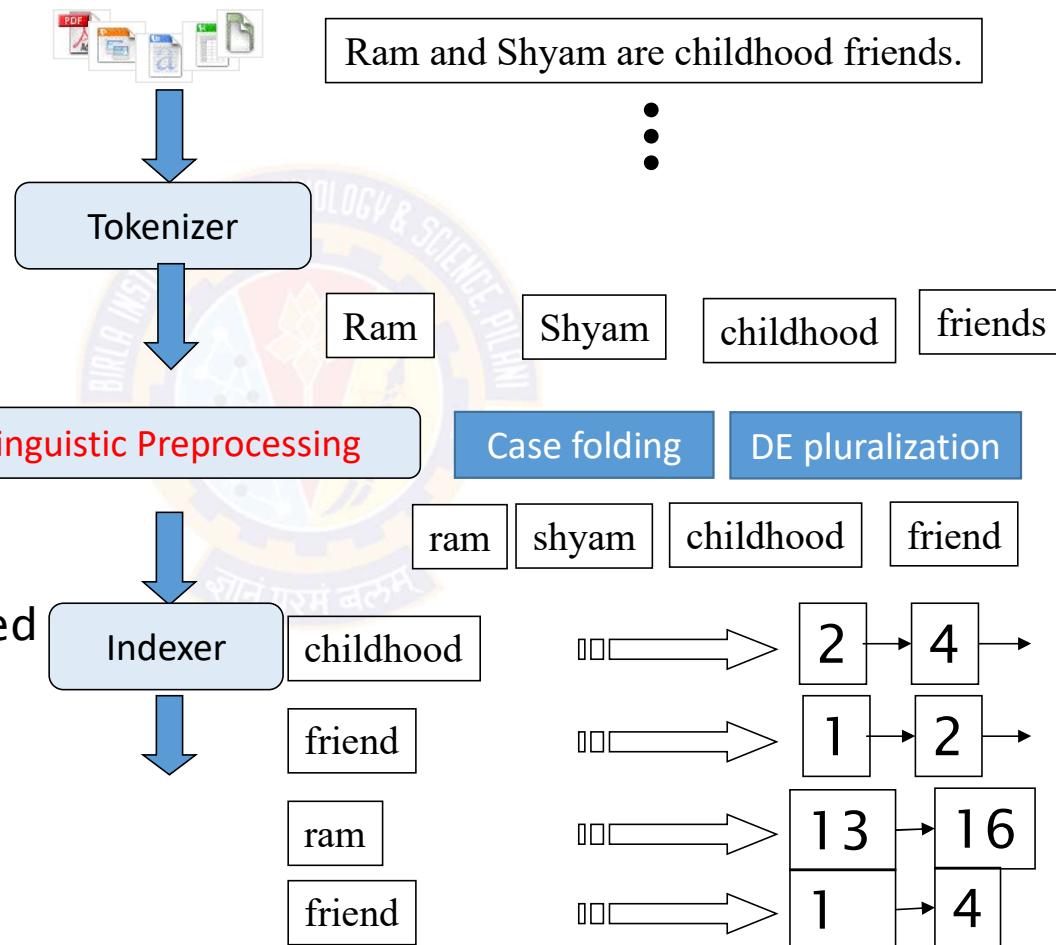
Tolerant Retrieval

- Some words may have different representations in the indexed documents.
- For example the word Ph.D may be written as Ph.D or PHD or PhD or phd
- When the user enters a query as PHD, documents contains all forms of this word must be returned as answer.



Information Retrieval Pipeline

Documents collected from various sources

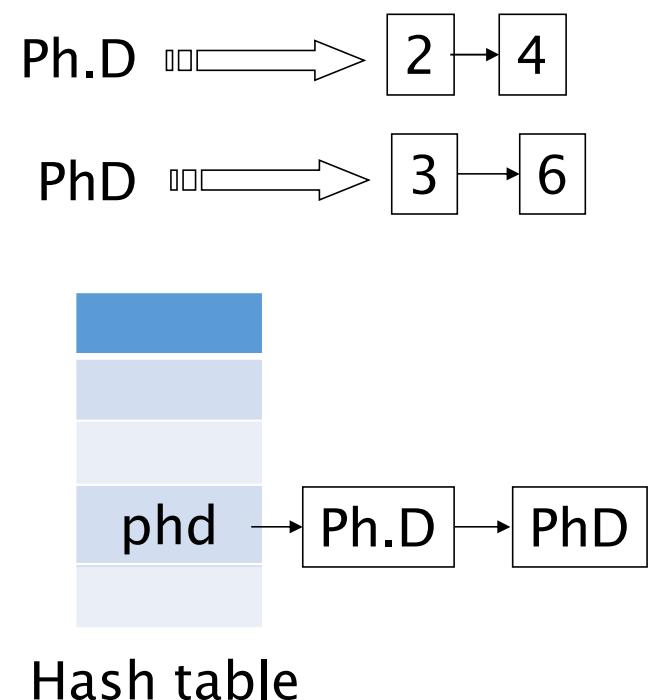


Normalization

- The process of normalization is to **reduce multiple tokens to the same canonical term**, such that matches occur despite superficial differences.
- For example suppose you have variant forms of writing the same word like Ph.D,PhD which gets mapped to a single token as phd.
- **Equivalence classing** is predominantly used technique for normalization. Deleting the periods, hyphens, Accents etc..

Query expansion / Asymmetric expansion

- For each term, t , in a query, expand the query with synonyms and related words of t from the thesaurus.
- Powerful but less efficient in terms of space
- These hand crafted terms is called as **Thesauri**.
- Handle Synonyms and Homonyms





Thank You!

In our next session: Stemming



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stemming

Prof. Aruna Malapati

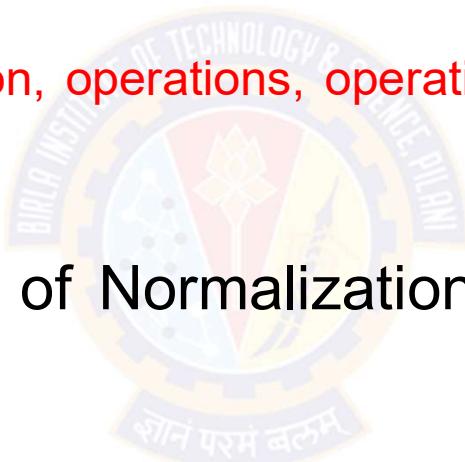
Learning objectives

- Explain stemming
- Apply Porter stemmer
- Analyze the effects of stemming



Stemming

- Stemming is the process of reducing inflectional form of words to their root form.
 - Example words like operation, operations, operational, operating can be reduce to operati (root word)
- Stemming is crude form of Normalization in which the suffixes are removed.
- The advantage of suffix stripping is to reduce the total number of terms in the inverted index resulting in a smaller size and complexity of the data in the system.



Porter Stemmer

- A consonant in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant.
- Any letter not a consonant is a Vowel.
- All the words in English are of the form C(VC)^mV where m is measure of any word or word part when represented in this form (VC).

Examples:

m=0 TR, EE, TREE, Y, BY.

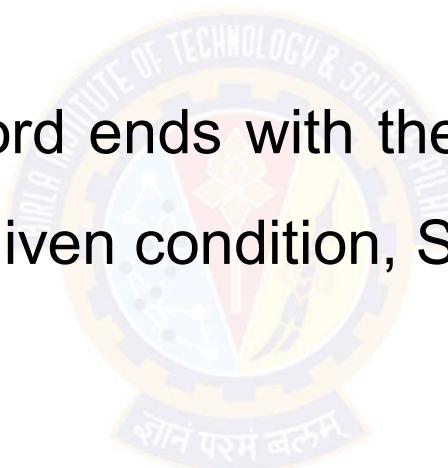
m=1 TROUBLE, OATS, TREES

m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

Porter Stemmer (contd..)

- The rules for removing a suffix will be given in the form
(condition) S1 -> S2
- This means that if a word ends with the suffix S1, and the stem
before S1 satisfies the given condition, S1 is replaced by S2.

(m > 1) EMENT ->



Porter Stemmer (contd..)

- The 'condition' part may also contain the following:
 - *S - the stem ends with S (and similarly for the other letters).
 - *v* - the stem contains a vowel.
 - *d - the stem ends with a double consonant (e.g. -TT, -SS).
 - *o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Porter Stemmer (contd..)

Step 1a

SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
	ties -> ti
SS -> SS	caress -> caress
S ->	cats -> cat

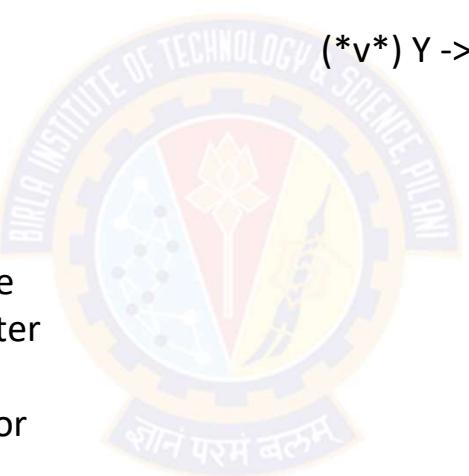
Step 1c

(*v*) Y -> I

happy -> happi
sky -> sky

Step 1b

(m>0) EED -> EE	feed -> feed
(*v*) ED ->	agreed -> agree
	plastered -> plaster
(*v*) ING ->	bled -> bled
	motoring -> motor
	sing -> sing



- Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Effect of stemming

- Suffix stripping of a vocabulary of 10,000 words

Number of words reduced in step 1: 3597

" 2: 766

" 3: 327

" 4: 2424

" 5: 1373

Number of words not reduced: 3650

- The resulting vocabulary of stems contained 6370 distinct entries.
- Thus the suffix stripping process **reduced the size of the vocabulary by about one third.**



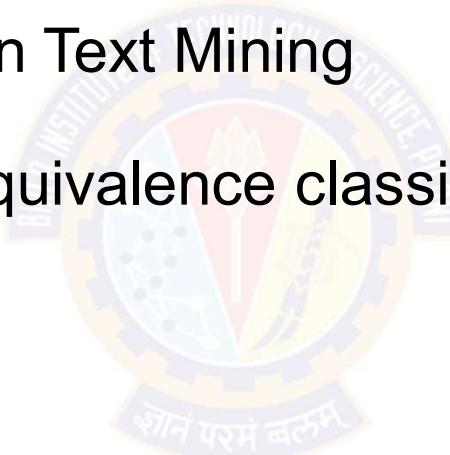
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Tolerant Retrieval using Normalization

Prof.Aruna Malapati

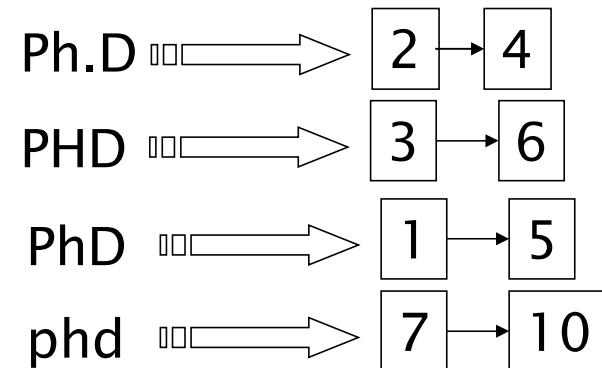
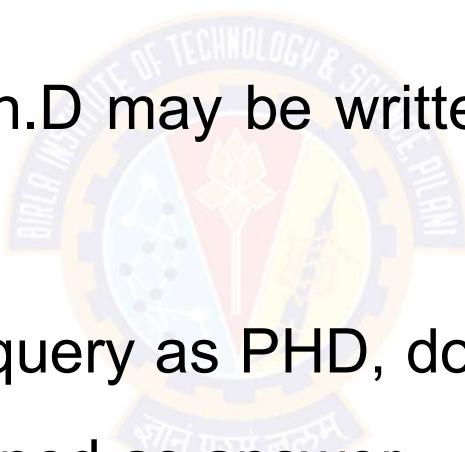
Learning objectives

- Explain Tolerant Retrieval
- Define Normalization in Text Mining
- Normalization using equivalence classing and query expansion



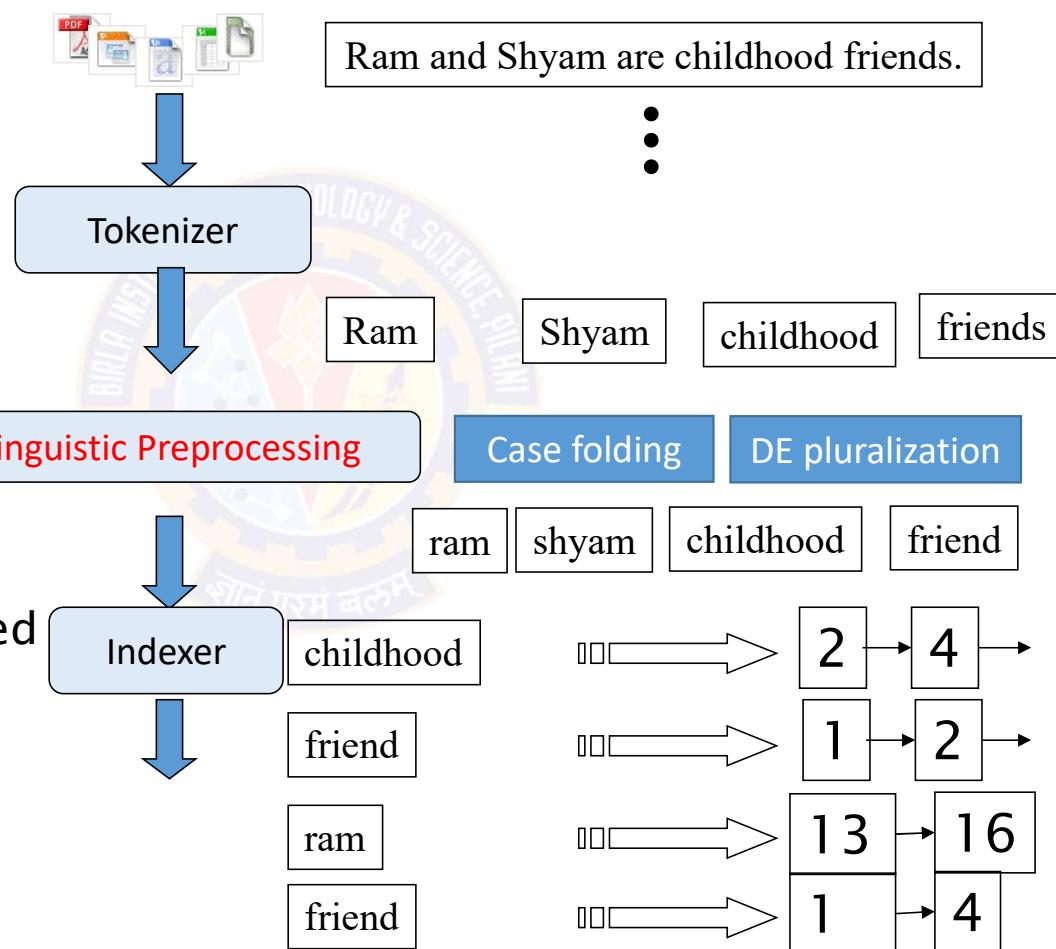
Tolerant Retrieval

- Some words may have different representations in the indexed documents.
- For example the word Ph.D may be written as Ph.D or PHD or PhD or phd
- When the user enters a query as PHD, documents contains all forms of this word must be returned as answer.



Information Retrieval Pipeline

Documents collected from various sources

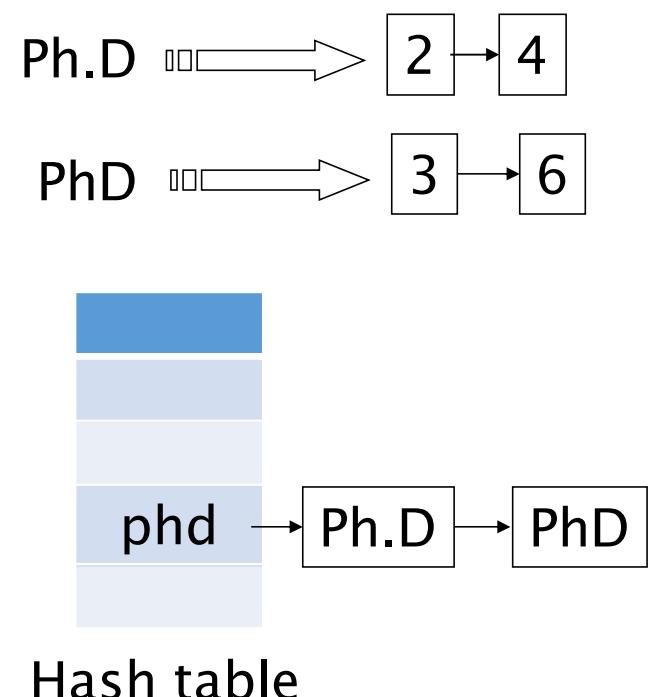


Normalization

- The process of normalization is to **reduce multiple tokens to the same canonical term**, such that matches occur despite superficial differences.
- For example suppose you have variant forms of writing the same word like Ph.D,PhD which gets mapped to a single token as phd.
- **Equivalence classing** is predominantly used technique for normalization. Deleting the periods, hyphens, Accents etc..

Query expansion / Asymmetric expansion

- For each term, t , in a query, expand the query with synonyms and related words of t from the thesaurus.
- Powerful but less efficient in terms of space
- These hand crafted terms is called as **Thesauri**.
- Handle Synonyms and Homonyms





Thank You!

In our next session: Stemming



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stemming

Prof. Aruna Malapati

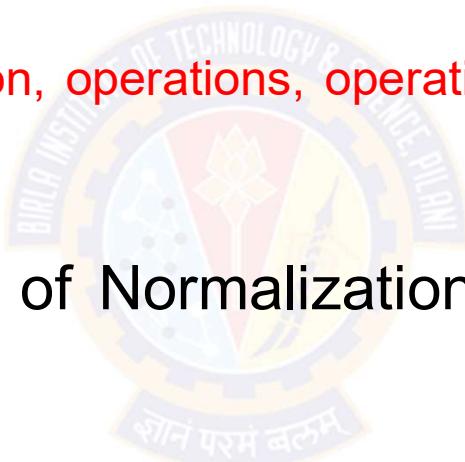
Learning objectives

- Explain stemming
- Apply Porter stemmer
- Analyze the effects of stemming



Stemming

- Stemming is the process of reducing inflectional form of words to their root form.
 - Example words like operation, operations, operational, operating can be reduce to operati (root word)
- Stemming is crude form of Normalization in which the suffixes are removed.
- The advantage of suffix stripping is to reduce the total number of terms in the inverted index resulting in a smaller size and complexity of the data in the system.



Porter Stemmer

- A consonant in a word is a letter other than A, E, I, O or U, and other than Y preceded by a consonant.
- Any letter not a consonant is a Vowel.
- All the words in English are of the form C(VC)^mV where m is measure of any word or word part when represented in this form (VC).

Examples:

m=0 TR, EE, TREE, Y, BY.

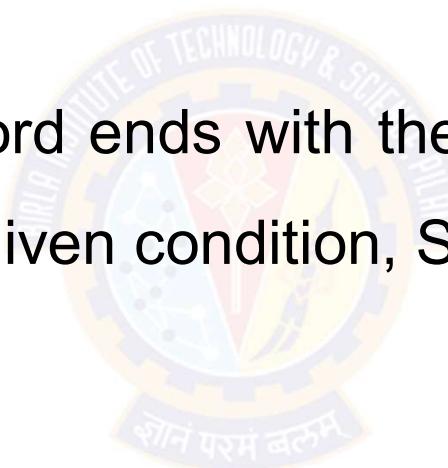
m=1 TROUBLE, OATS, TREES

m=2 TROUBLES, PRIVATE, OATEN, ORRERY.

Porter Stemmer (contd..)

- The rules for removing a suffix will be given in the form
(condition) S1 -> S2
- This means that if a word ends with the suffix S1, and the stem
before S1 satisfies the given condition, S1 is replaced by S2.

(m > 1) EMENT ->



Porter Stemmer (contd..)

- The 'condition' part may also contain the following:
 - *S - the stem ends with S (and similarly for the other letters).
 - *v* - the stem contains a vowel.
 - *d - the stem ends with a double consonant (e.g. -TT, -SS).
 - *o - the stem ends cvc, where the second c is not W, X or Y (e.g. -WIL, -HOP).

Porter Stemmer (contd..)

Step 1a

SSES -> SS	caresses -> caress
IES -> I	ponies -> poni
	ties -> ti
SS -> SS	caress -> caress
S ->	cats -> cat

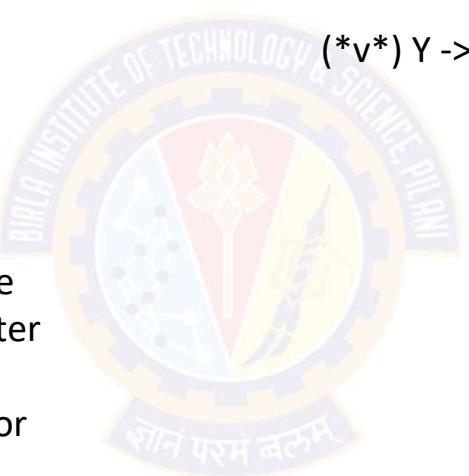
Step 1c

(*v*) Y -> I

happy -> happi
sky -> sky

Step 1b

(m>0) EED -> EE	feed -> feed
	agreed -> agree
(*v*) ED ->	plastered -> plaster
	bled -> bled
(*v*) ING ->	motoring -> motor
	sing -> sing



- Step 1 deals with plurals and past participles. The subsequent steps are much more straightforward.

Effect of stemming

- Suffix stripping of a vocabulary of 10,000 words

Number of words reduced in step 1: 3597

" 2: 766

" 3: 327

" 4: 2424

" 5: 1373

Number of words not reduced: 3650

- The resulting vocabulary of stems contained 6370 distinct entries.
- Thus the suffix stripping process **reduced the size of the vocabulary by about one third.**



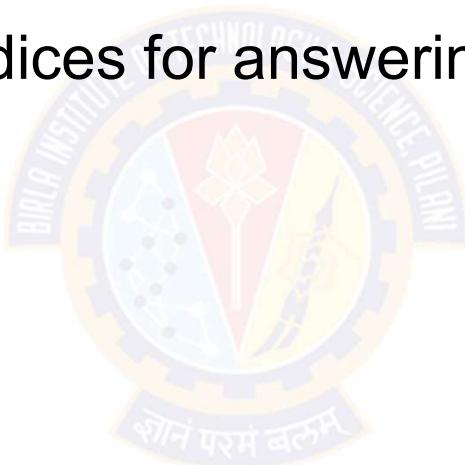
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Types of queries and indexing

Prof.Aruna Malapati

Learning objectives

- List variety of queries that search engine handles
- Identify appropriate indices for answering variant queries



Variety of queries handled by search engine

➤ Boolean queries - *Inverted Index*

➤ Phrase queries - " "

➤ Positional queries - " "  k " " Hyderabad

➤ Wild card queries - a*

Phrase Queries

- Queries of the form “BITS HYDERABAD” where the word order needs to maintained.
- Two approaches
 - Biword Index
 - Positional Index



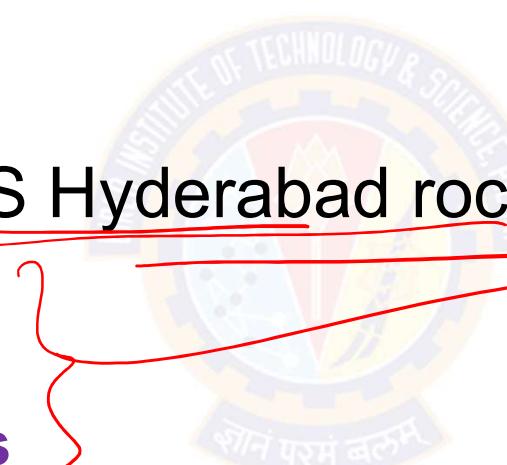
Biword Index

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

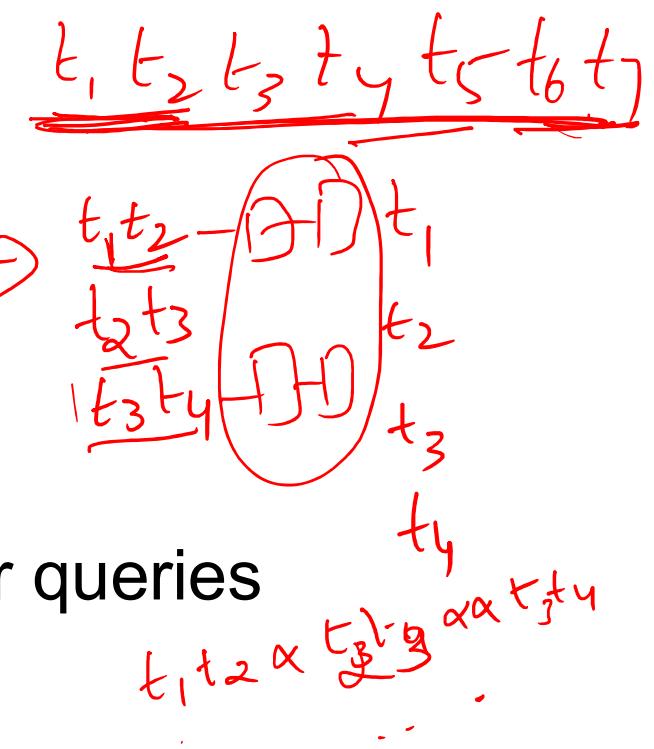
- For example “BITS Hyderabad rocks”

➤ BITS Hyderabad

➤ Hyderabad rocks



- Disadvantage: False positives for longer queries

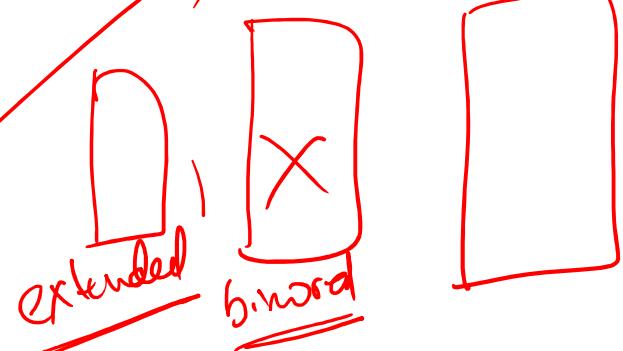


Extended biwords

- Parse the indexed text and perform part-of-speech-tagging.
- Bucket the terms into (say) Nouns (N) and articles/prepositions (X).
- Call any string of terms of the form NX*N an extended biword.
 - Each such extended biword is now made a term in the dictionary.
- Example: percyJackson and the torch

N	X	X	N
---	---	---	---
- Query processing: parse it into N's and X's
 - Segment query into enhanced biwords
 - Look up in index: percyJackson torch

t - percy Jackson torch





Thank You!

In our next session: Positional Index





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Positional Index

Prof.Aruna Malapati

Positional index

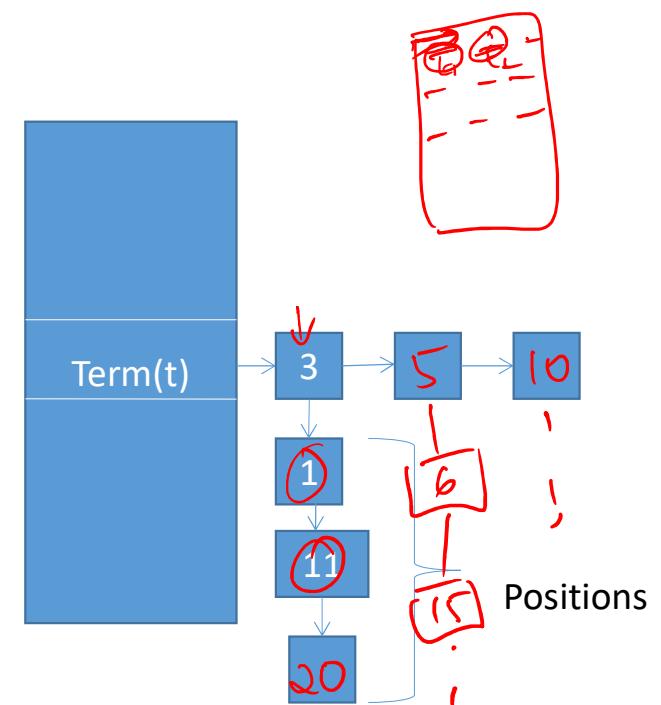
- In the postings, store for each **term** the position(s) in which tokens of it appear:

t , number of docs containing **term**

doc_1 : position₁, position₂ ... ;

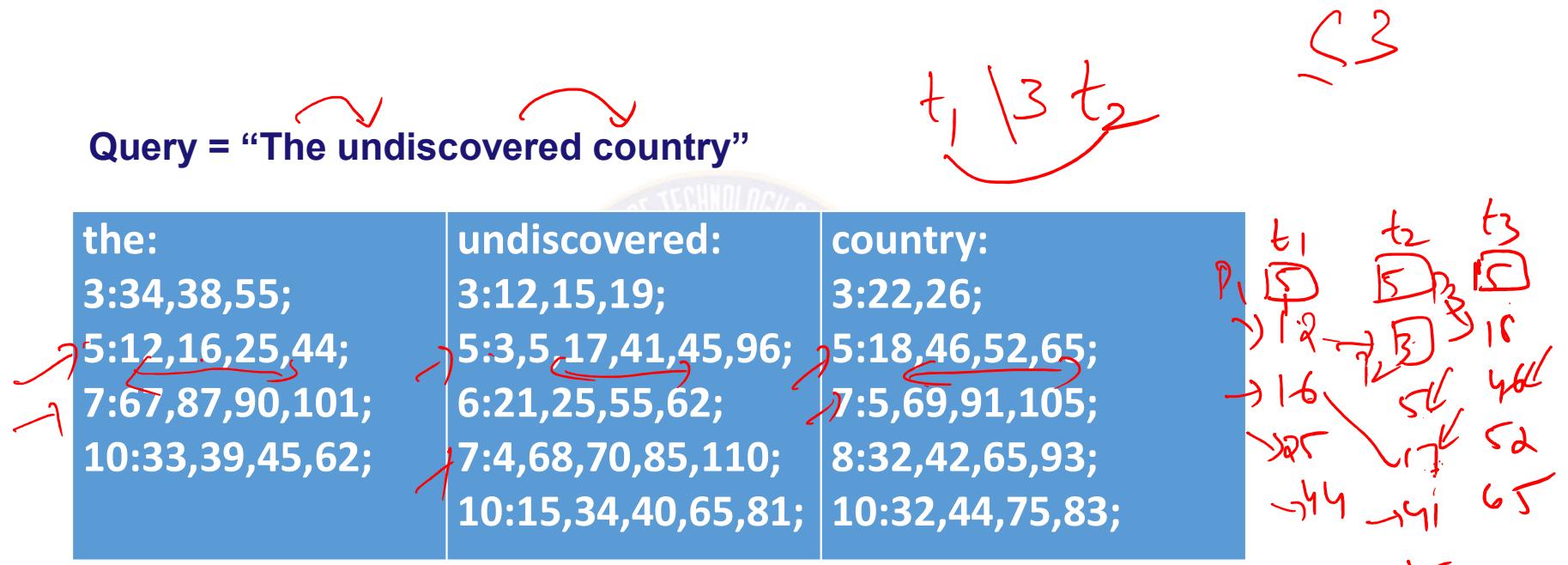
doc_2 : position₁, position₂ ... ;

etc. >



Example

Query = “The undiscovered country”



The occurrence of “The undiscovered country” is found in the following
Doc 5 (16,17,18) and (44,45,46)
Doc 7 (67,68,69)

5 (16,17,18)
(44,45,46)



Thank You!

In our next session: Wildcard queries



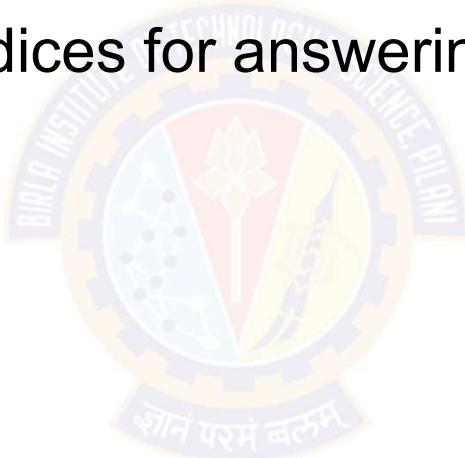
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Wildcard Queries

Prof.Aruna Malapati

Learning objectives

- List and define wildcard queries
- Identify appropriate indices for answering wildcard queries



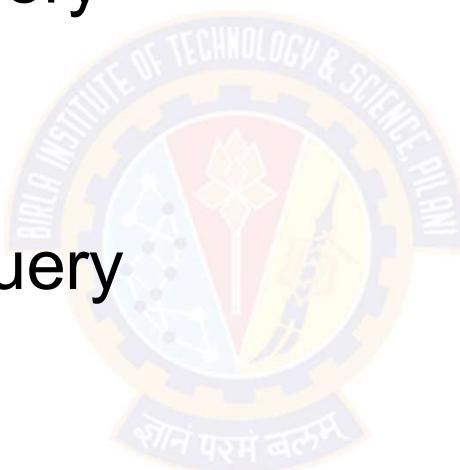
Wildcard queries

➤ Trailing wildcard query

➤ Ex: a^{*}

➤ Leading wildcard query

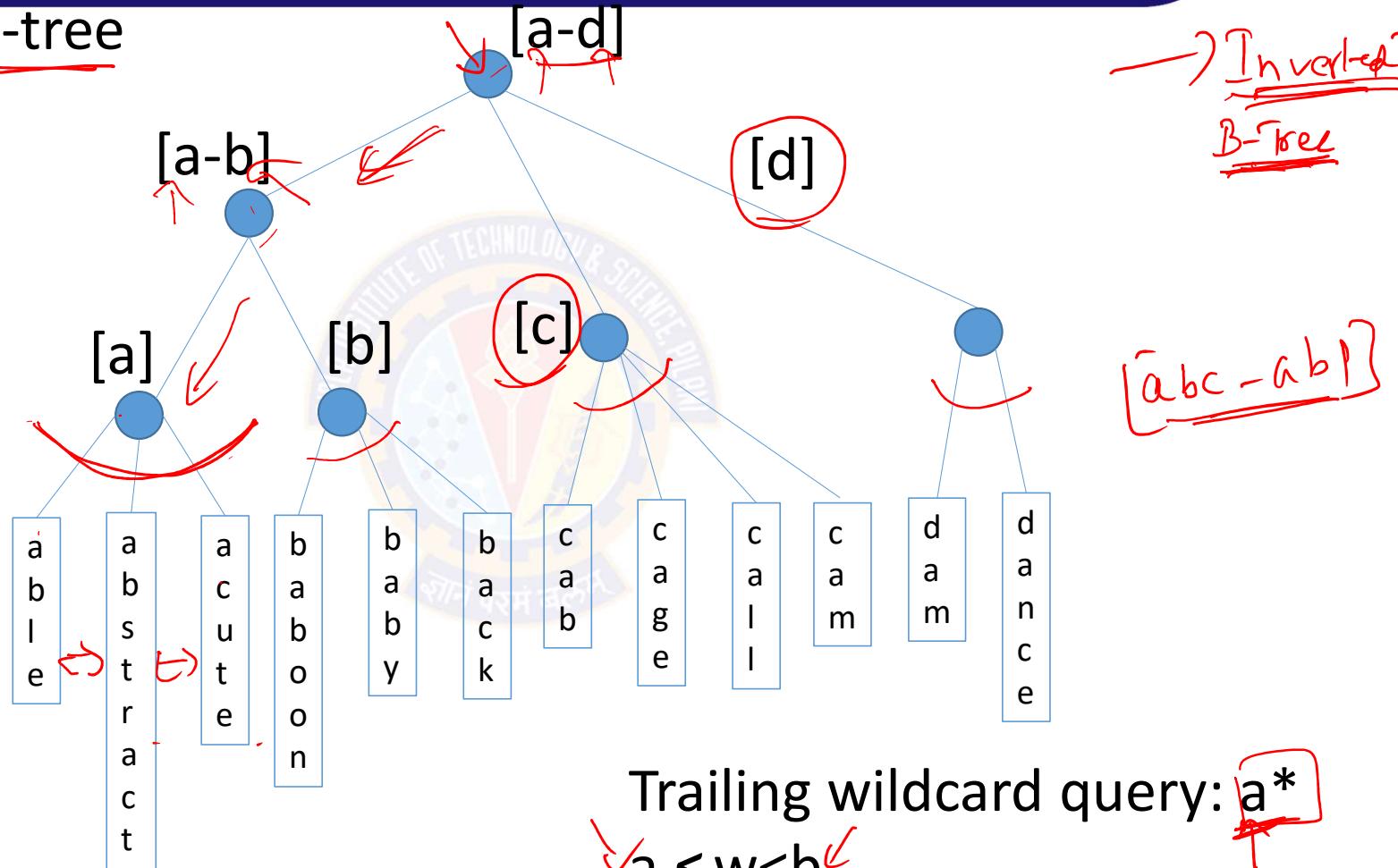
➤ Ex *a



Trailing wildcard query

able
abstract
acute
baboon
baby
back
cab
cage
call
cam
dam
dance

[2-4] B-tree



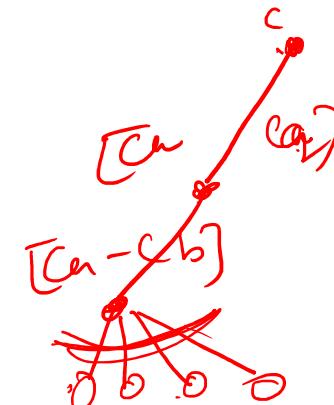
Trailing wildcard query: a^*
 $\checkmark a \leq w < b \checkmark$

Wildcard queries

Trailing wild card queries

➤ ca* (e.g cab,cal,cam,can,cap,car,cat)

➤ Walk down the tree following c,a



➤ Retrieve all words w such that: ca ≤ w < cb (i.e. all the words having prefix "ca")

➤ Let set of these terms be W.

➤ Use inverted index to retrieve documents containing terms in W



Thank You!

In our next session: Leading Wildcard queries



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Leading Wildcard Queries

Prof.Aruna Malapati

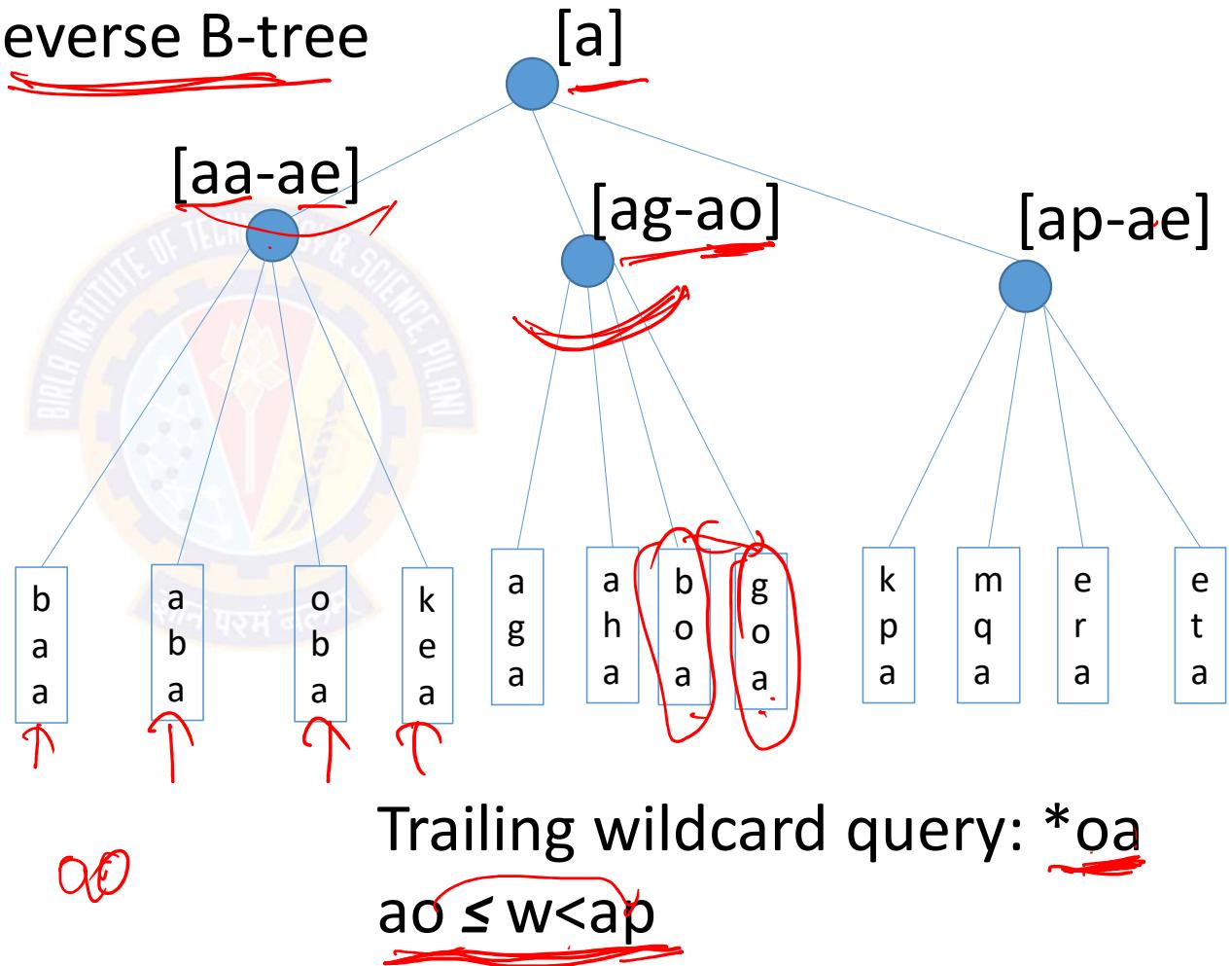
Learning objectives

- Define a leading wildcard query
- Identify appropriate indices for leading answering wildcard queries



Leading wildcard queries

aba		aab
aga		aba
aha		abo
baa	goa goa	aek
boa		aga
era		aha
eta	Reverse the Terms and sort	aob
goa		aog
kea		apk
kpa		aqm
mqa		are
oba		ate





Thank You!

In our next session: K-Gram Index



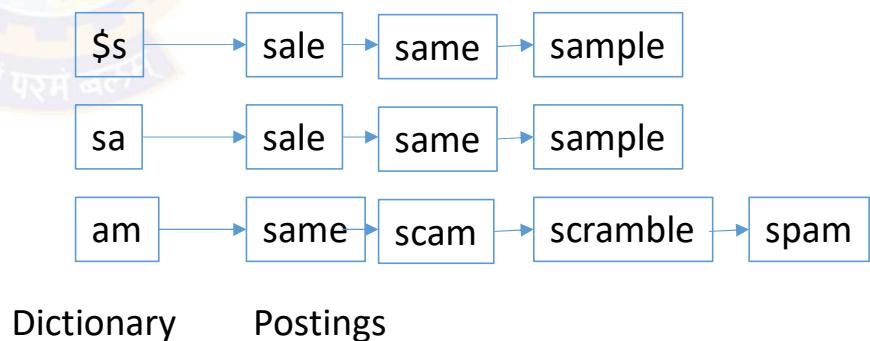
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

K-Gram Index

Prof.Aruna Malapati

K-gram index

- Query: \$sam*
- 2-grams from the query {\$s and sa and am}
- Fetch all the terms found in the posting list of the three K-grams using the merge algorithm



Postprocessing

- Consider using the 3-gram index described for the query
red*





Thank You!

In our next session: Ranked Retrieval using vector space model



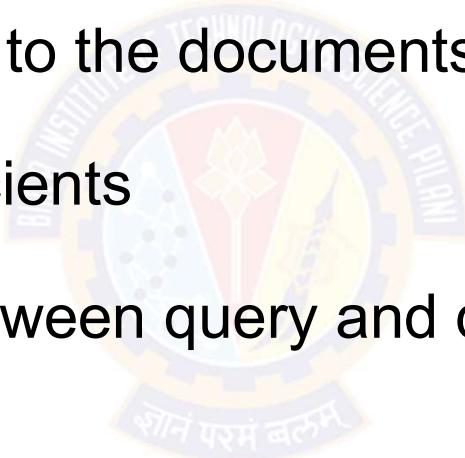
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked Retrieval

Prof. Aruna Malapati

Learning objectives

- Define Ranked Retrieval and vector space model
- Relate vector notation to the documents and queries
- Types of vector coefficients
- Compute similarity between query and documents



Ranked Retrieval

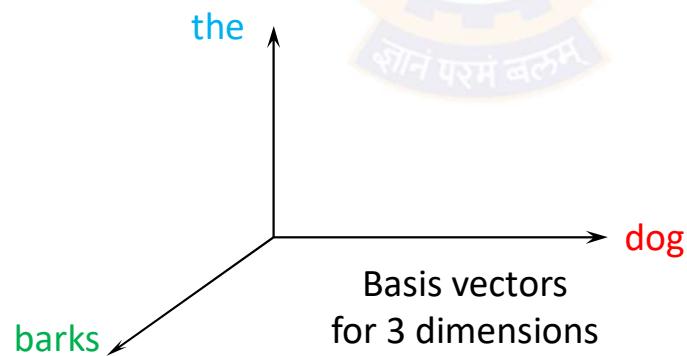
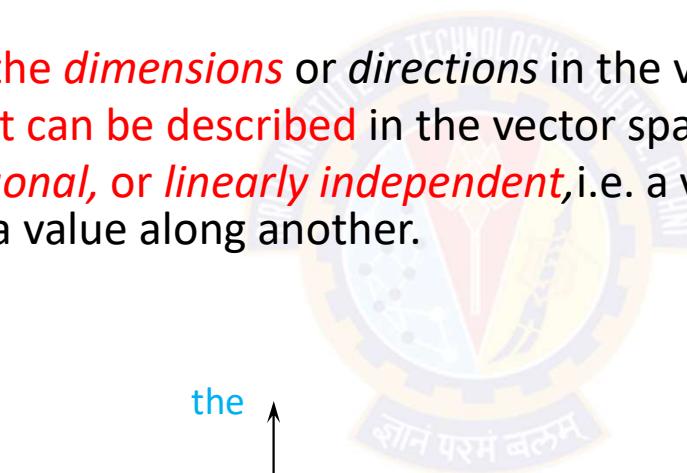
- The user enters a **free text query** and we would like to return the results in a ranked order.
- Hence we need a **scoring scheme** to compute the relevance of the document to the user query.
- A simple scoring scheme in the range [0-1].

