Syntagmatic Relation Discovery: Entropy

Syntagmatic Relation = Correlated Occurrences

Whenever "**eats**" occurs, what **other words** also tend to occur?

My cat **eats** fish on Saturday
His cat **eats** turkey on Tuesday
My dog **eats** meat on Sunday
His dog **eats** turkey on Tuesday
...

My ___ **eats** ___ on Saturday
His ___ **eats** ___ on Tuesday
My ___ **eats** ___ on Sunday
His ___ **eats** ___ on Tuesday
...

What words tend to occur to the **left** of "**eats**"?

What words are to the **right?**

Word Prediction: Intuition and Formal Definition

Prediction questions:

Is word **W** present/ absent in a text segment (sentence, paragraph, document)?

Are some words easier to predict than others:

1) W = "meat" 2) W="the" 3) W="unicorn"

Binary Random Variable : $X_w \in \{0, 1\}$

$$X_w = \begin{cases} 1 & w \text{ is present} \\ 0 & w \text{ is absent} \end{cases}$$

$$p(X_w = 1) + p(X_w = 0) = 1$$

The more random X_w, the more difficult the prediction.

How to quantitatively measure the "randomness" of a random variable like Xw?

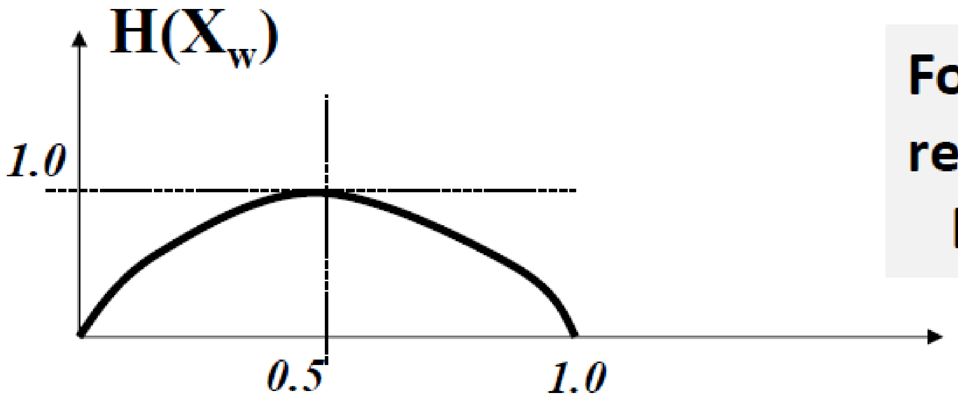
Entropy H(X) Measures Randomness of X

$$H(X_w) = \sum_{v \in \{0,1\}} -p(X_w = v)\log_2 p(X_w = v)$$

$$X_w = \begin{cases} 1 & \text{w is present} \\ 0 & \text{w is absent} \end{cases}$$

$$= -p(X_w = 0)\log_2 p(X_w = 0) - p(X_w = 1)\log_2 p(X_w = 1)$$

Define $0\log_2 0 = 0$



or equivalently P(Xw=0) (Why?)

For what X_w, does $H(X_w)$ reach maximum/minimum?

E.g., $P(X_w=1)=1$? $P(X_w=1)=0.5$?

Entropy H(X): Coin Tossing

$$H(X_{coin}) = -p(X_{coin}=0)\log_2 p(X_{coin}=0) - p(X_{coin}=1)\log_2 p(X_{coin}=1)$$

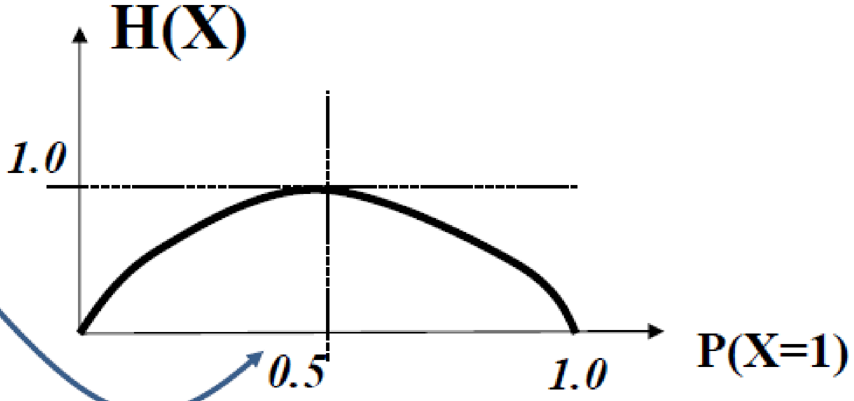
X_{coin}: tossing a coin $\quad X_{coin} = \begin{cases} 1 & \text{Head} \\ 0 & \text{Tail} \end{cases}$