Vector Space Model

- Any text object can be represented by a term vector
 - Examples: Documents, queries, sentences,
- Similarity is determined by distance in a vector space
 - Example: The cosine of the angle between the vectors

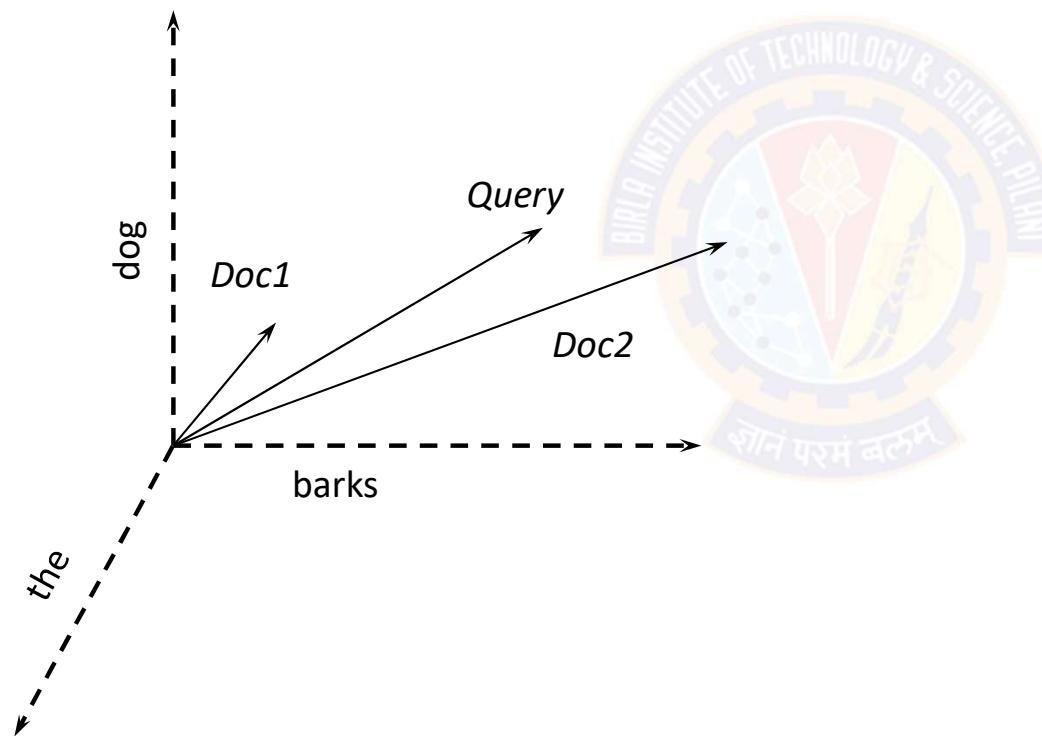
Vector Space Representation from linear algebra perspective

- Formally, a vector space is defined by a set of **linearly independent basis vectors**.
- Basis vectors:
 - correspond to the *dimensions* or *directions* in the vector space;
 - determine what can be described in the vector space; and
 - must be *orthogonal*, or *linearly independent*, i.e. a value along one dimension implies nothing about a value along another.



Vector Coefficients

- How to represent the documents and queries?



Doc1: the dog barks <1 1 1>

Doc2: the dog dog barks barks barks <1 2 3>

Query: the dog dog barks barks <1 2 2>

Vector Coefficients

- The coefficients (vector elements, term weights) represent term presence, importance, or “representativeness”
- The vector space model does not specify how to set term weights.
- Commonly used coefficients are :
 - Raw term frequency(tf)
 - Term Frequency – Inverse Document Frequency (TF-Idf)

Vector Space Similarity

Sim(X,Y)

Inner product

(# nonzero dimensions)

Dice coefficient

(Length normalized

Inner Product)

Cosine coefficient

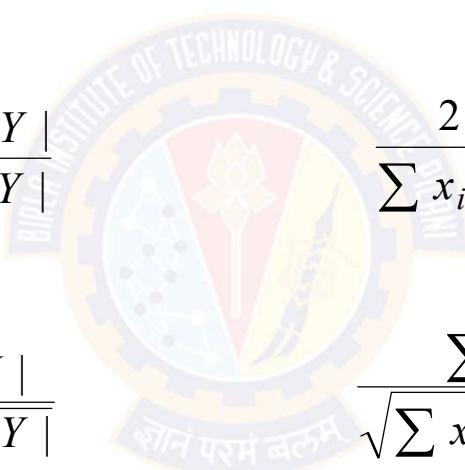
(like Dice, but lower
penalty with diff # features)

Jaccard coefficient

(like Dice, but penalizes
low overlap cases)

Binary Term Vectors

$$|X \cap Y|$$



Weighted Term Vectors

$$\sum x_i \cdot y_i$$

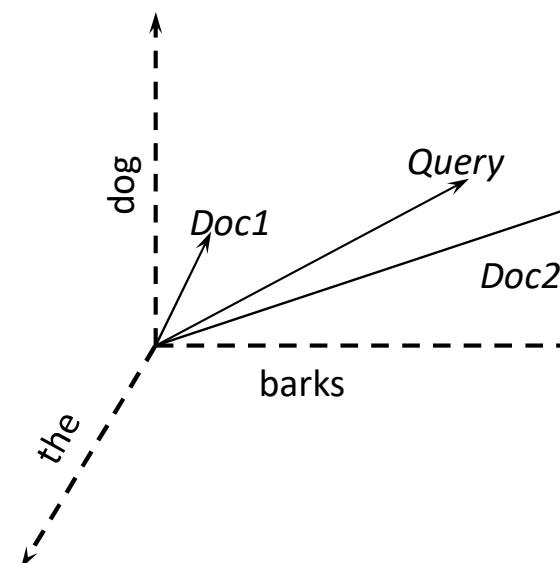
$$\frac{2 \sum x_i y_i}{\sum x_i^2 + \sum y_i^2}$$

$$\frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$$

$$\frac{|X \cap Y|}{\sqrt{|X|} \sqrt{|Y|}}$$

$$\frac{|X \cap Y|}{|X| + |Y| - |X \cap Y|}$$

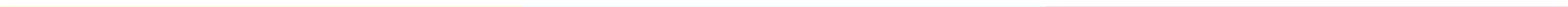
$$\frac{\sum x_i \cdot y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i \cdot y_i}$$





Thank You!

In our next session: Vector Space Model





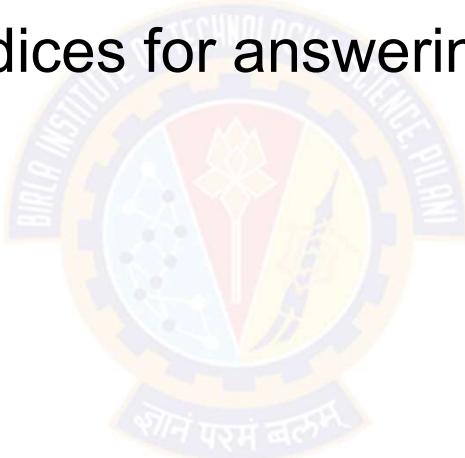
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Wildcard Queries

Prof.Aruna Malapati

Learning objectives

- List and define wildcard queries
- Identify appropriate indices for answering wildcard queries



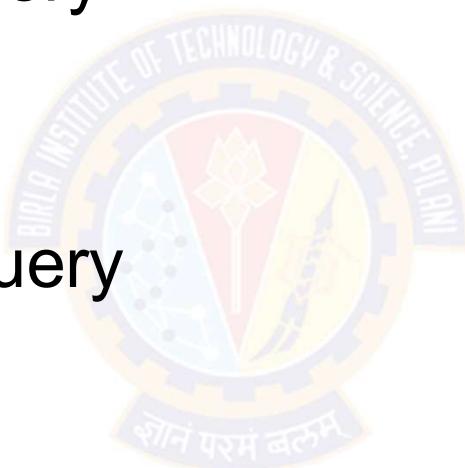
Wildcard queries

➤ Trailing wildcard query

➤ Ex: a*

➤ Leading wildcard query

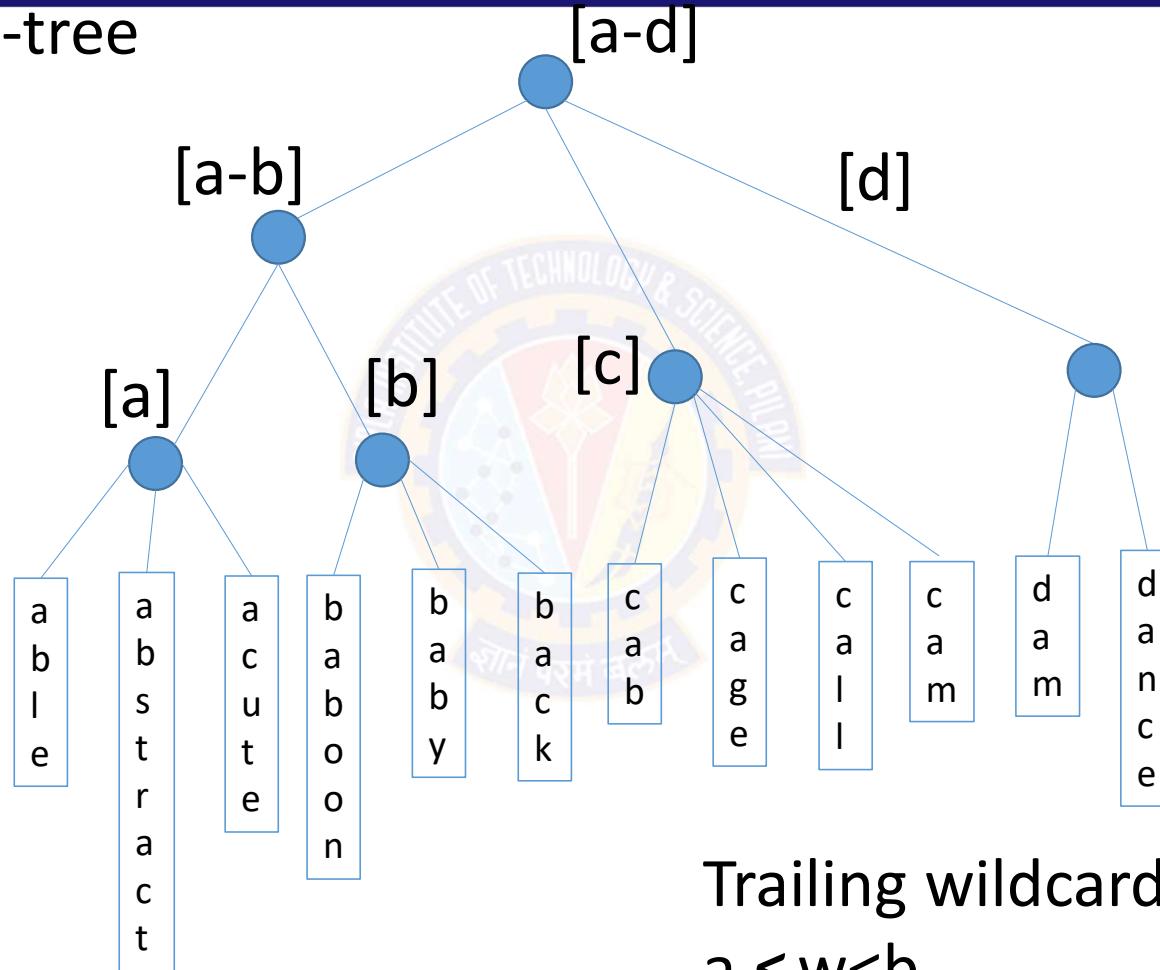
➤ Ex *a



Trailing wildcard query

able
abstract
acute
baboon
baby
back
cab
cage
call
cam
dam
dance

[2-4] B-tree



Wildcard queries

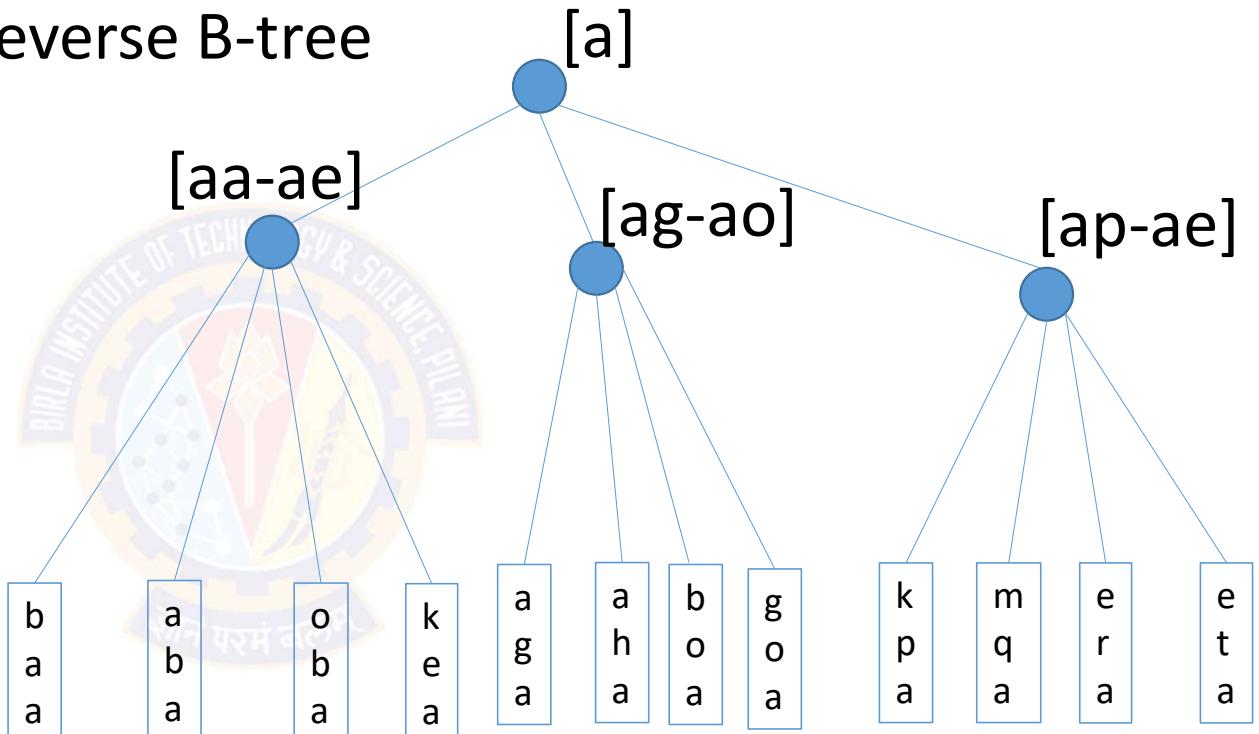
Trailing wild card queries

- ca* (e.g cab,cal,cam,can,cap,car,cat)
- Walk down the tree following c,a
- Retrieve all words w such that: $ca \leq w < cb$ (i.e. all the words having prefix “ca”)
- Let set of these terms be W.
- Use inverted index to retrieve documents containing terms in W

Leading wildcard queries

aba	aab
aga	aba
aha	abo
baa	ae
boa	ag
era	ah
eta	ao
goa	aog
kea	apk
kpa	aqm
mqa	are
oba	ate

[2-4] Reverse B-tree



Reverse
the
Terms
and
sort

Trailing wildcard query: *oa
ao ≤ w < ab



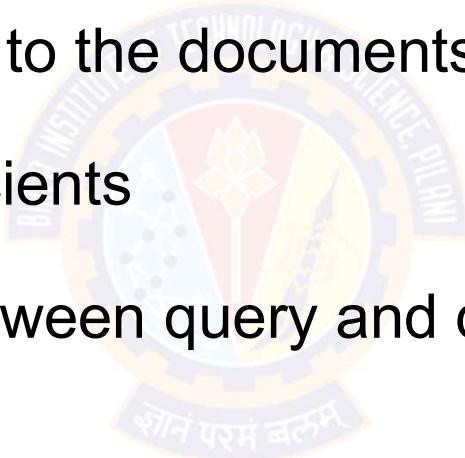
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked Retrieval

Prof. Aruna Malapati

Learning objectives

- Define Ranked Retrieval and vector space model
- Relate vector notation to the documents and queries
- Types of vector coefficients
- Compute similarity between query and documents



Ranked Retrieval

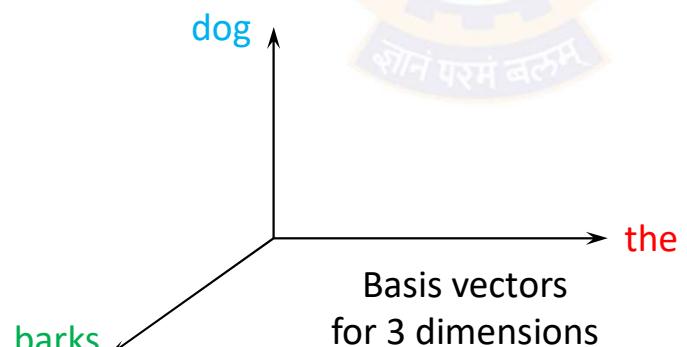
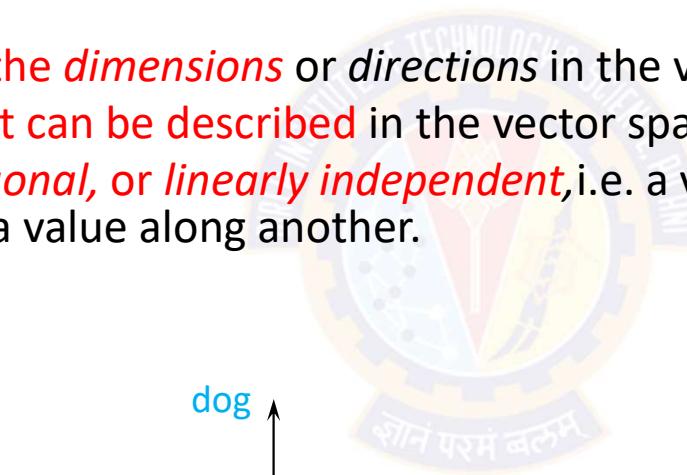
- The user enters a **free text query** and we would like to return the results in a ranked order.
- Hence we need a **scoring scheme** to compute the relevance of the document to the user query.
- A simple scoring scheme in the range [0-1].

Vector Space Model

- Any text object can be represented by a term vector
 - Examples: Documents, queries, sentences,
- Similarity is determined by distance in a vector space
 - Example: The cosine of the angle between the vectors

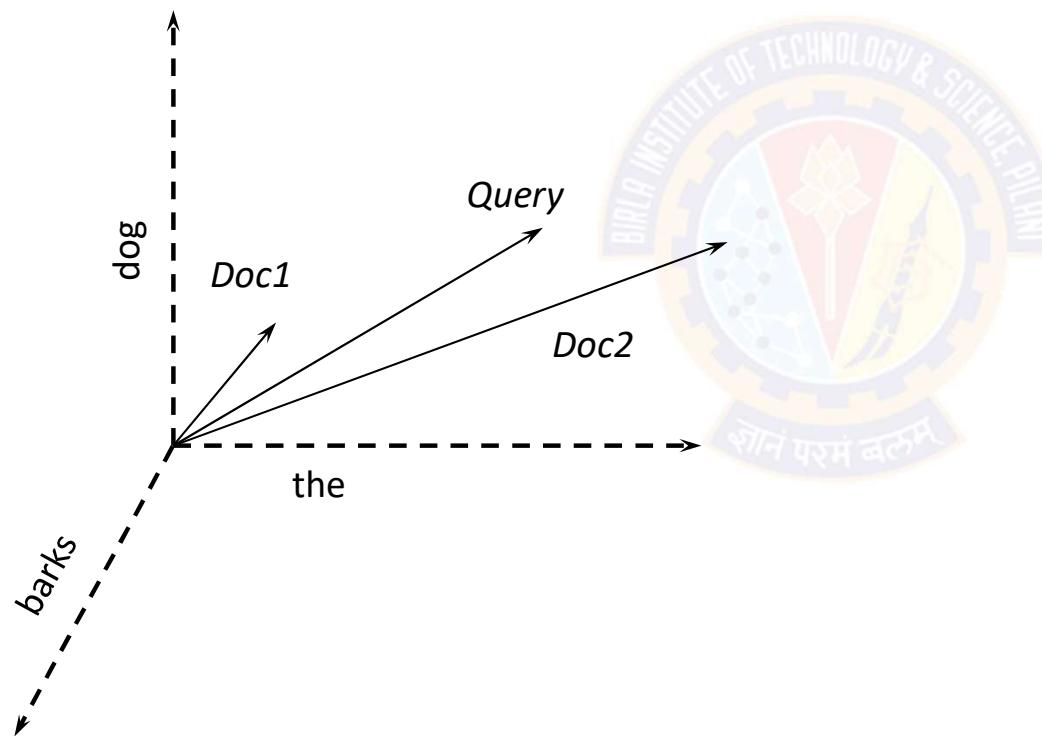
Vector Space Representation from linear algebra perspective

- Formally, a vector space is defined by a set of **linearly independent basis vectors**.
- Basis vectors:
 - correspond to the *dimensions* or *directions* in the vector space;
 - determine what can be described in the vector space; and
 - must be *orthogonal*, or *linearly independent*, i.e. a value along one dimension implies nothing about a value along another.



Vector Coefficients

- How to represent the documents and queries?



Doc1: the dog barks <1 1 1>

Doc2: the dog dog barks barks barks <1 2 3>

Query: the dog dog barks barks <1 2 2>

Vector Coefficients

- The coefficients (vector elements, term weights) represent term presence, importance, or “representativeness”
- The vector space model does not specify how to set term weights.
- Commonly used coefficients are :
 - Raw term frequency(tf)
 - Term Frequency – Inverse Document Frequency (TF-IDF)
 - Collection Frequency(CF)

Vector Space Similarity

Sim(X,Y)

Inner product

Dice coefficient

Cosine coefficient

Jaccard coefficient

Weighted Term Vectors

$$\sum x_i \cdot y_i$$

$$\frac{2 \sum x_i y_i}{\sum x_i^2 + \sum y_i^2}$$

$$\frac{\sum x_i \cdot y_i}{\sqrt{\sum x_i^2} \cdot \sqrt{\sum y_i^2}}$$

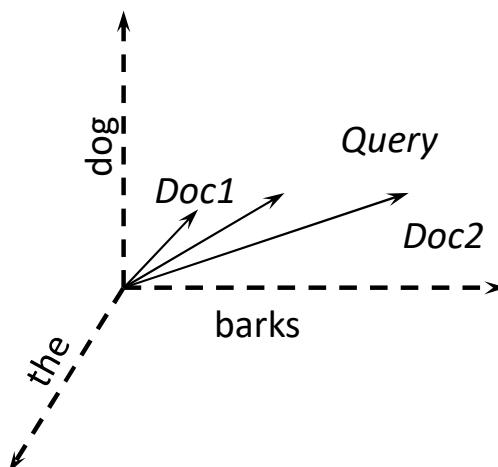
$$\frac{\sum x_i \cdot y_i}{\sum x_i^2 + \sum y_i^2 - \sum x_i \cdot y_i}$$

Assume that the query vector is denoted by X and Y is the document vector.

Example of similarity with vector coefficient as term frequency

	Term Weights		
Query	1	2	2

	Term Weights		
Doc 1	1	1	1
Doc 2	1	2	3



$$\text{Inner product } S(Q, D1) = 1 * 1 + 2 * 1 + 2 * 1 = 5$$

$$\text{Inner product } S(Q, D2) = 1 * 1 + 2 * 2 + 2 * 3 = 11$$

$$\text{Dice coefficient } S(Q, D1) = \frac{2(1 * 1 + 2 * 1 + 2 * 1)}{(1^2 + 2^2 + 2^2) + (1^2 + 1^2 + 1^2)}$$

$$\text{Dice coefficient } S(Q, D2) = \frac{2(1 * 1 + 2 * 2 + 2 * 3)}{(1^2 + 2^2 + 2^2) + (1^2 + 2^2 + 3^2)}$$

$$\text{Cosine coefficient } S(Q, D1) = \frac{1 * 1 + 2 * 1 + 2 * 1}{\sqrt{(1^2 + 2^2 + 2^2)} * \sqrt{(1^2 + 1^2 + 1^2)}}$$

$$\text{Cosine coefficient } S(Q, D2) = \frac{1 * 1 + 2 * 2 + 2 * 3}{\sqrt{(1^2 + 2^2 + 2^2)} * \sqrt{(1^2 + 2^2 + 3^2)}}$$

Advantages and Disadvantages

- Simplicity: Easy to implement
- Ability to incorporate any kind of term weights
- Can measure similarities between almost anything:
 - documents and queries, documents and documents, queries and queries, sentences and sentences, etc.
- The vector space model is the most **popular retrieval model (today)**
- Assumes independence relationship among terms.
- The weighting is subjective.



Thank You!

In our next session: Ranked Retrieval using TF-IDF



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Ranked retrieval

Prof. Aruna Malapati

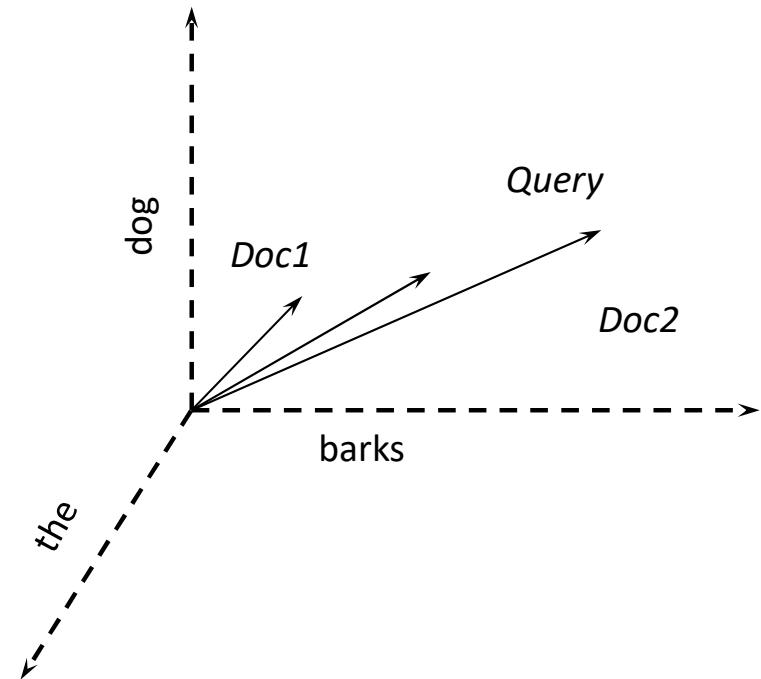
Learning objectives

- List and problems with Term Frequency as the vector co-efficients
- Define and apply TF-IDF vector co-efficients



Limitations of Term frequency and cosine angle

- Term Frequency represents the relative importance of the word in the document.
- A term appearing too often is not very important. For example stop words like a,an,the,etc...



TF Weighting

- The log frequency weight of term t in document d is

$$w_{t,d} = \begin{cases} 1 + \log_{10} tf_{t,d}, & \text{if } tf_{t,d} > 0 \\ 0, & \text{otherwise} \end{cases}$$

$0 \rightarrow 0, 1 \rightarrow 1, 2 \rightarrow 1.3, 10 \rightarrow 2, 1000 \rightarrow 4, \text{ etc.}$

- Score for a document-query pair: sum over terms t in both q and d :

$$\text{Score}(q,d) = \sum_{t \in q \cap d} W_{t,d}$$

Inverse Document Frequency (IDF)

- The presence of a rare word is more important than a stop word.
- The rarity of a term is quantified using IDF.
- df_t is the document_frequency of t : the number of documents that contain t
- We define the idf (inverse document frequency) of t by

$$\text{idf}_t = \log_{10} (N/\text{df}_t)$$

tf-idf weighting

- The tf-idf weight of a term is the product of its tf weight and its idf weight.

$$w_{t,d} = (1 + \log tf_{t,d}) \times \log_{10}(N / df_t)$$

- Best known weighting scheme in information retrieval
- Increases with the number of occurrences within a document
- Increases with the rarity of the term in the collection

tf-idf weighting for document and queries

Example :D1: The restaurants are in the city.

D2: The resorts are in the outskirts.

V={are, city, in, outskirts, resorts, restaurants, the}

Q= {resorts,outskirts}

$$\text{Score}(q, d) = \sum_{t \in q \cap d} \text{tf.idf}_{t,d}$$

	TF-D1	IDF-D1	TF-IDF(D1)	TF-D2	IDF-D2	TF-IDF(D2)	TF-Query	IDF-Query	TF-IDF Query
are	1+log 1=1	log2/2=0		0	1+log 1=1	log2/2=0	0	0	log2/2=0
city	1+log 1=1	log2/1=0.3		0.3	0	log2/1=0.3	0	0	log2/1=0.3
in	1+log 1=1	log2/2=0		0	1+log 1=1	log2/2=0	0	0	log2/2=0
outskirts		0 log2/1=0.3		0	1+log 1=1	log2/1=0.3	0.3	1+log 1=1	log2/1=0.3
resorts		0 log2/1=0.3		0	1+log 1=1	log2/1=0.3	0.3	1+log 1=1	log2/1=0.3
restaurants	1+log 1=1	log2/1=0.3		0.3	0	log2/1=0.3	0	0	log2/1=0.3
the	1+log 2=1.3	log2/2=0		0	1+log 2=1.3	log2/2=0	0	0	log2/2=0

tf-idf weighting has many variants

Term frequency	Document frequency	Normalization
n (natural) $tf_{t,d}$	n (no) 1	n (none) 1
I (logarithm) $1 + \log(tf_{t,d})$	t (idf) $\log \frac{N}{df_t}$	c (cosine) $\frac{1}{\sqrt{w_1^2 + w_2^2 + \dots + w_M^2}}$
a (augmented) $0.5 + \frac{0.5 \times tf_{t,d}}{\max_t(tf_{t,d})}$	p (prob idf) $\max\{0, \log \frac{N - df_t}{df_t}\}$	u (pivoted unique) $1/u$
b (boolean) $\begin{cases} 1 & \text{if } tf_{t,d} > 0 \\ 0 & \text{otherwise} \end{cases}$		b (byte size) $1/CharLength^\alpha, \alpha < 1$
L (log ave) $\frac{1 + \log(tf_{t,d})}{1 + \log(\text{ave}_{t \in d}(tf_{t,d}))}$		

- SMART Notation: denotes the combination in use in an engine, with the notation *ddd.ddd*



Thank You!

In our next session: Computing scores



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

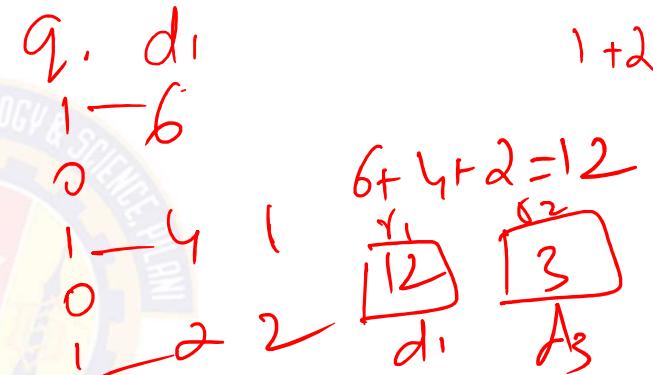
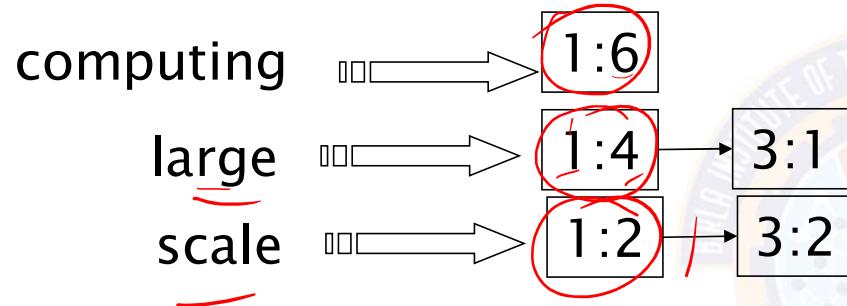
Ranked retrieval

Prof.Aruna Malapati

Document at a Time

Vocabulary {computing, data, large, mining, scale}

Query = <1 0 1 0 1>



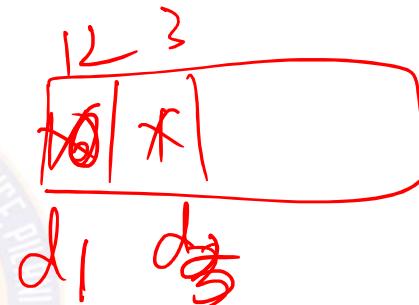
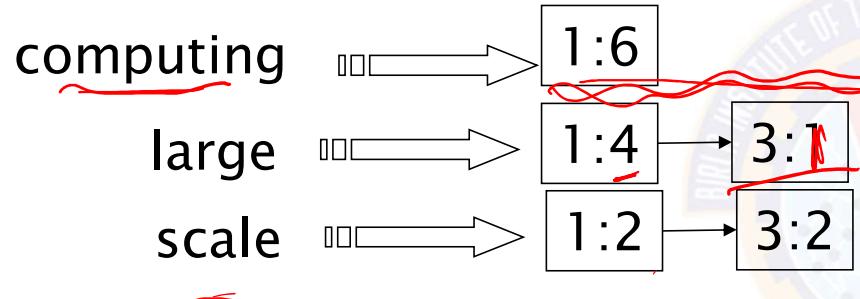
Scanning posting lists of computing, large, scale in parallel

1. Find the smallest docId, say d, among all lists
2. Compute the pointers for all postings whose docid= d
3. Forward the pointers for all postings whose docid = d
4. Repeat steps 1- 3 for next doc

Term at a time

Vocabulary {computing, data, large, mining, scale}

Query = <1 0 1 0 1>



- Scanning posting lists of computing, large, scale in one by one
- Sim(q,d) is available after scanning the last term
- Accumulates partial score when each term is scanned



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Parts of Speech Tagging

Prof.Aruna Malapati

Learning objectives

- Define the problem of Parts Of Speech tagging (POS)
- Challenges while POS tagging
- Introduce to the benchmarked PennTree dataset
- Applications
- Measuring performance of POS taggers
- Two approaches to POS tagging

Parts Of Speech Tagging (POS)

Aruna saw the saw.

NNP

VB

DT

NN

➤ Annotate each word in a sentence with a part of speech.

Sequence Data

- Till now, we assumed that the data instances are classified independently.
- More precisely, we assumed that the data is iid (identically and independently distributed)
- In many applications, the data arrives sequentially and the classes are correlated – E.g., weather prediction, speech recognition, activity recognition, etc..

What is the challenge in PoS Tagging?

- Tag ambiguous words
- Solve the lexical ambiguities
- The/DT wind/NN was/VB too/ADV strong/ADJ to/PRP
wind/VB the/DT sail/NN.
- Tag unknown words
- The/DT rural/JJ Babbitt/??? who/WP bloats/???
about/IN progress/NN and/CC growth/NN

Category of classes

- Open: vast number of new members
 - Nouns, Verbs, Adjectives, Adverbs
- Closed: small set of words
 - Determiners: **a, an, the**
 - Pronouns: **she, he, i, you, we**
 - Prepositions: **on, under, over, near, by,...**

Choosing a tagset

- Need to choose a standard set of tags to do POS tagging.
- Could pick very coarse tagset – N, V, Adj, Adv, Prep.
- More commonly used set is finer-grained.
 - Penn TreeBank II tagset has 36 word tags
 - PRP, PRP\$, VBG, VBD, JJR, JJS ...

Penn TreeBank PoS Tagset

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	<i>and, but, or</i>	SYM	symbol	<i>+, %, &</i>
CD	cardinal number	<i>one, two, three</i>	TO	“to”	<i>to</i>
DT	determiner	<i>a, the</i>	UH	interjection	<i>ah, oops</i>
EX	existential ‘there’	<i>there</i>	VB	verb, base form	<i>eat</i>
FW	foreign word	<i>mea culpa</i>	VBD	verb, past tense	<i>ate</i>
IN	preposition/sub-conj	<i>of, in, by</i>	VBG	verb, gerund	<i>eating</i>
JJ	adjective	<i>yellow</i>	VBN	verb, past participle	<i>eaten</i>
JJR	adj., comparative	<i>bigger</i>	VBP	verb, non-3sg pres	<i>eat</i>
JJS	adj., superlative	<i>wildest</i>	VBZ	verb, 3sg pres	<i>eats</i>
LS	list item marker	<i>1, 2, One</i>	WDT	wh-determiner	<i>which, that</i>
MD	modal	<i>can, should</i>	WP	wh-pronoun	<i>what, who</i>
NN	noun, sing. or mass	<i>llama</i>	WP\$	possessive wh-	<i>whose</i>
NNS	noun, plural	<i>llamas</i>	WRB	wh-adverb	<i>how, where</i>
NNP	proper noun, singular	<i>IBM</i>	\$	dollar sign	<i>\$</i>
NNPS	proper noun, plural	<i>Carolinas</i>	#	pound sign	<i>#</i>
PDT	predeterminer	<i>all, both</i>	“	left quote	<i>‘ or “</i>
POS	possessive ending	<i>'s</i>	”	right quote	<i>’ or ”</i>
PRP	personal pronoun	<i>I, you, he</i>	(left parenthesis	<i>[, (, {, <</i>
PRP\$	possessive pronoun	<i>your, one's</i>)	right parenthesis	<i>],), }, ></i>
RB	adverb	<i>quickly, never</i>	,	comma	<i>,</i>
RBR	adverb, comparative	<i>faster</i>	.	sentence-final punc	<i>. ! ?</i>
RBS	adverb, superlative	<i>fastest</i>	:	mid-sentence punc	<i>: ; ... --</i>
RP	particle	<i>up, off</i>			

POS Tagging performance evaluation

- Percentage of tags predicted correctly.
- Baseline approach: Tag every word with its most common tag and rest of the words are tagged as Noun.

Applications of POS tagging

- Text to speech conversion
- Useful as a preprocessing step of parsing



Two Methods for PoS Tagging

- Rule-based systems
- Statistical sequence models
- Hidden Markov Models





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stochastic Language Models

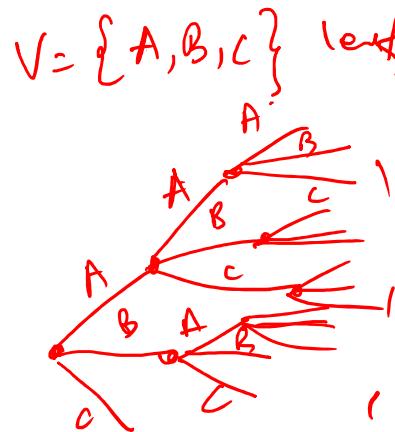
Prof.Aruna Malapati

Learning objectives

- Express the need for language models
- Jargons
- Formulate a naïve language model



Language Modelling



A=the

B=dog

C=banks

D=boy

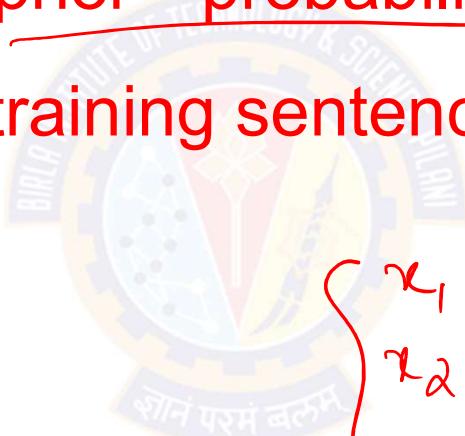
the boy banks



- The task of a language model is to **express the restrictions imposed** on the way in which words can be combined to form sentences.

Stochastic Language Model

➤ The task of a stochastic language model is to provide estimates of the prior probability of the sentence generation using the training sentences.



$$\left. \begin{array}{l} x_1 = w_1, w_2, \dots, w_n \\ x_2 = w_n, w_{12}, w_6, \dots, w_n \\ \vdots \\ \vdots \end{array} \right\}$$

Jargons

- Corpus: - set of training sentences.
- Vocabulary: finite set of words used in writing sentences.
 - For example $V = \{\text{the, dog, barks, cat, smiles, boy, play..}\}$
- V' = Set of all possible sentences in this language.

the dog barks STOP(well formed sentence)

the cat cat STOP(ill formed sentence)

the the the STOP(ill formed sentence)

the cat smiles STOP(well formed sentence)

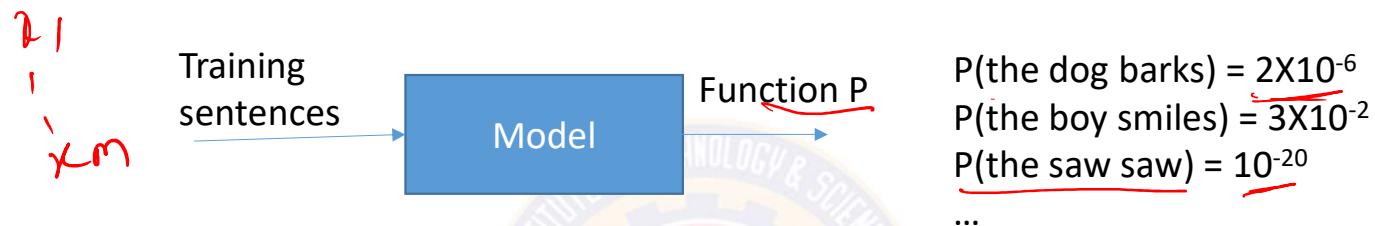
STOP

Stochastic Language Model (Contd..)

- Assume that we have a training sentences in English.
- These could be easily collected from online newspapers or webpages.
- Given these training samples the task of Language Model is learn a distribution \underline{P} over sentences in our language.
- P is going to be a function which must satisfy the following two constraints

- 1) For any sentence $s \in \mathcal{V}^+$ $P(s) \geq 0$
- 2) $\sum_{s \in \mathcal{V}^+} P(s) = 1$

Stochastic Language Model (Contd..)



- The task of function P is to **assign probability to every sentence in the language.**
- We would prefer a good **language model** to **assign high probability to a sentence which occurs in English.**

A naïve language model

- Given N training sentences, the task is to **learn** a distribution P over sentences in the language.
- For any sentence w_1, w_2, \dots, w_n let $c(w_1, w_2, \dots, w_n)$ denote the number of times this sentence is seen in the corpus.

$$P(S) = \frac{c(w_1, \dots, w_n)}{N}$$



Thank You!

In our next session: Types of Language Models





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Stochastic Language Models

Prof. Aruna Malapati

Learning objectives

- Express the Language model using the joint probability distribution
- Types of Language models



Markov Process



First Order Markov Process



Second Order Markov Process



How to model variable length sequences?





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Generative model for POS tagging

Prof.Aruna Malapati

Discriminative classification models

- Given a set of training examples $\langle x_i, y_i \rangle$ for $i=1..m$ where each x_i is the input vector and y_i is the class label.
- The modeling task as to **learn a function f mapping the input x to labels $f(x)$.**
- The classification models you have learnt till now are discriminative models where each sample input was considered independent of each other.

Generative model for sequence labelling

- Given a set of training examples $\langle x_i, y_i \rangle$ for $i=1..m$ where each x_i is the input vector and y_i is the class label.
- In the POS tagging problem we have
 - $x_1 = \text{the dog barks}, y_1 = \text{DT NN VB}$
 - $x_2 = \text{the boy smiles}, y_2 = \text{DT NN VB}$
 -
- The task is to learn a function f that maps input x to its corresponding labels $f(x)$.

Generative model for sequence labelling

Using Bayes rule

$$P(t_1, \dots, t_n | w_1, \dots, w_n) = \frac{P(t_1, \dots, t_n)P(w_1, \dots, w_n | t_1, \dots, t_n)}{P(w_1, \dots, w_n)}$$



Submodels:

1. Prior: $P(t_1, \dots, t_n)$
2. Likelihood: $P(w_1, \dots, w_n | t_1, \dots, t_n)$
3. Marginal: $P(w_1, \dots, w_n)$ – can be ignored in argmax search

Markov Assumption

➤ Context model (prior)

$$P(t_1, \dots, t_n) = \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1})$$

➤ Lexical model (likelihood)

$$P(w_1, \dots, w_n | t_1, \dots, t_n) = \prod_{i=1}^n P(w_i | t_i)$$

Model Parameters

- Contextual probabilities : $P(t_i|t_{i-k}, \dots, t_{i-1})$
- Lexical probabilities : $P(w_i|t_i)$
- We can estimate these probabilities from a tagged corpus:

$$\hat{P}_{MLE}(w_i|t_i) = \frac{c(w_i, t_i)}{c(t_i)} \quad \hat{P}_{MLE}(t_i|t_{i-k}, \dots, t_{i-1}) = \frac{c(t_{i-k}, \dots, t_{i-1}, t_i)}{c(t_{i-k}, \dots, t_{i-1})}$$

Computing Probabilities

- The probability of a tagging:

$$P(t_1, \dots, t_n, w_1, \dots, w_n) = \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$



- Finding the most probable tagging:

$$\operatorname{argmax}_{t_1, \dots, t_n} \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$



Two fundamental problems in HMM

- Decoding:
 - How do we compute the best tag sequence given parameters?
- Learning:
 - How do we estimate the parameters?



Example

- Given a sentence of length 3, * * the dog barks STOP and the tag sequence * * DT NN VB * then
- $P(w_1 w_2 w_3, y_1, y_2, y_3) = T(DT|*, *) \times T(NN|*, DT) \times T(VB|DT NN) \times T(STOP|NN VB) \times E(*|*) \times E(*|*) E(\text{the}|DT) \times E(\text{dog}|NN) \times E(\text{barks}|VB) \times E(STOP|*)$
- We can also define $y_{-1} = ^*$ and $y_0 = ^*$ as special symbols.



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Parts of Speech Tagging using HMM

Prof.Aruna Malapati

Learning objectives

- Define Markov chains
- Define Hidden Markov Model



Markov Chains

- A Markov chain is a model that tells us something about the **probabilities of sequences of random variables**, states, each of which can take on values from some set.
- A Markov Model is a finite state machine with probabilistic state transitions.
- Markov assumption that next state only depends on the current state and independent of previous history.

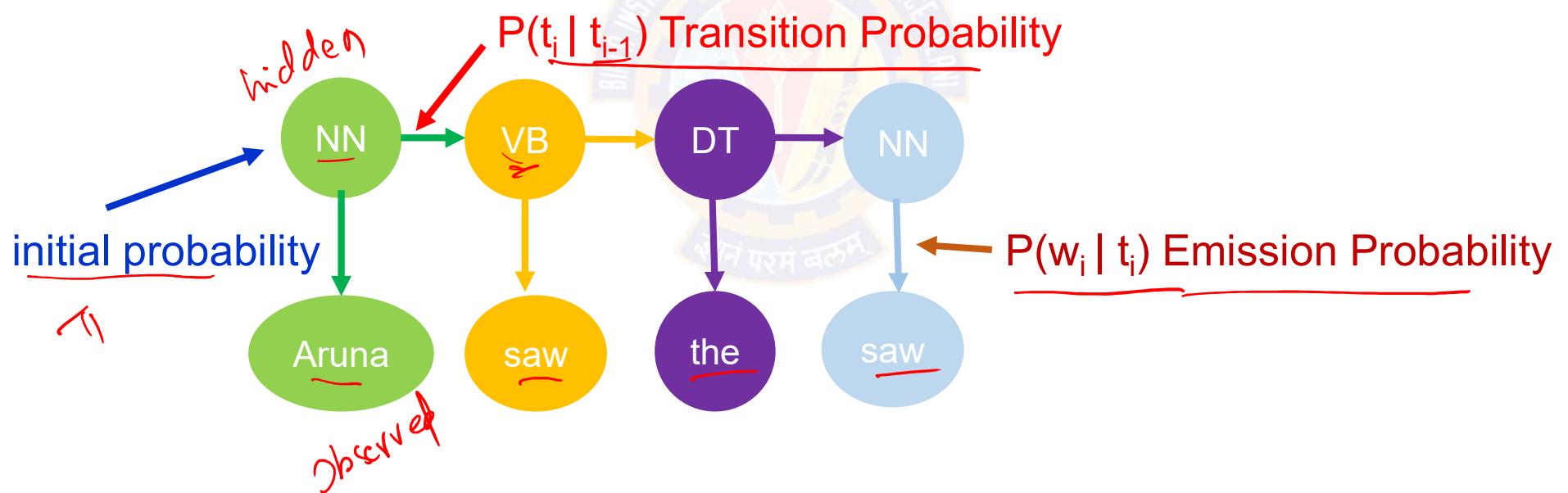
Markov Chain

➤ Formally, a Markov chain is specified by the following components:

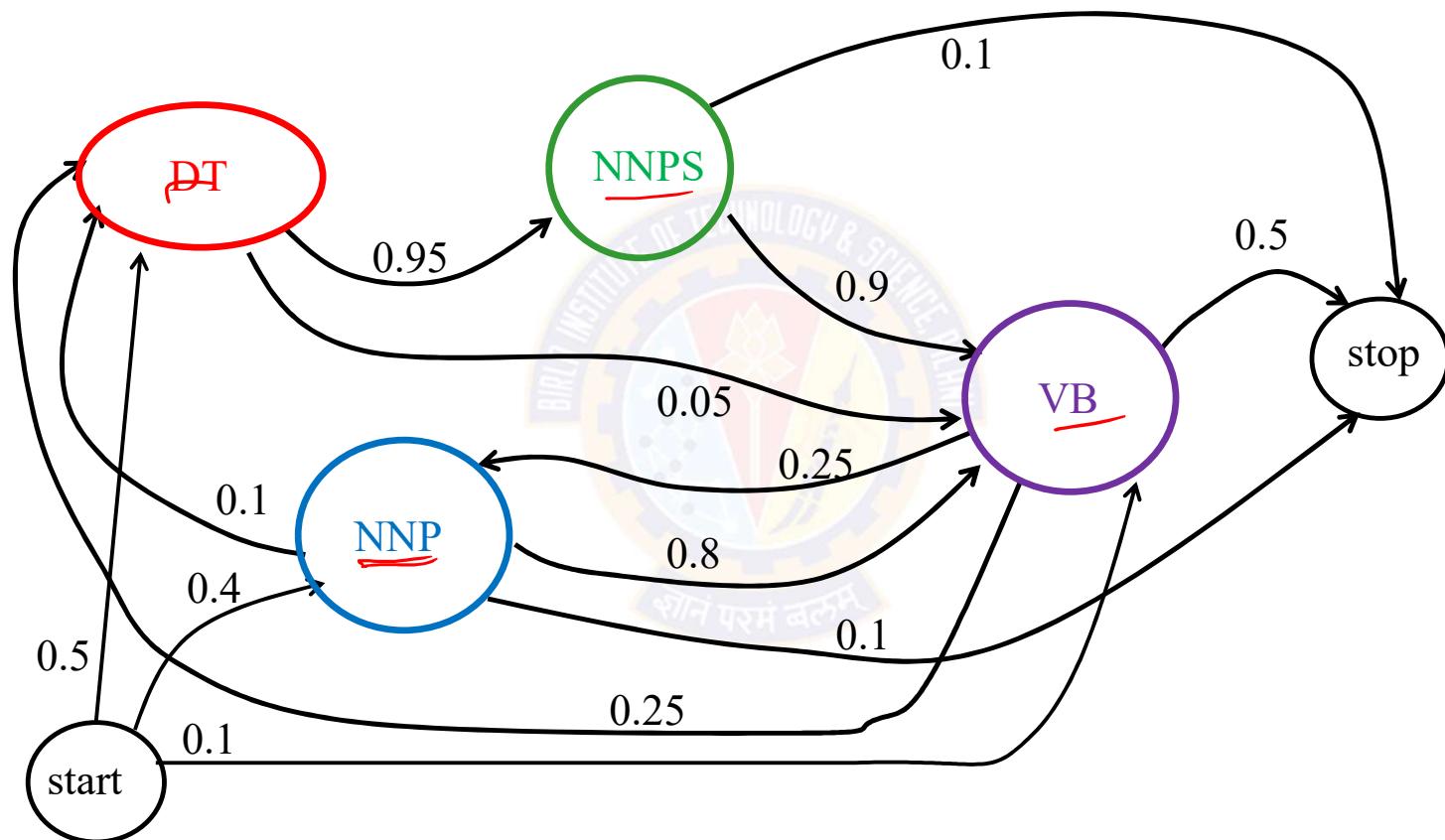
$\underline{Q = q_1 q_2 \dots q_N}$ a set of N states	A set of N states
$\underline{A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}}$	A transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, $\sum_{j=1}^n a_{ij} = 1 \forall i$
$\underline{\pi = \pi_1, \pi_2, \pi_3, \dots, \pi_n}$	An initial probability distribution over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. $\sum_{i=1}^n \pi_i = 1$

The Hidden Markov Model

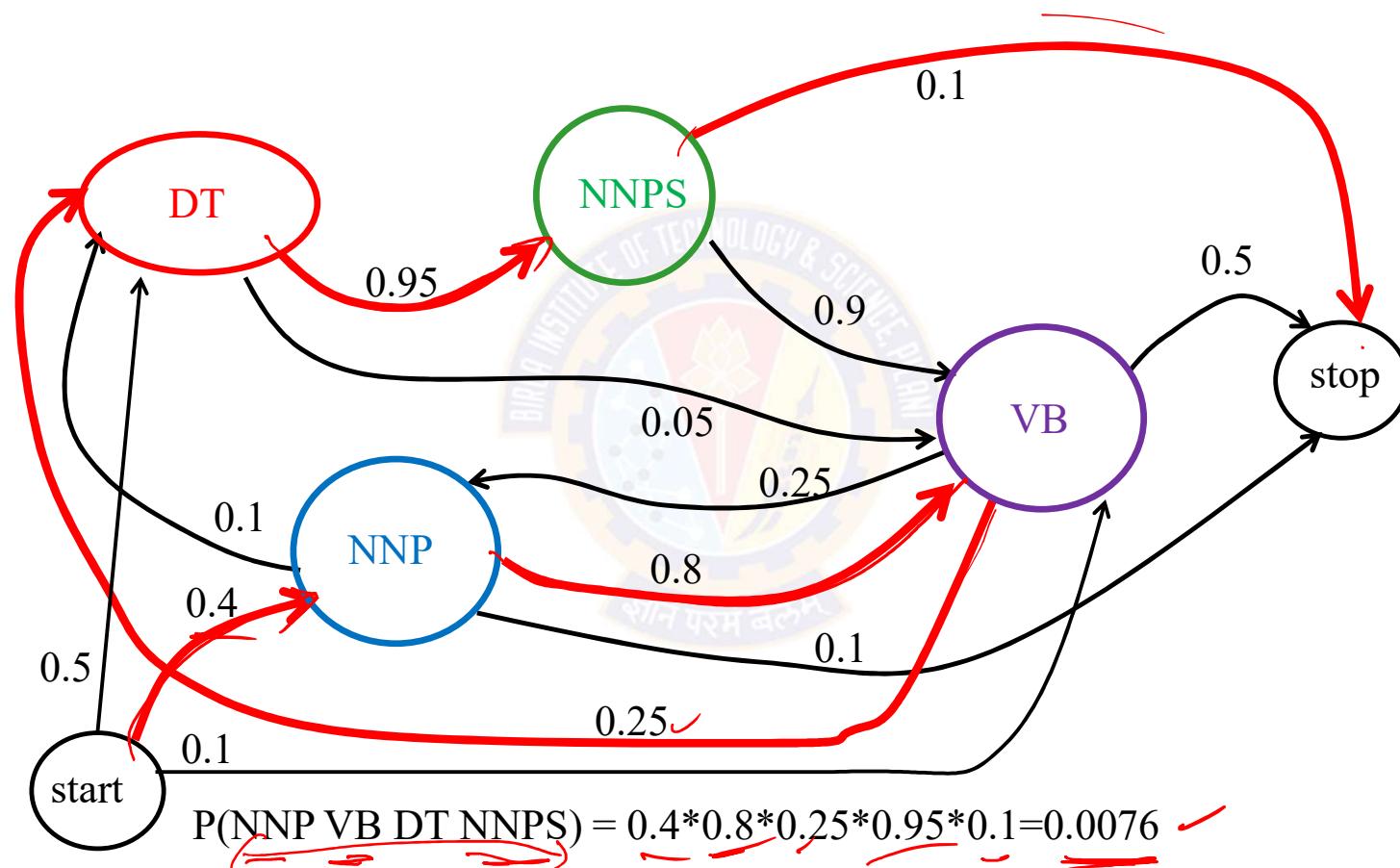
- In many cases the events we are interested in are hidden: we don't observe them directly.



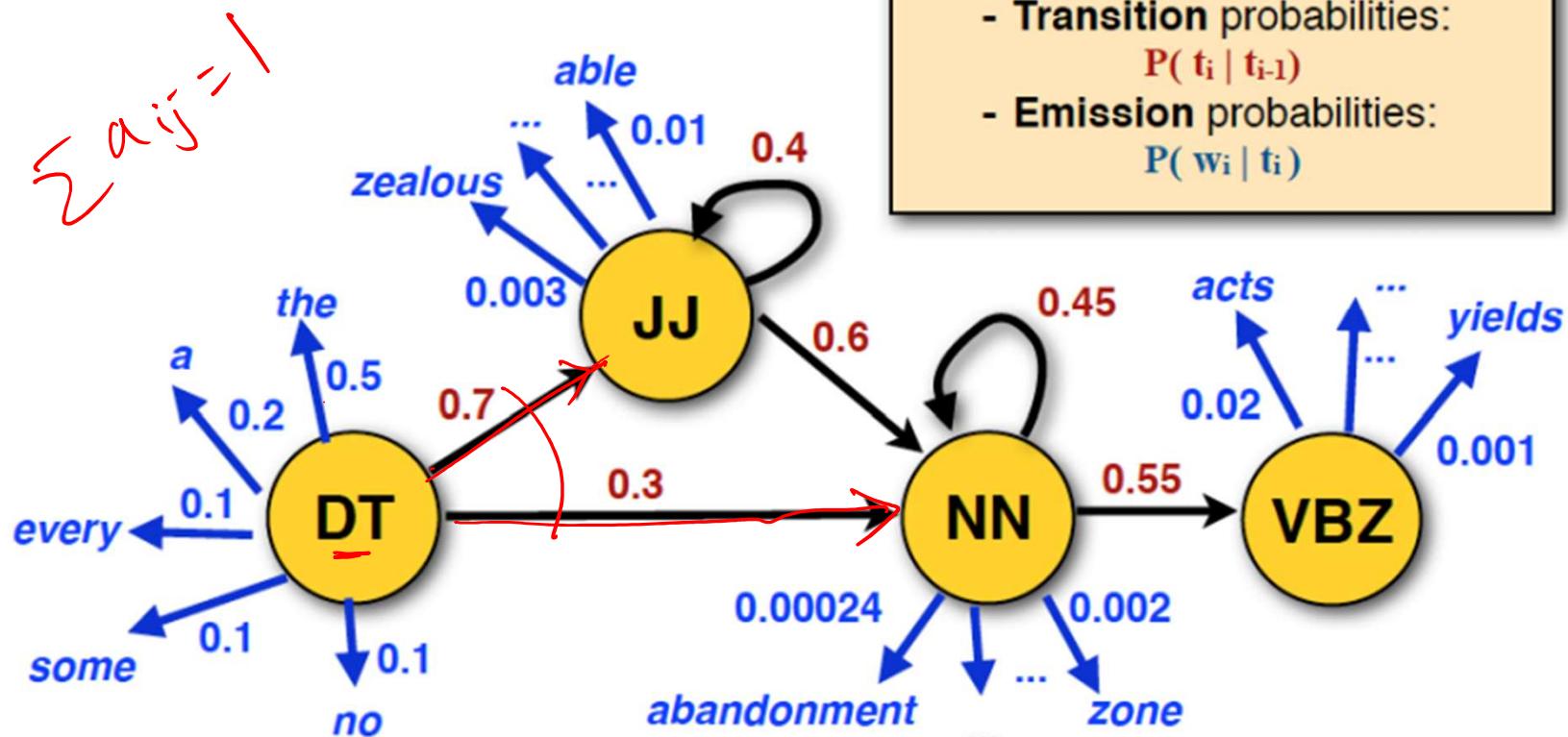
HMMs as probabilistic FSA



HMMs as probabilistic FSA



HMMs as probabilistic FSA



- **Transition** probabilities:
 $P(t_i | t_{i-1})$
- **Emission** probabilities:
 $P(w_i | t_i)$

Hidden Markov Models (formal)

- States $T = t_1, t_2 \dots t_N$;
- Observations $W = w_1, w_2 \dots w_N$;
 - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \dots v_V\}$
- Transition probabilities
 - Transition probability matrix $A = \{a_{ij}\}$
$$a_{ij} = P(t_i = j | t_{i-1} = i) \quad 1 \leq i, j \leq N$$
- Observation likelihoods
 - Output probability matrix $B = \{b_i(k)\}$
$$b_i(k) = P(w_i = v_k | t_i = i) \quad ?(v_k | i)$$
- Special initial probability vector π $\pi_i = P(t_1 = i) \quad 1 \leq i \leq N$

HMM tagging as decoding

➤ HMM model contains hidden variables, the task of determining the hidden variables sequence corresponding to the sequence of observations decoding is called decoding.

➤ Find our best estimate of the sequence that maximizes $P(t_1 \dots t_n | w_1 \dots w_n)$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

HMM tagging as decoding (Contd...)

Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$



Drop denominator since is the same for all tags we consider

HMM tagging as decoding (Contd...)

➤ A1: P(w) depends only on its own POS

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^n P(w_i | t_i)$$

➤ A2: P(t) depends only on P(t-1)

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \underbrace{\widehat{P}(w_1^n | t_1^n)}_{\text{likelihood}} \underbrace{\widehat{P}(t_1^n)}_{\text{prior}}$$

HMM tagging as decoding (Contd...)

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n) \approx \operatorname{argmax}_{t_1^n} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

Now we have two probabilities to calculate:

- Probability of a word occurring given its tag
- Probability of a tag occurring given a previous tag
- We can calculate each of these from a POS-tagged corpus



Thank You!

In our next session: HMM Example



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

HMM Example

Prof. Aruna Malapati

Learning Objectives

- Construction of Transaction and Emission Matrices



Example

Emission Matrix B

	*	DT	NNS	VB	NN	IN	STOP
*	1						
the		3/4					
employees			3/4				
pass				2/4			
an	1/4						
exam					1		
wait			1/4				
for					1		
employers		1/4					
fire			1/4				
:						1	

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
*	2/3	1/3					
DT			2/4	1/4	1/4		
NNS				3/4			1/4
VB		1/4	1/4			1/4	1/4
NN							1
IN					1		
STOP							



$P(DT|X)$

$P(\text{the } DT)$

$P(DT)$

$P(NNS|DT)$

$P(\text{the } DT)$

$P(DT)$

$P(\text{the } DT)$

$P(DT)$

$P(NNS)$

$P(VB)$

$P(NN)$

$P(IN)$

$P(STOP)$

S1: the Employees pass an exam .

T1: DT NNS VB DT NN STOP

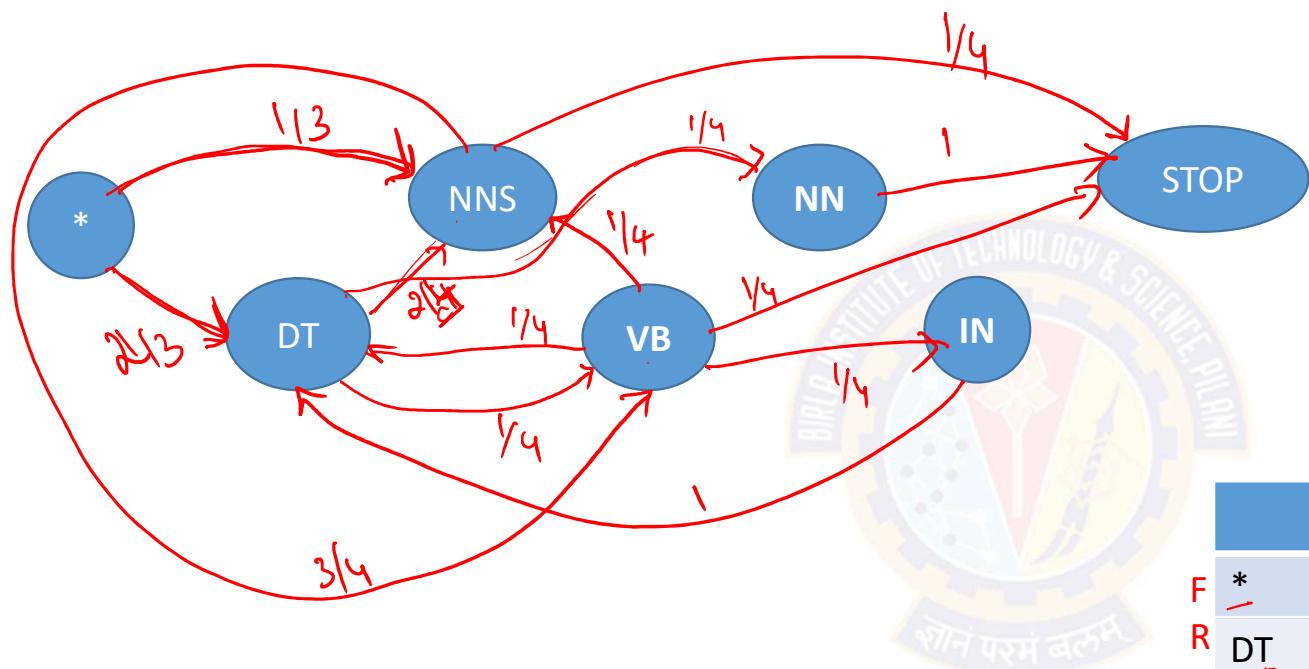
S2: the employees wait for the pass .

T2: DT NNS VB IN DT VB STOP

S3: employers fire employees .

T3: NNS VB NNS STOP

Transition diagram



$$\frac{P(DT | *))}{P(NNS | *)}$$

$$P(NNS | DT)$$

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
F	*		<u>2/3</u>	<u>1/3</u>			
R	DT			<u>2/4</u>	<u>1/4</u>	<u>1/4</u>	
I							
S	NNS				<u>3/4</u>		<u>1/4</u>
T	VB		<u>1/4</u>	<u>1/4</u>			<u>1/4</u>
N	NN						<u>1</u>
A	IN				<u>1</u>		
G	STOP						



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Parts of Speech Tagging using HMM

Prof.Aruna Malapati

Learning objectives

- Define Markov chains
- Define Hidden Markov Model



Markov Chains

- A Markov chain is a model that tells us something about the **probabilities of sequences of random variables**, states, each of which can take on values from some set.
- A Markov Model is a finite state machine with probabilistic state transitions.
- Markov assumption that next state only depends on the current state and independent of previous history.

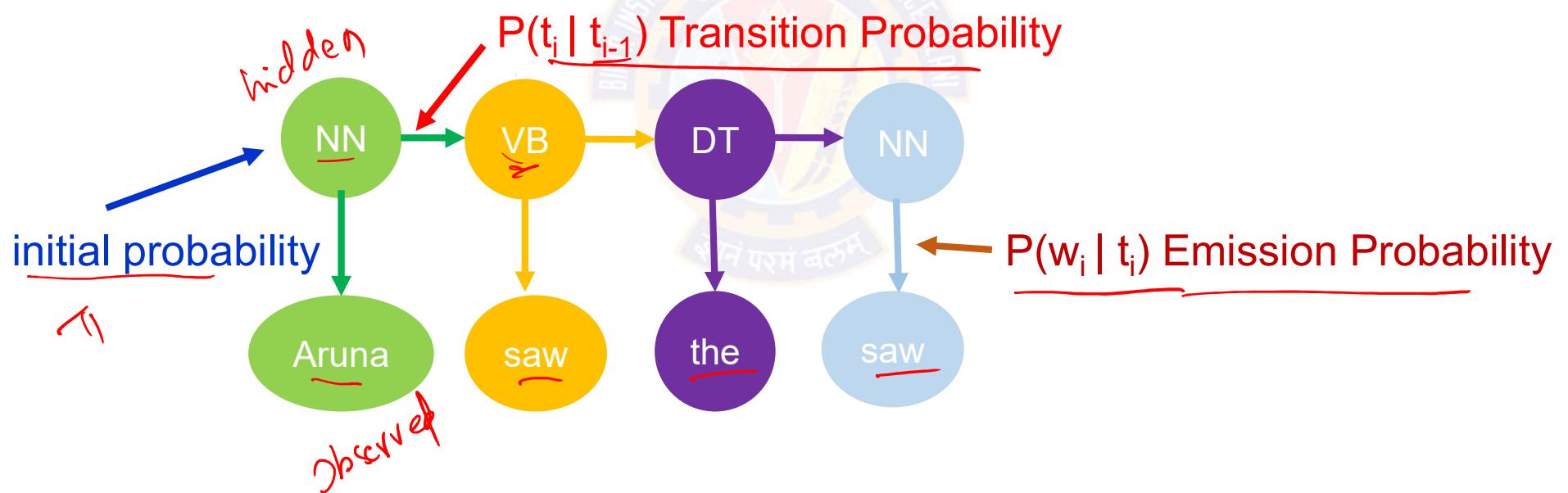
Markov Chain

➤ Formally, a Markov chain is specified by the following components:

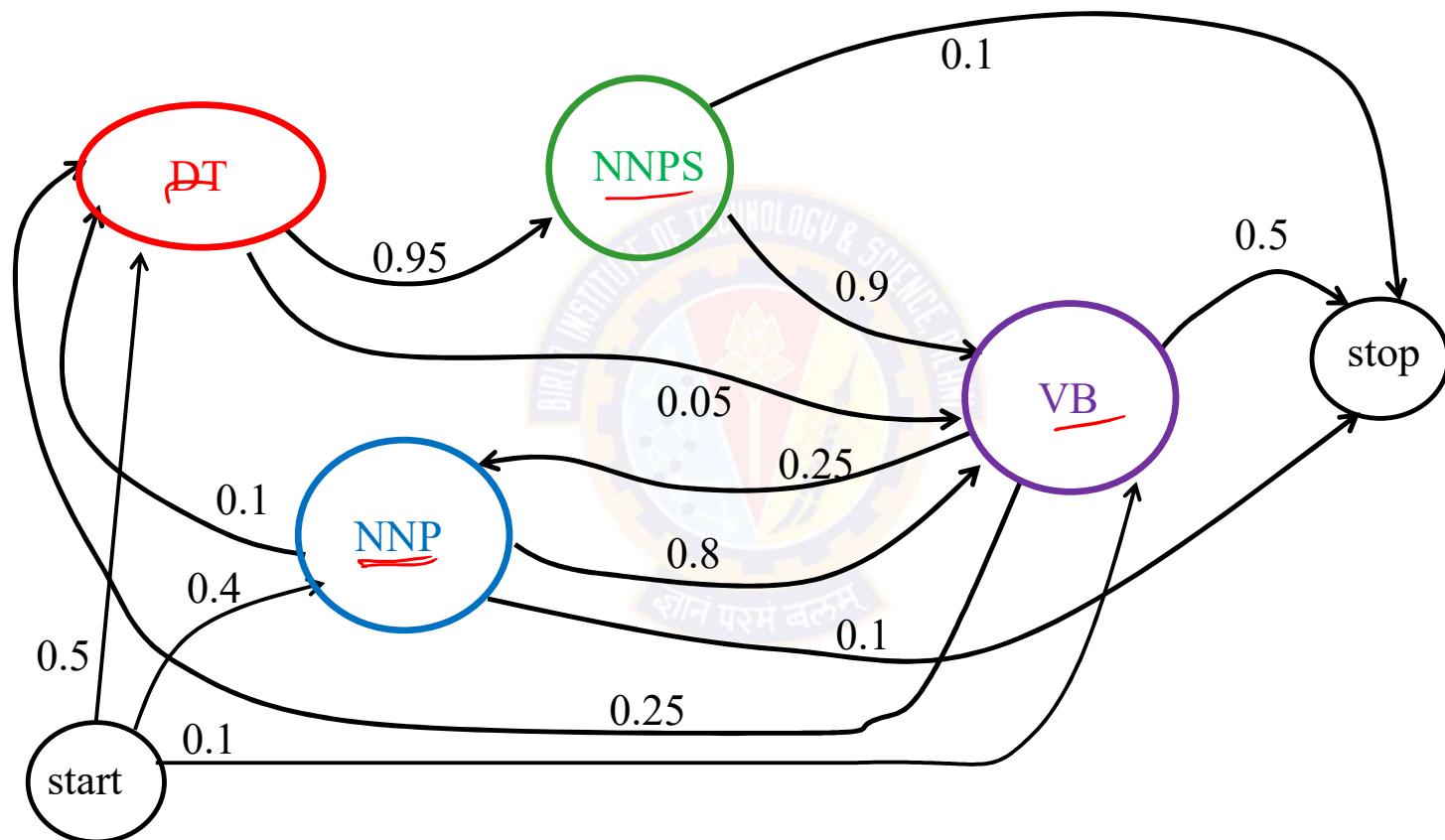
$\underline{Q = q_1 q_2 \dots q_N}$ a set of N states	A set of N states
$\underline{A = a_{11} a_{12} \dots a_{n1} \dots a_{nn}}$	A transition probability matrix A, each a_{ij} representing the probability of moving from state i to state j, $\sum_{j=1}^n a_{ij} = 1 \forall i$
$\underline{\pi = \pi_1, \pi_2, \pi_3, \dots, \pi_n}$	An initial probability distribution over states. π_i is the probability that the Markov chain will start in state i. Some states j may have $\pi_j = 0$, meaning that they cannot be initial states. $\sum_{i=1}^n \pi_i = 1$

The Hidden Markov Model

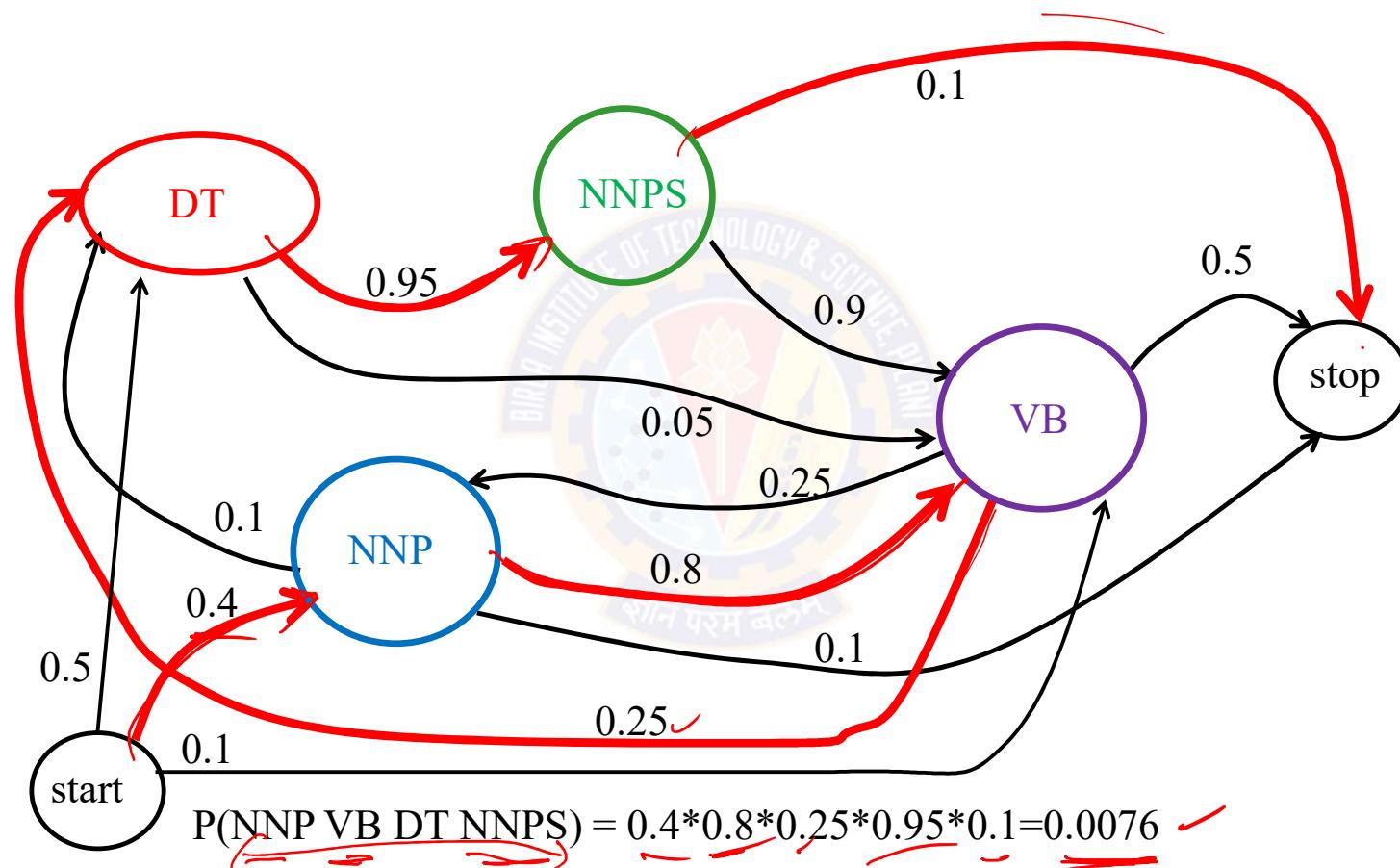
- In many cases the events we are interested in are hidden: we don't observe them directly.



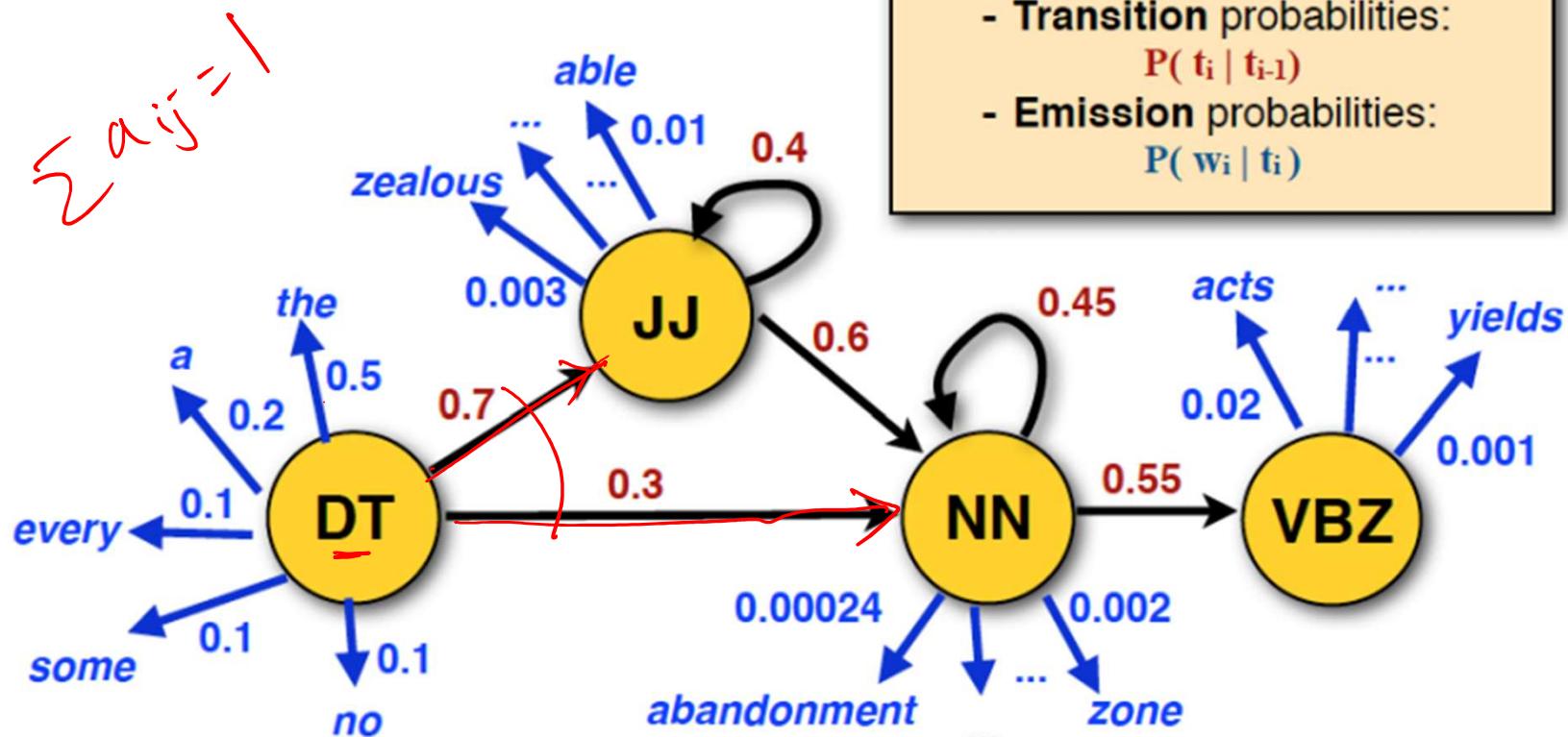
HMMs as probabilistic FSA



HMMs as probabilistic FSA



HMMs as probabilistic FSA



- **Transition** probabilities:
 $P(t_i | t_{i-1})$
- **Emission** probabilities:
 $P(w_i | t_i)$

Hidden Markov Models (formal)

- States $T = t_1, t_2 \dots t_N$;
- Observations $W = w_1, w_2 \dots w_N$;
 - Each observation is a symbol from a vocabulary $V = \{v_1, v_2, \dots v_V\}$
- Transition probabilities
 - Transition probability matrix $A = \{a_{ij}\}$
$$a_{ij} = P(t_i = j | t_{i-1} = i) \quad 1 \leq i, j \leq N$$
- Observation likelihoods
 - Output probability matrix $B = \{b_i(k)\}$
$$b_i(k) = P(w_i = v_k | t_i = i) \quad ?(v_k | i)$$
- Special initial probability vector π $\pi_i = P(t_1 = i) \quad 1 \leq i \leq N$

HMM tagging as decoding

➤ HMM model contains hidden variables, the task of determining the hidden variables sequence corresponding to the sequence of observations decoding is called decoding.

➤ Find our best estimate of the sequence that maximizes $P(t_1 \dots t_n | w_1 \dots w_n)$

$$\hat{t}_1^n = \underset{t_1^n}{\operatorname{argmax}} P(t_1^n | w_1^n)$$

HMM tagging as decoding (Contd...)

Bayes Rule

$$P(x|y) = \frac{P(y|x)P(x)}{P(y)}$$

$$P(t_1^n | w_1^n) = \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \frac{P(w_1^n | t_1^n)P(t_1^n)}{P(w_1^n)}$$



Drop denominator since is the same for all tags we consider

HMM tagging as decoding (Contd...)

➤ A1: P(w) depends only on its own POS

$$P(w_1^n | t_1^n) \approx \prod_{i=1}^n P(w_i | t_i)$$

➤ A2: P(t) depends only on P(t-1)

$$P(t_1^n) \approx \prod_{i=1}^n P(t_i | t_{i-1})$$

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} \underbrace{\widehat{P}(w_1^n | t_1^n)}_{\text{likelihood}} \underbrace{\widehat{P}(t_1^n)}_{\text{prior}}$$

HMM tagging as decoding (Contd...)

$$\hat{t}_1^n = \operatorname{argmax}_{t_1^n} P(t_1^n | w_1^n) \approx \operatorname{argmax}_{t_1^n} \prod_{i=1}^n P(w_i | t_i) P(t_i | t_{i-1})$$

Now we have two probabilities to calculate:

- Probability of a word occurring given its tag
- Probability of a tag occurring given a previous tag
- We can calculate each of these from a POS-tagged corpus



Thank You!

In our next session: HMM Example



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

HMM Example

Prof. Aruna Malapati

Learning Objectives

- Construction of Transaction and Emission Matrices



Example

Emission Matrix B

	*	DT	NNS	VB	NN	IN	STOP
*	1						
the		3/4					
employees			3/4				
pass				2/4			
an	1/4						
exam					1		
wait			1/4				
for					1		
employers		1/4					
fire			1/4				
:						1	

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
*	2/3	1/3					
DT			2/4	1/4	1/4		
NNS				3/4			1/4
VB		1/4	1/4			1/4	1/4
NN							1
IN					1		
STOP							

$$P(NNS|DT)$$

$$P(\text{the}|DT) = \frac{P(\text{the}, DT)}{P(DT)}$$

$$P(DT|*)$$

$$P(\text{the}, DT) = P(DT) \cdot P(\text{the}|DT)$$

$$P(DT|*)$$

$$P(DT) = P(DT|*) \cdot P(*)$$

$$P(*)$$

$$P(*) = 1$$

S1: the Employees pass an exam .

T1: DT NNS VB DT NN STOP

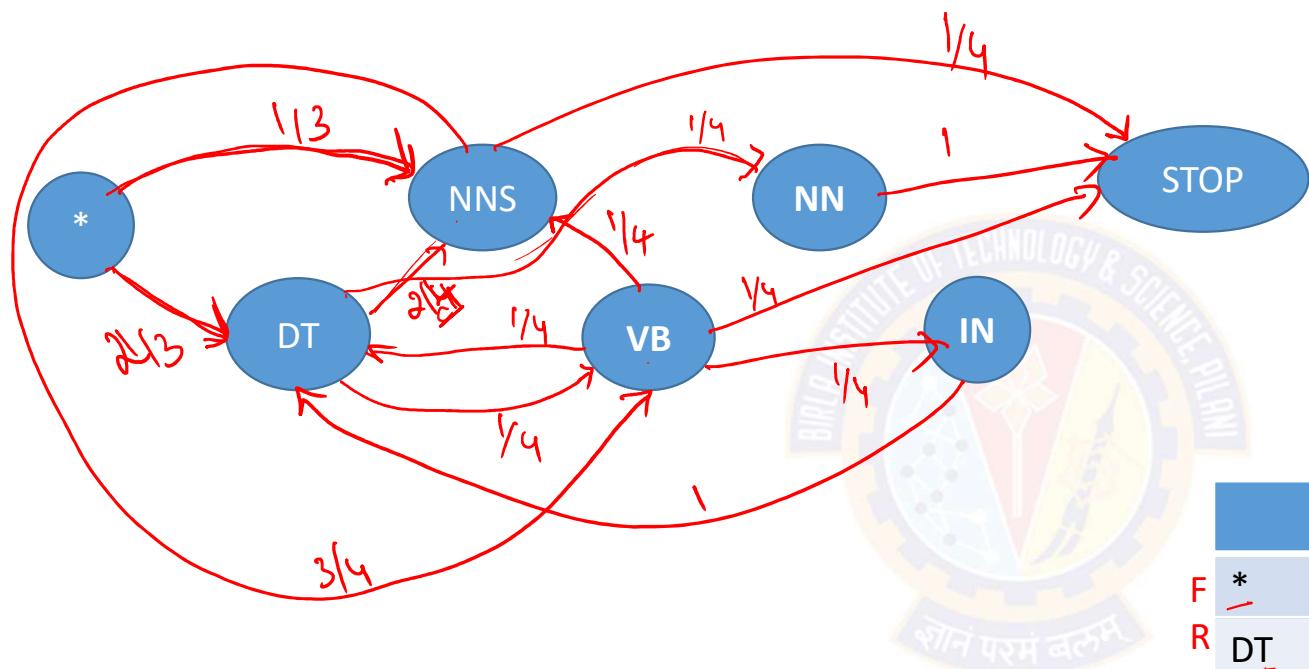
S2: the employees wait for the pass .

T2: DT NNS VB IN DT VB STOP

S3: employers fire employees .

T3: NNS VB NNS STOP

Transition diagram



$$\begin{aligned} P(\text{DT} | \star, \star) \\ P(\text{NNS} | \star, \text{DT}) \end{aligned}$$

Tag Translation Matrix

	*	DT	NNS	VB	NN	IN	STOP
F	*	<u>2/3</u>	<u>1/3</u>				
R	DT		<u>2/4</u>	<u>1/4</u>	<u>1/4</u>		
I							
S	NNS				<u>3/4</u>		<u>1/4</u>
T	VB		<u>1/4</u>	<u>1/4</u>			<u>1/4</u>
A	NN						1
G	IN		1				
	STOP						



Thank You!

In our next session: Viterbi Algorithm





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Viterbi Algorithm

Prof. Aruna Malapati

Learning Objectives

- Motivation for Viterbi
- HMM decoding using Viterbi algorithm
- Example



The intuition behind Viterbi

- Finding the most probable tagging sequence for a test sentence $\langle w_1 \ w_2 \ w_3 \dots, w_n \rangle$

$$\underset{t_1, \dots, t_n}{\operatorname{argmax}} \prod_{i=1}^n P(t_i | t_{i-k}, \dots, t_{i-1}) P(w_i | t_i)$$

- The argmax is taken over all sequences y_1, y_2, \dots, y_n such that $y_i \in S$ for $i = 1, 2, \dots, n$ and $y_{n+1} = \text{STOP}$
- Lets assume that our tagging model is a bigram model

Brute force search over tag sequences

➤ Input sentence <the employees wait for an exam>

➤ S = { DT NNS VB IN NN }

➤ All possible tag sequences

DT DT DT DT DT DT STOP $\rightarrow 0.002$

DT DT DT DT DT NNS STOP $\rightarrow 0.003 \times 10^{-12}$

DT DT DT DT DT VB STOP $\rightarrow 0.0006$

5^6
 $|S|^{12}$

....

Viterbi algorithm

➤ Create a table V with $N+2$ rows and T columns:

➤ N – the number of states/tags

➤ T – the length of the sequence/sentence

➤ Initialise the first column

➤ For each tag t in the tagset compute:

$$\underline{V[t, 1]} = \underline{P(t|start)} \underline{P(w_1|t)}$$

➤ For each column $j = 2$ to T in the table V :

➤ For each tag t in the tagset compute:

$$V[t, j] = \max_{t'} V[t', j-1] \underline{P(t|t')} \underline{P(w_j|t)}$$

Example

Transition matrix: $P(t_i|t_{i-1})$

	NOUN	Verb	Det	Prep	ADV	STOP
<S>	.3	.1	.3	.2	.1	0
Noun	.2	.4	.01	.3	.04	.05
Verb	.3	.05	.3	.2	.1	.05
Det	.9	.01	.01	.01	.07	0
Prep	.4	.05	.4	.1	.05	0
Adv	.1	.5	.1	.1	.1	.1

Emission matrix: $P(w_i|t_i)$

	a	cat	doctor	in	is	the	very
Noun	0	.5	.4	0	0.1	0	0
Verb	0	0	.1	0	.9	0	0
Det	.3	0	0	0	0	.7	0
Prep	0	0	0	1.0	0	0	0
Adv	0	0	0	.1	0	0	.9

| S^{nt})
 | (1E)
 | (1S)
 +, <S>, start

$$V[t, 1] = P(t|start)P(w_1|t)$$

	w1=the	w2=doctor	w3=is	w4=in	STOP
Noun	0				
Verb	0				
Det	.21				
Prep	0				
Adv	0				

$$\begin{aligned}
 V(\text{Noun, the}) &= P(\text{Noun}|<\text{S}>)P(\text{the}|\text{Noun}) = .3 \times 0 = 0 \\
 V(\text{Verb, the}) &= P(\text{Verb}|<\text{S}>)P(\text{the}|\text{Verb}) = 0 \times 0 = 0 \\
 V(\text{Det, the}) &= P(\text{Det}|<\text{S}>)P(\text{the}|\text{Det}) = .3 \times .7 = .21 \\
 V(\text{Prep, the}) &= P(\text{Prep}|<\text{S}>)P(\text{the}|\text{Prep}) = .2 \times 0 = 0 \\
 V(\text{Adv, the}) &= P(\text{Adv}|<\text{S}>)P(\text{the}|\text{Adv}) = .2 \times 0 = 0
 \end{aligned}$$

Example (Contd..)

$$V(\text{Noun}, \text{doctor}) = \max_{t'} V(t', \text{the}) \underbrace{XP(\text{Noun}|t')}_{\text{P(doctor|Noun)}} \\ = \max \{ 0, 0, .21 (.3 \times .4), 0, 0 \} = .0756$$

$$V(\text{Verb}, \text{doctor}) = \max_{t'} V(t', \text{the}) \underbrace{XP(\text{Verb}|t')}_{\text{P(doctor|Verb)}} \\ = \max \{ 0, 0, .21(.01 \times .1), 0, 0 \} = .00021$$



	w1=-the	w2=doctor	w3=is	w4=in	STOP
Noun	0	.0756			
Verb	0	.00021			
Det	.21	0			
Prep	0	0			
Adv	0	0			

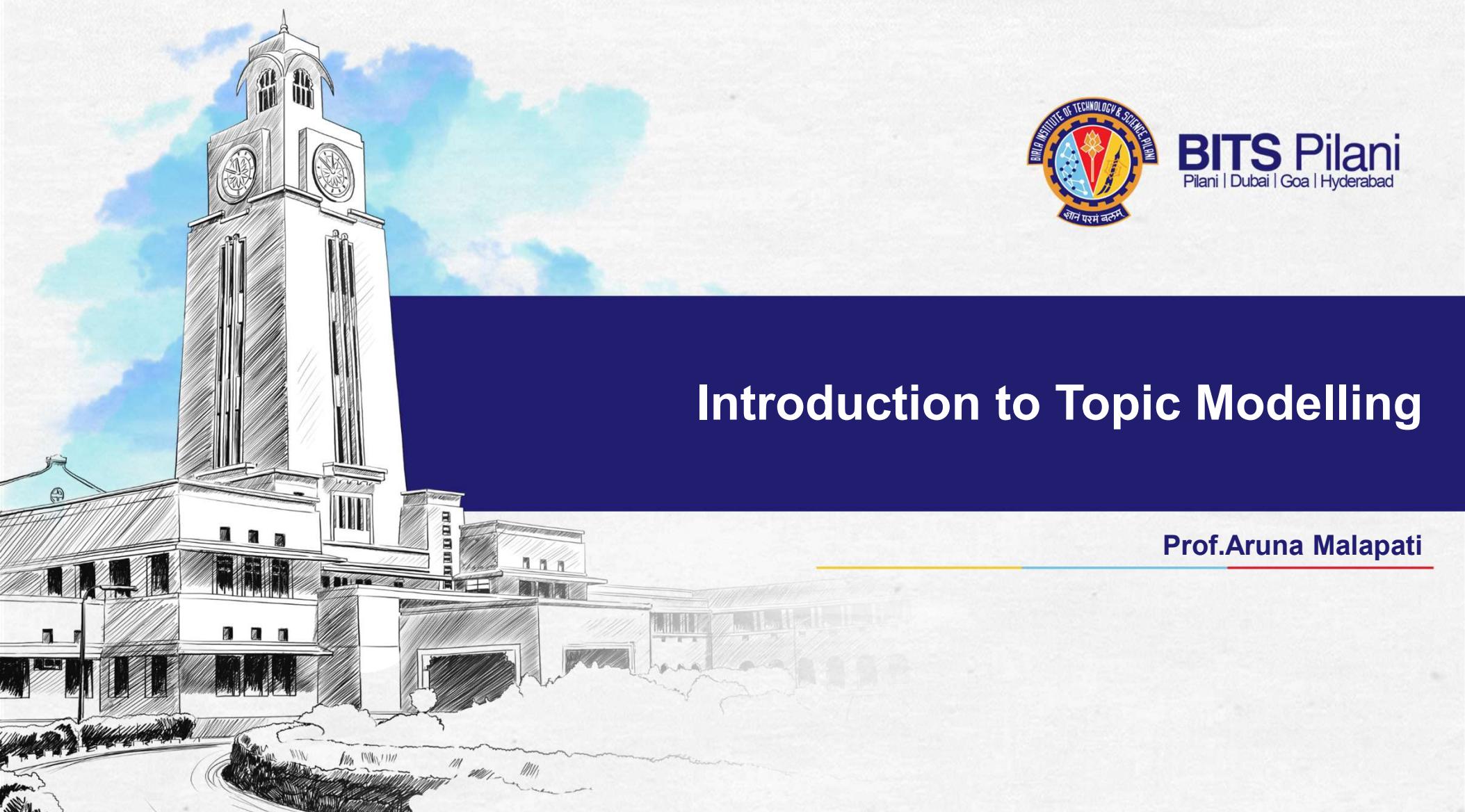
Completed Viterbi matrix

	w1--the	w2=doctor	w3=is	w4=in	STOP
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.0000272
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

Backtracking the Viterbi Matrix

	w1--the	w2=doctor	w3=is	w4=in	STOP
Noun	0	.0756	.001512	0	
Verb	0	.00021	.027216	0	
Det	.21	0	0	0	.0000272
Prep	0	0	0	.005443	
Adv	0	0	0	.000272	

Det Noun Verb Prep STOP



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Topic Modelling

Prof. Aruna Malapati

Learning Objectives

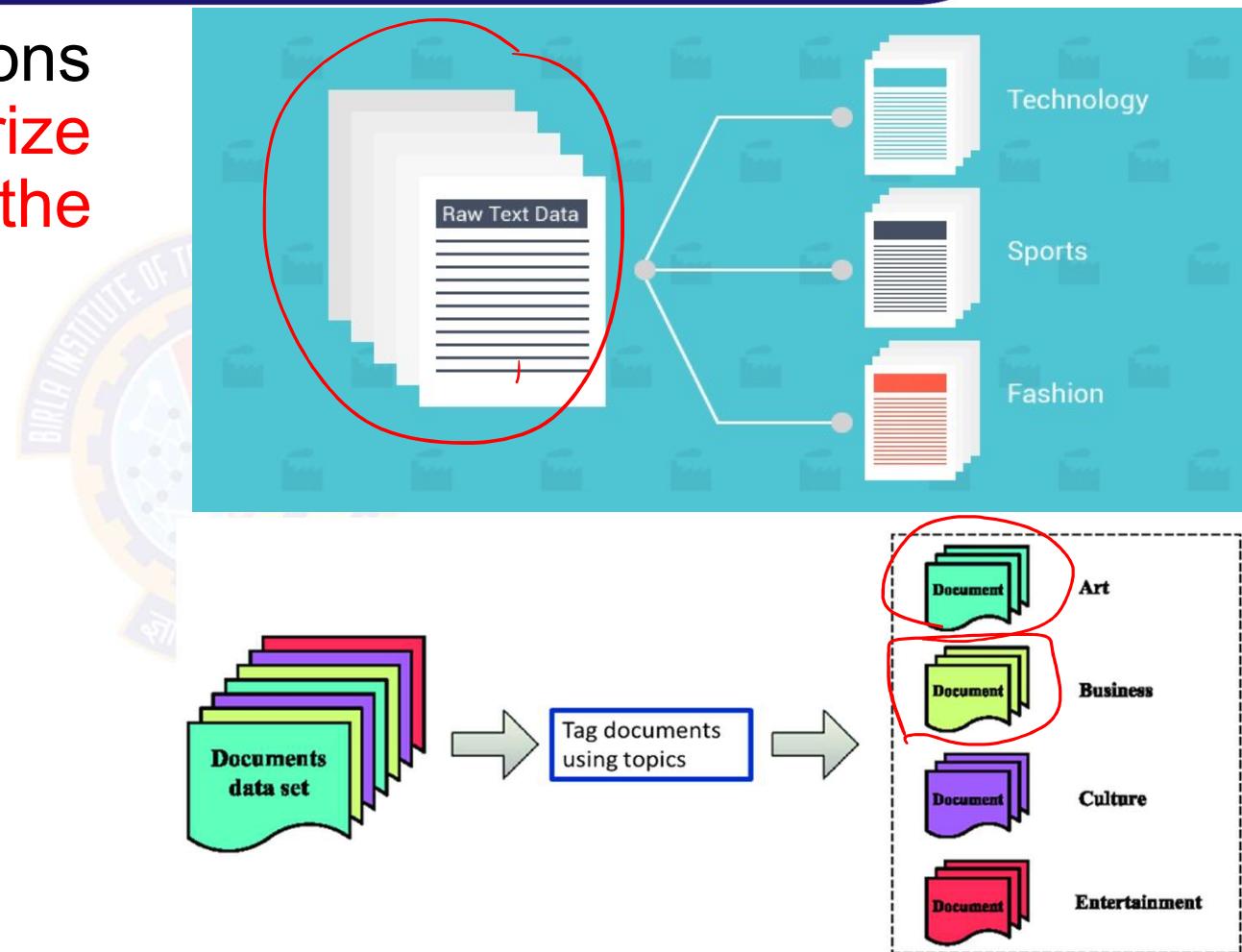
- Motivation for Topic Modelling
- Objectives of Topic Modelling
- Generative model for Topic Model
- The posterior distribution

Motivation for Topic Modelling



Objectives of Topic Modelling

- Use these annotations to organize, summarize and search the documents.