Fair coin: p(X=1)=p(X=0)=1/2

$$H(X) = -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1$$

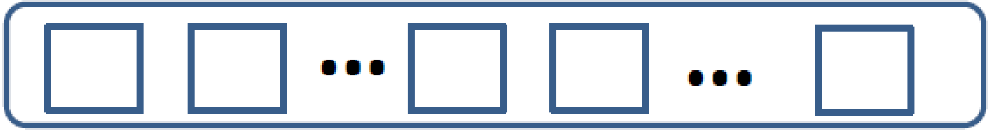
Completely biased: p(X=1)=1

$$H(X) = -0*\log_2 0 - 1*\log_2 1 = 0$$



Entropy for Word Prediction

Is word **W** present (or absent) in this segment?



1) W = "meat"   2) W = "the"   3) W = "unicorn"

Which is **high/low**? **H(X_{meat}), H(X_{the}), or H($X_{unicorn}$)?**

H(X_{the})≈0 → no uncertainty since p(X_{the}=1) ≈ 1

High entropy words are harder to predict!

Syntagmatic Relation Discovery: Conditional Entropy

What If We Know More About a Text Segment?

Does presence of "**eats**" help predict the presence of "**meat**"?

Does it **reduce** the uncertainty about "meat", i.e., **H(X_{meat})?**

What if we know of the absence of "eats"? Does it also help?

Know nothing about the segment | **Know "eats" is present (X_{eats} =1)**

$$p(X_{meat}=1) \dashrightarrow p(X_{meat}=1 \mid X_{eats}=1)$$

$$p(X_{meat}=0) \dashrightarrow p(X_{meat}=0 \mid X_{eats}=1)$$

$$H(X_{meat}) = -p(X_{meat}=0)\log_2 p(X_{meat}=0) - p(X_{meat}=1)\log_2 p(X_{meat}=1)$$

$$\Downarrow$$

$$H(X_{meat} \mid X_{eats}=1) = -p(X_{meat}=0 \mid X_{eats}=1)\log_2 p(X_{meat}=0 \mid X_{eats}=1)$$
$$- p(X_{meat}=1 \mid X_{eats}=1)\log_2 p(X_{meat}=1 \mid X_{eats}=1)$$

$H(X_{meat} \mid X_{eats}=0)$ **can be defined similarly**

Conditional Entropy: Complete Definition, Capturing Syntagmatic Relation

$$H(X_{meat} \mid X_{eats}) = \sum_{u\in\{0,1\}} [p(X_{eats}=u)\ H(X_{meat} \mid X_{eats}=u)]$$

$$= \sum_{u\in\{0,1\}} [p(X_{eats}=u) \sum_{v\in\{0,1\}} [-p(X_{meat}=v \mid X_{eats}=u) \log_2 p(X_{meat}=v \mid X_{eats}=u)]]$$

In general, for any discrete random variables X and Y, we have H(X) $\geq$ H(X|Y)

What's the minimum possible value of H(X|Y)?

$$H(X_{meat} \mid X_{eats}) = \sum_{u\in\{0,1\}} [p(X_{eats}=u)\ H(X_{meat} \mid X_{eats}=u)]$$

$$H(X_{meat} \mid X_{meat}) = ?$$

Which is smaller? $H(X_{meat}|X_{the})$ or $H(X_{meat}|X_{eats})$?
For which word w, does $H(X_{meat}|X_w)$ reach its minimum (i.e., 0)?
For which word w, does $H(X_{meat}|X_w)$ reach its maximum, $H(X_{meat})$?

Mining Syntagmatic Relations with Conditional Entropy

- For each word W1
  - For **every other word** W2, compute $H(X_{W1}|X_{W2})$
  - Sort all $H(X_{W1}|X_{W2})$ in **ascending order**
  - **Top-ranked** candidate words - **potential syntagmatic** relations with W1
  - Need to use a **threshold** for each W1
- However, while $H(X_{W1}|X_{W2})$ and $H(X_{W1}|X_{W3})$ are comparable, $H(X_{W1}|X_{W2})$ and $H(X_{W3}|X_{W2})$ aren't!

How can we mine the strongest K syntagmatic relations from a collection?