Sample output from the LDA

- Four topics learned from the S&P 500 stock market data
- Goal is to find groups of stocks that tend to move together.

Topic 1	Topic 2	Topic 3	Topic 4
Southwestern Energy Range Resources Cabot Oil & Gas EOG Resources Chesapeake Energy Pioneer Resources Devon Energy Peabody Energy Anadarko Petroleum Massey Energy	Penneys Macys Kohls Nordstrom Target Limited Lowes Home Depot American Express Abercrombie	Capital One BNY Mellon Discover Northern Trust Janus JPMorgan Chase State Street Wells Fargo PPL T. Rowe Price	Simon Property Kimco Realty Equity Residential AvalonBay Communities Apartment Investment Vornado Realty Trust Boston Properties Public Storage Host Hotels HCP Inc.

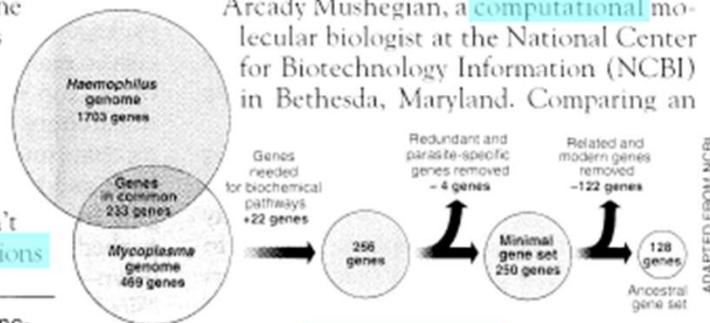
- The topic model does not provide any label to these group of words.

Seeking Life's Bare (Genetic) Necessities

COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

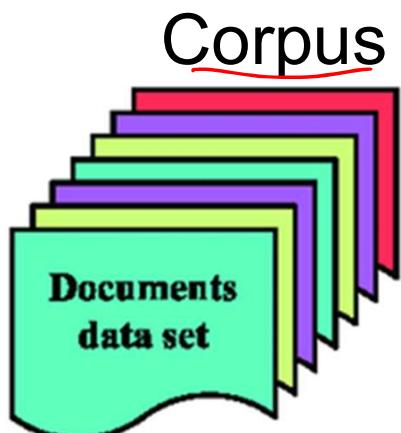
* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

Genetics

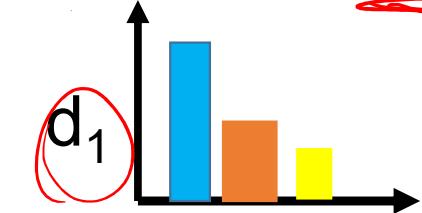
Evolutionary
biology

Data Analysis

Overall schematic

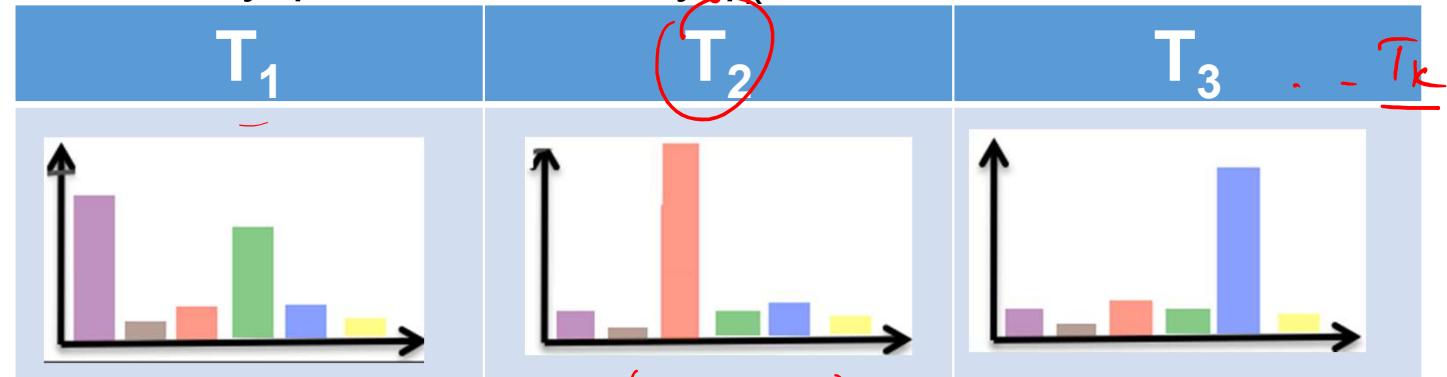


Documents($d_1 \dots d_n$)



Each document has a distribution over K topics

Each topic is defined as a Multinomial distribution over the vocabulary, parameterized by ϕ_k



K-Topics (Hyper Parameter)

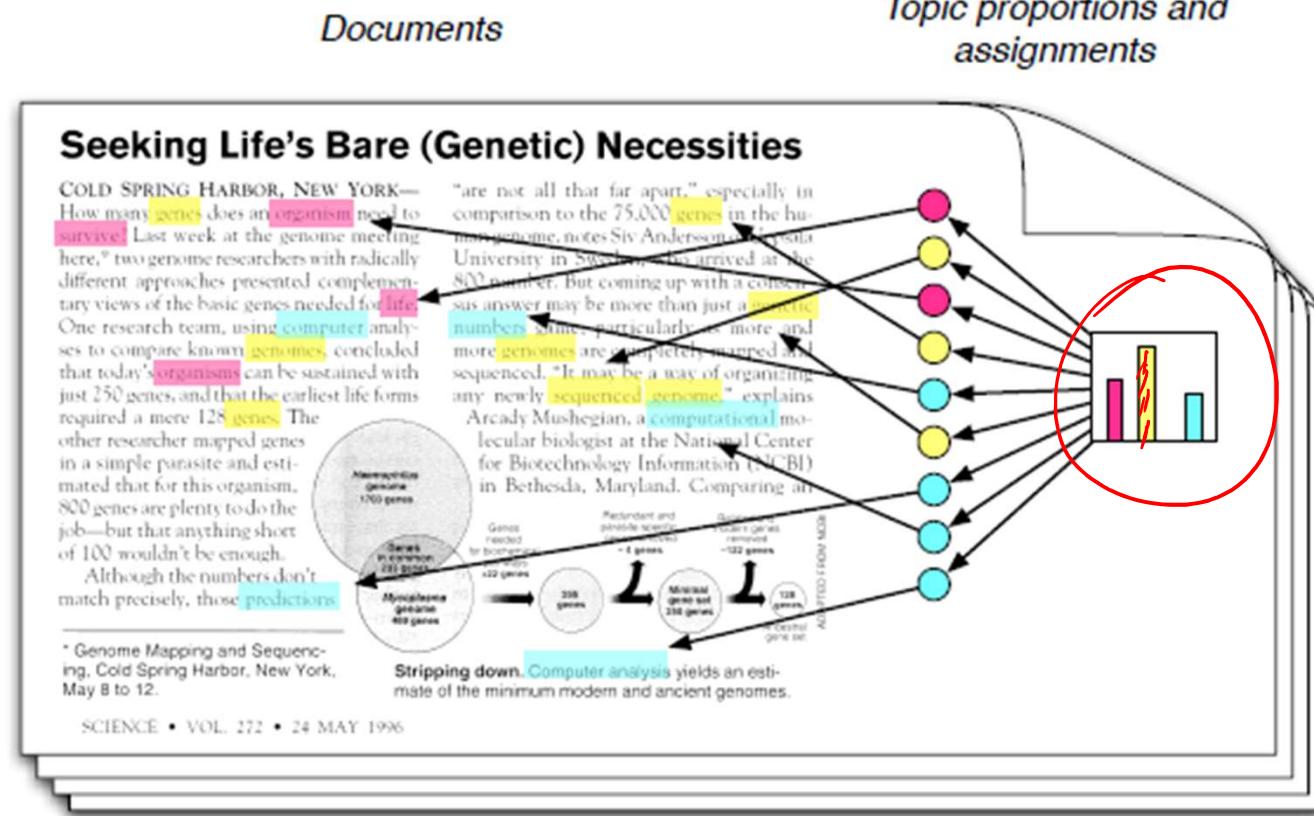
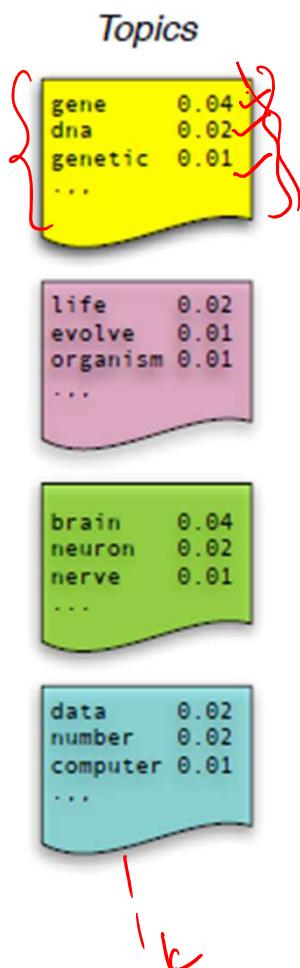
$$\begin{bmatrix} w_1 \\ w_2 \\ w_3 \\ \vdots \\ w_m \end{bmatrix} = \begin{bmatrix} 0.06 & 0.1 \\ 0.0000002 & 0.2 \\ 0.2 & \dots \end{bmatrix}$$

$$w_i = \frac{1}{m}$$

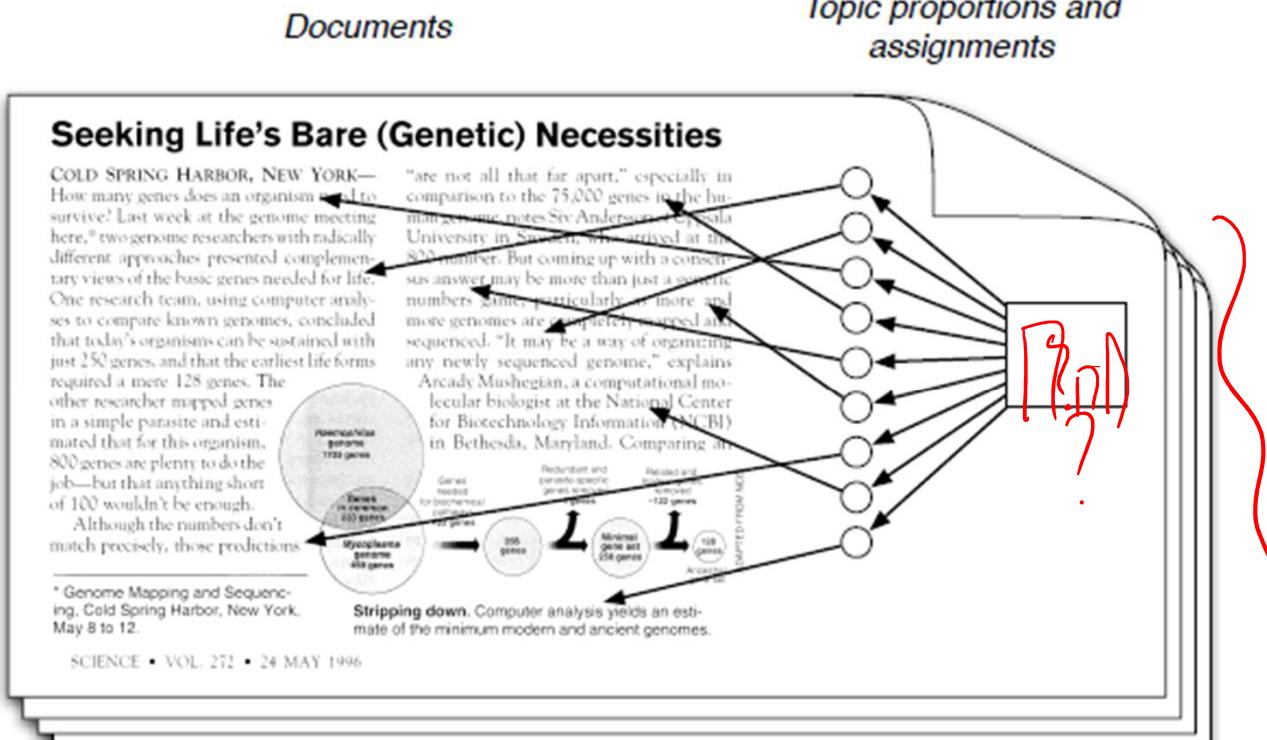


Vocabulary($W_1 \dots W_m$)

Generative model for Topic Model



The posterior distribution





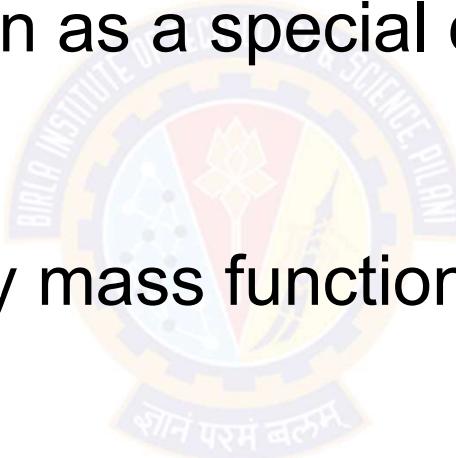
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Probability Density Functions

Prof.Aruna Malapati

Learning Objectives

- Bernoulli trial
- Bernoulli distribution as a special case of Binomial Distribution
- Bernoulli probability mass function
- Beta Distribution

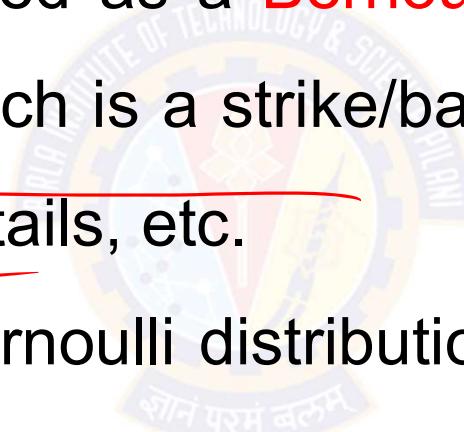


Bernoulli Trial

➤ Any single trial with two possible outcomes can be modeled as a Bernoulli trial: team wins/loses, pitch is a strike/ball, coin comes up heads or tails, etc.

➤ A Bernoulli trial uses Bernoulli distribution to calculate the probability of either outcome.

Bernoulli trial



$$P(X=1) = \theta \quad P(X=0) = (1-\theta)$$

Bernoulli: A Special Case of the Binomial Distribution

**Binomial Trail: Chance of getting
n heads in a row(n=3)**



**Bernoulli Trail: Chance of getting
a heads on a single flip**

Bernoulli - Distribution Notation

- The probability mass function of the Bernoulli distribution is

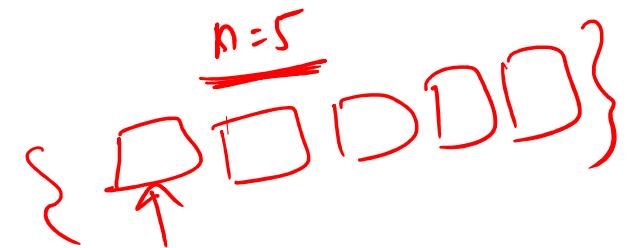
$$P(X=1) = \theta \quad P(X=0) = (1-\theta)$$

$$f(x) = P(X=k) = \theta^k (1-\theta)^{1-k}, \quad k=\{0,1\}$$

- The only parameter of the bernoulli distribution is θ , which defines the probability of success during a bernoulli trial.

Binomial distribution

$$k=0 \\ P(X=0) = \theta^0 (1-\theta)^{1-0} = 1-\theta$$

{  }
n=5

$$k=1 \\ P(X=1) = \theta^1 (1-\theta)^{1-1} = \underline{\theta}$$

$$P(X_1=1, X_2=1, X_3=0) = \theta \times \theta (1-\theta) = \underline{\theta^2(1-\theta)}$$

$$P\left(\sum_{i=1}^{n=3} X_i = 2\right) = P(1,1,0) + P(1,0,1) + P(0,1,1) \\ = \theta^2 (1-\theta) + \theta^2 (1-\theta) + \theta^2 (1-\theta) \\ = 3\theta^2 (1-\theta)$$

$$P\left(\sum_{i=1}^n X_i = k\right) = \binom{n}{k} \theta^k (1-\theta)^{n-k} \quad \boxed{\binom{n}{k} = \frac{n!}{k!(n-k)!}}$$

Beta Distribution

- The probability distribution function for the beta distribution

$$f(\theta; \alpha, \beta) = \frac{\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}}{B(\alpha, \beta)} \propto \theta^{(\alpha-1)}(1-\theta)^{\beta-1}$$

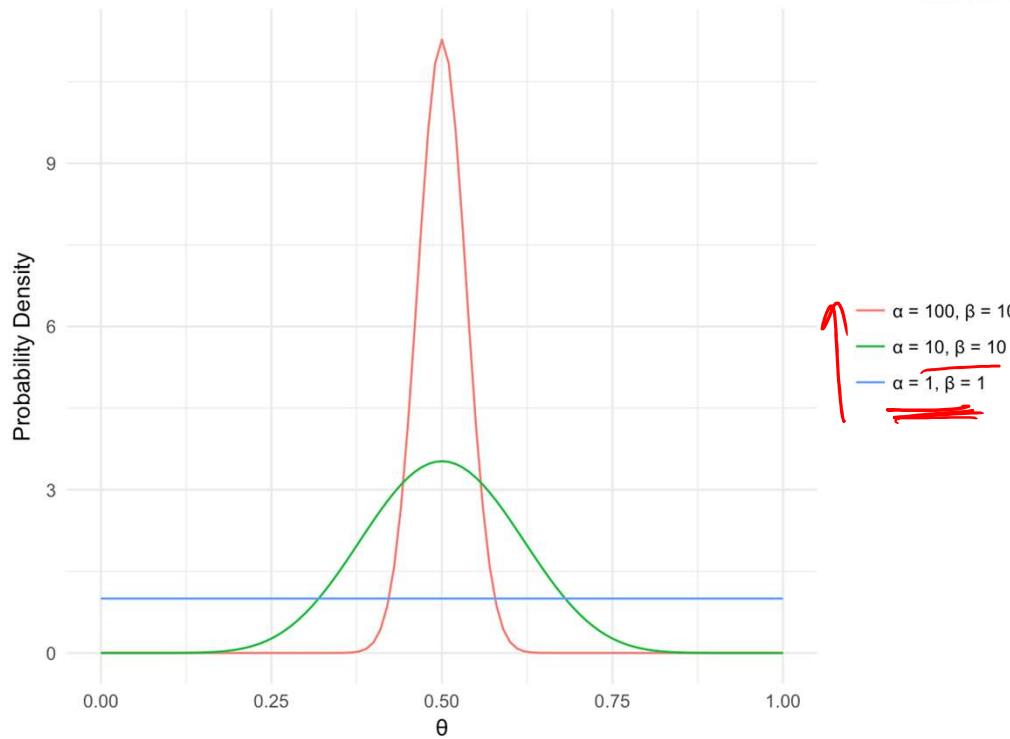
$\theta \in [0, 1]$

$$B(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}$$

$$\Gamma(a) = (a-1)!$$

Beta Distribution

➤ The beta distribution can be thought of as a **probability distribution of probabilities.**



Beta function as a function of Gamma





Thank You!

In our next session: Multinomial Distribution





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Multinomial Distribution

Prof.Aruna Malapati

Learning Objectives

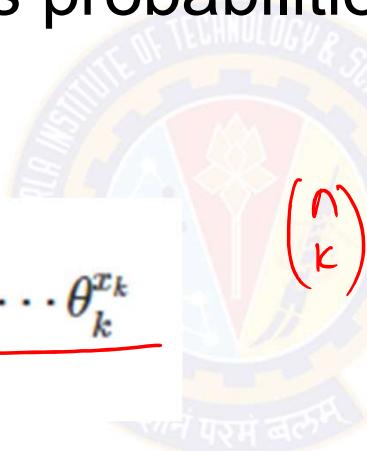
- Multinomial distribution
- Parameter Estimation



From Dice to words

➤ Suppose we roll our die of words having k sides(vocabulary) where each side takes probabilities $\Theta_1, \dots, \Theta_k$ respectively.

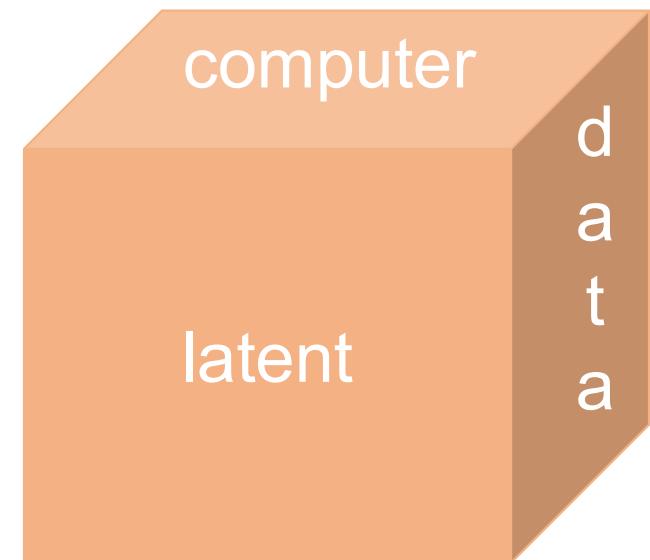
$$f(x) = \frac{n!}{x_1!x_2!\dots x_k!} \theta_1^{x_1} \theta_2^{x_2} \dots \theta_k^{x_k}$$



$$\binom{n}{k} \theta^k (1-\theta)^{n-k}$$

k - number of sides on the die

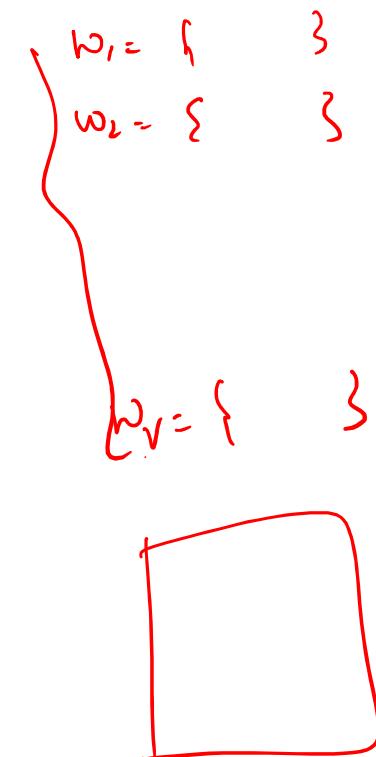
n - number of times the die will be rolled



The Building Blocks of inferring the parameters

- Parameter estimation

$$\underbrace{p(\theta|D)}_{\text{posterior}} = \frac{\overbrace{p(D|\theta) p(\theta)}^{\text{likelihood prior}}}{\underbrace{p(D)}_{\text{evidence}}}$$



- Maximum Likelihood
- Maximum a Posterior (MAP)
- Bayesian Inference ✓



Thank You!

In our next session: Conjugate Prior





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Conjugate Prior

Prof.Aruna Malapati

Conjugate Prior

$$\underbrace{p(\theta|D)}_{\text{posterior}} = \frac{\overbrace{p(D|\theta) p(\theta)}^{\text{likelihood prior}}}{\underbrace{p(D)}_{\text{evidence}}}$$

$$P(D|\theta) \sim \text{Normal} \text{ and } P(\theta) \sim \text{Normal}$$

$$\Rightarrow P(\theta|D) \sim \text{Normal}$$

$$P(D|\theta) \sim \text{Normal} \text{ and } P(\theta) \sim \text{Gamma}$$

$$\Rightarrow P(\theta|D) \propto \text{Normal/gamma}$$

$$P(D|\theta) \sim \text{Bernoulli} \quad \theta^k (1-\theta)^{N-k}$$
$$\text{prior} \sim \overline{\theta^{a-1} (1-\theta)^{b-1}}$$

$$\text{Posterior} \sim \text{Beta}$$



Thank You!

In our next session: Dirichelet distribution



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Dirichlet distributions

Prof. Aruna Malapati

Dirichlet distributions

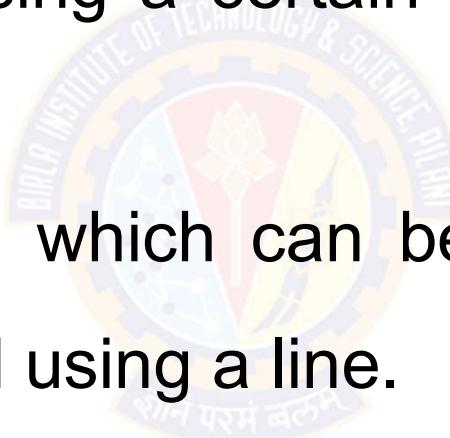
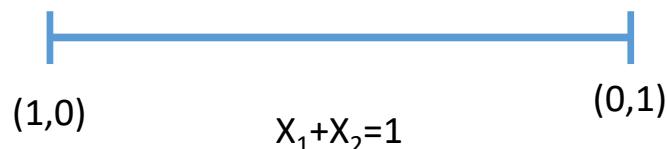
- Dirichlet distributions are probability distributions over multinomial parameter vectors
- They are called Beta distributions when $k = 2$
- The Dirichlet probability density function is defined as

$$Dir(\vec{\theta} | \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K \theta_i^{\alpha_i-1}$$

$$\therefore Dir(\vec{\theta} | \vec{\alpha}) = \frac{1}{B(\alpha)} \prod_{i=1}^K \theta_i^{\alpha_i-1} \quad \text{where} \quad \frac{1}{B(\alpha)} = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)}$$

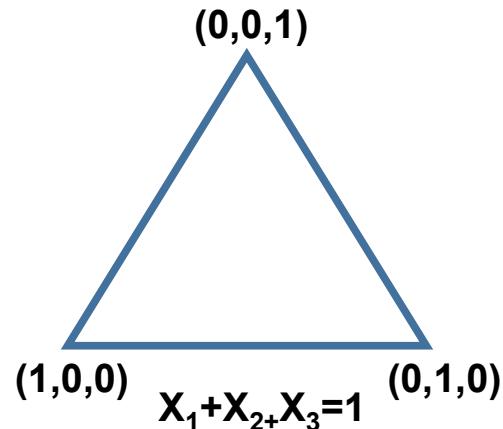
Visualization of the simplex

- This is often referred as **simplex** and a most convenient way to visualize this is using a certain shapes depending upon the number of topics.
- Suppose K=2 topics which can be modeled as 1-simplex and can be visualized using a line.



Visualization of the simplex(Contd...)

- Suppose K=3 topics which can be modeled as 2-simplex and can be visualized using a triangle.



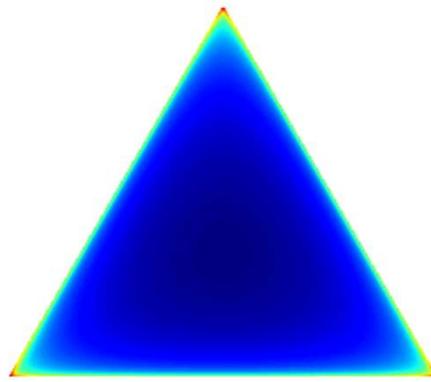
- If we have **K** topics this can be generated using **K-1 simplex**.

Dirichlet distribution is parametrized by α

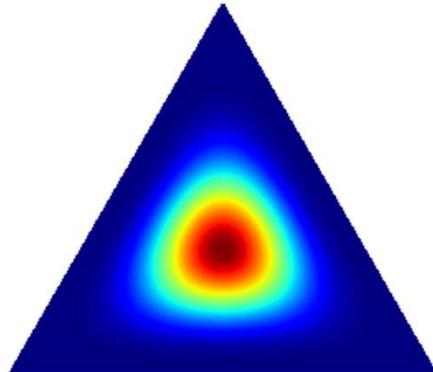


Shape of the Dirichlet distribution

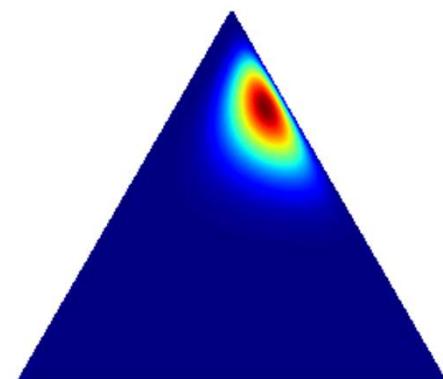
Dirichlet(0.999, 0.999, 0.999)

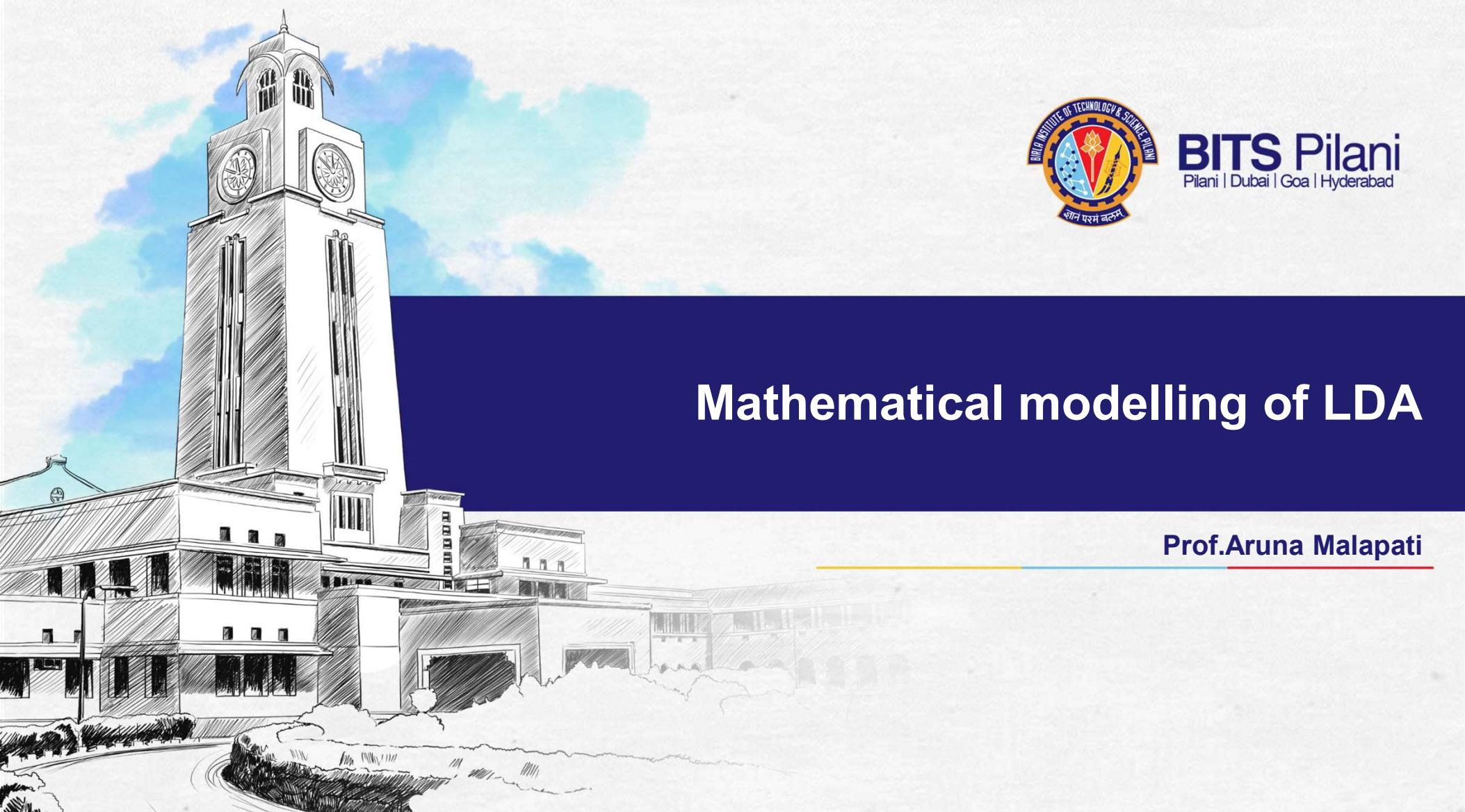


Dirichlet(5, 5, 5)



Dirichlet(2, 5, 15)





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical modelling of LDA

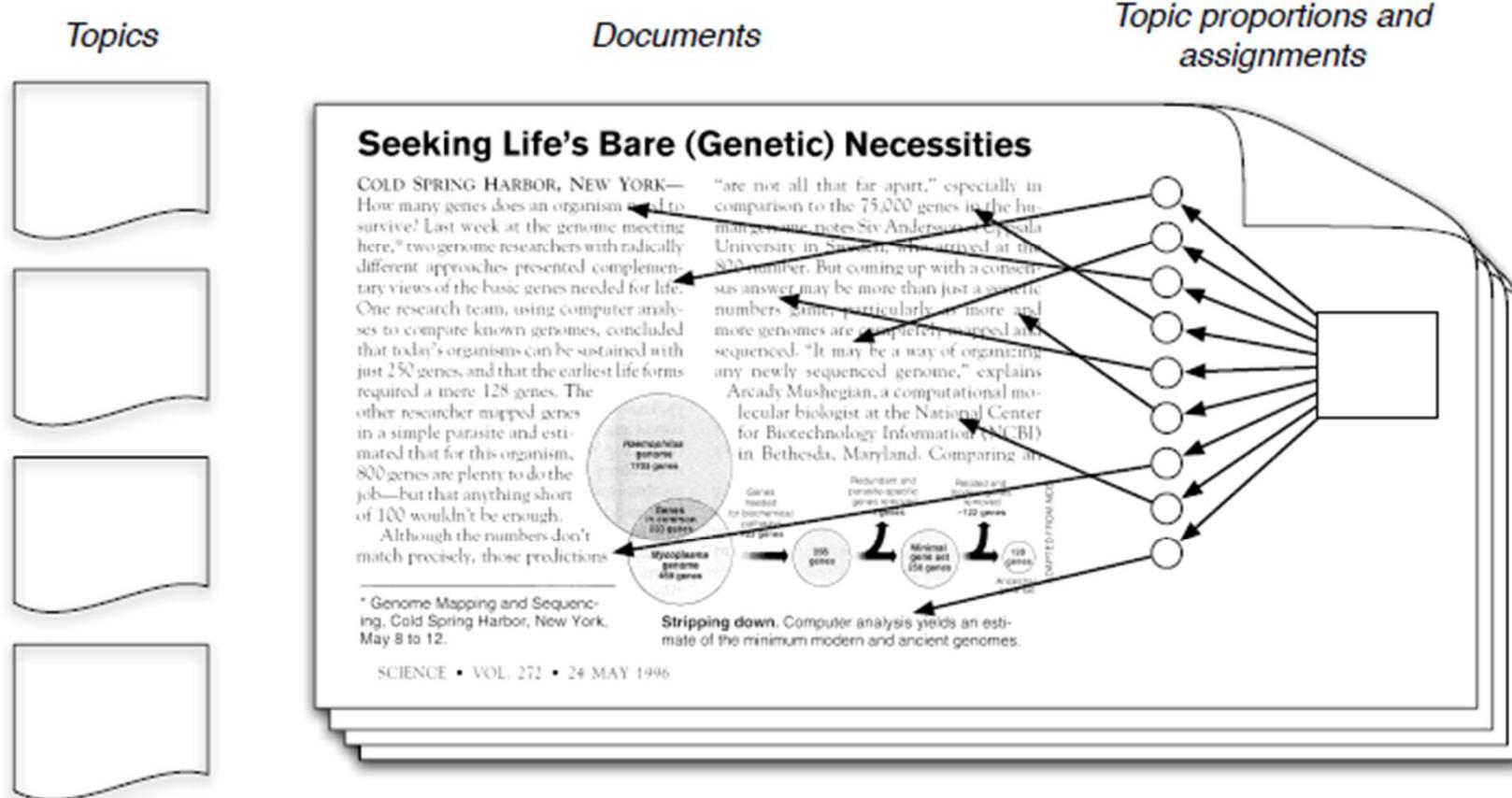
Prof.Aruna Malapati

Learning Objectives

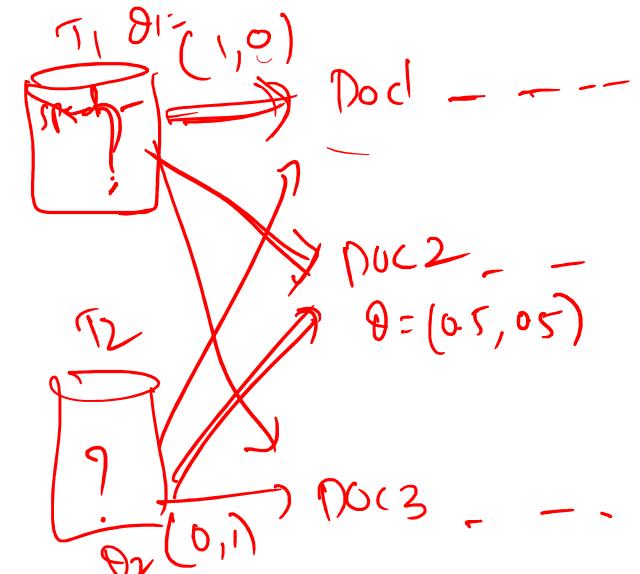
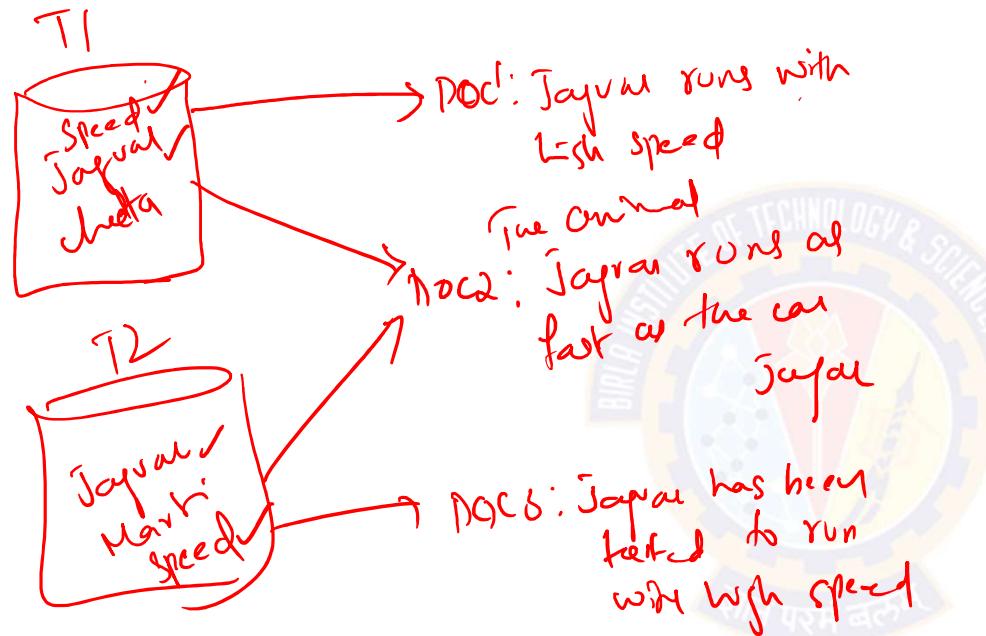
- Generative process of modelling LDA



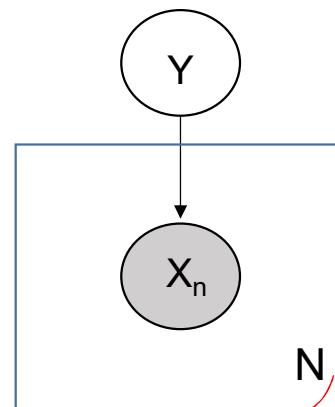
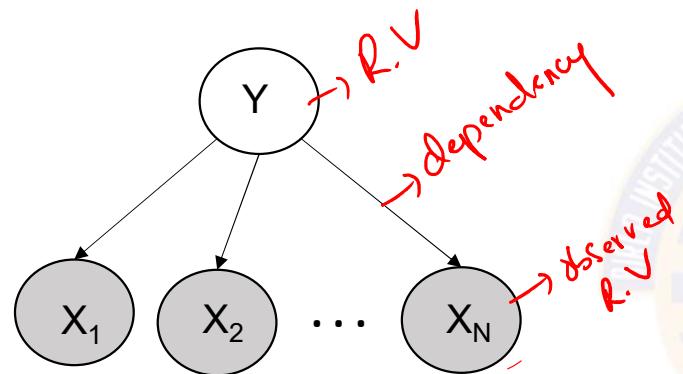
The posterior distribution



Statistical Inference ✓

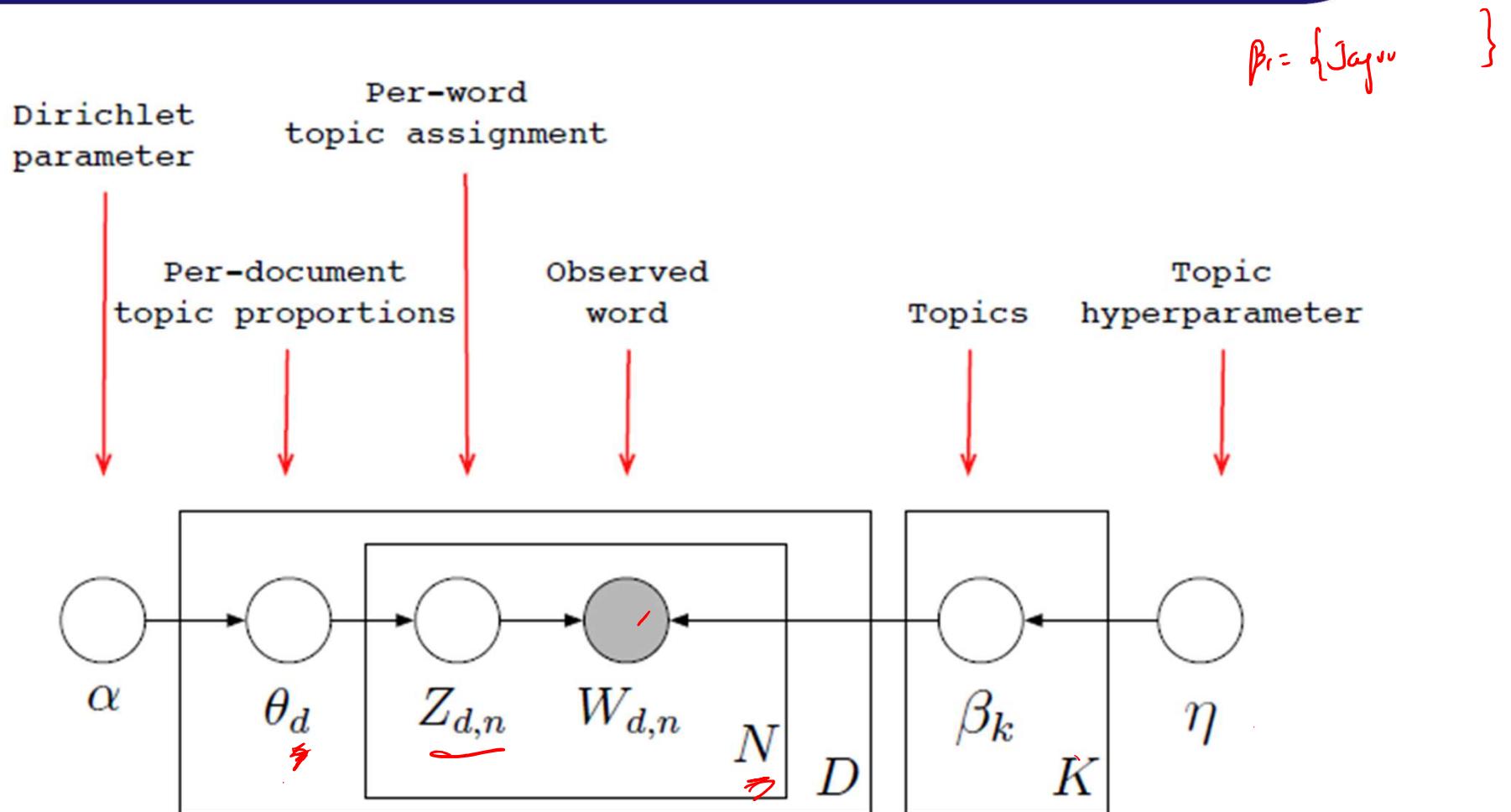


Directed graphical model

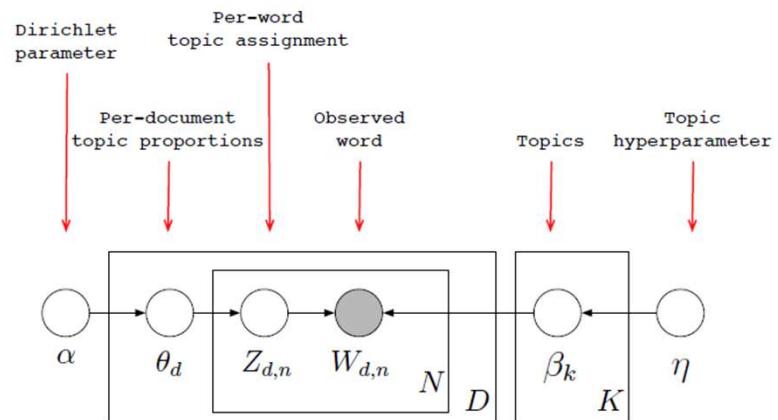


$$P(Y, x_1, x_2, \dots, x_N) = P(Y) \prod_{n=1}^N P(x_n|y)$$

Directed graphical model of LDA



Directed graphical model of LDA (Contd..)



$$\prod_{k=1}^K P(\beta_k | \eta) \left(\prod_{d=1}^D P(\theta_d | \alpha) \right) \left(\prod_{n=1}^N P(Z_{d,n} | \theta_d) \right) P(W_{d,n} | Z_{d,n}, \beta_1, \dots, \beta_K)$$

$\sim \text{Dir}$ $\sim \text{Dir}$

multi

Lion - 0.2	Jayat - 0.4	Cheetah - 0.8	Animals
------------	-------------	---------------	---------

➤ Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$

➤ For each document

➤ Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$

➤ Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$

➤ Draw $W_{d,n} \sim \text{Multinomial}(\beta_{Z_{d,n}})$

$$t_1 = \beta_1 = \dots = \beta_K \sim \text{Dir}(\eta)$$

$$t_2 = \beta_2 = \dots = \beta_K$$

$$\vdots$$

$$t_K = \beta_K$$

$$Z_{d,n} \sim \text{Multinomial}(\theta_d)$$

$$\theta_d = 2$$

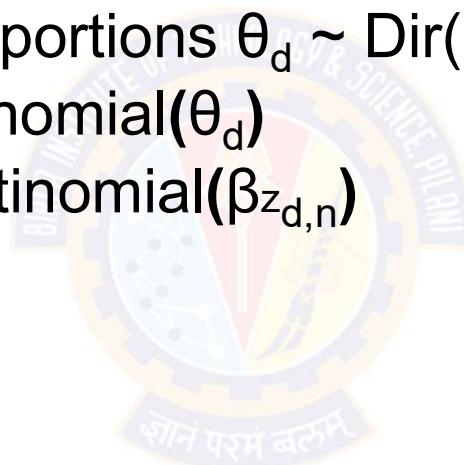
$$d_1 = \theta_1 = (1, 1, 1)^T$$

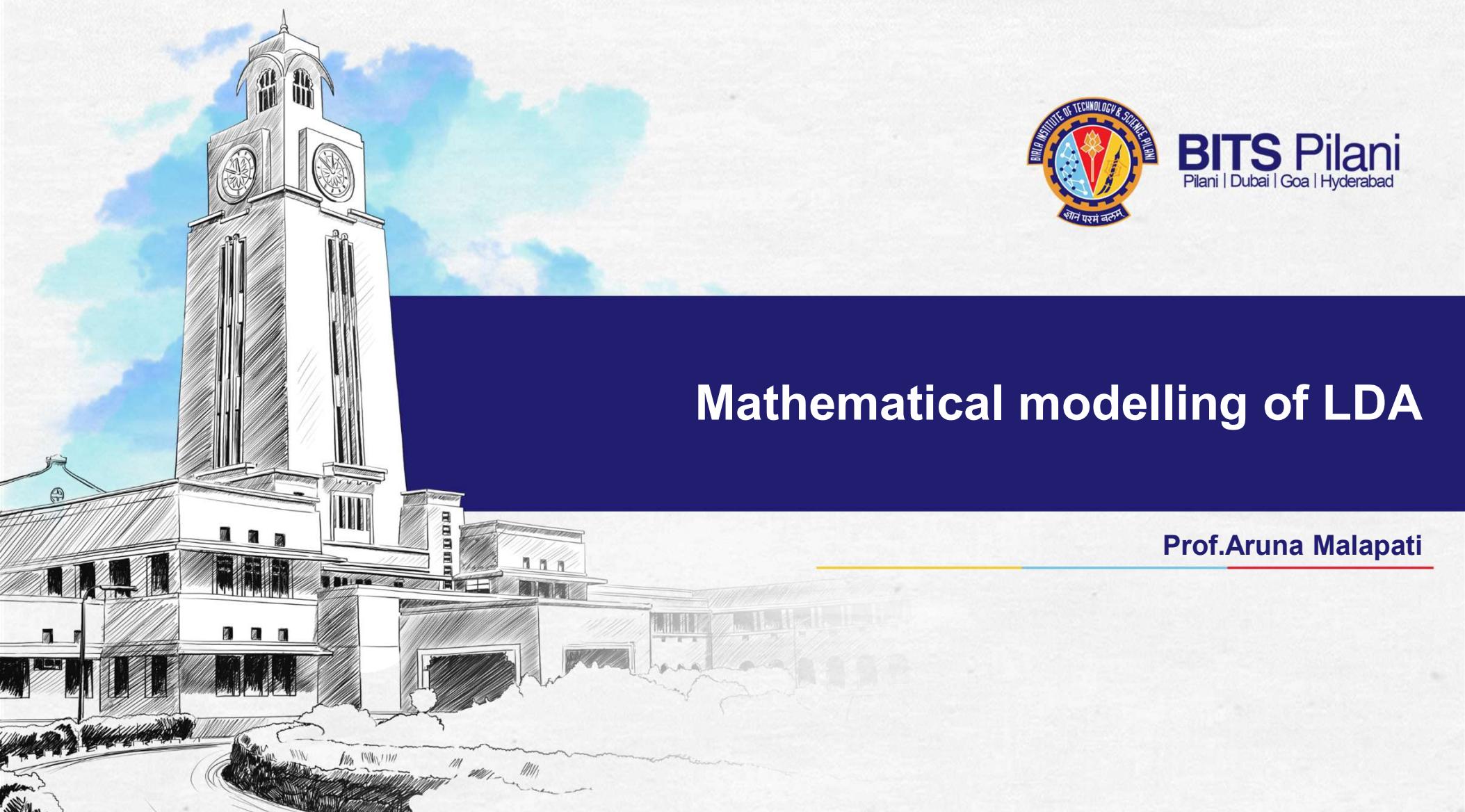
$$d_2 = \theta_2 = (1, 1, 1)^T$$

$d_p \theta_d$

Directed graphical model of LDA (Contd..)

- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical modelling of LDA

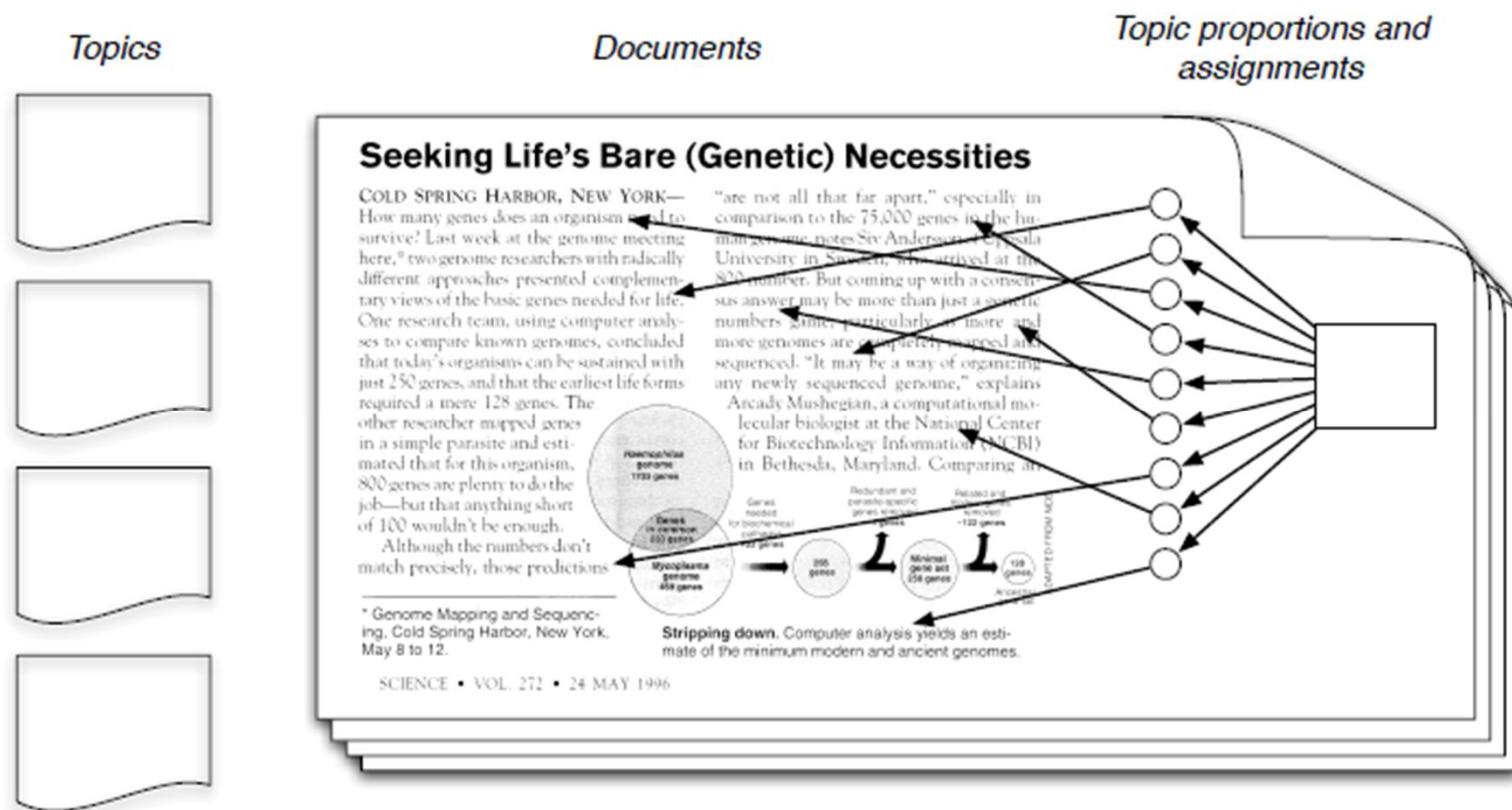
Prof. Aruna Malapati

Learning Objectives

- Generative process of modelling LDA

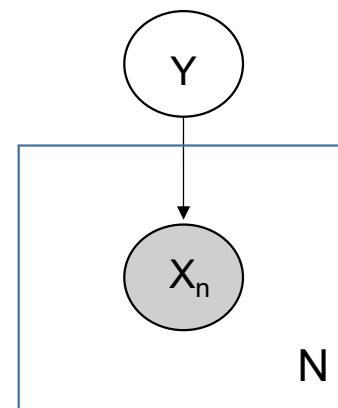
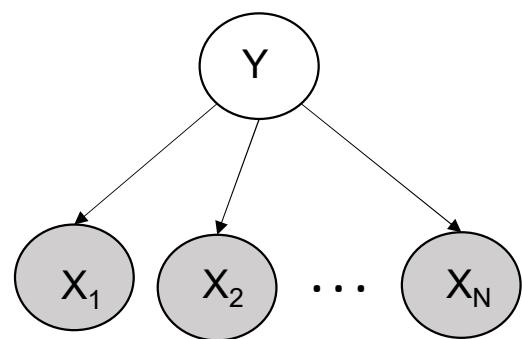


The posterior distribution

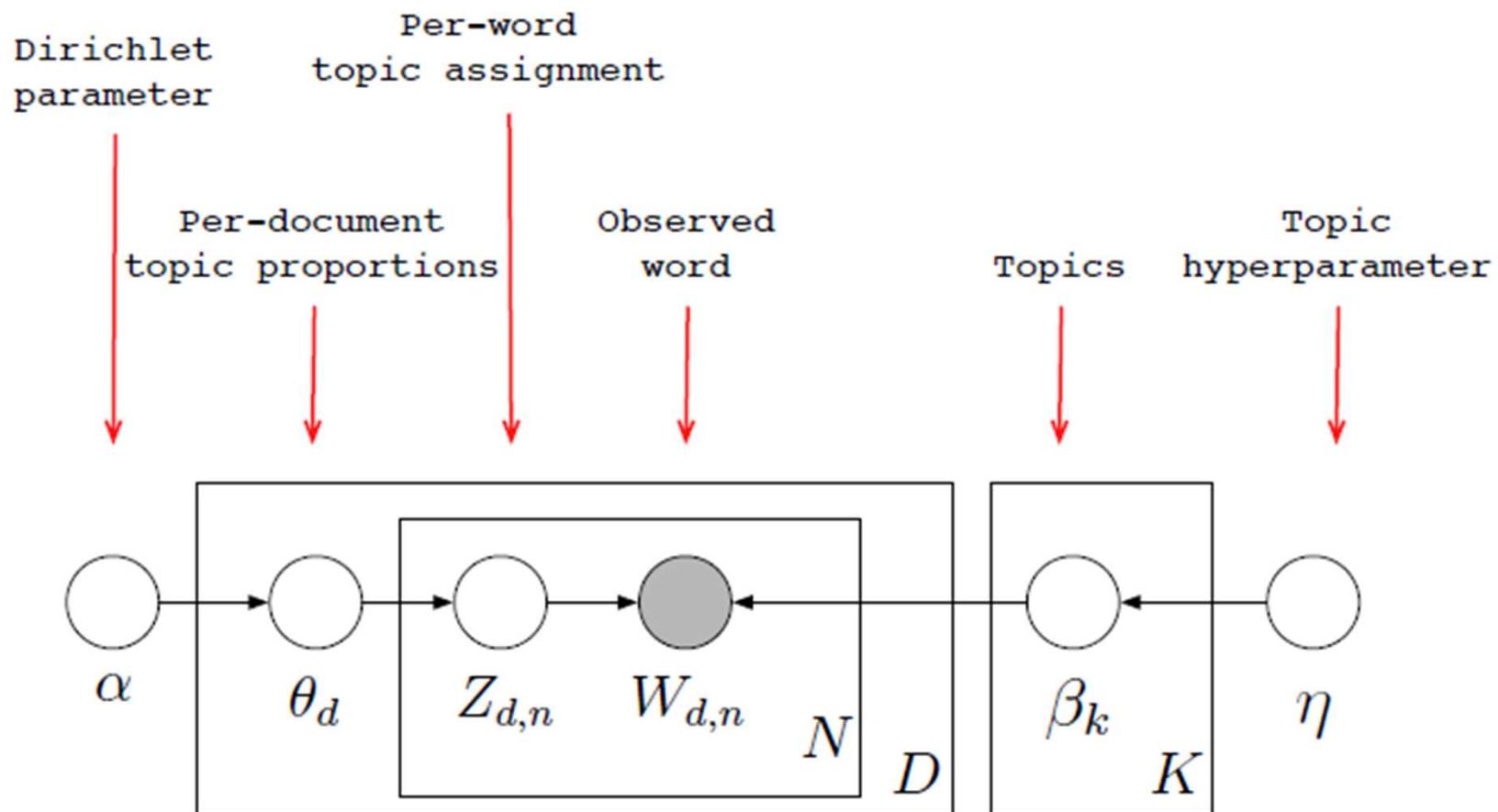




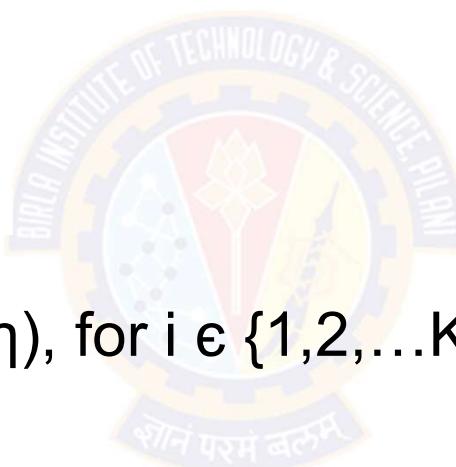
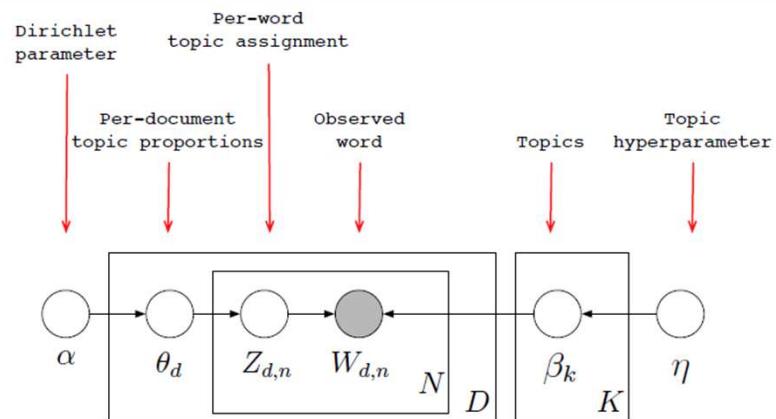
Directed graphical model



Directed graphical model of LDA



Directed graphical model of LDA (Contd..)



- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$

- Draw each topic $\beta_i \sim \text{Dir}(\eta)$, for $i \in \{1, 2, \dots, K\}$
- For each document
 - Draw each topic proportions $\theta_d \sim \text{Dir}(\alpha)$
 - Draw $Z_{d,n} \sim \text{Multinomial}(\theta_d)$
 - Draw $W_{d,n} \sim \text{Multinomial}(\beta_{z_{d,n}})$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Gibbs Sampling for Parameter Estimation

Prof.Aruna Malapati







Algorithm

- Step1: Assign a random topic [1...T] for each word
- Step2: For each word token, a new topic is sampled as per $P(z_i=j|z_{-i}, w_i, d_i)$ and the matrices C_{wt} and C_{dt} are updated.
- One iteration over all word token in the document is a Gibbs Sample
- Each iteration may have correlation with the next hence these samples are saved at spaced intervals.

Estimate for θ and β





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Sentiment Analysis

Prof. Aruna Malapati

Learning Objectives

- Motivation for Sentiment Analysis
- Facts Vs Opinions
- Sentiment Analysis definition
- Applications of Sentiment Analysis



Motivation for Sentiment Analysis

➤ What others think has always been an important piece of information.

“Which mobile should I buy?”

“Which colleges should I apply to?”

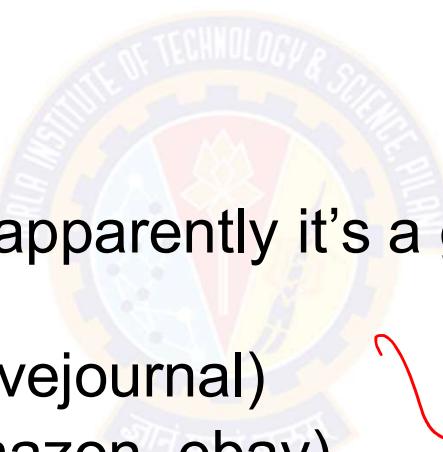
“Which employer is best to work?”

“Whom should I vote for?”



Whom should I ask?

- Pre Web
 - Friends and relatives
 - Acquaintances
 - Consumer Reports
- Post Web
 - "...I don't know who..but apparently it's a good phone. It has good battery life and..."
 - Blogs (google blogs, livejournal)
 - E-commerce sites (amazon, ebay)
 - Review sites (CNET, PC Magazine)
 - Discussion forums forums.craigslist.org, forums.macrumors.com)
 - Friends and Relatives (occasionally)



Too many opinions?????



Different terms for the sentiment analysis

Review
Mining

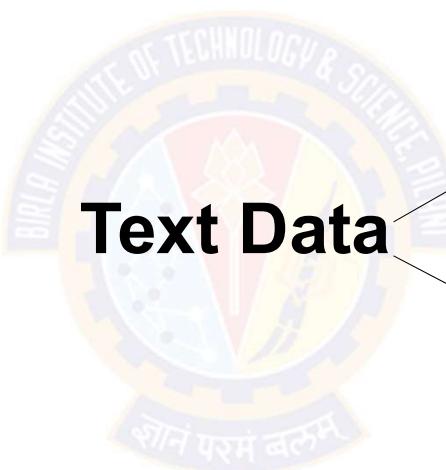
Subjectivity Analysis

Sentiment Analysis

Appraisal
Extraction

Opinion Mining

Facts Vs Opinions



Text Data

Facts

Opinions

Sentiment Analysis

- Computational study of opinions, sentiments, evaluations, attitudes, appraisal, affects, views, emotions, subjectivity, etc., expressed in text.



Lots of applications

- Businesses and organizations
- Individuals
- Ads placements
- Election campaigns
- Policy Acceptance





Thank You!

In our next session: Modelling Sentiment Analysis



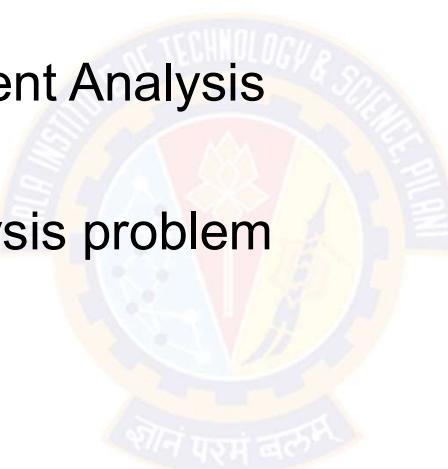
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Modelling Sentiment Analysis

Prof.Aruna Malapati

Learning Objectives

- Sentiment Analysis Problem definition
- Different Levels of Sentiment Analysis
- Modelling Sentiment Analysis problem



The Problem of Sentiment Analysis

“(1) I bought an *iPhone* a few days ago. (2) It was such a *nice phone*. (3) The *touch screen* was really *cool*. (4) The *voice quality* was *clear* too. (5) Although the *battery life* was *not long*, that is ok for me. (6) However, *my mother* was mad with me as I did not tell her before I bought it. (7) She also thought the *phone* was too *expensive*, and wanted me to return it to the shop. ... ”



Definition: An opinion is a quadruple (**target**, **sentiment**, **holder**, **time**)

Practical Definition: (**entity**, **aspect**, **sentiment**, **holder**, **time**)

e.g., (iPhone, touch_screen, +, John, 29-01-2020)

The goal of sentiment analysis is to **mine all quintuples** from the opinion documents.

Different Levels of Sentiment Analysis

- Three levels of granularity
 - Document level
 - Sentence level
 - Entity and Feature/Aspect level



Modelling Sentiment Analysis problem

- The solution to the Sentiment Analysis problem depends on the granularity of the sentiment
- Positive / Negative or 1 to 5 stars : (Binary Classification / Multiclass classification)
 - Naïve Bayes, and support vector machines (SVM), Logistic regression and Maximum Entropy etc..
- Regression if the Sentiment is a continuous value between 1 to 5



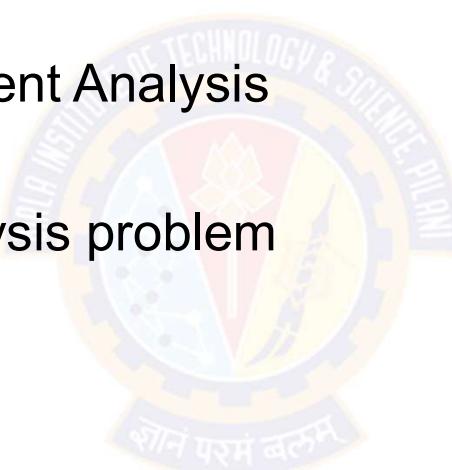
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Modelling Sentiment Analysis

Prof.Aruna Malapati

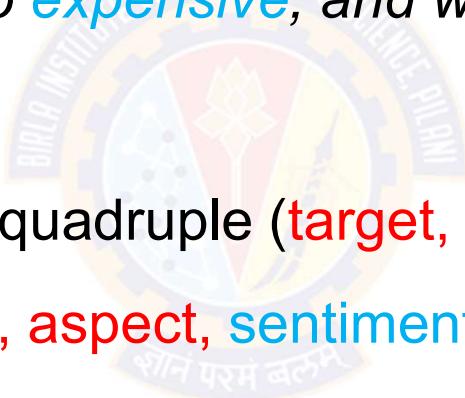
Learning Objectives

- Sentiment Analysis Problem definition
- Different Levels of Sentiment Analysis
- Modelling Sentiment Analysis problem



The Problem of Sentiment Analysis

“(1) I bought an *iPhone* a few days ago. (2) It was such a *nice phone*. (3) The *touch screen* was really *cool*. (4) The *voice quality* was *clear* too. (5) Although the *battery life* was *not long*, that is ok for me. (6) However, my mother was mad with me as I did not tell her before I bought it. (7) She also thought the *phone* was too *expensive*, and wanted me to return it to the shop. ... ”



Definition: An opinion is a quadruple (**target**, **sentiment**, **holder**, **time**)

Practical Definition: (**entity**, **aspect**, **sentiment**, **holder**, **time**)

e.g., (iPhone, touch_screen, +, John, 29-01-2020)

The goal of sentiment analysis is to **mine all quintuples** from the opinion documents.

Different Levels of Sentiment Analysis

- Three levels of granularity
 - Document level
 - Sentence level
 - Entity and Feature/Aspect level



Modelling Sentiment Analysis problem

- The solution to the Sentiment Analysis problem depends on the granularity of the sentiment
- Positive / Negative or 1 to 5 stars : (Binary Classification / Multiclass classification)
 - Naïve Bayes, and support vector machines (SVM), Logistic regression and Maximum Entropy etc..
- Regression if the Sentiment is a continuous value between 1 to 5

The Multinomial Naive Bayes' Classifier

Given a document $d = \{w_1, \dots, w_n\}$
Class $C \in \{C_0, C_1\}$

$$P(\text{Sentiment} | w_1, \dots, w_n) = \frac{P(\text{sentiment}) \prod_{i=1}^n P(w_i | \text{sentiment})}{P(w_1, \dots, w_n)}$$

$$P(\text{sentiment} | w_1, \dots, w_n) \propto P(\text{sentiment}) \prod_{i=1}^n P(w_i | \text{sentiment})$$

$$\log P(\text{Sentiment} | w_1, \dots, w_n) \propto \log P(\text{sentiment}) + \log \prod_{i=1}^n P(w_i | \text{sentiment})$$

$$\stackrel{i < 1 \text{ or } i > 1}{P(C_i | d)} \propto P(C_i) \propto \frac{N_c}{N_{\text{doc}}}$$

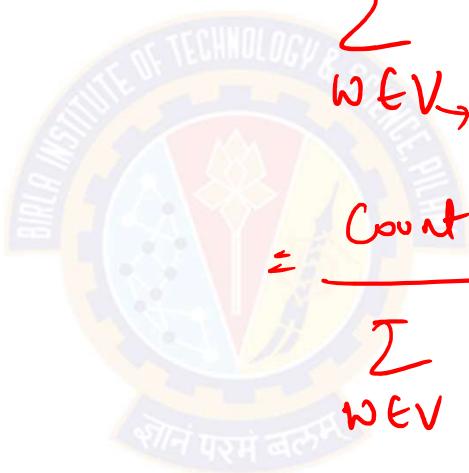
$$(d_1, C_1), (d_2, C_2), \dots, (d_n, C_n)$$

$$P(C_i) = \frac{\text{no. of documents of class } c}{\text{total no. of documents in the training dataset}}$$

$$= \frac{N_c}{N_{\text{doc}}}$$

The Multinomial Naive Bayes' Classifier

$$\begin{aligned} P(w_i|c) &= \frac{\text{Count}(w_i, c)}{\sum_{w \in V} \text{Count}(w, c)} \\ &= \frac{\text{Count}(w_i, c) + 1}{\sum_{w \in V} \text{Count}(w, c) + |V|} \\ &= \frac{\text{Count}(w_i, c) + 1}{\sum_{w \in V} \text{Count}(w, c) + |V|} \end{aligned}$$





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Sentiment Lexicon Resources

Prof.Aruna Malapati

Learning Objectives

- Lexicons and their use
- Resources for sentiment lexicons



Lexicon

- Many sentiment applications rely on lexicons to supply features to a model.
- A lexicon is a **resource with information about words**.
- A sentiment lexicon has information such as list of words which are positive and negative.

General Inquirer (GI)

- Harvard General Inquirer Database (Stone, ~~1966~~)
 - Total of ~~11,788~~ terms
 - http://www.wjh.harvard.edu/~inquirer/spreadsheet_guide.htm
 - <http://www.wjh.harvard.edu/~inquirer/homecat.htm>
 - Positive (1915 words) vs Negative (2291 words)
 - Strong vs Weak ✓
 - Active vs Passive
 - Overstated versus Understated
 - Pleasure, Pain, Virtue, Vice
 - Motivation, Cognitive Orientation, etc

1-5
12345

MPQA Subjectivity Cues Lexicon

➤ Home page:

http://www.cs.pitt.edu/mpqa/subj_lexicon.html

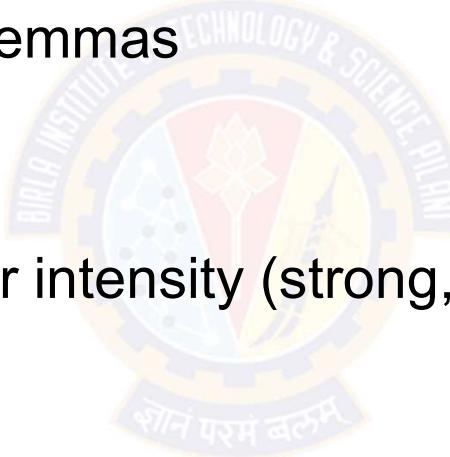
➤ ~~6885 words from 8221 lemmas~~

➤ 2718 positive

➤ 4912 negative

➤ Each word annotated for intensity (strong, weak)

➤ GNU GPL



Theresa Wilson, Janyce Wiebe, and Paul Hoffmann (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proc. of HLT-EMNLP-2005.

Riloff and Wiebe (2003). Learning extraction patterns for subjective expressions. EMNLP-2003.

LIWC Linguistic Inquiry & Word Count

- Home Page: <http://www.liwc.net/>
- 2300 word > 70 classes
- Affective Processes
- Negative emotion (bad, weird, hate, problem,tough) ✓
- Positive emotion (love,nice,sweet) ✓
- Cognitive Processes

Bing Liu Opinion Lexicon

- [Bing Liu's Page on Opinion Mining](#)
- <http://www.cs.uic.edu/~liub/FBS/opinion-lexicon-English.rar>
- 6786 words
 - 2006 positive
 - 4783 negative



Minqing Hu and Bing Liu. Mining and Summarizing Customer Reviews. ACM SIGKDD-2004.

Disagreements between polarity lexicons

	Opinion Lexicon	General Inquirer	SentiWordNet
MPQA ✓	33/5402 (0.6%)	49/2867 (2%)	1127/4214 (27%)
Opinion Lexicon ✓		32/2411 (1%)	1004/3994 (25%)
General Inquirer ✓			520/2306 (23%)
SentiWordNet ✓			

Christopher Potts, [Sentiment Tutorial](#), 2011



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Generating custom based sentiment lexicons

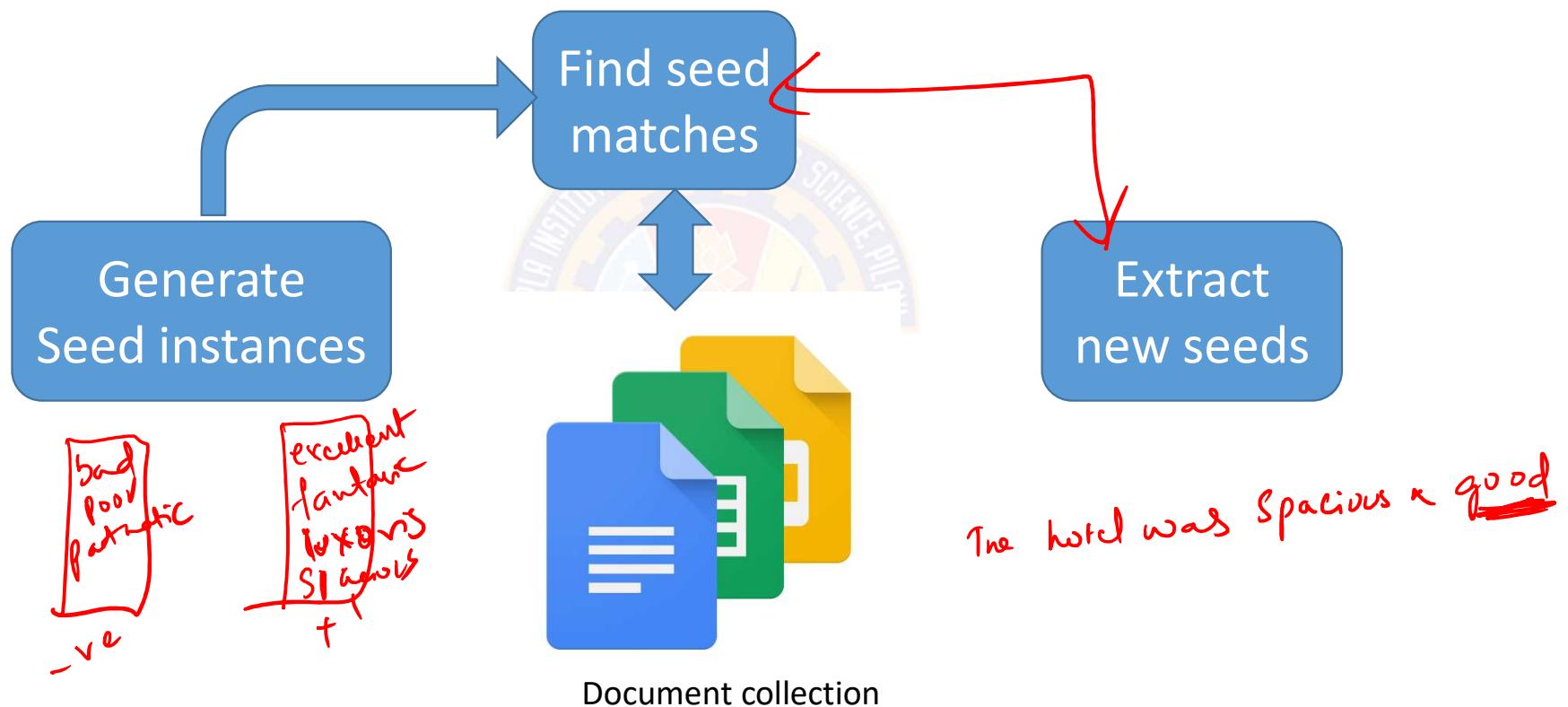
Prof.Aruna Malapati

Learning Objectives

- Bootstrapping
- Corpus based lexicon generation



Bootstrapping Architecture



Corpus-based lexicon generation

- A more sophisticated technique is a corpus-based approach which relies on syntactic or co-occurrence patterns together with a seed list of opinion words.
- The technique starts with a list of seed opinion adjective words, and uses them and a set of linguistic constraints or conventions on connectives to identify additional adjective opinion words and their orientations.
- For example “This house is beautiful and spacious”

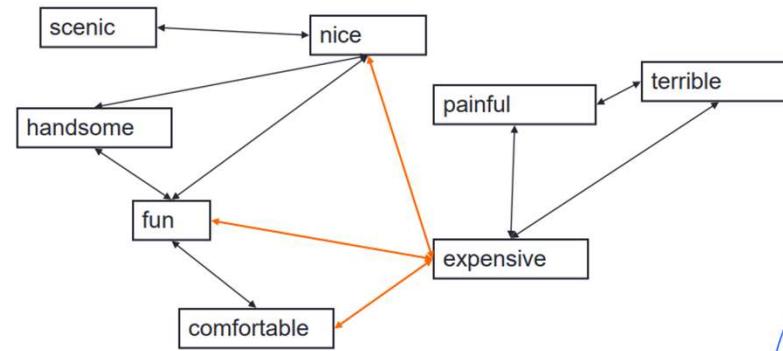
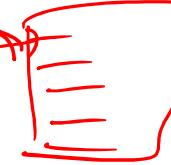
*spacious and lux
good but bad pair Adjective And —*



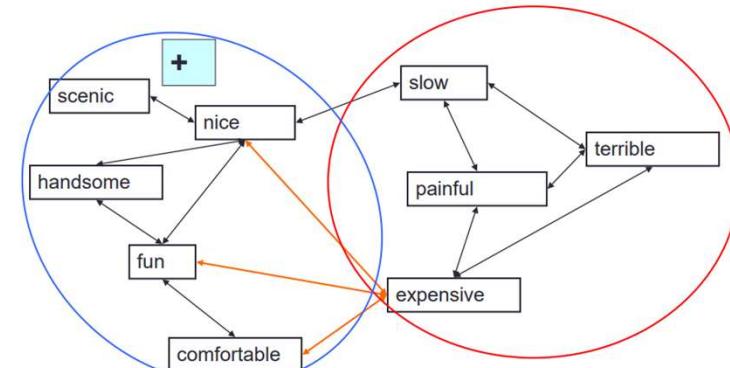
Algorithm

- Generate a Labeled seed set of adjectives
- Expand seed set to conjoined adjectives by looking up in a corpus/web search
- A supervised learning algorithm builds a graph of adjectives linked by the same or different semantic orientation

~~Spacious and
but~~



1977



- A clustering algorithm partitions the adjectives into two subsets

Turney Algorithm

- Extract a phrasal lexicon from reviews
- Learn polarity of each phrase
- Rate a review by the average polarity of its phrases

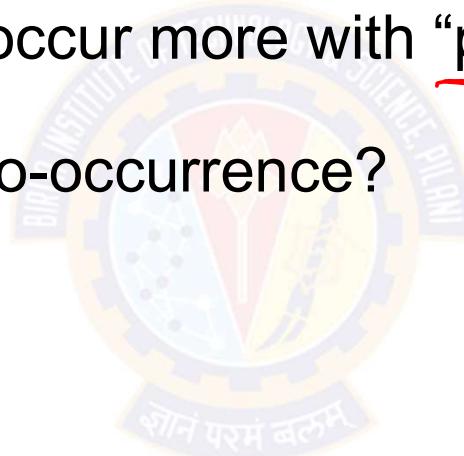


First Word	Second Word	Third Word (not extracted)
JJ <i>Adj</i>	NN or NNS	<u>anything</u>
RB, RBR, RBS <i>Adv</i>	JJ <i>Adj</i>	Not NN nor NNS <u> </u>
JJ <i>Adj</i>	JJ <i>Adj</i>	<u>Not NN or NNS</u>
NN or NNS	JJ	Not NN nor NNS
RB, RBR, or RBS	VB, VBD, VBN, VBG <i>Verb</i>	anything

Two-word phrases with adjectives

How to measure polarity of a phrase?

- Positive phrases co-occur more with “excellent”
- Negative phrases co-occur more with “poor”
- But how to measure co-occurrence?



Pointwise Mutual Information

- Pointwise mutual information: How much more do events x and y co-occur than if they were independent?

$$\text{PMI}(\underline{\text{word}_1}, \underline{\text{word}_2}) = \log_2 \frac{P(\text{word}_1, \text{word}_2)}{P(\text{word}_1)P(\text{word}_2)}$$

- If two words are statistically independent, $\text{PMI} = 0$ $\log_2 \frac{P(w_1)P(w_2)}{P(w_1)P(w_2)} = \log_2 1 = 0$
- If two words tend to not at all co-occur , PMI is negative $\log_2 \frac{P(w_1 \sim w_2)}{P(w_1)P(w_2)} = -\infty$
- If two words tend to co-occur , PMI is positive

Does phrase appear more with “poor” or “excellent”?

➤ Polarity(phrase) = PMI(phrase, “excellent”) - PMI(phrase, “poor”)

$$\text{SO}(\text{phrase}) = \log_2 \left[\frac{\text{hits}(\text{phrase NEAR “excellent”}) \text{ hits}(“poor”)}{\text{hits}(\text{phrase NEAR “poor”}) \text{ hits}(“excellent”)} \right]$$



Spacious rooms are excellent

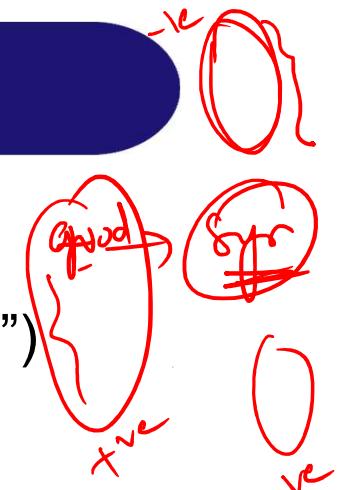
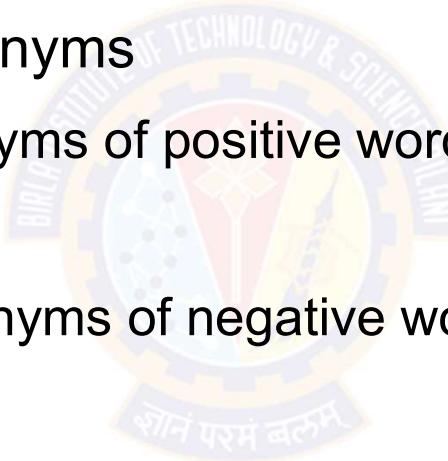
Two reviews for Positive and Negative phrases

Phrase	POS tags	Polarity
online service	JJ NN	2.8
online experience	JJ NN	2.3
direct deposit	JJ NN	1.3
local branch	JJ NN	0.42
...		
low fees	JJ NNS	0.33
true service	JJ NN	-0.73
other bank	JJ NN	-0.85
inconveniently located	JJ NN	-1.5
Average		0.32

Phrase	POS tags	Polarity
direct deposits	JJ NNS	5.8
online web	JJ NN	1.9
very handy	RB JJ	1.4
...		
virtual monopoly	JJ NN	-2.0
lesser evil	RBR JJ	-2.3
other problems	JJ NNS	-2.8
low funds	JJ NNS	-6.8
unethical practices	JJ NNS	-8.5
Average		-1.2

Wordnet based polarity estimation

- WordNet: online thesaurus indexing words by synonyms
- Create positive ("good") and negative seed-words ("terrible")
- Find Synonyms and Antonyms
 - Positive Set: Add synonyms of positive words ("well") and antonyms of negative words
 - Negative Set: Add synonyms of negative words ("awful") and antonyms of positive words ("evil")
- Repeat, following chains of synonyms
- Filter





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Aspect Based Sentiment Analysis

Prof.Aruna Malapati

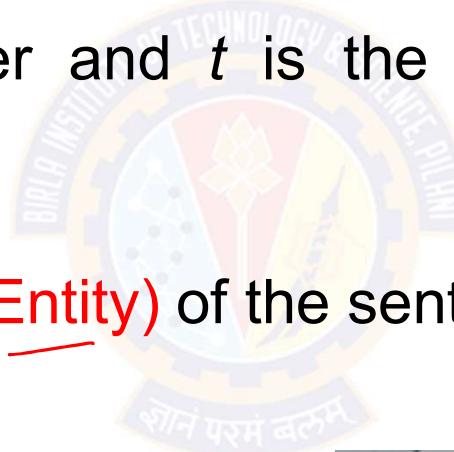
Learning Objectives

- Aspect Based Sentiment Analysis (ABSA)
- Frequency based Aspect Extraction



Aspect Based Sentiment Analysis (ABSA)

- Each opinion is defined as quintuple (e, a, s, h, t), where e is an entity and a is one of its aspects, s is the sentiment on the aspect a, h is the opinion holder and t is the time when the opinion is expressed.
- Find the target(Aspect/Entity) of the sentiment.

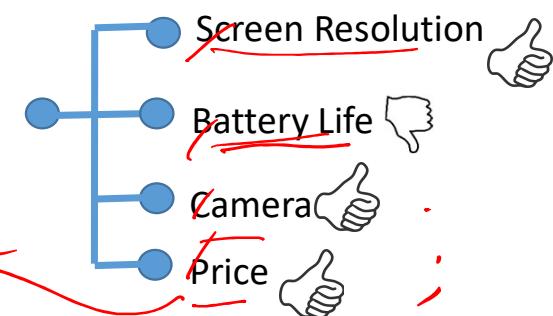


↑ ↑ ↑
I bought an iphone and the
voice quality was extremely good. }

- Two approaches

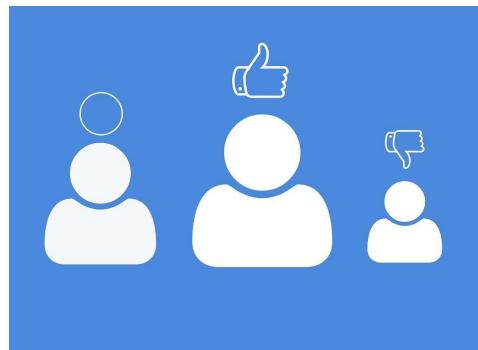
- Find most common noun phrases

- Build a classifier



Frequency-Based Aspect Extraction

- A key characteristic is that an **opinion always has a target**.
- Exploit **syntactic structures** to depict opinion and target relationships



Review corpus

Association Rule
Mining



Screen Size – 100/500 ✓
Camera Resolution – ~~300/500~~
Battery Life – 350/500
Price -450/500
Voice clarity – 325/500

t_1 ↘
↙
↙
↙

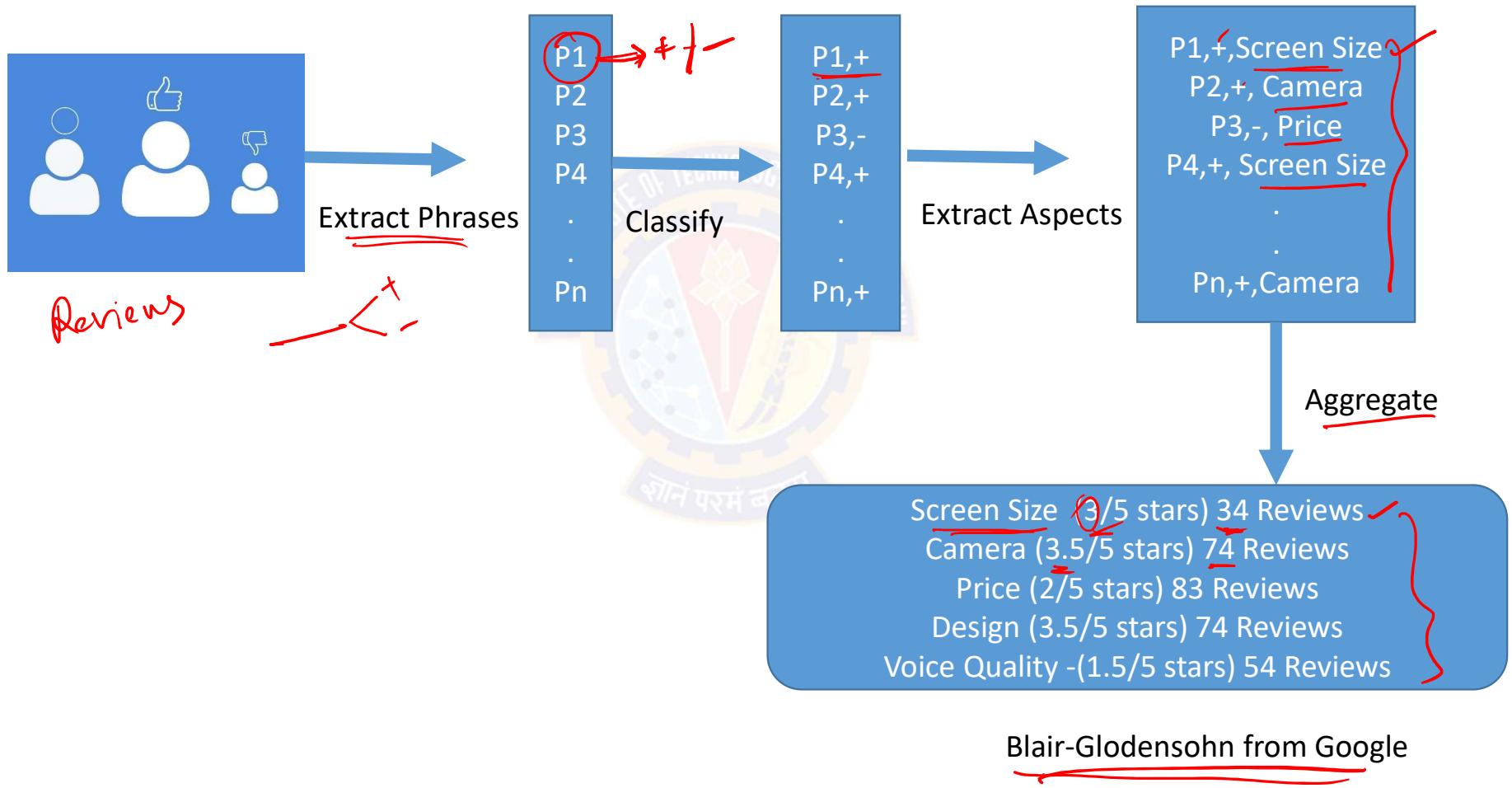


Examples of aspects extracted

Entity	Aspects extracted
Casino	Casino, <u>buffet</u> , <u>pool</u> , <u>resort</u> , <u>beds</u>
Department store	Selection, department, sales, shop, clothing
Greek Restaurant	Food, Wine, Service, Appetizer, lamb

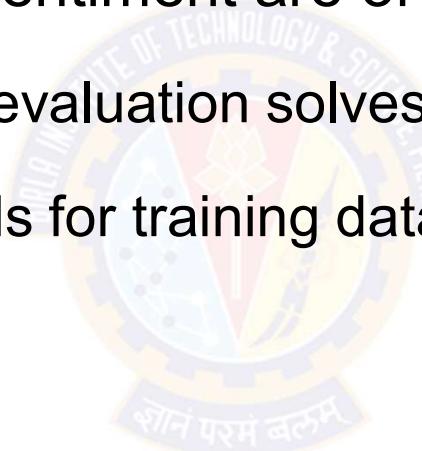
- Those **candidate aspects with the highest frequency counts** are almost always the most **important aspects** of the product.
- Assumption: Corpus has **reasonable number of reviews** and belong to same product.

Architecture for Aspect Based Sentiment Analysis

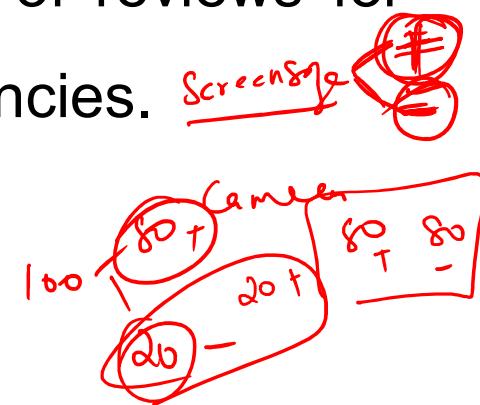


Assumptions

- The baseline algorithm assumes that the number of reviews for Positive and Negative sentiment are of equal frequencies.
- Usage of F-Score for evaluation solves this problem.
- Use Sampling methods for training data



ScreenShot



How to deal with star ratings?

⑧

- Binarization of the star ratings. $\begin{cases} 0 & \text{if } r < 2.5 \\ 1 & \text{if } r \geq 2.5 \end{cases}$
- Use regression instead of a binary classifier.

$\begin{cases} 0 & \text{if } r < t \\ 1 & \text{if } r \geq t \end{cases}$





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Aspect Based Sentiment Analysis

Prof.Aruna Malapati

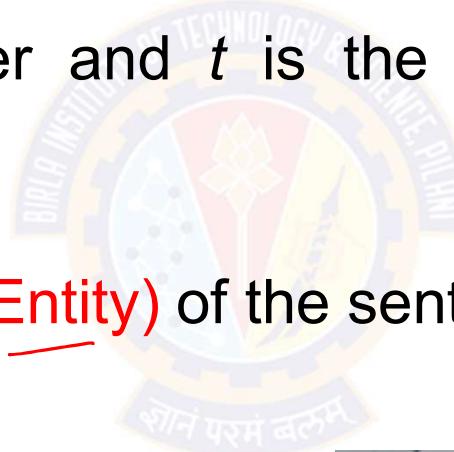
Learning Objectives

- Aspect Based Sentiment Analysis (ABSA)
- Frequency based Aspect Extraction



Aspect Based Sentiment Analysis (ABSA)

- Each opinion is defined as quintuple (e, a, s, h, t), where e is an entity and a is one of its aspects, s is the sentiment on the aspect a, h is the opinion holder and t is the time when the opinion is expressed.
- Find the target(Aspect/Entity) of the sentiment.

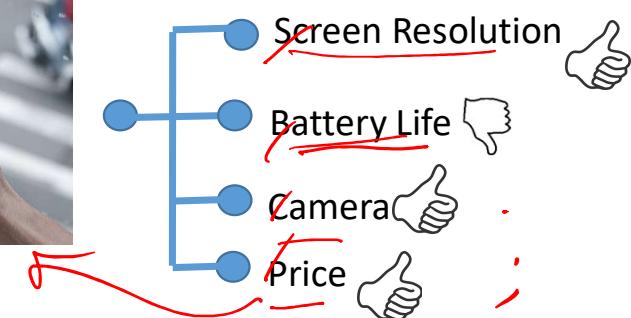


↑ ↑ ↑
I bought an iphone and the
voice quality was extremely good. }
} {

- Two approaches

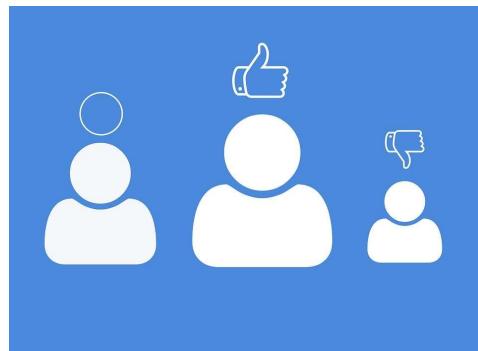
- Find most common noun phrases

- Build a classifier



Frequency-Based Aspect Extraction

- A key characteristic is that an **opinion always has a target**.
- Exploit **syntactic structures** to depict opinion and target relationships



Review corpus

Association Rule Mining



Screen Size – 100/500 ✓
Camera Resolution – ~~300/500~~
Battery Life – 350/500
Price -450/500
Voice clarity – 325/500

t_1 ↘
↙
↙
↙

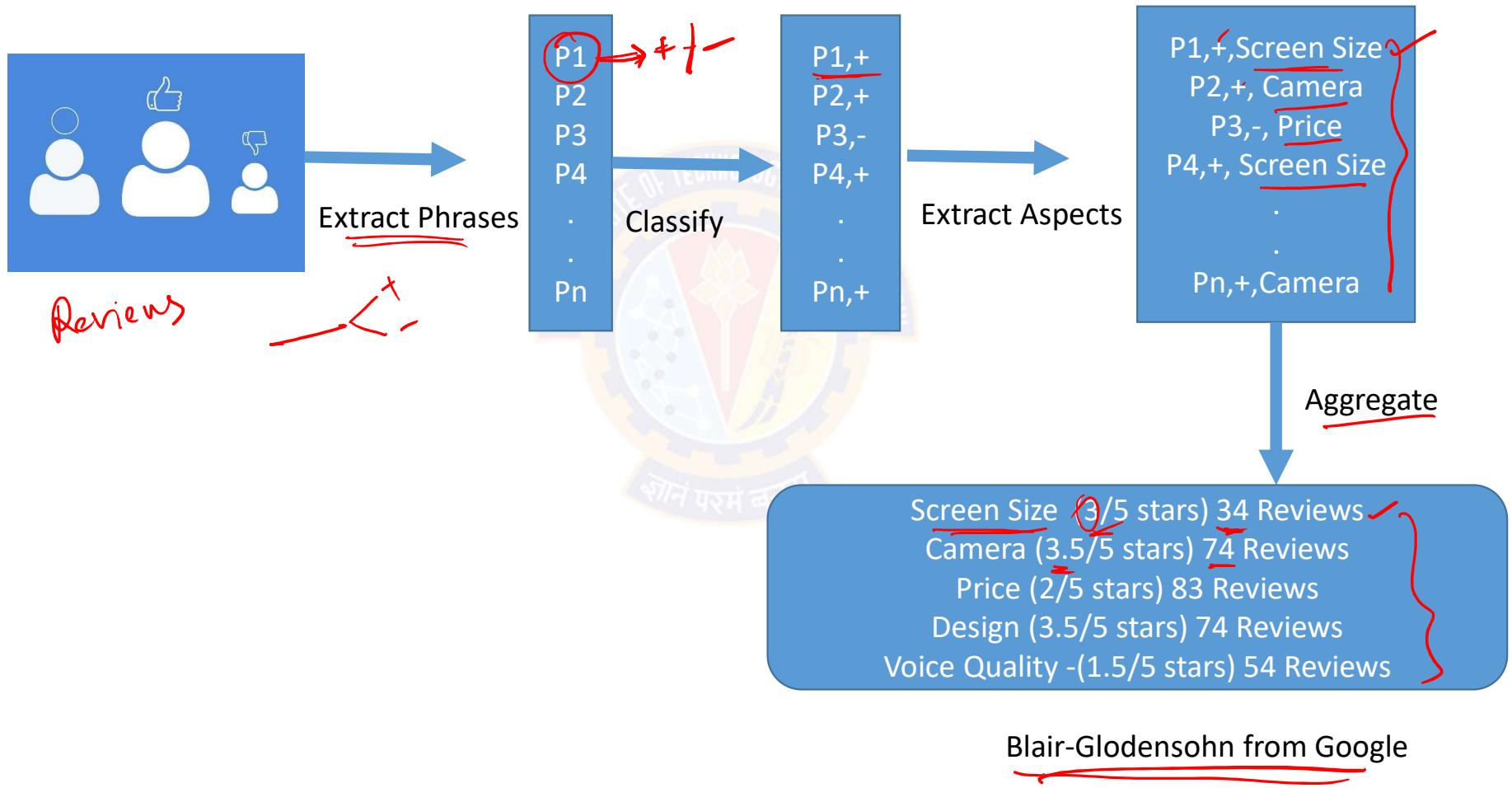
{

Examples of aspects extracted

Entity	Aspects extracted
Casino	Casino, <u>buffet</u> , <u>pool</u> , <u>resort</u> , <u>beds</u>
Department store	Selection, department, sales, shop, clothing
Greek Restaurant	Food, Wine, Service, Appetizer, lamb

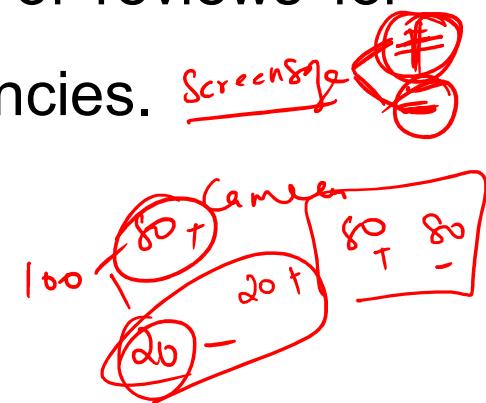
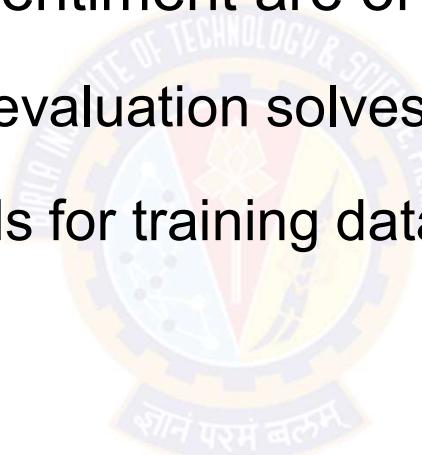
- Those **candidate aspects with the highest frequency counts** are almost always the most **important aspects** of the product.
- Assumption: Corpus has **reasonable number of reviews** and belong to same product.

Architecture for Aspect Based Sentiment Analysis



Assumptions

- The baseline algorithm assumes that the number of reviews for Positive and Negative sentiment are of equal frequencies.
- Usage of F-Score for evaluation solves this problem.
- Use Sampling methods for training data



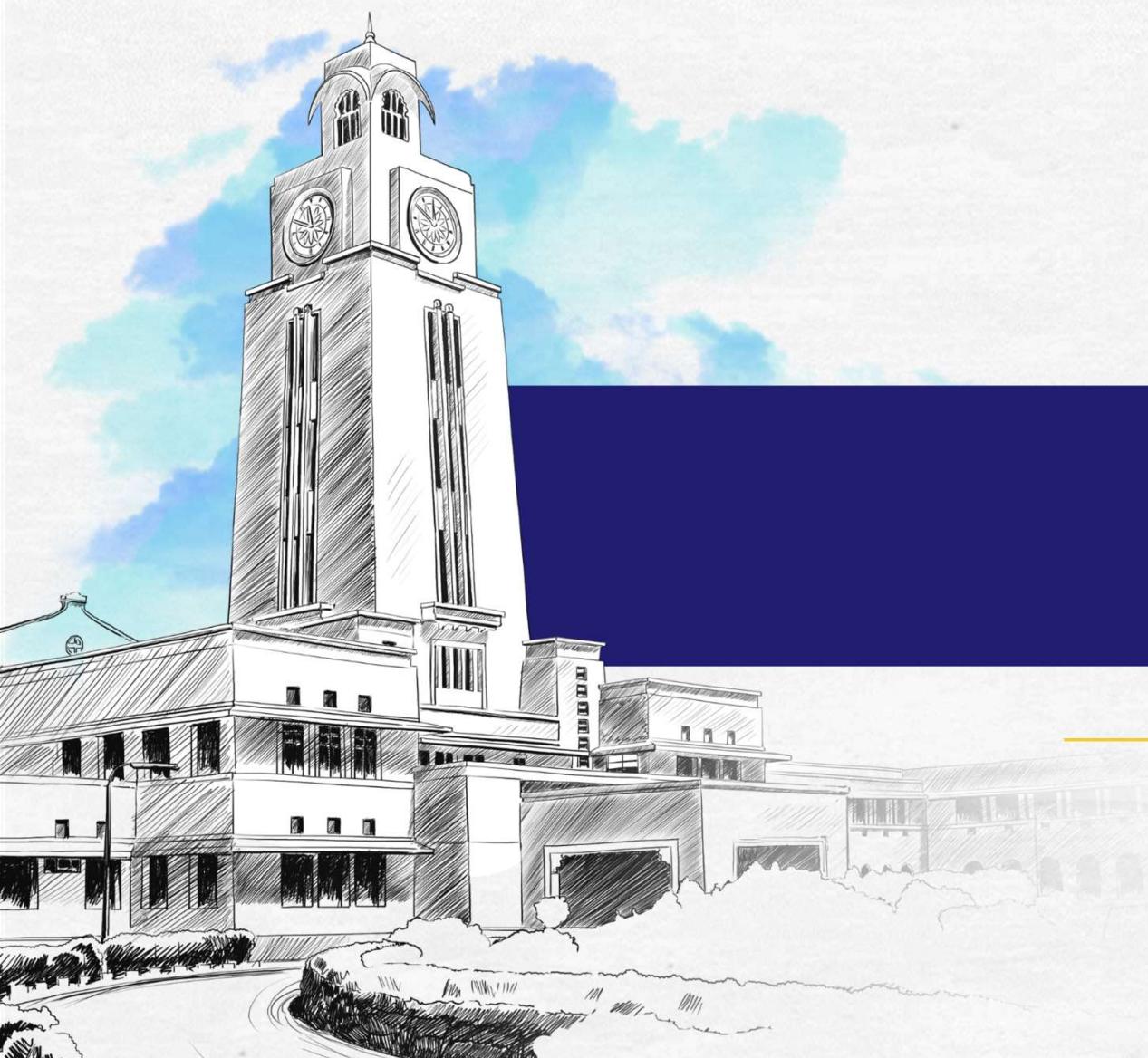
How to deal with star ratings?

⑧

- Binarization of the star ratings. $\begin{cases} 0 & \text{if } r \leq 2.5 \\ 1 & \text{if } r > 2.5 \end{cases}$
- Use regression instead of a binary classifier.

$\begin{cases} 0 & \text{if } r \leq t \\ 1 & \text{if } r > t \end{cases}$





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Opinion Spamming

Prof.Aruna Malapati

Opinion Spamming

➤ Types of Spam

➤ Type 1 (fake reviews)

➤ Type 2 (reviews about brands only)

➤ Type 3 (non-reviews)



Types of Data, Features and Detection

- Three main types of data have been used for review spam detection:
 - Review content
 - Meta-data about the review
 - Product information



Supervised Spam Detection

- Opinion spam detection can be formulated as a classification problem with two classes, fake and non-fake.
- Due to the fact that there is no labeled training data for learning, Jindal and Liu (2008) exploited duplicate reviews.
- In their study of 5.8 million reviews and 2.14 million reviewers from amazon.com, a large number of duplicate and near-duplicate reviews were found.

Four categories to handle duplicates and near duplicates

- Duplicates from the same user-id on the same product
- Duplicates from different user-ids on the same product
- Duplicates from the same user-id on different products
- Duplicates from different user-ids on different products

Feature engineering for fake reviews

- Review centric features
- Reviewer centric features
- Product centric features



Some interesting observations from the study

- Only reviews of some products are likely to be fake.
- Top-ranked reviewers are more likely to be fake reviewers.
- Products of lower sales ranks are more likely to be spammed.



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Introduction to Recommender Systems

Prof.Aruna Malapati

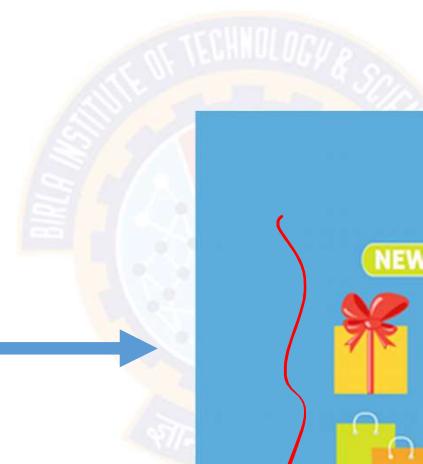
Learning Objectives

- Motivation for Recommender Systems
- Modelling the problem of Recommender Systems



➤ Motivation for Recommender Systems

Search Vs Recommender systems



Commercial Interests for Recommender systems

- Netflix: 2/3 of the movies watched
- Amazon: 35% sales
- Google news: recommendations ⇒ 38% more click through
- Choicestream: 28% of people would buy more music if they found they like the recommendation.

Modelling Recommender Systems

➤ $U = \{\underline{\text{USERS}}\}$

U I

➤ $I = \{\underline{\text{ITEMS}}\}$

➤ F is a utility function, measures the usefulness of items I to user U .

$F: U \times I \rightarrow \underline{R}$ where R is the rating $0, 1$
 $0 - 5$

➤ Characteristics of a good Utility function

➤ Personalized

➤ Diverse

➤ Serendipity

Netflix Utility Matrix



Movie/User	Movie-1	Movie-2	Movie-3	Movie-4
User-1	4	3.5	5	5
User-2	4	3	4.5	?
User-3	4.5	4	2	3

2006-2009

Improvement in RMSE by 10%

Prize: \$1 Million

Netflix dataset: over 17K movies and 500K+ Users!

Problems faced while building recommender systems

- Gathering the rating from users.
Explicit (1-5)
Implicit
- Developing right models to learn function from known ratings
- Evaluating the models for unknown rating

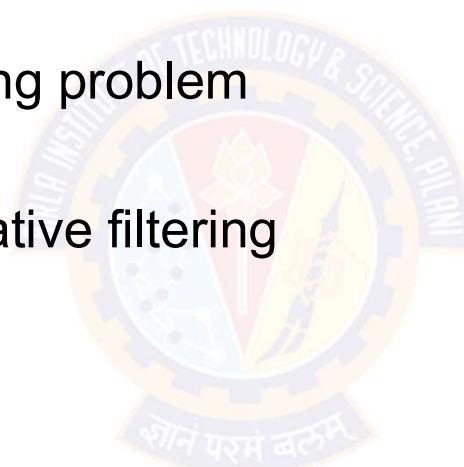


Collaborative Filtering

Prof.Aruna Malapati

Learning Objectives

- Baseline method for predicting the ratings
- Define collaborative filtering problem
- User/Item based collaborative filtering



Baseline approach for rating prediction

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

Now Avg - μ

$$\text{Predicted Rating}(E, \text{Equilibrium}) = \mu + b_E + b_{\text{Equilibrium}}$$

Where μ is the global average

b_E is the deviation of the user E

$b_{\text{Equilibrium}}$ is the deviation of the Movie Equilibrium

$$\text{Predicted Rating} = 3.78 + 0.22 + 0.22 = 4.22$$

Collaborative Filtering

- The task of predicting user preferences on new items is by collecting taste of similar users.

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

- Each user has expresses an opinion for some items
 - Explicit opinion: Rating score ✓
 - Implicit: Purchase records or listen to the tracks ✗

User Based Collaborative Filtering

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?

- Identify the set of items rated by the target user.
- Identify which other users have rated the same items as target user.
- Compute similarity of each user to the target user.
- Select top K similar users

Finding similar users

➤ Let r_x and r_y be the vector of users x and y 's ratings

➤ **Jaccard similarity measure between x and y**

➤ **Problem:** Ignores the value of the rating

➤ **Cosine similarity measure**

$$\text{sim}(x, y) = \cos(r_x, r_y) = \frac{r_x \cdot r_y}{\|r_x\| \cdot \|r_y\|}$$

➤ **Problem:** Treats missing ratings as “negative”

➤ **Pearson correlation coefficient**

➤ S_{xy} = items rated by both users x and y

$$Sim(x, y) = \frac{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)(r_{ys} - \bar{r}_y)}{\sqrt{\sum_{s \in S_{xy}} (r_{xs} - \bar{r}_x)^2} \sqrt{\sum_{s \in S_{xy}} (r_{ys} - \bar{r}_y)^2}}$$

\bar{r}_x, \bar{r}_y ... avg.
rating of x, y

any
 $x \cup y$

$k=2$
 $\hookrightarrow A, D$

Estimating the Predicted Rating

- Let N be the set of k users most similar to x who have rated item i
- Prediction for item s of user x :

$k=2$
 $N = \{A, D\}$

$$r_{xi} = \frac{1}{k} \sum_{y \in N} r_{yi} \rightarrow$$

$$r_{xi} = \frac{\sum_{y \in N} s_{xy} \cdot r_{yi}}{\sum_{y \in N} s_{xy}}$$

Item Based Collaborative Filtering

User/Movie	Batman	Alice in Wonderland	Dumb and Dumber	Equilibrium
User A	4		3	5
User B		5	4	
User C	5	4	2	
User D	2	4		3
User E	3	4	5	?



User Based Vs Item Based

- User based Similarity is more dynamic and can be used if
 - Item base is smaller than user base
 - Item base rapidly changes rapidly
- Item based Similarity is static and recommends new items that were also liked by the same users
 - Good if the use base is small

||| |||

Issues in implementing Collaborative Filtering

- Many Items to choose from
- Few data per user
- No data for new users
- Very large datasets





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Prerequisite for Matrix factorization-1

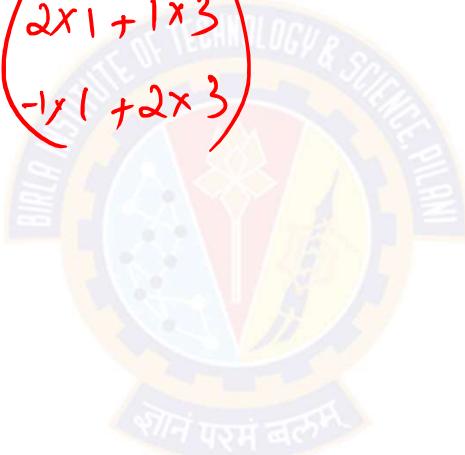
Prof.Aruna Malapati

Matrix vector multiplication

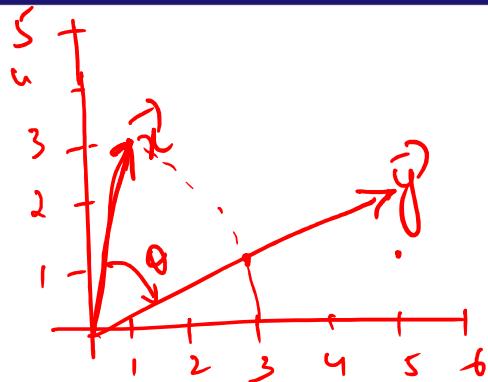
$$\vec{x} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \quad M = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix}$$

$$\vec{y} = M\vec{x} = \begin{pmatrix} 2 & 1 \\ -1 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} 2 \times 1 + 1 \times 3 \\ -1 \times 1 + 2 \times 3 \end{pmatrix}$$

$$\vec{y} = \begin{pmatrix} 5 \\ 2 \end{pmatrix}$$



Geometric intuition



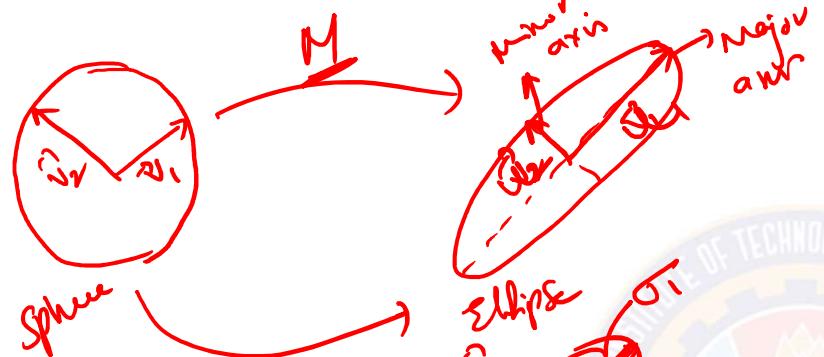
$$M = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$
$$M = \begin{pmatrix} \alpha & 0 \\ 0 & \alpha \end{pmatrix} \quad \alpha \rightarrow \text{stretching}$$
$$\alpha > 1 \quad \alpha < 1$$

$$\vec{y} = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix} \begin{pmatrix} 1 \\ 3 \end{pmatrix} = \begin{pmatrix} \alpha \times 1 + 0 \times 3 \\ 0 \times 1 + \alpha \times 3 \end{pmatrix} = \begin{pmatrix} 2 \\ 6 \end{pmatrix}$$

$$\alpha = 0$$



Matrix vector multiplication in higher dimensions



new coordinate space

$\vec{v}_1, \vec{v}_2 \dots \vec{v}_n \xrightarrow{\text{Matrix multr}} \vec{u}_1, \vec{u}_2 \dots \vec{u}_n$ Principal axis

$\sigma_1, \sigma_2 \dots \sigma_n$ Unit stretch factor

$$M \vec{v}_1 = \sigma_1 \vec{u}_1$$

$$A \vec{x} = \lambda \vec{x}$$

$$M \vec{v}_j = \sigma_j \vec{u}_j \quad j=1 \dots n$$

Matrix vector multiplication in higher dimensions(contd..)

$$[\quad] [v_1 \quad \dots \quad v_n]_{n \times n} = [\tilde{v}_1 \tilde{v}_2 \dots \tilde{v}_n] [\sigma_1 \sigma_2 \dots \sigma_n]$$

$$A V = \hat{U} \sum \hat{V}$$

↓ rotation
 ↓ stretch

$$\begin{aligned}
 V^{-1} &= V^* \\
 \hat{U} &= V^* \\
 A &= \hat{U} \sum V^* \\
 &= \hat{U} \sum V^*
 \end{aligned}$$

$\sigma_1 \geq \sigma_2 \geq \dots \geq \sigma_n \geq 0$

$$A_{m \times n} = U \sum V^T$$

↓
 unitary matrix

$$A^T A = I$$

$$A A^T = \bar{U}$$



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Using SVD

Prof.Aruna Malapati

SVD Theorem

$$\underline{\mathbf{A}}_{[m \times n]} = \mathbf{U}_{[m \times r]} \Sigma_{[r \times r]} (\mathbf{V}_{[n \times r]})^T$$

A: Input data matrix

- $m \times n$ matrix (e.g., m users, n movies)

U: Left singular vectors

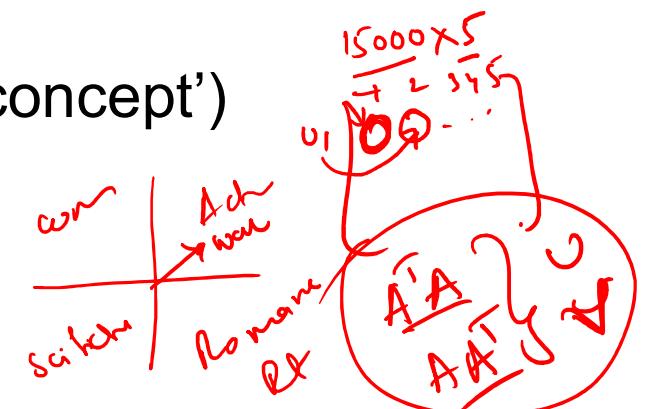
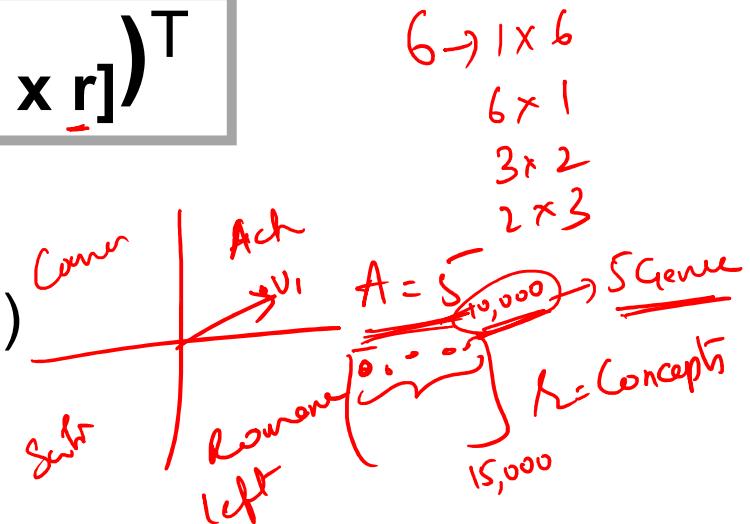
- $m \times r$ matrix (m users, r concepts)

Σ : Singular values

- $r \times r$ diagonal matrix (strength of each ‘concept’)
(r : rank of the matrix \mathbf{A})

V: Right singular vectors

- $n \times r$ matrix (n movies, r concepts)



Singular Value Decomposition

- The key issue in an SVD decomposition is to find a lower dimensional feature space where the new features represent “concepts” and the strength of each concept in the context of the collection is computable.
- The core of the SVD algorithm lies in the following theorem
- It is always possible to decompose a given matrix A into $A = U \Sigma V^T$

Example

$\overset{A}{\rightarrow} \begin{bmatrix} 5 & 1 & 1 & 4 \\ 1 & 4 & 2 & 0 \\ 2 & 1 & 4 & 5 \end{bmatrix} \quad 3 \times 4$

User to Movie Utility Matrix

$= \begin{bmatrix} u_1 & u_2 \end{bmatrix} \begin{bmatrix} c_1 & c_2 & c_3 \end{bmatrix}$

User to Concept 3×3

$\times \begin{bmatrix} 8.87 & 0.00 & 0.00 & 0.00 \\ 0.00 & 4.01 & 0.03 & 0.00 \\ 0.00 & 0.00 & 2.51 & 0.00 \end{bmatrix} \quad 4 \times 4$

Strength of each concept 4×4

$\times \begin{bmatrix} -0.47 & -0.28 & -0.47 & 0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \\ -0.71 & -0.23 & 0.66 & 0.13 \\ -0.52 & 0.39 & -0.53 & 0.55 \end{bmatrix} \quad 4 \times 4$

Movie to Concept

$\|A\|_F = \sqrt{\sum_{ij} A_{ij}^2} = \sqrt{4^2 + 1^2 + 1^2 + 4^2 + 1^2 + 2^2 + 0^2 + 2^2 + 1^2 + 4^2 + 5^2}$

B

$\begin{bmatrix} 2.69 & 0.57 & 2.22 & 4.25 \\ 0.78 & 3.93 & 2.21 & 0.04 \\ 3.17 & 1.38 & 2.92 & 4.78 \end{bmatrix}$

Reconstructed Matrix

$= \begin{bmatrix} -0.61 & 0.28 \\ -0.29 & -0.95 \\ -0.74 & 0.14 \end{bmatrix} \times \begin{bmatrix} 8.87 & 0.00 \\ 0.00 & 4.01 \end{bmatrix} \times \begin{bmatrix} -0.47 & -0.28 & -0.47 & -0.69 \\ 0.11 & -0.85 & -0.27 & 0.45 \end{bmatrix}$

User to Concept 3×2

Strength of each concept 2×2

Movie to Concept

$\rightarrow \begin{bmatrix} -0.23 & -0.81 \end{bmatrix} \quad 3 \times 2$

$\approx \sqrt{\sum_{ij} B_{ij}^2} \quad 3 \times 2$

$\|A-B\|_F = \sqrt{\sum_{ij} (A_{ij}-B_{ij})^2} \quad 3 \times 4$

Recommendations for new User

➤ How to use SVD for recommendations?

$$\mathbf{u}_{new} = \mathbf{u} \times \mathbf{V}_{m \times r} \times \mathbf{S}^{-1}_{r \times r}$$

$$\begin{bmatrix} -0.23 & -0.89 \end{bmatrix} = \begin{bmatrix} 1 & 4 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} -0.47 & 0.11 \\ -0.28 & -0.85 \\ -0.47 & -0.27 \\ -0.69 & 0.45 \end{bmatrix} \times \begin{bmatrix} 0.11 & 0.00 \\ 0.00 & 0.25 \end{bmatrix}$$

U $m \times 4$ V 4×2 S^{-1}

(new user)

$$\begin{bmatrix} 8.87 & 0.00 \\ 0.00 & 4.01 \\ \cancel{0.00} & \cancel{0.00} \end{bmatrix} \Sigma$$

Strength of each concept

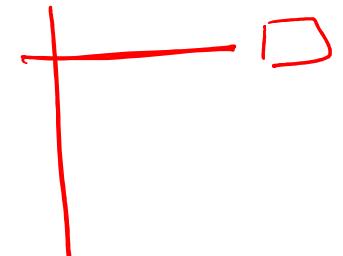
$$\begin{bmatrix} v_1 & -0.47 & -0.28 & -0.47 & -0.69 \\ v_2 & 0.11 & -0.85 & -0.27 & 0.45 \end{bmatrix}$$

Movie to Concept $\sqrt{\cdot}$

$$\begin{aligned} & [1x -0.47 + 4x -0.28 + 1x -0.47 + 0x -0.69] \\ & [2.06 - 3.54] S^{-1} \\ & 1x 2 = [-0.23 -0.89] \end{aligned}$$

Drawbacks of SVD

- Conventional SVD is undefined for incomplete matrices!
- Imputation to fill in missing values
- Increases the amount of data
- We need an approach that can simply ignore missing ratings





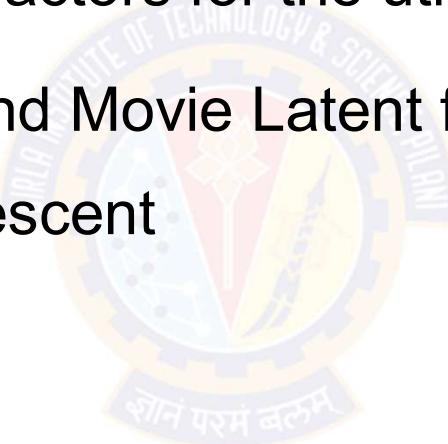
BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Recommender Systems Using Latent Factor Models

Prof.Aruna Malapati

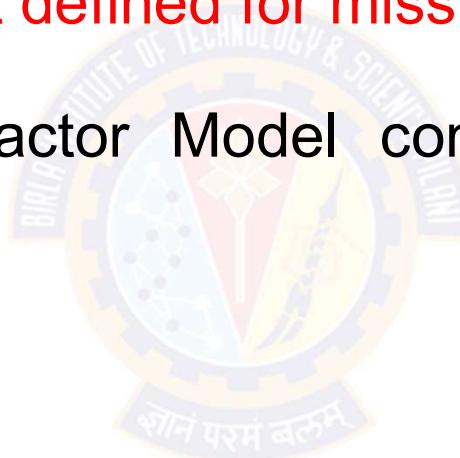
Learning Objectives

- Motivation for Latent Factor Models
- Extracting the Latent Factors for the utility matrix
- Computing the User and Movie Latent factors using Stochastic Gradient Descent



Drawback of SVD

- Though SVD gives us the best rank approximation, the major problem is that it is **not defined for missing entries.**
- Hence the Latent Factor Model comes handy which draws inspiration from SVD.



User – Movie interactions in real world

$U_2 = U_3 + U_4$

dependent

User/ Movie	M1	M2	M3	M4	M5
User1	1	3	2	5	4
User2	2	1	1	1	5
User3	3	2	3	1	5
User4	2	4	1	5	2

User/ Movie	M1	M2	M3	M4	M5
User1	4	4	4	4	4
User2	4	4	4	4	4
User3	4	4	4	4	4
User4	4	4	4	4	4

User/ Movie	M1	M2	M3	M4	M5
User1	1	3	2	5	4
User2	2	1	1	1	5
User3	3	2	3	1	5
User4	2	4	1	5	2

$U_1 = U_2 + U_3 + U_4$ $M_1 = M_2 \cup M_3 \cup M_4$

independent

User/ Movie	M1	M2	M3	M4	M5
User1	3	1	1	3	1
User2	1	2	4	1	3
User3	3	1	1	3	1
User4	4	3	5	4	4

$$U_1 = U_2 + U_3 \\ M_1 = M_2 \cup M_3$$

$$\frac{U_2 + U_3}{M_5} = U_4 \\ \overline{M_5} : Avg(M_2, M_3)$$

Movie / User Features

➤ Movie Features



Comedy



Action



Horror



Drama

➤ User Preferences

- Likeliness of genre of movies (0/1)

Movie / User Features

Movie/User	Comedy	Action
M1	3	1
M2	1	2
M3	1	4
M4	3	1
M5	1	3

f_1 f_2

User/Movie	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

f_1 f_2

$$1 \times 1 + 1 \times 3 + 1 + 3 = 4$$

$$\begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} 1 \\ 4 \end{bmatrix}$$

movie 1 3 1
 3 1 3

$$U_{11} = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \end{bmatrix}$$

$$3 \times 1 + 1 \times 0 = 3$$

$$U_{31} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \end{bmatrix}$$

$$3 \times 1 + 1 \times 0 = 3$$

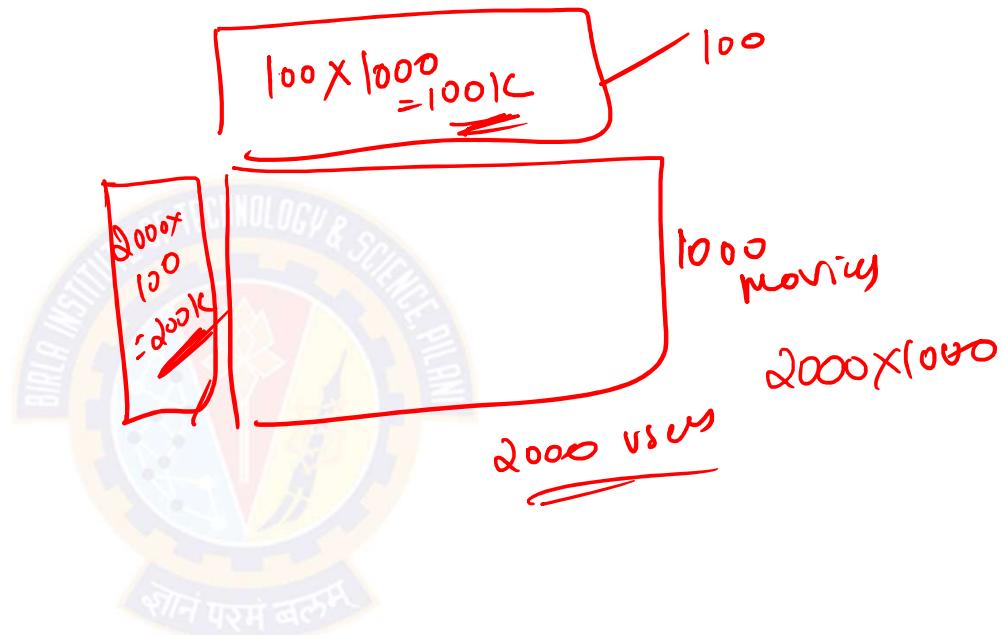
Latent Factors using Matrix Factorization

Movie/ Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

User/ Movie	M1	M2	M3	M4	M5
User1	3	1	1	3	1
User2	1	2	4	1	3
User3	3	1	1	3	1
User4	4	3	5	4	4

Benefit of Matrix Factorization-Storage



Stochastic Gradient Descent (SGD)

The diagram illustrates the relationship between three types of matrices used in machine learning, specifically in the context of Stochastic Gradient Descent (SGD) for recommendation systems.

Left Matrix: User/Features matrix. Rows represent users (User1 to User4) and columns represent features (F1, F2). Red annotations show circled values: 1.2 (under Comedy for User1) and 0.5 (under F2 for User1).

Middle Matrix: Movie/Features matrix. Rows represent movies (Comedy, Action) and columns represent features (M1 to M5). Red annotations show circled values: 1.2 (under M1 for Comedy) and 2.4 (under M1 for Action).

Right Matrix: User/Movie matrix. Rows represent users (User1 to User4) and columns represent movies (M1 to M5). Red annotations show circled values: 3 (under M1 for User1), 1 (under M2 for User1), 1 (under M3 for User2), 2 (under M2 for User2), 4 (under M4 for User3), 1 (under M3 for User3), 3 (under M4 for User4), and 4 (under M5 for User4).

Annotations:

- A red double-headed arrow connects the middle matrix to the right matrix, labeled "coupled".
- A red double-headed arrow connects the left matrix to the middle matrix, labeled "utility matx".
- A red double-headed arrow connects the left matrix to the right matrix.
- Red handwritten text "utility matx" is written next to the right matrix.
- Red handwritten text "coupled" is written next to the middle matrix.
- Red handwritten text "comedy" and "Action" are written next to the circled value 1.2 in the middle matrix.
- Red handwritten text "F1" and "F2" are written next to the circled values in the left matrix.
- Red handwritten text "User1" is written next to the circled value 0.5 in the left matrix.
- Red handwritten text "User1" is written next to the circled value 1.2 in the middle matrix.
- Red handwritten text "User1" is written next to the circled value 3 in the right matrix.
- Red handwritten text "User2" is written next to the circled value 0.5 in the left matrix.
- Red handwritten text "User2" is written next to the circled value 1.32 in the middle matrix.
- Red handwritten text "User2" is written next to the circled value 1 in the right matrix.
- Red handwritten text "User3" is written next to the circled value 0.3 in the left matrix.
- Red handwritten text "User3" is written next to the circled value 2.76 in the middle matrix.
- Red handwritten text "User3" is written next to the circled value 3 in the right matrix.
- Red handwritten text "User4" is written next to the circled value 0.4 in the left matrix.
- Red handwritten text "User4" is written next to the circled value 1.68 in the middle matrix.
- Red handwritten text "User4" is written next to the circled value 4 in the right matrix.

$$\text{Error} = (3 - 1.44)^2 + (1 - 1.37)^2 + \dots$$

derivative

Rating predictions

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

m × k

Movie/Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

k × n

User/Movie	M1	M2	M3	M4	M5
User1	3		1	3	1
User2	1		4	1	
User3	3	1		3	1
User4		3		4	4

U, M_y

$$(1 \cdot 0) \begin{bmatrix} 3 \\ 1 \end{bmatrix} = 3 \cdot 1 + 1 \cdot 0 = 3$$

m × n



Thank You!

In our next session: Mathematical formulation of Latent Factor Model



BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Mathematical formulation of Latent Factor Model

Prof. Aruna Malapati

Learning Objectives

- Formulate the objective function for Latent Factor Model
- Apply Stochastic Gradient Descent to solve the objective function



Objective function

$$R \approx P Q^T = \hat{R}$$

P Q

Movie/ Features	M1	M2	M3	M4	M5
Comedy	3	1	1	3	1
Action	1	2	4	1	3

$$\hat{r}_{vi} = P_v Q_i^T = \sum_{k=1}^K p_{vk} q_{ki}^T$$

User/Features	Comedy	Action
User1	1	0
User2	0	1
User3	1	0
User4	1	1

User/ Movie	M1	M2	M3	M4	M5
User1	3		1		1
User2	1		4	1	
User3	3	1		3	1
User4		3		4	4

Our goal is to find two matrices P and Q such that the following objective function is minimized on test data:

$$e_{ui}^2 = (r_{vi} - \hat{r}_{vi})^2$$

$$= \left(r_{vi} - \sum_{k=1}^K p_{vk} q_{ki}^T \right)^2$$

Stochastic Gradient Descent

$$\frac{\partial e_{ij}^2}{\partial p_{uk}} = -2(\underline{h}_{ui} - \hat{h}_{ui})(q_{ki}) \\ = -2e_{ui}q_{ki}$$

$$\frac{\partial e_{ij}^2}{\partial q_{ki}} = -2(\underline{h}_{ui} - \hat{h}_{ui})(\underline{p}_{uk}) \\ = -2\underline{e}_{ui}\underline{p}_{uk}$$

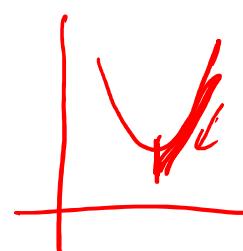
$$p^{new} = p^{old} - \eta \circledcirc p$$

$$q^{new} = q^{old} - \eta \circledcirc q$$

$$e_{ui}^2 = (\underline{h}_{ui} - \hat{h}_{ui})^2$$

$$p^{new} = p^{old} - \eta (-2e_{ui}q_{ki}) \\ = p^{old} + 2\eta e_{ui}q_{ki}$$

$$q^{new} = q^{old} - \eta (-2e_{ui}p_{uk}) \\ = q^{old} + 2\eta e_{ui}p_{uk}$$



Stochastic Gradient Descent (Contd..)

6 param

$$e_{ui} = \left(r_{ui} - \sum_{k=1}^K p_{uk} q_{ki} \right)^2 + \lambda \left(\| p_u \|^2 + \| q_i \|^2 \right)$$

$$p_u = \begin{pmatrix} p_{u1} \\ p_{u2} \\ p_{u3} \end{pmatrix} \quad q_i = \begin{pmatrix} q_{i1} \\ q_{i2} \\ q_{i3} \end{pmatrix}$$

$$p_{uk}^{new} = p_{uk}^{old} + \eta \frac{\partial e_{ui}}{\partial p_{uk}} = p_{uk} + \eta (2e_{ui} q_{ki} - \lambda p_{uk})$$

$$= q_{ki} + \eta (2e_{ui} p_{ui} - \lambda q_{ki})$$

2006-2009

$$\hat{r}_{ui} = \mu + b_u + b_i + \sum_{k=1}^K p_{uk} q_{ki}$$

Other Parameters

$$r_{ij} = \mu + b_v(t) + b_i(t) + \sum_{k=1}^K p_{vk} q_{kj}$$

~~0.5
520~~





BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Metrics used for evaluating Recommender Systems

Prof.Aruna Malapati

Accuracy of estimated rating / Error Based

- Mean Absolute Error (MAE)
- Mean square error (MSE)
- Root Mean Squared Error(RMSE)

User/Movie	Movie1	Movie2	Movie3	Movie4	Movie5	Movie6
User1✓	2			4	4✓	
User2	5		4			1
User3			5		2	
User4		1		5		4
User5			4			2
User6	4	5	1			

Training Set Test Set

Accuracy of estimated rating / Error Based (Contd..)

User Id	Movie Id	Actual	Predicted	MAE	MSE	RMSE ✓
User1 ✓	4 ✓	4 ✓	2	2	4	4
User1	5	4	1	3	9	9
User2	6	1	1.5	0.5	0.25	0.25
User3	5	2	2	0	0	0
User4	4	5	4.5	0.5	0.25	0.25
User4	6	4	3	1	1	1
User5	6	2	2	0	0	0
User6	4	1	3	2	4	4
				9/8	18.5/8	Sqrt(18.5/8)

test set

$$MAE = \frac{\sum |P - R|}{\# \text{ ratings}}$$

$$MSE = \frac{\sum (P - R)^2}{\# \text{ ratings}}$$

$$RMSE = \sqrt{\frac{\sum (P - R)^2}{\# \text{ ratings}}}$$

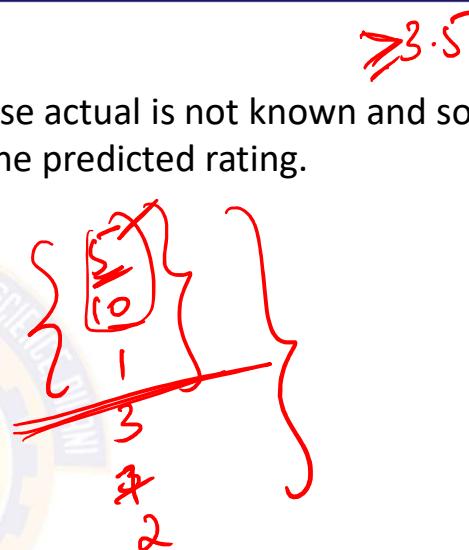
\bar{Z}

Precision at rank K

Movie	User 1	Actual	Predicted
1	1	4	2.3
2	1	2	3.6
3	1	3	3.4
4	1	?	4.4
5	1	5	4.5
6	1	?	2.3
7	1	2	4.9
8	1	?	4.3
9	1	?	3.3
10	1	4	4.3

Ignore the values whose actual is not known and sort the items in Descending order of the predicted rating.

~~note~~ ↓
 Item7 2/4.9
 item5 5/4.5
 item10 4/4.3
 item2 2/3.6
 item3 3/3.4
 item1 4/2.3



Let's assume that all rating ≥ 3.5 are relevant then the recommender system will suggest the following

Item7 2/4.9
 item5 5/4.5
 item10 4/4.3

compute the precision-at-Rank 3
 Recall 7/3 5/3 10/3
 Prec@ 1/1 1/2 2/3





BITS Pilani
Hyderabad Campus

C5: Text Mining

Dr. Chetana Gavankar, Ph.D,
IIT Bombay-Monash University Australia
Chetana.gavankar@pilani.bits-pilani.ac.in



BITS Pilani
Pilani Campus



Session 3
Date – 9th June 2024
Time – 10 am to 12.15 pm

These slides are prepared by the instructor, with grateful acknowledgement of Prof. Luis G. Serrano,
Prof. Andrew Ng and many others who made their course materials freely available online.



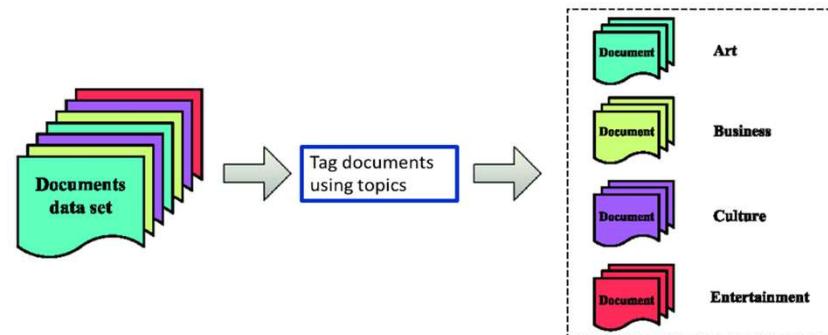
Session Content

- Objective of Latent Dirichlet Allocation (LDA)
- Intuition behind LDA
- LDA Generative model
- Mathematical foundations for LDA : Bernoulli Trial, Binomial distribution, Multinomial distribution
- Mathematical foundations for LDA : Beta distribution, Conjugate Prior and Dirichlet distributions
- Dirichlet Visualization using Simplex
- Probabilistic Graphical LDA Model
- Mathematical modelling of LDA
- Gibbs Sampling Algorithm
- Case Study
- Implementing LDA in Python



Objectives of Topic Modelling

- Use these annotations to organize, summarize and search the documents.
- Topic Model can be defined as an unsupervised technique to discover topics across various text documents





Sample output from the LDA

- Four topics learned from the S&P 500 stock market data
- Goal is to find groups of stocks that tend to move together.

Topic 1	Topic 2	Topic 3	Topic 4
Southwestern Energy Range Resources Cabot Oil & Gas EOG Resources Chesapeake Energy Pioneer Resources Devon Energy Peabody Energy Anadarko Petroleum Massey Energy	Penneys Macys Kohls Nordstrom Target Limited Lowes Home Depot American Express Abercrombie	Capital One BNY Mellon Discover Northern Trust Janus JPMorgan Chase State Street Wells Fargo PPL T. Rowe Price	Simon Property Kimco Realty Equity Residential AvalonBay Communities Apartment Investment Vornado Realty Trust Boston Properties Public Storage Host Hotels HCP Inc.

- The topic model does not provide any label to these group of words.



Sample output from the LDA

Seeking Life's Bare (Genetic) Necessities

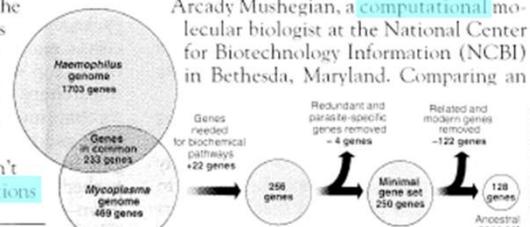
COLD SPRING HARBOR, NEW YORK—How many genes does an organism need to survive? Last week at the genome meeting here,* two genome researchers with radically different approaches presented complementary views of the basic genes needed for life. One research team, using computer analyses to compare known genomes, concluded that today's organisms can be sustained with just 250 genes, and that the earliest life forms required a mere 128 genes. The other researcher mapped genes in a simple parasite and estimated that for this organism, 800 genes are plenty to do the job—but that anything short of 100 wouldn't be enough.

Although the numbers don't match precisely, those predictions

* Genome Mapping and Sequencing, Cold Spring Harbor, New York, May 8 to 12.

"are not all that far apart," especially in comparison to the 75,000 genes in the human genome, notes Siv Andersson of Uppsala University in Sweden, who arrived at the 800 number. But coming up with a consensus answer may be more than just a genetic numbers game, particularly as more and more genomes are completely mapped and sequenced. "It may be a way of organizing any newly sequenced genome," explains

Arcady Mushegian, a computational molecular biologist at the National Center for Biotechnology Information (NCBI) in Bethesda, Maryland. Comparing an



Stripping down. Computer analysis yields an estimate of the minimum modern and ancient genomes.

SCIENCE • VOL. 272 • 24 MAY 1996

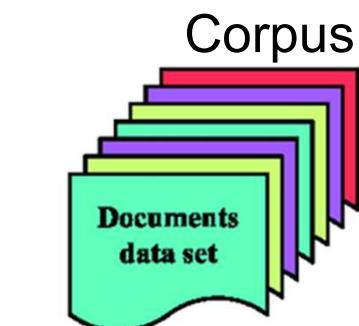
Genetics

Evolutionary biology

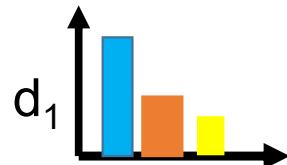
Data Analysis



Overall schematic

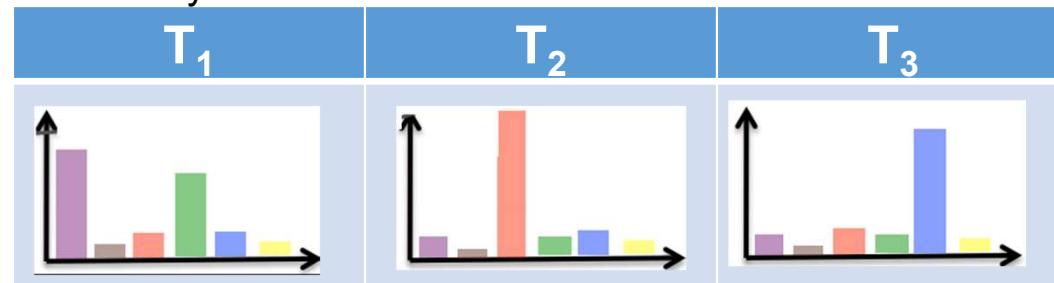


Documents($d_1 \dots d_n$)



Each document has a distribution over K topics

Each topic is defined as a Multinomial distribution over the vocabulary

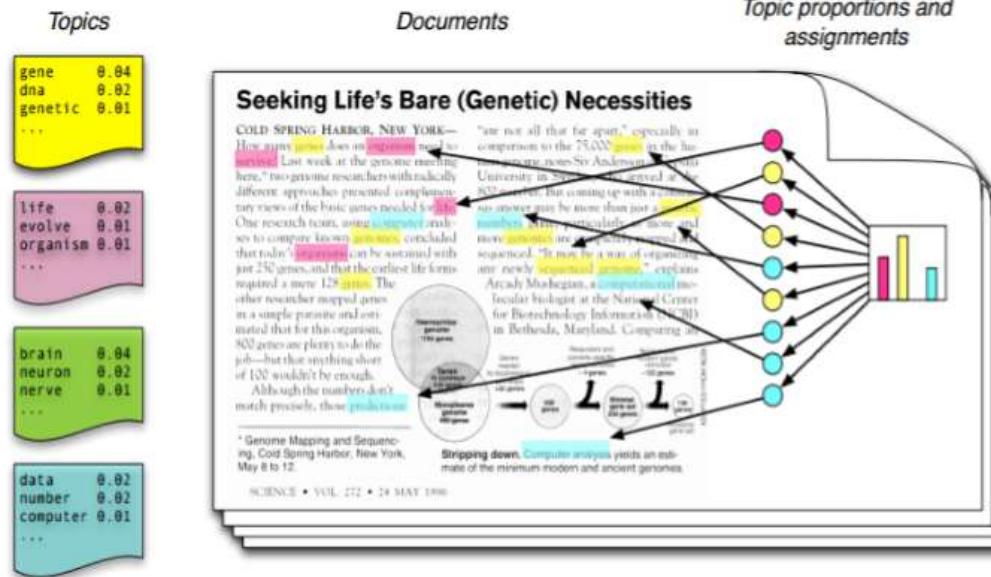


K-Topics (Hyper Parameter)



Vocabulary($W_1 \dots W_m$)

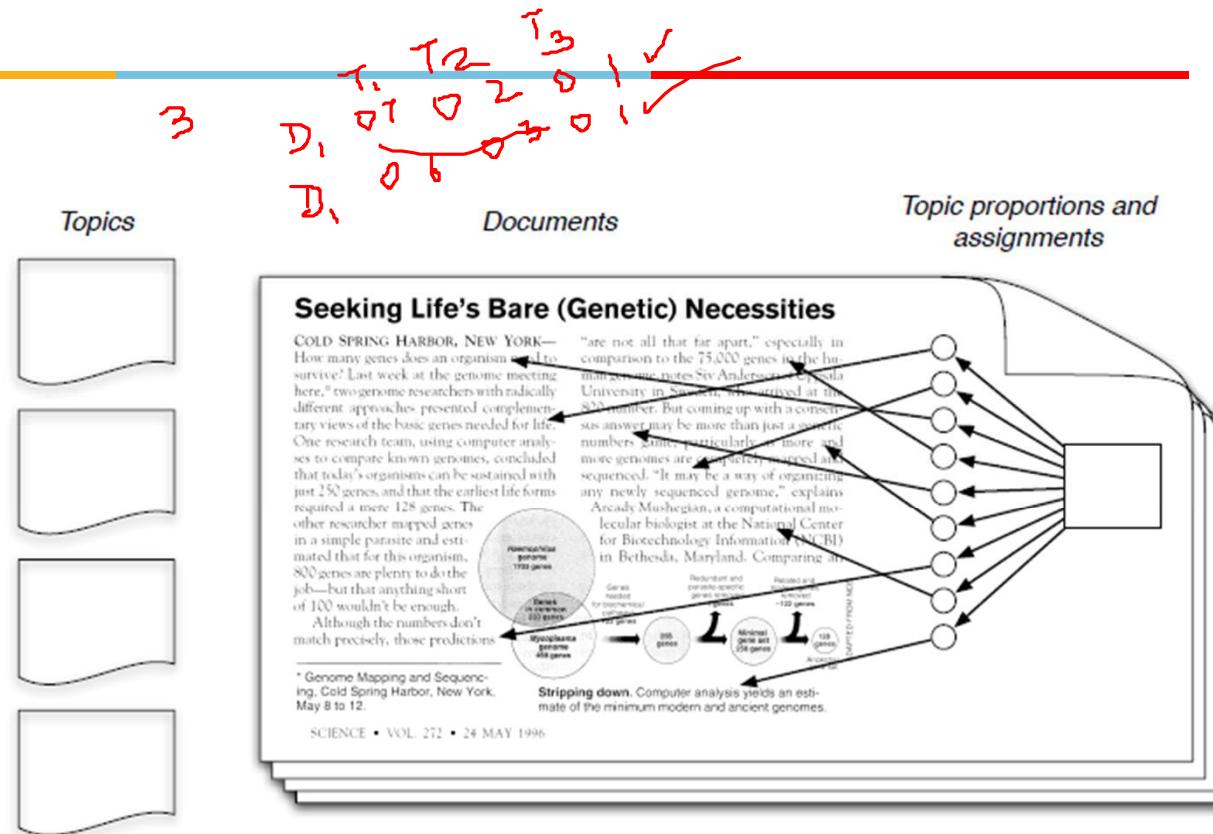
LDA Generative model



- Each **topic** is a distribution over words
- Each **document** is a mixture of corpus-wide topics
- Each **word** is drawn from one of these topics
- We only observe the words within the documents and the other structure are **hidden variables**.



The posterior distribution



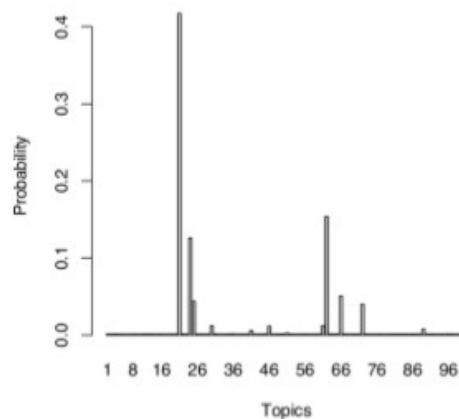


LDA model

A 100-topic LDA model was fitted to **17,000 articles from the *Science* journal**.

At right are **the top 15 most frequent words** from the most frequent topics.

At left are the **inferred topic proportions** for the example article from previous slide.



“Genetics”	“Evolution”	“Disease”	“Computers”
human	evolution	disease	computer
genome	evolutionary	host	models
dna	species	bacteria	information
genetic	organisms	diseases	data
genes	life	resistance	computers
sequence	origin	bacterial	system
gene	biology	new	network
molecular	groups	strains	systems
sequencing	phylogenetic	control	model
map	living	infectious	parallel
information	diversity	malaria	methods
genetics	group	parasite	networks
mapping	new	parasites	software
project	two	united	new
sequences	common	tuberculosis	simulations



LDA Generative Process

LDA assumes that new documents are created in the following way:

1. Determine number of words in document
2. Choose a topic mixture for the document over a fixed set of topics (i.e. 20% topic A, 30% topic B, 50% topic C)
3. Generate the words in the document by:
 - First pick a topic based on the document's multinomial distribution above.
 - Next pick a word based on the topic's multinomial distribution.



LDA Generative Process

-Say we have a group of articles and we assume that all of those articles can be characterized by three topics: Animals, Cooking, and Politics.

-Each of those topics can be described by the following words:

- Animals: dog, chicken, cat, nature, zoo
- Cooking: oven, food, restaurant, plates, taste, delicious
- Politics: Republican, Democrat, Congress, ineffective, divisive

-Say we want to generate a new document that is 80% about animals and 20% about cooking.

1. We choose the length of the article (say, 1000 words)
2. We choose a topic based on our specified mixture (so, out of our 1000 words, roughly 800 will come from the topic "animals")
3. We choose a word based on the word distribution for each topic (i.e.



Intuition behind LDA

- Suppose you have a corpus of documents
- You want LDA to learn the topic representation of K topics in each document and the word distribution of each topic.
- LDA backtracks from the document level to identify topics that are likely to have generated the corpus.



Intuition behind LDA

- Randomly assign each word in each document to one of the K topics.
- For each document d :
 - Assume that all topic assignments except for the current one are correct.
 - Calculate two proportions:
 1. Proportion of words in document d that are currently assigned to topic t = $p(\text{topic } t \mid \text{document } d)$
 2. Proportion of assignments to topic t over all documents that come from this word w = $p(\text{word } w \mid \text{topic } t)$
 - Multiply those two proportions and assign w a new topic based on that probability. $p(\text{topic } t \mid \text{document } d) * p(\text{word } w \mid \text{topic } t)$
 - Eventually we'll reach a steady state where assignments make sense



Bernoulli Trial

- Any **single trial** with **two possible outcomes** can be modeled as a **Bernoulli trial**: team wins/loses, pitch is a strike/ball, coin comes up heads or tails, etc.
- A Bernoulli trial uses Bernoulli distribution to calculate the probability of either outcome.



Bernoulli trial



Bernoulli: A Special Case of the Binomial Distribution

Binomial Trial: Chance of getting n heads in a row($n=3$)



Bernoulli Trial: Chance of getting a heads on a single flip



Bernoulli - Distribution Notation

- The probability mass function of the Bernoulli distribution is

$$f(x) = P(X=k) = \theta^k (1-\theta)^{1-k}, \quad k=\{0,1\}$$

- The only parameter of the bernoulli distribution is θ , which defines the probability of success during a bernoulli trial.



Binomial distribution

$$P(X=0) = \theta^0 (1-\theta)^{1-0} = 1-\theta$$

$$P(X=1) = \theta^1 (1-\theta)^{1-1} = \underline{\theta}$$

$$P(x_1=1, x_2=1, x_3=0) = \theta \times \theta (1-\theta) = \theta^2 (1-\theta)$$

$$P\left(\sum_{i=1}^{n=3} x_i = 2\right) = P(1,1,0) + P(1,0,1) + P(0,1,1)$$

$$= \theta^2 (1-\theta) + \theta^2 (1-\theta) + \theta^2 (1-\theta)$$

$$= 3\theta^2 (1-\theta)$$

$$P\left(\sum_{i=1}^n x_i = k\right) = \binom{n}{k} \theta^k (1-\theta)^{n-k}$$

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$



Binomial distribution

If p is the probability of heads, the probability of getting exactly k heads in n independent yes/no trials is given by the binomial distribution $\text{Bin}(n,p)$:

$$\begin{aligned} P(k \text{ heads}) &= \binom{n}{k} p^k (1-p)^{n-k} \\ &= \frac{n!}{k!(n-k)!} p^k (1-p)^{n-k} \end{aligned}$$



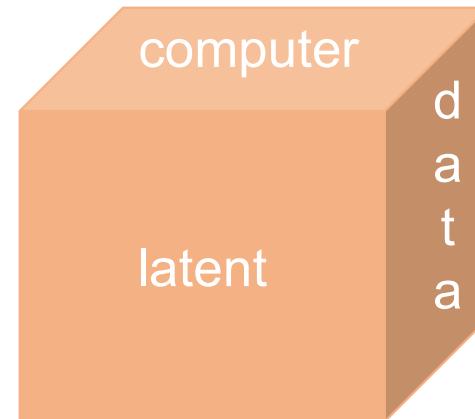
Multinomial Distribution

- Suppose we roll our die of words having k sides(vocabulary) where each side takes probabilities $\Theta_1, \dots, \Theta_k$ respectively.
- Probability mass function

$$f(x) = \frac{n!}{x_1!x_2!\cdots x_k!} \theta_1^{x_1}\theta_2^{x_2}\cdots\theta_k^{x_k}$$

k - number of sides on the die

n - number of times the die will be rolled





Multinomial Distribution

Suppose that we observe an experiment that has k possible outcomes $\{O_1, O_2, \dots, O_k\}$ independently n times.

Let p_1, p_2, \dots, p_k denote probabilities of O_1, O_2, \dots, O_k respectively.

Let X_i denote the number of times that outcome O_i occurs in the n repetitions of the experiment.

Then the joint probability function of the random variables X_1, X_2, \dots, X_k is

$$p(x_1, \dots, x_n) = \frac{n!}{x_1! x_2! \dots x_k!} p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$$



Multinomial Distribution

Note: $p_1^{x_1} p_2^{x_2} \dots p_k^{x_k}$

is the probability of a sequence of length n containing

x_1 outcomes O_1

x_2 outcomes O_2

...

x_k outcomes O_k

$$\frac{n!}{x_1! x_2! \dots x_k!} = \binom{n}{x_1 \ x_2 \ \dots \ x_k}$$

is the number of ways of choosing the positions for the x_1 outcomes O_1 , x_2 outcomes O_2 , ..., x_k outcomes O_k



Beta Distribution

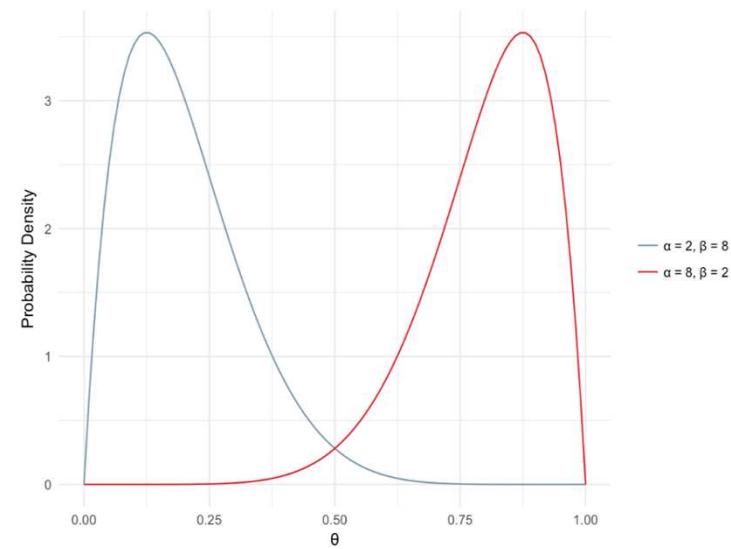
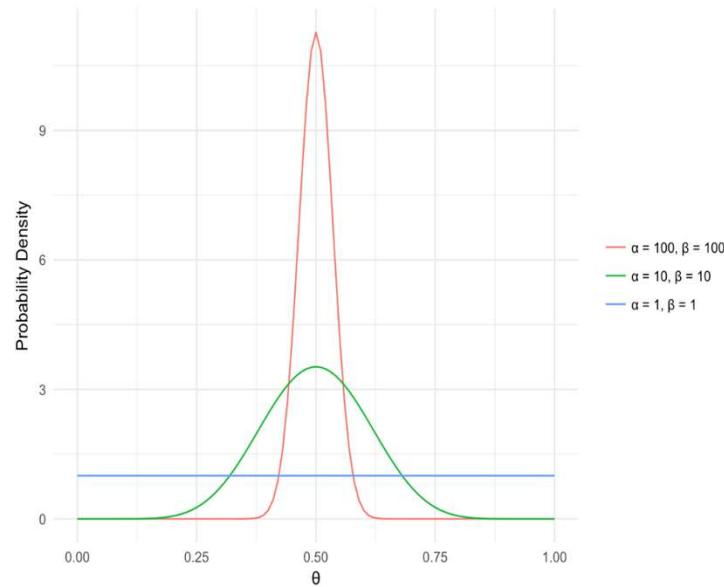
- The probability distribution function for the beta distribution

$$f(\theta; \alpha, \beta) = \frac{\theta^{(\alpha-1)}(1-\theta)^{(\beta-1)}}{B(\alpha, \beta)}$$



Beta Distribution

- The beta distribution can be thought of as a **probability distribution of probabilities**.



The Building Blocks of inferring the parameters



➤ Parameter estimation

$$\underbrace{p(\theta|D)}_{posterior} = \frac{\overbrace{p(D|\theta) p(\theta)}^{likelihood\ prior}}{\underbrace{p(D)}_{evidence}}$$



Dirichlet distributions

- Dirichlet distributions are probability distributions over multinomial parameter vectors
- They are called Beta distributions when k = 2
- The Dirichlet probability density function is defined as

$$Dir(\vec{\theta} | \vec{\alpha}) = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)} \prod_{i=1}^K \theta_i^{\alpha_i - 1}$$

➤ $Dir(\vec{\theta} | \vec{\alpha}) = \frac{1}{B(\alpha)} \prod_{i=1}^K \theta_i^{\alpha_i - 1}$ where $\frac{1}{B(\alpha)} = \frac{\Gamma(\sum_{i=1}^K \alpha_i)}{\prod_{i=1}^K \Gamma(\alpha_i)}$



Conjugate Prior

In Bayesian probability theory,

- if the posterior distributions $p(\vartheta | x)$ are in the same probability distribution family as the prior probability distribution $p(\theta)$, the prior and posterior are then called conjugate distributions, and the prior is called a conjugate prior for the likelihood function

$$p(\theta|D) = \frac{\underbrace{p(D|\theta)}_{\text{posterior}} \underbrace{p(\theta)}_{\text{likelihood prior}}}{\underbrace{p(D)}_{\text{evidence}}}$$

- conjugate prior gives a closed-form expression for the posterior; otherwise numerical integration may be necessary.
- Also may give intuition, by more transparently showing how a likelihood function updates a prior distribution

$$\begin{aligned} P(D|\theta) &\sim \text{Normal} \quad \text{and} \quad P(\theta) \sim \text{Normal} \\ \Rightarrow P(\theta|D) &\sim \text{Normal} \end{aligned}$$

$$\begin{aligned} P(D|\theta) &\sim \text{Normal} \quad \text{and} \quad P(\theta) \sim \text{Gamma} \\ \Rightarrow P(\theta|D) &\sim \text{Normal/gamma} \end{aligned}$$

$$\begin{aligned} P(D|\theta) &\sim \text{Bernoulli} \quad \theta^k (1-\theta)^{N-k} \\ \text{Prior} &\sim \theta^{a-1} (1-\theta)^{b-1} \end{aligned}$$

$$\text{Posterior} \sim \text{Beta}$$



Beta distribution as Conjugate Prior for Binomial distribution

Given a **prior** $P(\theta | \alpha, \beta) = \text{Beta}(\alpha, \beta)$, and **data** $D=(H, T)$, what is our posterior?

$$\begin{aligned} P(\theta | \alpha, \beta, H, T) &\propto P(H, T | \theta) P(\theta | \alpha, \beta) \\ &\propto \theta^H (1 - \theta)^T \theta^{\alpha-1} (1 - \theta)^{\beta-1} \\ &= \theta^{H+\alpha-1} (1 - \theta)^{T+\beta-1} \end{aligned}$$

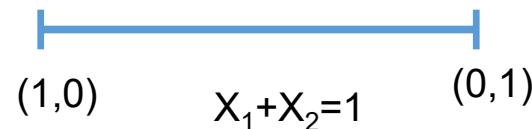
With normalization

$$\begin{aligned} P(\theta | \alpha, \beta, H, T) &= \frac{\Gamma(H + \alpha + T + \beta)}{\Gamma(H + \alpha)\Gamma(T + \beta)} \theta^{H+\alpha-1} (1 - \theta)^{T+\beta-1} \\ &= \text{Beta}(\alpha + H, \beta + T) \end{aligned}$$



Visualization of the simplex

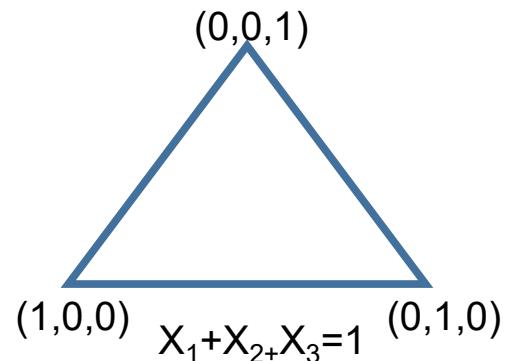
- This is often referred as **simplex** and a most convenient way to visualize this is using a certain shapes depending upon the number of topics.
- Suppose K=2 topics which can be modeled as 1-simplex and can be visualized using a line.





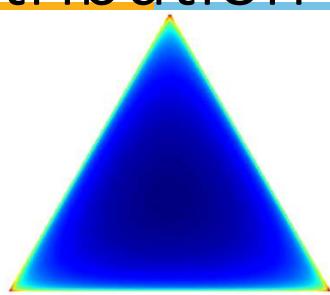
Visualization of the simplex

- Suppose K=3 topics which can be modeled as 2-simplex and can be visualized using a triangle.

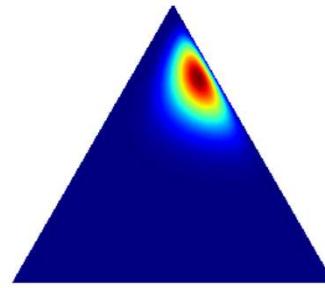


- If we have **K topics** this can be generated using **K-1 simplex**.

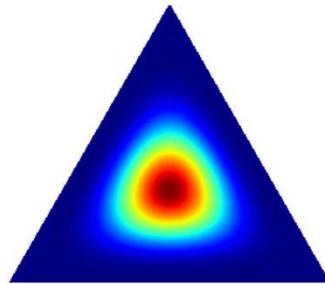
Shape of the Dirichlet distribution



Dirichlet(0.999, 0.999, 0.999)



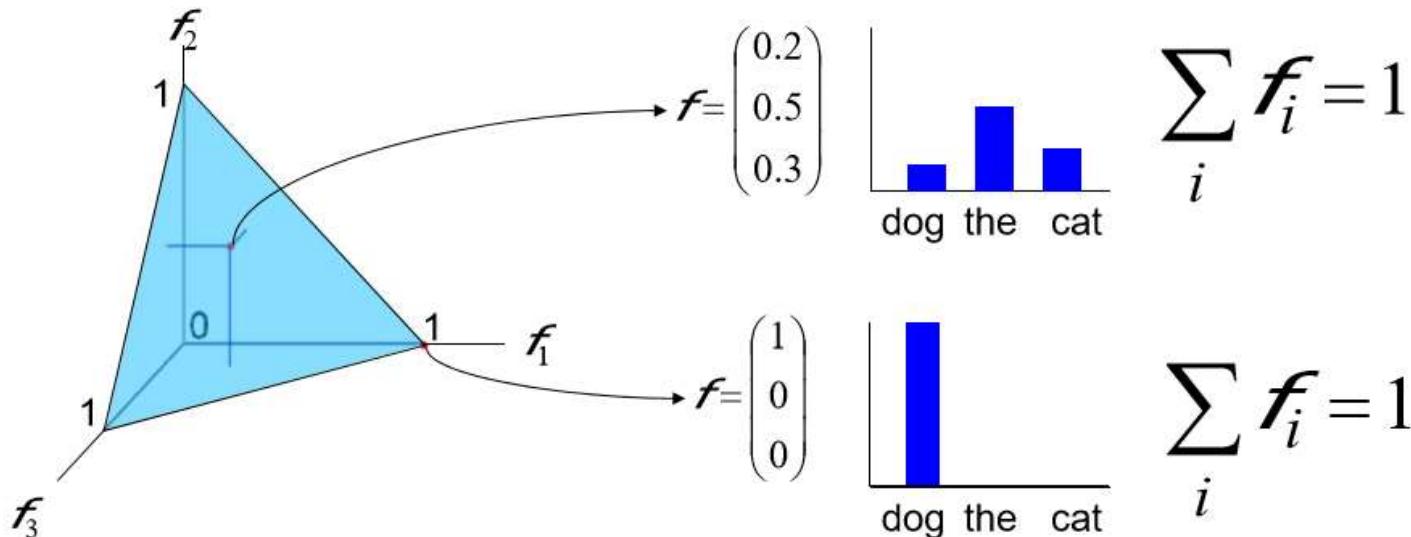
Dirichlet(2, 5, 15)



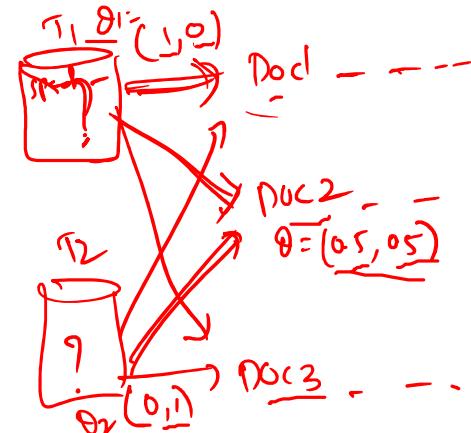
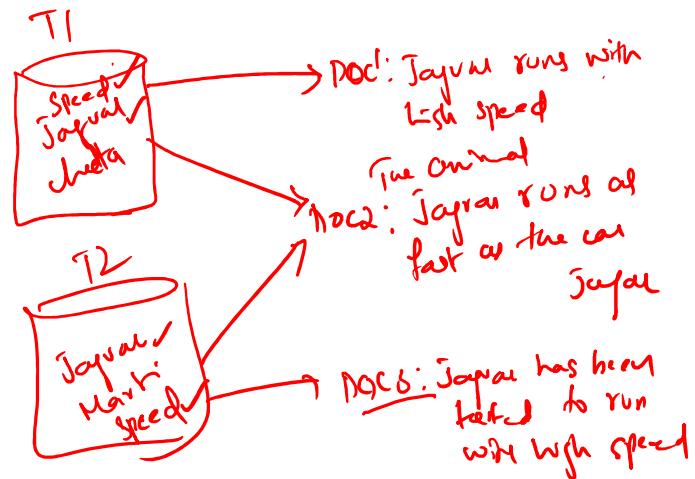
Dirichlet(5, 5, 5)

Dirichlet distributions

Each point on a k dimensional simplex is a multinomial probability distribution:



Statistical Inference





Topic Modelling Example - LDA



Sports



Politics



Science



Science

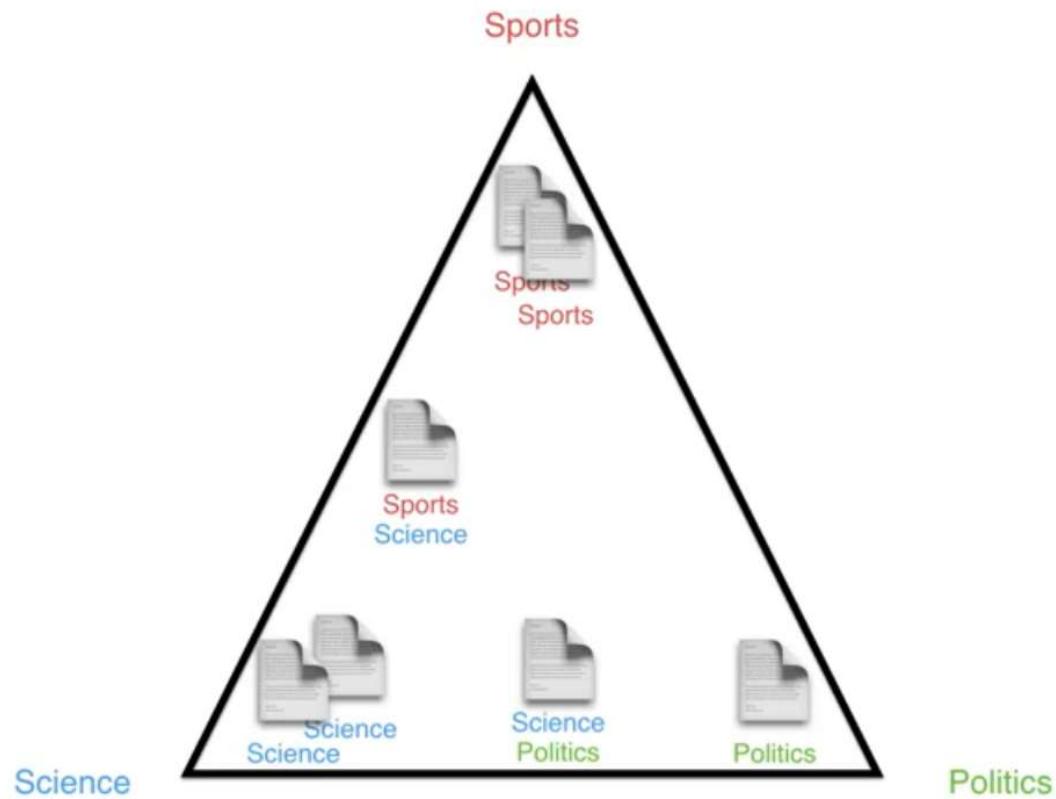
Science

Politics

Sports

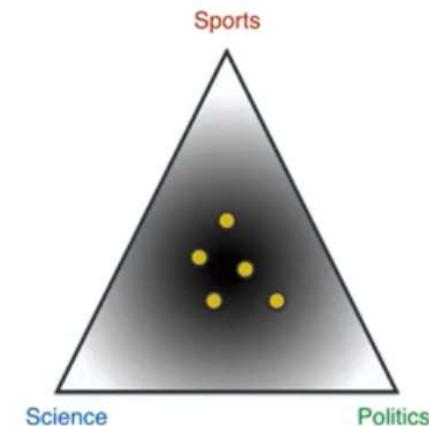
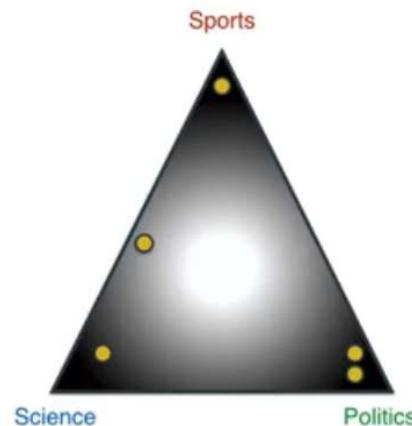
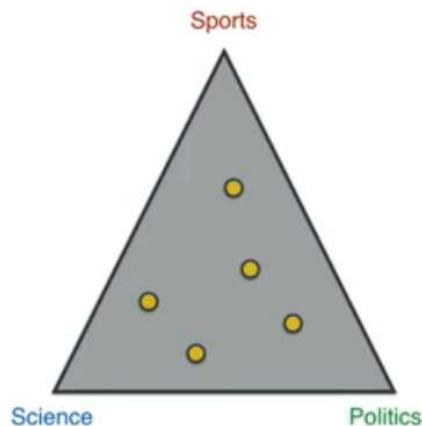


LDA



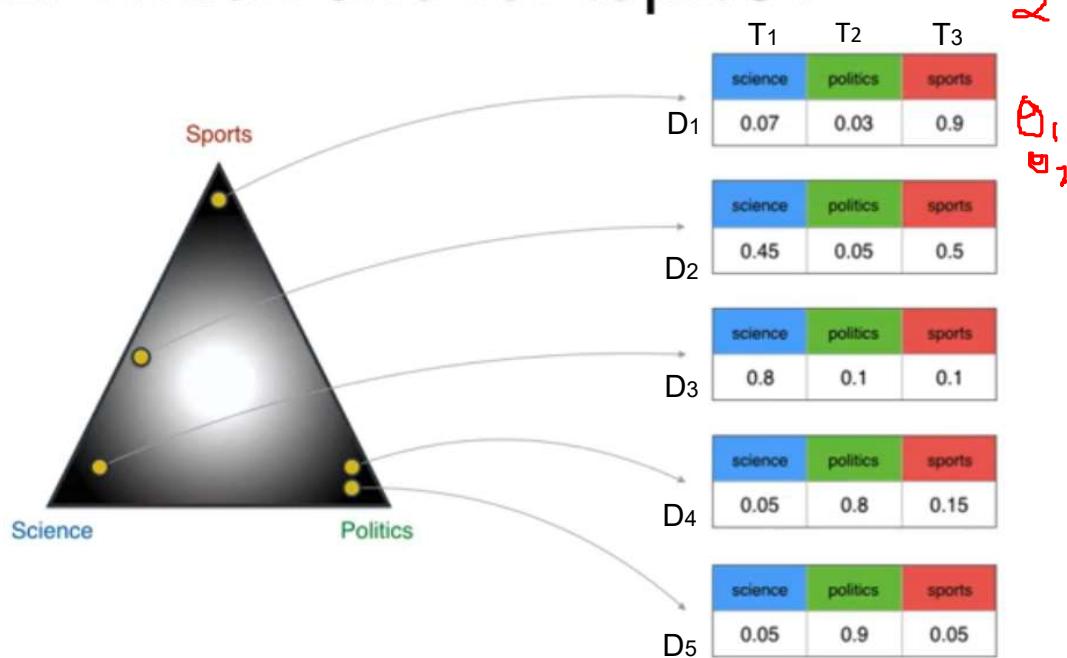


Quiz: Which one for topics?



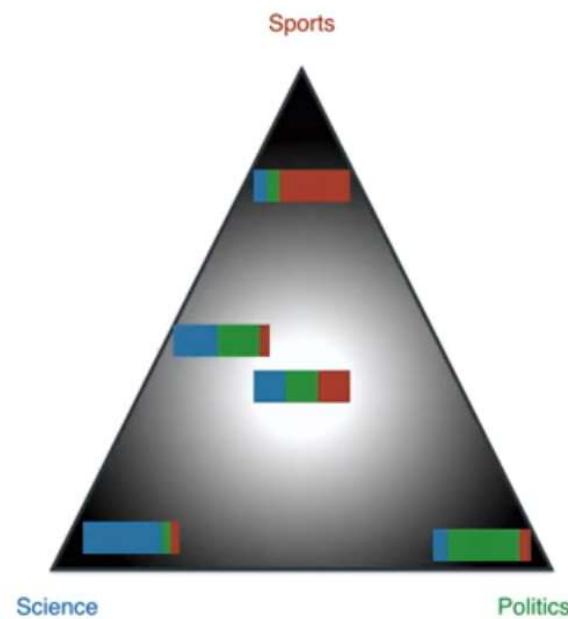


Quiz: Which one for topics?



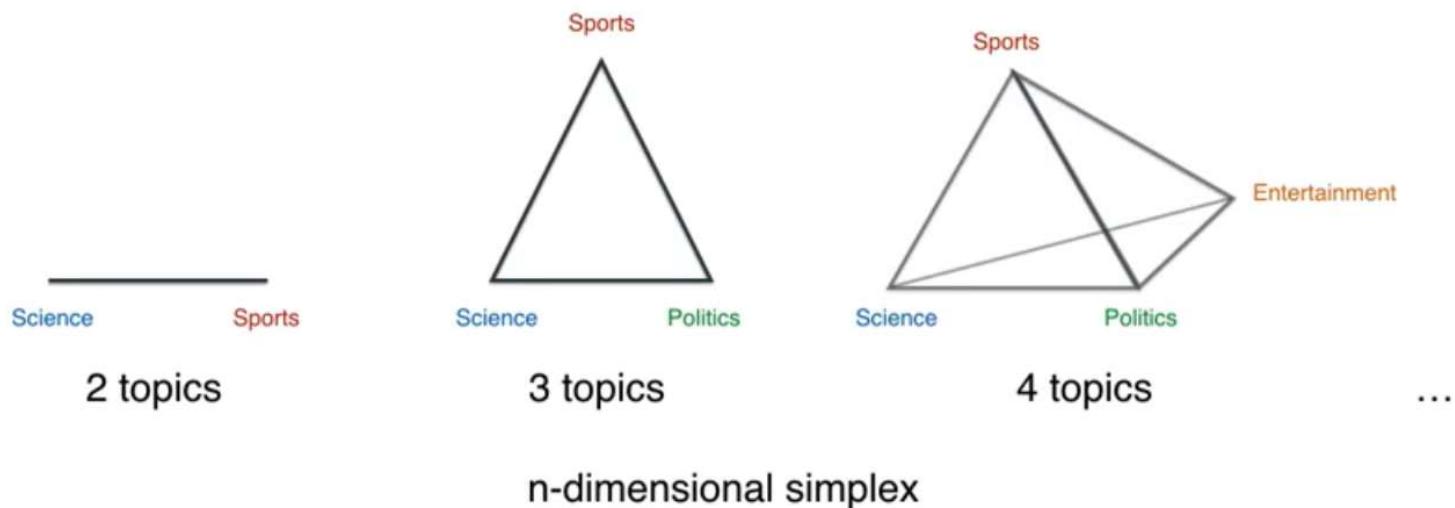


A distribution of distributions



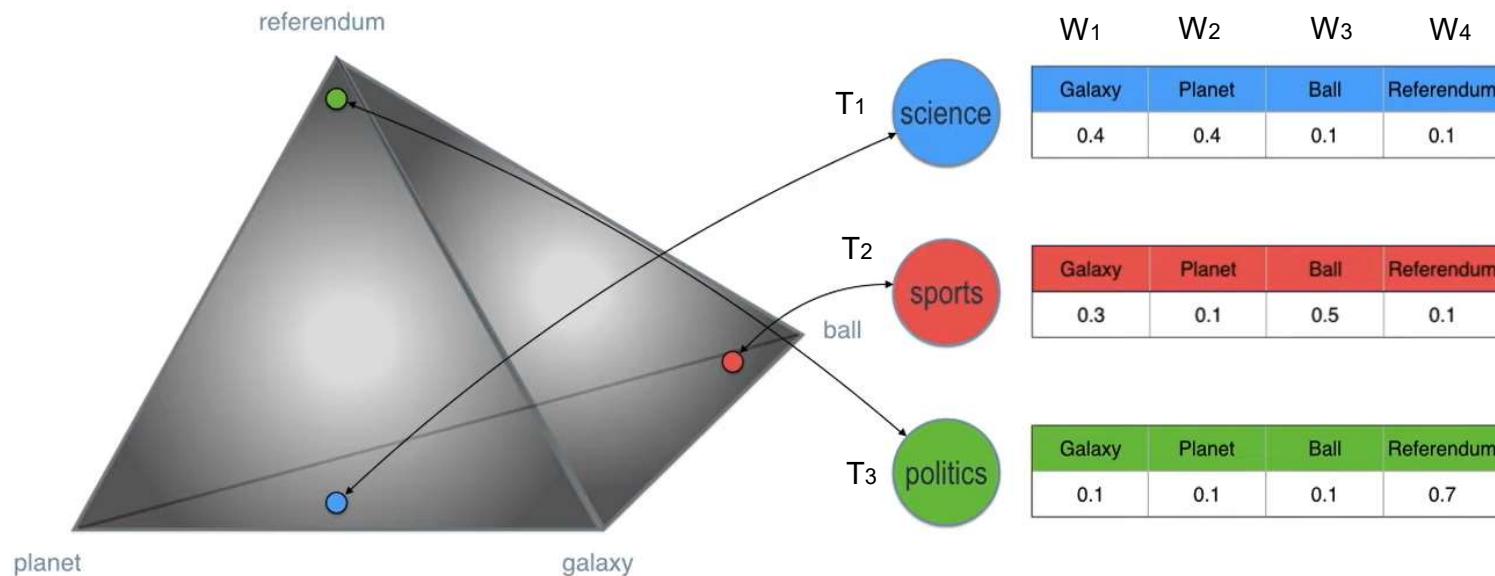


More topics? More dimensions



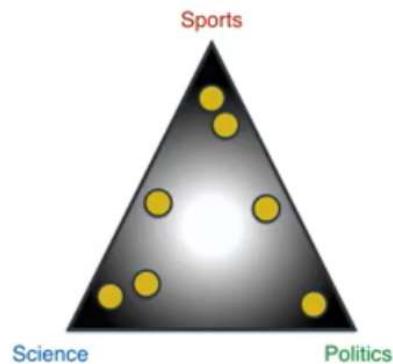


Word distributions

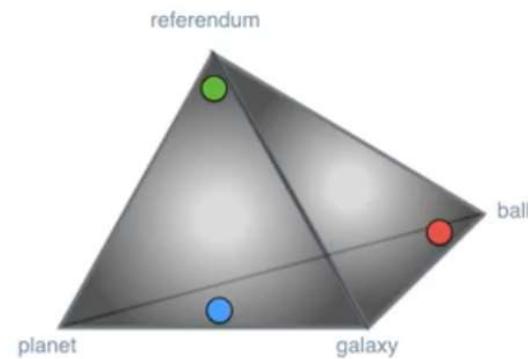




Two Dirichlet distributions



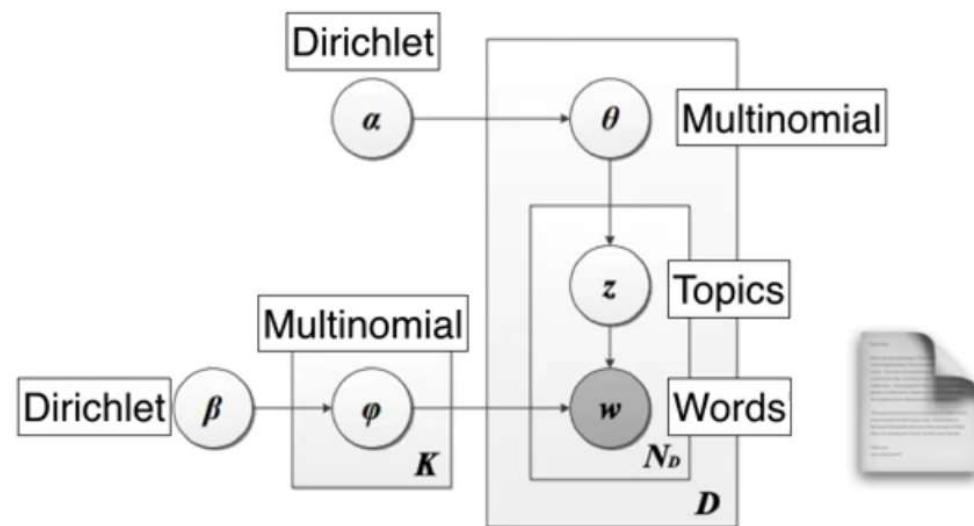
Documents-Topics



Topics-Words



Blueprint for the LDA machine

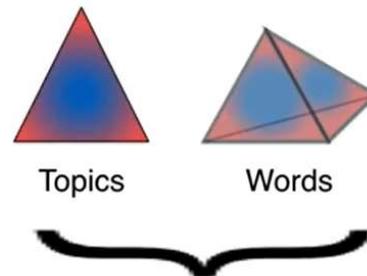




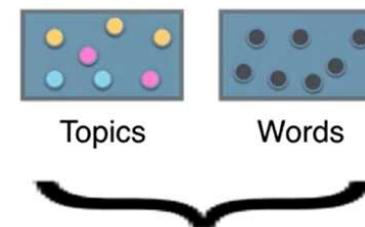
Mathematical modelling of LDA

Probability of a document

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\varphi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \varphi_{Z_{j,t}})$$



Dirichlet
Distributions

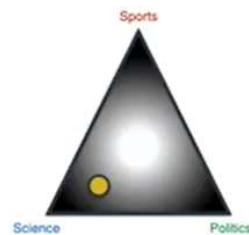


Multinomial
Distributions

Mathematical modelling of LDA

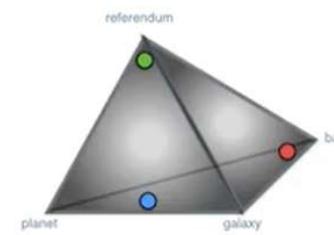


$$\prod_{j=1}^M P(\theta_j; \alpha)$$



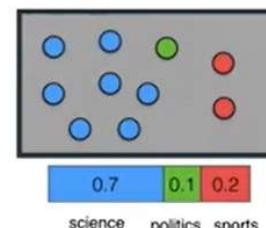
science	politics	sports
0.7	0.1	0.2

$$\prod_{i=1}^K P(\varphi_i; \beta)$$

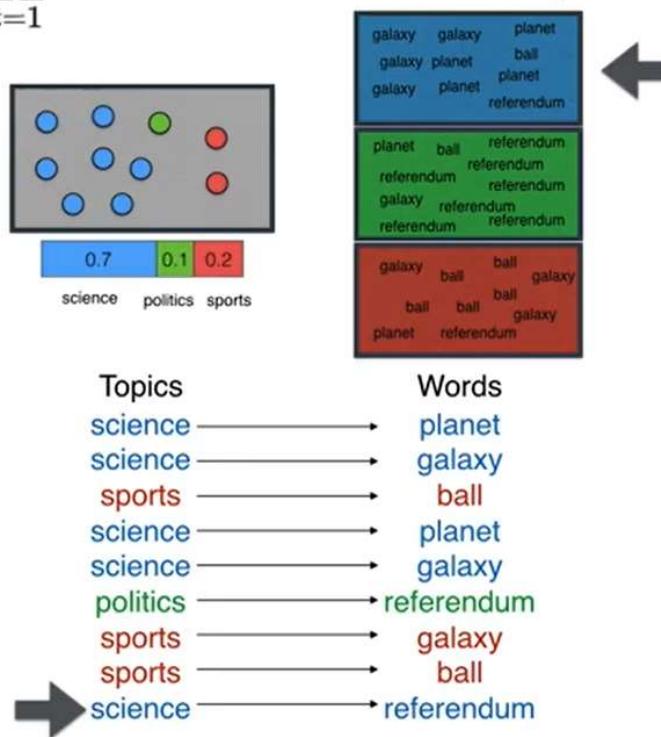


Galaxy	Planet	Ball	Referendum
0.4	0.4	0.1	0.1
Galaxy	Planet	Ball	Referendum
0.1	0.1	0.1	0.7
Galaxy	Planet	Ball	Referendum
0.3	0.1	0.5	0.1

$$\prod_{t=1}^N P(Z_{j,t} \mid \theta_j)$$



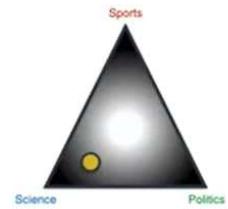
$$P(W_{j,t} \mid \varphi_{Z_{j,t}})$$





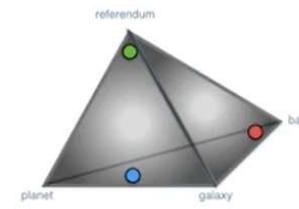
Mathematical modelling of LDA

$$\prod_{j=1}^M P(\theta_j; \alpha)$$



science	politics	sports
0.7	0.1	0.2

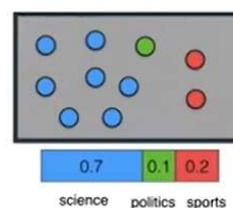
$$\prod_{i=1}^K P(\varphi_i; \beta)$$



Galaxy	Planet	Ball	Referendum
0.4	0.4	0.1	0.1
Galaxy	Planet	Ball	Referendum
0.1	0.1	0.1	0.7

Galaxy	Planet	Ball	Referendum
0.3	0.1	0.5	0.1

$$\prod_{t=1}^N P(Z_{j,t} | \theta_j)$$



Topics
 science
 science
 sports
 science
 science
 politics
 sports
 sports
 science

$$P(W_{j,t} | \varphi_{Z_{j,t}})$$

galaxy	galaxy	planet
galaxy	planet	ball
galaxy	planet	planet
		referendum
planet	ball	referendum
referendum	referendum	referendum
galaxy	referendum	referendum
referendum	referendum	referendum
galaxy	ball	ball
planet	ball	galaxy
galaxy	ball	ball
planet	referendum	galaxy



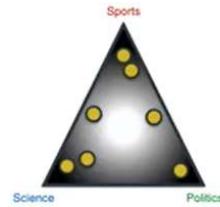
planet	galaxy	ball
galaxy	planet	referendum
planet	galaxy	referendum
referendum	galaxy	ball
referendum	ball	referendum



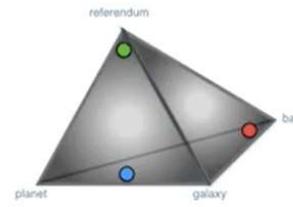


Mathematical modelling of LDA

$$\prod_{j=1}^M P(\theta_j; \alpha)$$

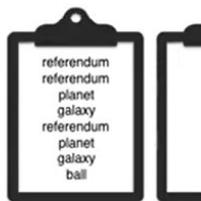
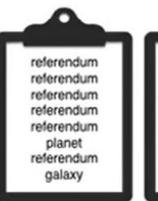


$$\prod_{i=1}^K P(\varphi_i; \beta)$$



$$\prod_{t=1}^N P(Z_{j,t} | \theta_j) \quad P(W_{j,t} | \varphi_{Z_{j,t}})$$

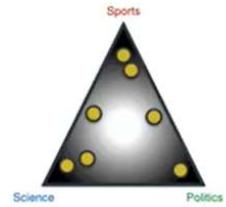
$P(\text{same articles}) = \text{low}$



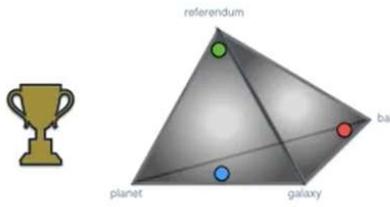


Mathematical modelling of LDA

$$\prod_{j=1}^M P(\theta_j; \alpha)$$

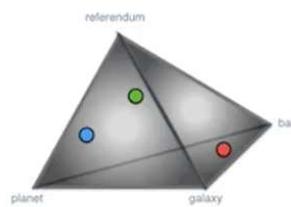
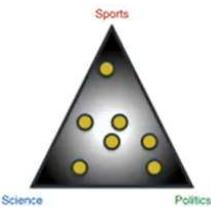


$$\prod_{i=1}^K P(\varphi_i; \beta)$$



$$\prod_{t=1}^N P(Z_{j,t} | \theta_j) \quad P(W_{j,t} | \varphi_{Z_{j,t}})$$

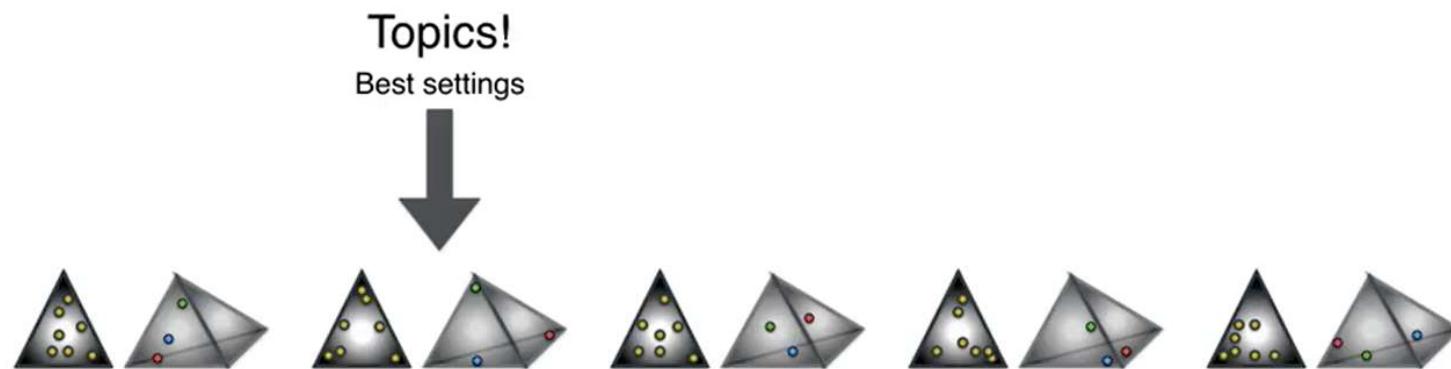
$P(\text{same articles}) = \text{low}$



$P(\text{same articles}) = \text{very low}$



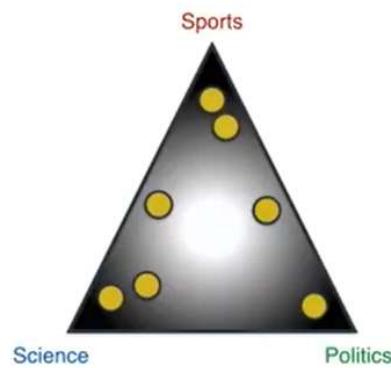
Best settings on the machine



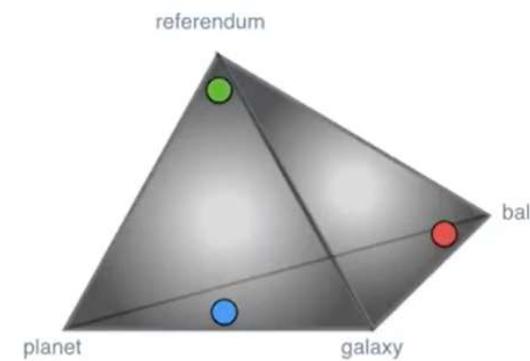


Mathematical modelling of LDA

The winning arrangements



Documents-Topics

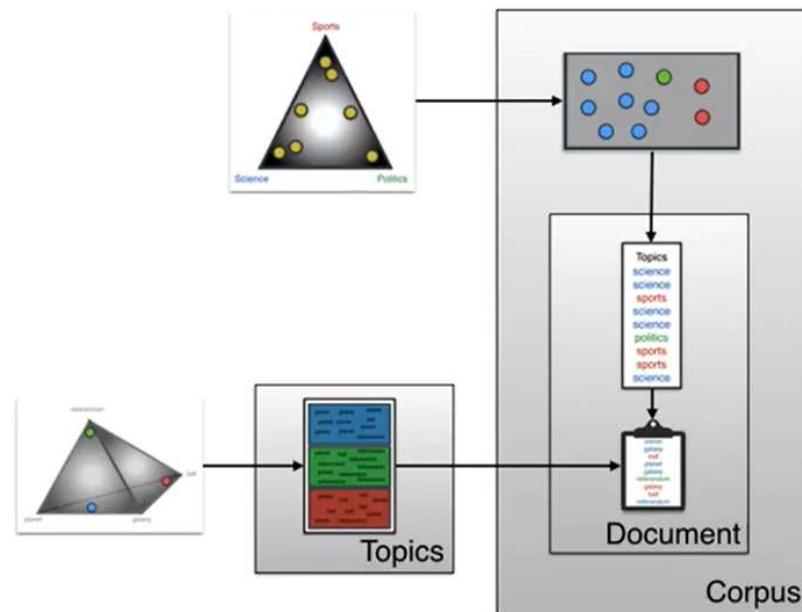
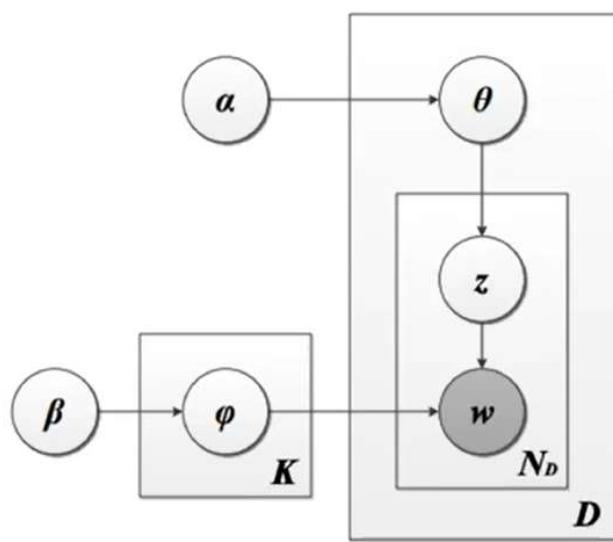


Topics-Words

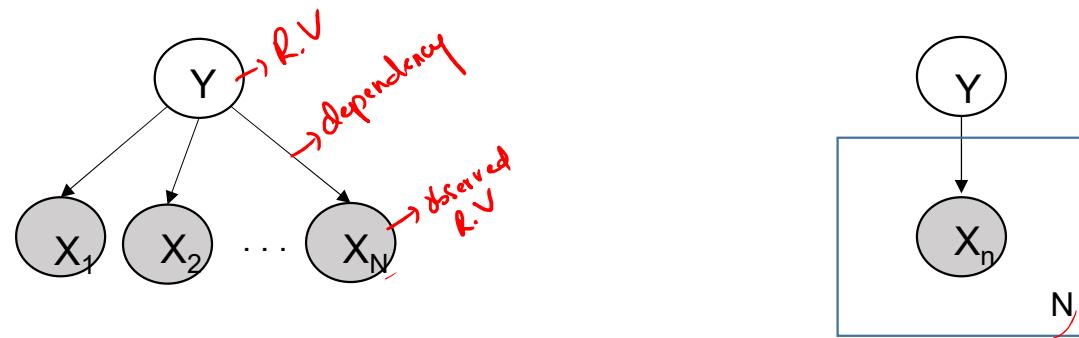


Length of the articles?
Poisson distribution

Latent Dirichlet Allocation



Directed graphical model

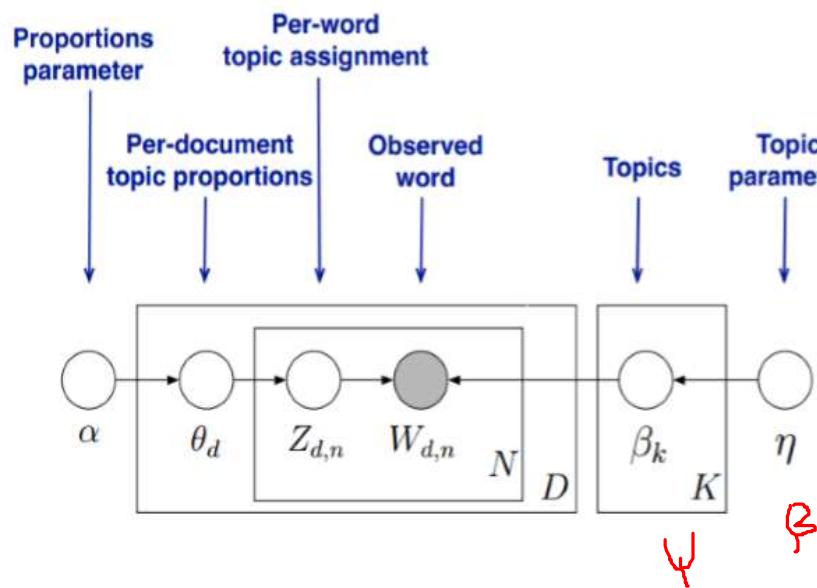


$$P(Y, x_1, x_2, \dots, x_N) = P(Y) \prod_{n=1}^N P(x_n|y)$$



Graphical LDA Model

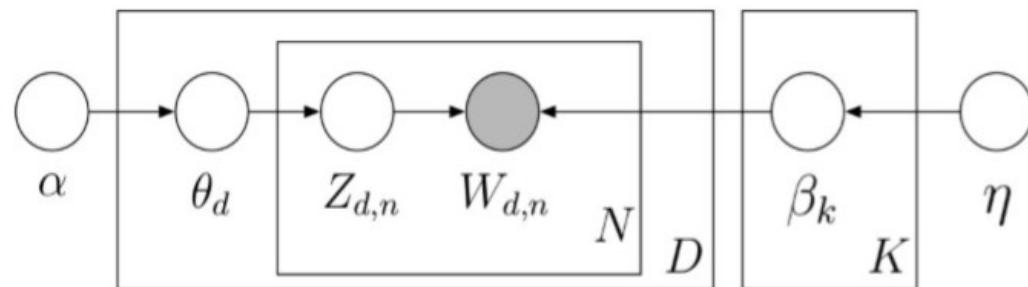
Our goal is to **infer** or **estimate** the hidden variables, i.e. computing their distribution conditioned on the documents. $\longrightarrow p(\text{topics, proportions, assignments} \mid \text{documents})$



- Nodes are RVs; edges indicate dependence.
- Shaded nodes are observed, and unshaded nodes are hidden.
- Plates indicate replicated variables.



Graphical LDA Model



K – total number of topics

β_k – topic, a distribution over the vocabulary

D – total number of documents

Θ_d – per-document topic proportions

N – total number of words in a document (it fact, it should be N_d)

$Z_{d,n}$ – per-word topic assignment

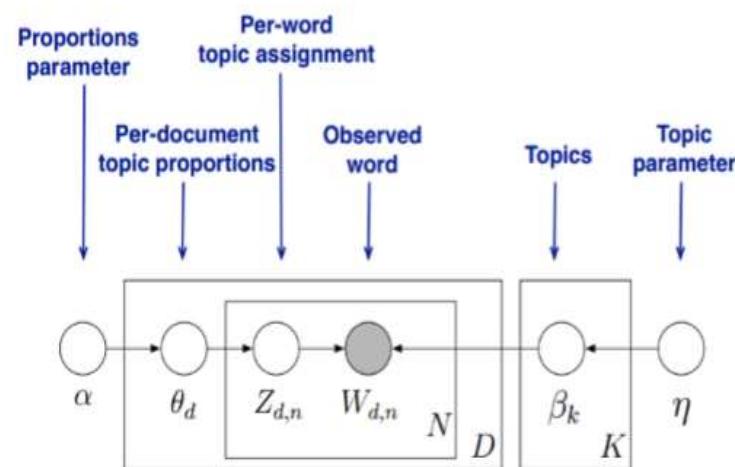
$W_{d,n}$ – observed word

α, η – Dirichlet parameters

- Several **inference algorithms** are available (e.g. sampling based)
- A few **extensions** to LDA were created:
 - Bigram Topic Model



Graphical LDA Model



$$p(\beta, \theta, z, w) = \left(\prod_{i=1}^K p(\beta_i | \eta) \right) \left(\prod_{d=1}^D p(\theta_d | \alpha) \prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

1) Draw each topic $\beta_i \sim Dir(\eta)$ for $i = 1, \dots, K$

2) For each document:

First, Draw topic proportions $\theta_d \sim Dir(\alpha)$

For each word within the document:

a) Draw $Z_{d,n} \sim Multi(\theta_d)$

b) Draw $W_{d,n} \sim Multi(\beta_{z_{d,n}})$



LDA Model

This joint distribution defines a posterior $p(\theta, z, \beta | w)$.

From a collection of documents we have to infer:

1. Per-word topic assignment $z_{d,n}$.
2. Per-document topic proportions θ_d .
3. Per-corpus topic distributions β_k .

Then use posterior expectations ($E\{\beta|w\}$ for the corpus, $E\{\theta_d|w\}$ for each document) to perform the task at hand: information retrieval, document similarity, exploration, and others.



LDA Matrix representation

$$\begin{matrix} & \text{documents} \\ \text{words} & \boxed{\mathbf{C}} \end{matrix} = \begin{matrix} & \text{topics} \\ \text{words} & \boxed{\boldsymbol{\Phi}} \end{matrix} \begin{matrix} & \text{documents} \\ \text{topics} & \boxed{\boldsymbol{\Theta}} \end{matrix}$$

normalized co-occurrence matrix mixture components mixture weights



Mathematical modelling of LDA

Formal definition of the model:

$$p(\beta, \theta, z, w) = \left(\prod_{i=1}^K p(\beta_i | \eta) \right) \left(\prod_{d=1}^D p(\theta_d | \alpha) \prod_{n=1}^N p(z_{d,n} | \theta_d) p(w_{d,n} | \beta_{1:K}, z_{d,n}) \right)$$

$$(\beta_d | \eta) \sim Dir(\beta)$$

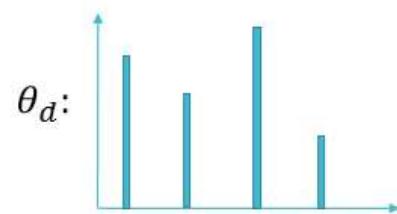
$$(\theta_d | \alpha) \sim Dir(\alpha)$$

$$Z_{d,n} \sim Multi(\theta_d)$$

$$W_{d,n} \sim Multi(\beta_{z_{d,n}})$$

$$p(z_{d,n} | \theta_d) = \theta_{d,z_{d,n}}$$

$$p(w_{d,n} | z_{d,n}, \beta_{1:K}) = \beta_{z_{d,n}, w_{d,n}}$$



β :

Topics	Word probabilities for each topic		



Train LDA - Gibbs Sampling



Sports



Politics



Science



Science

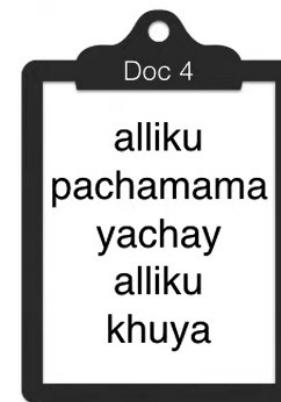
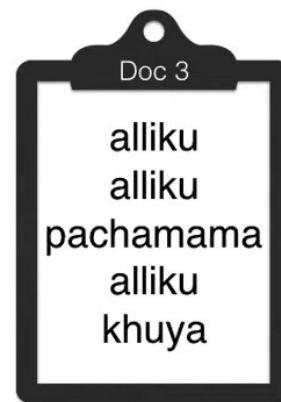
Science

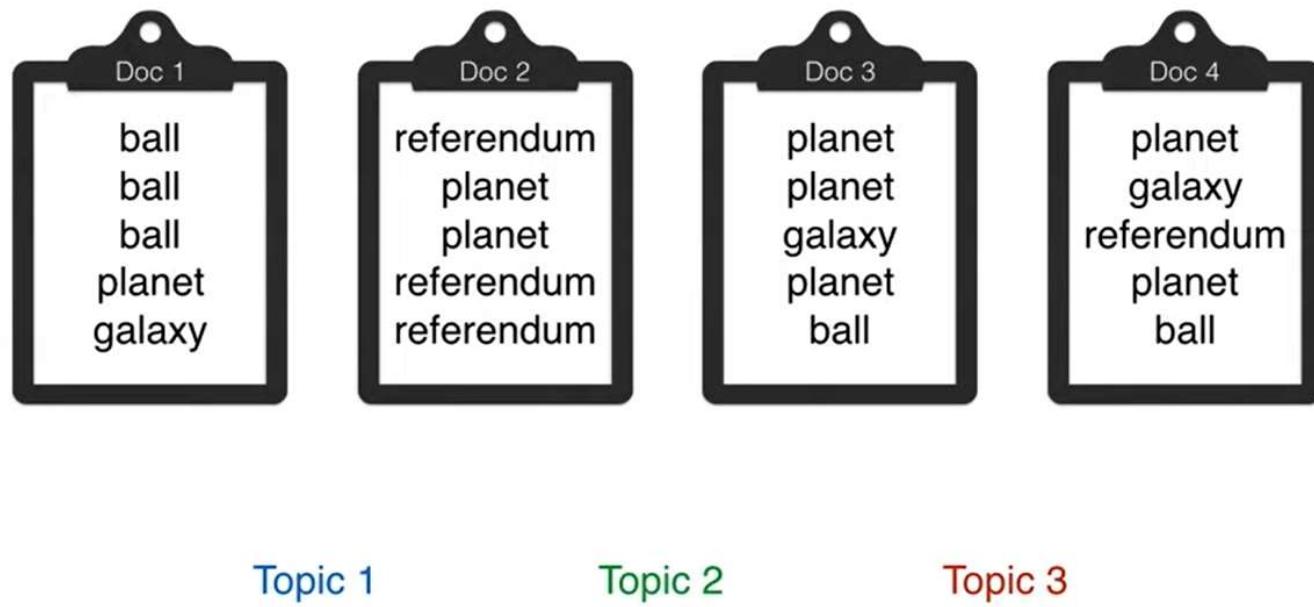
Politics

Sports



Assign Topics





Topic 1

Topic 2

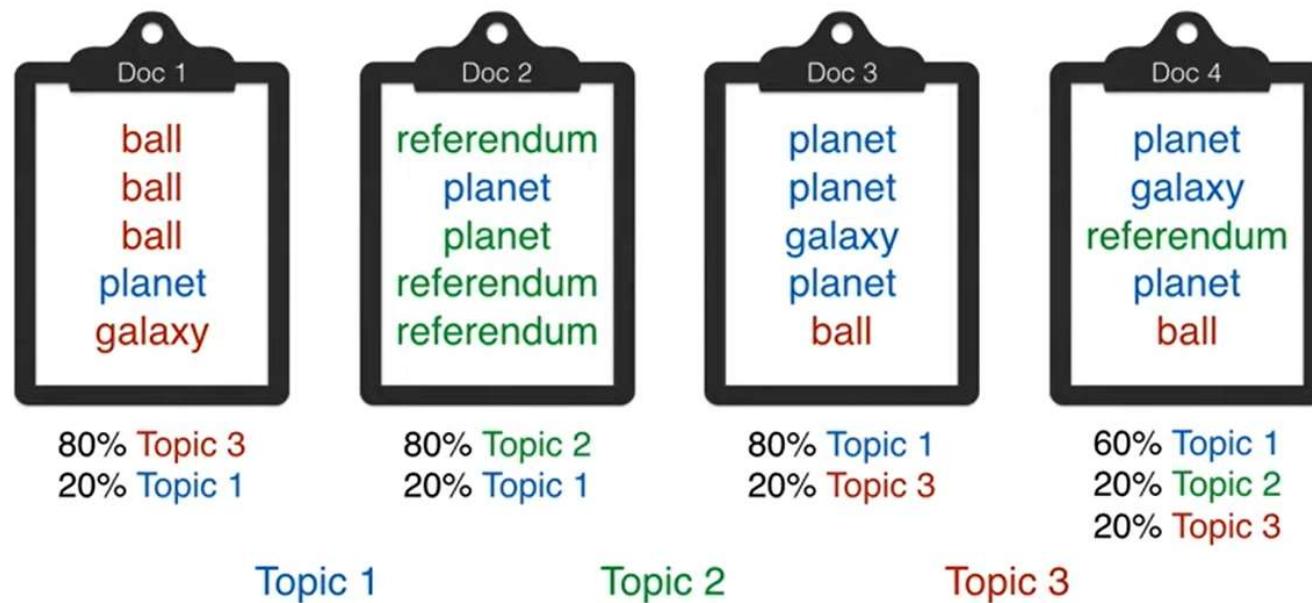
Topic 3



Topic 1

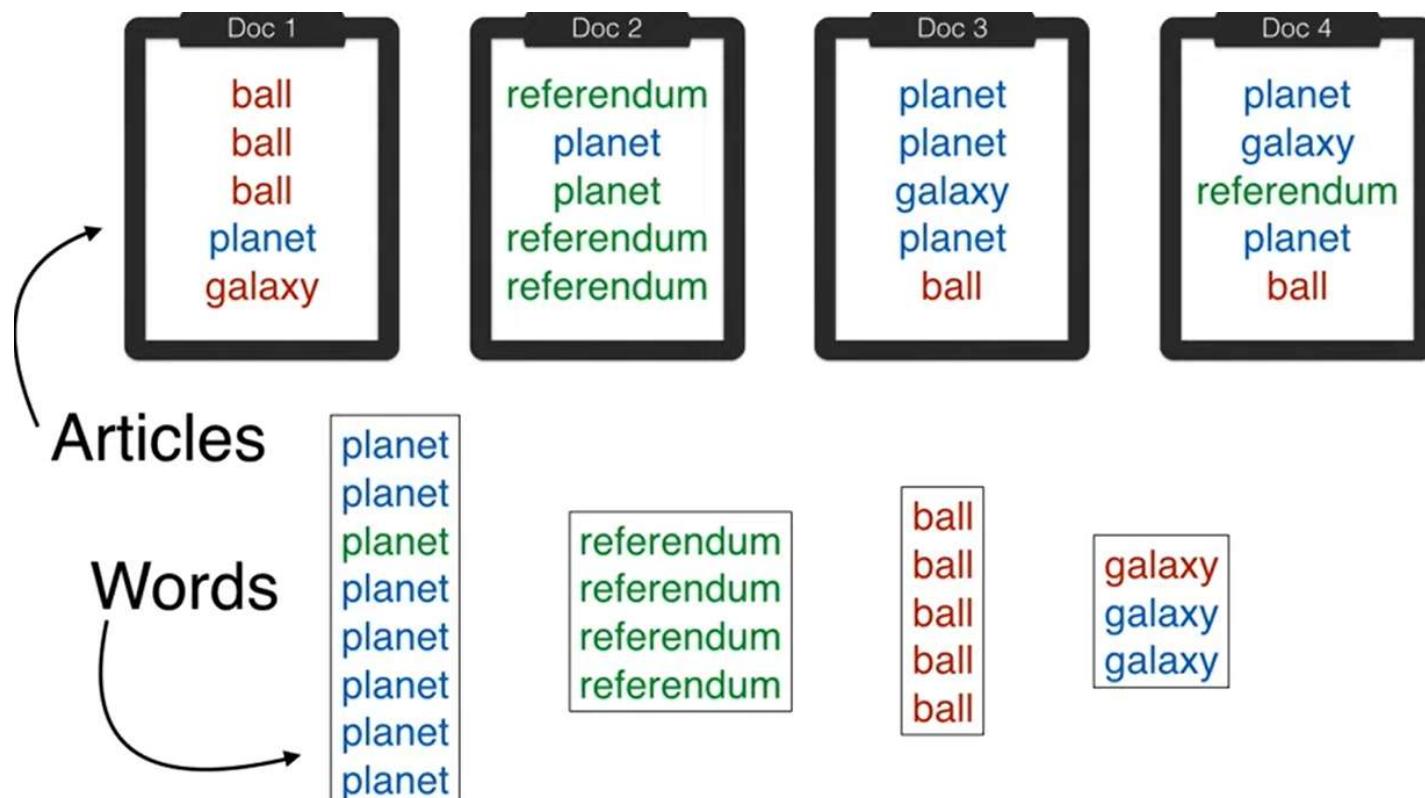
Topic 2

Topic 3





Property 1:
Articles are as monochromatic as possible





Property 2: Words are as monochromatic as possible

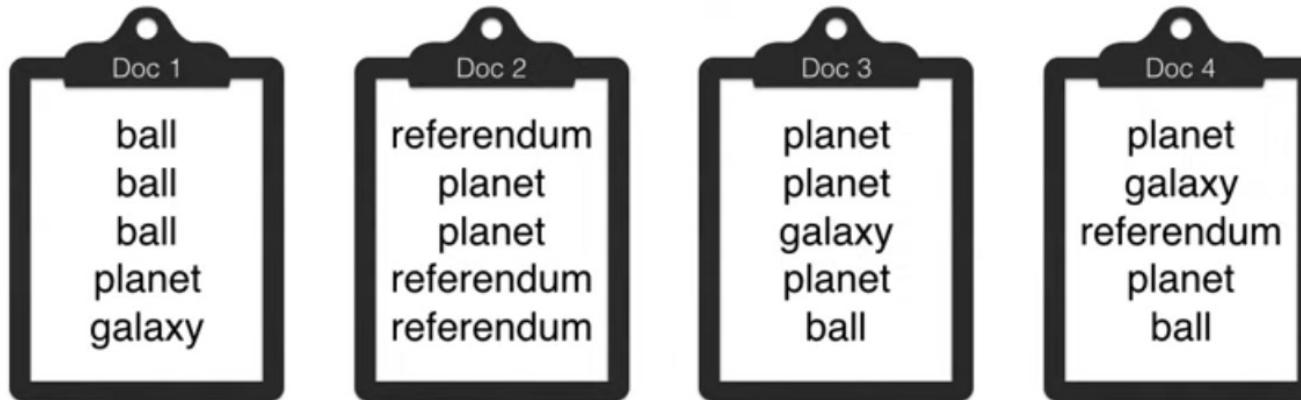
Words

planet
planet
planet
planet
planet
planet
planet
planet

referendum
referendum
referendum
referendum

ball
ball
ball
ball
ball

galaxy
galaxy
galaxy



Goal: Color each word with **blue, green, red**

1. Each article is as monochromatic as possible
2. Each word is as monochromatic as possible



Topic 1



Topic 2



Doc 4

planet
galaxy
referendum
planet
ball

ball

Topic 3

How much is Topic 1 in Doc 1?

2

How much is 'ball' in Topic 1?

0

Product: 0

How much is Topic 2 in Doc 1?

0

How much is 'ball' in Topic 2?

1

Product: 0

How much is Topic 3 in Doc 1?

2

How much is 'ball' in Topic 3?

3

Product: 6



Topic 1

How much is Topic 1 in Doc 1?

$$2 + \alpha$$

How much is 'ball' in Topic 1?

$$0 + \beta$$

Topic 2

How much is Topic 2 in Doc 1?

$$0 + \alpha$$

How much is 'ball' in Topic 2?

$$1 + \beta$$

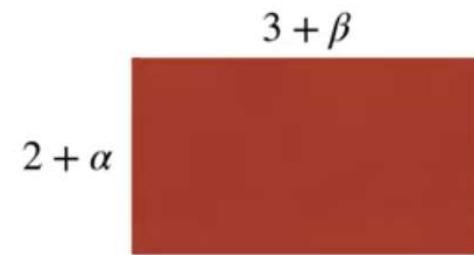
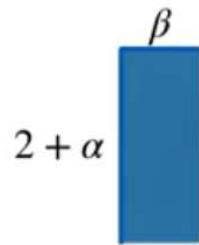
Topic 3

How much is Topic 3 in Doc 1?

$$2 + \alpha$$

How much is 'ball' in Topic 3?

$$3 + \beta$$



Topic 1

How much is Topic 1 in Doc 1?

$$2 + \alpha$$

How much is 'ball' in Topic 1?

$$0 + \beta$$

Topic 2

How much is Topic 2 in Doc 1?

$$0 + \alpha$$

How much is 'ball' in Topic 2?

$$1 + \beta$$

Topic 3

How much is Topic 3 in Doc 1?

$$2 + \alpha$$

How much is 'ball' in Topic 3?

$$3 + \beta$$



80% Topic 3
20% Topic 1



80% Topic 2
20% Topic 1



80% Topic 1
20% Topic 3



60% Topic 1
20% Topic 2
20% Topic 3

Science

Topic 1
planet (7)
galaxy (2)

Politics

Topic 2
referendum (4)
planet (1)

Sports

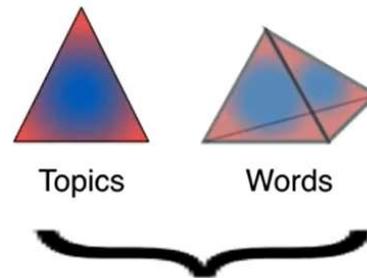
Topic 3
ball (5)
galaxy (1)



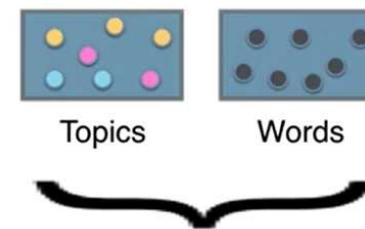
Mathematical modelling of LDA

Probability of a document

$$P(\mathbf{W}, \mathbf{Z}, \boldsymbol{\theta}, \boldsymbol{\varphi}; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\varphi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \varphi_{Z_{j,t}})$$



Dirichlet
Distributions



Multinomial
Distributions



Probability of a document

$$P(\mathbf{W}, \mathbf{Z}, \theta, \phi; \alpha, \beta) = \prod_{j=1}^M P(\theta_j; \alpha) \prod_{i=1}^K P(\phi_i; \beta) \prod_{t=1}^N P(Z_{j,t} | \theta_j) P(W_{j,t} | \phi_{Z_{j,t}})$$

↑ ↑
Topics Words

Gibbs sampling



Gibbs Sampling Algorithm

- Step1: Assign a random topic [1...T] for each word
- Step2: For each word token, a new topic is sampled as per $P(z_i=j|z_{-i}, w_i, d_i)$ and the matrices C_{wt} (word-topic) and C_{dt} (document-topic) are updated.
- One iteration over all word token in the document is a Gibbs Sample
- Each iteration may have correlation with the next hence these samples are saved at spaced intervals.



LDA Summary

Documents are probability distributions over latent topics.

Topics are probability distributions over words.

LDA takes a number of documents. It assumes that the words in each document are related. It then tries to figure out the “recipe” for how each document could have been created. We just need to tell the model how many topics to construct and it uses that “recipe” to generate topic and word distributions over a corpus. Based on that output, we can identify similar documents within the corpus.



LDA Summary

ADVANTAGES

LDA is an effective tool for topic modeling.

Easy to understand conceptually

Has been shown to produce good results over many domains.

New applications

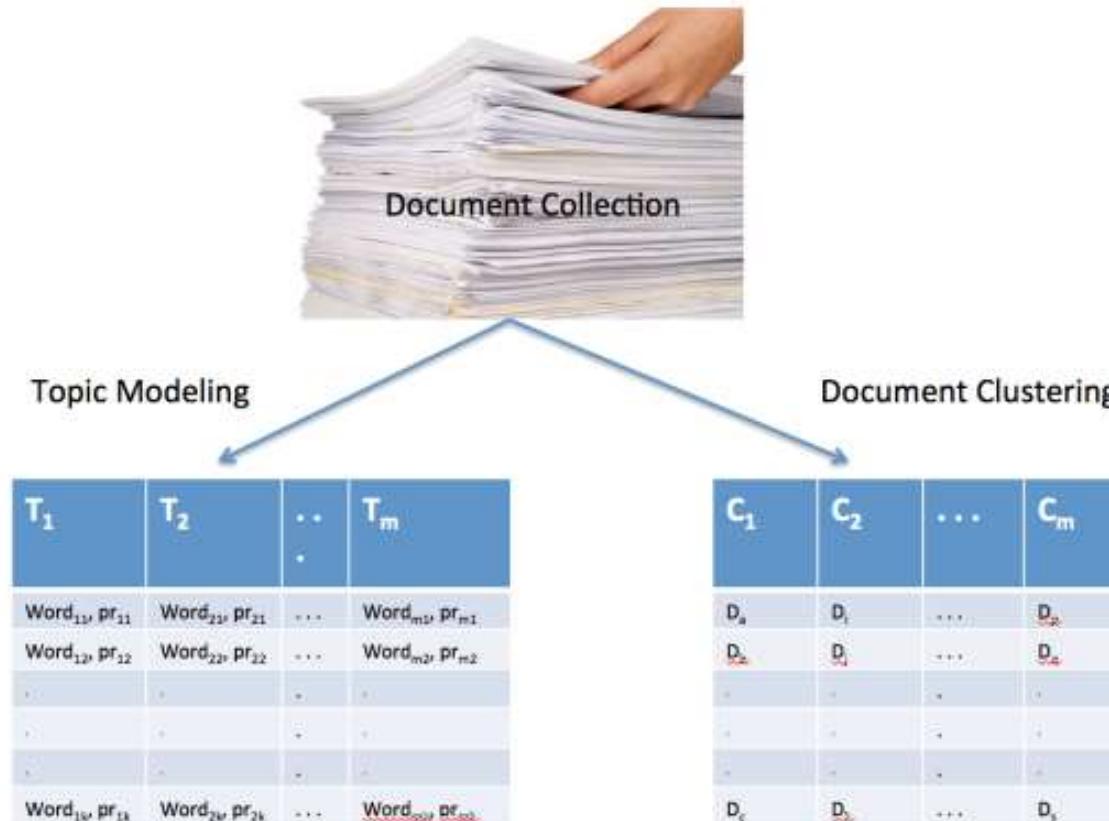
LIMITATIONS

Must know the number of topics K in advance

Dirichlet topic distribution cannot capture correlations among topics



Difference between document clustering and topic modeling



<https://iksinc.online/2016/05/16/topic-modeling-and-document-clustering-whats-the-difference/>



Implementation Tools

Tooling



gensim: topic modeling for humans

- Free python library
- Memory independent
- Distributed computing

<http://radimrehurek.com/gensim>



Stanford Topic Modeling Toolbox

<http://nlp.stanford.edu/software/tmt>



MAchine Learning for LanguagE Toolkit (MALLET) is a Java-based package for:

- statistical natural language processing
- document classification
- Clustering
- topic modeling
- information extraction
- and other machine learning applications to text.

<http://mallet.cs.umass.edu>



References

Bernoulli trial, binomial and multinomial distribution:

<https://www.askiitians.com/iit-jee-algebra/probability/bernoulli-trials-and-binomial-distribution/>

Beta Distribution:

- https://www.youtube.com/watch?v=v1uUgTcInQk&feature=emb_logo

Conjugate Prior:

- https://www.youtube.com/watch?time_continue=2&v=aPNrhR0dFi8&feature=emb_logo
- <https://www.youtube.com/watch?v=qpNAXnmy0GU>

Topic Models

- <https://www.youtube.com/watch?v=fCmlceNqVog>

Gibbs Sampling

- <https://www.youtube.com/watch?v=u7I5hhmdc0M>
- <https://medium.com/analytics-vidhya/topic-modeling-using-lda-and-gibbs-sampling-explained-49d49b3d1045>



References

LDA

- [1] Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." Journal of machine Learning research (2003): 993-1022.
- <https://www.youtube.com/watch?v=3mHy4OSyRf0>
- <https://www.coursera.org/learn/text-mining/lecture/dmpQ0/2-5-topic-mining-and-analysis-motivation-and-task-definition>
- <https://www.youtube.com/watch?v=NYkbqzTIW3w>
- <https://github.com/adashofdata/nlp-in-python-tutorial>
- https://www.youtube.com/watch?time_continue=3&v=Cpt97Bpl-t4&feature=emb_logo
- <https://github.com/bhattbhavesh91>
- <https://www.youtube.com/watch?v=T05t-SqKArY>
- https://www.youtube.com/watch?v=BaM1uiCpj_E
- <https://livebook.manning.com/book/grokking-machine-learning/>