Syntagmatic Relation Discovery: Mutual Information I(X;Y)

Mutual Information I(X;Y): Entropy Reduction & Sintagm Rel Mining

If we know Y, how much reduction in the entropy of X can we get?

Mutual Information: $I(X; Y) = H(X) - H(X|Y) = H(Y) - H(Y|X)$

Properties:

- **Non-negative**: $I(X;Y) \geq 0$
- **Symmetric**: $I(X;Y) = I(Y;X)$
- $I(X;Y) = 0$ iff X & Y are **independent**

With fixed X and different Ys: same order of I(X;Y) and H(X|Y), but **I(X;Y) allows to compare different (X,Y) pairs**.

Whenever "**eats**" occurs, what **other words** also tend to **occur**?
Which **words** have high mutual information with "**eats**"?

$$I(X_{eats}; X_{meats}) = I(X_{meats}; X_{eats}) > I(X_{eats}; X_{the}) = I(X_{the}; X_{eats})$$

$$I(X_{eats}; X_{eats}) = H(X_{eats}) \geq I(X_{eats}; X_w)$$

Rewriting Mutual Information (MI) Using KL-divergence

The <u>observed</u> joint distribution of X_{W1} and X_{W2}

$$I(X_{w1};X_{w2}) = \sum_{u\in\{0,1\}} \sum_{v\in\{0,1\}} p(X_{w1}=u, X_{w2}=v) \log_2 \frac{p(X_{w1}=u, X_{w2}=v)}{p(X_{w1}=u)p(X_{w2}=v)}$$

The <u>expected</u> joint distribution of X_{W1} and X_{W2} if X_{W1} and X_{W2} were independent

MI measures the **divergence** of the **actual joint distribution** <u>from the expected</u> distribution under the independence assumption. The **larger the divergence** is, the **higher the MI** would be.

Mutual Information: Probabilities and Relations Between Them

$$I(X_{w1};X_{w2}) = \sum_{u\in\{0,1\}} \sum_{v\in\{0,1\}} p(X_{w1}=u, X_{w2}=v) \log_2 \frac{p(X_{w1}=u, X_{w2}=v)}{p(X_{w1}=u)p(X_{w2}=v)}$$

Presence & absence of w1: $p(X_{W1}=1) + p(X_{W1}=0) = 1$

Presence & absence of w2: $p(X_{W2}=1) + p(X_{W2}=0) = 1$

Co-occurrences of w1 and w2:

$$\underbrace{p(X_{W1}=1, X_{W2}=1)}_{\text{Both w1 \& w2 occur}} + \underbrace{p(X_{W1}=1, X_{W2}=0)}_{\text{Only w1 occurs}} + \underbrace{p(X_{W1}=0, X_{W2}=1)}_{\text{Only w2 occurs}} + \underbrace{p(X_{W1}=0, X_{W2}=0)}_{\text{None of them occurs}} = 1$$

Co-occurrences of w1 and w2:

$p(X_{W1}=1, X_{W2}=1) + p(X_{W1}=1, X_{W2}=0)+p(X_{W1}=0, X_{W2}=1) + p(X_{W1}=0, X_{W2}=0) = 1$

Constraints:

$p(X_{W1}=1, X_{W2}=1) + p(X_{W1}=1, X_{W2}=0) = p(X_{W1}=1)$

$p(X_{W1}=0, X_{W2}=1) + p(X_{W1}=0, X_{W2}=0)= p(X_{W1}=0)$

$p(X_{W1}=1, X_{W2}=1) + p(X_{W1}=0, X_{W2}=1) = p(X_{W2}=1)$

$p(X_{W1}=1, X_{W2}=0) + p(X_{W1}=0, X_{W2}=0) = p(X_{W2}=0)$

Computation of Mutual Information

Presence & absence of w1: $\boxed{p(X_{w1}=1)} + p(X_{w1}=0) = 1$

Presence & absence of w2: $\boxed{p(X_{w2}=1)} + p(X_{w2}=0) = 1$

Co-occurrences of w1 and w2:

$$\boxed{p(X_{w1}=1, X_{w2}=1)} + p(X_{w1}=1, X_{w2}=0) + p(X_{w1}=0, X_{w2}=1) + p(X_{w1}=0, X_{w2}=0) = 1$$

$$\boxed{p(X_{w1}=1, X_{w2}=1)} + p(X_{w1}=1, X_{w2}=0) = \boxed{p(X_{w1}=1)}$$

$$p(X_{w1}=0, X_{w2}=1) + p(X_{w1}=0, X_{w2}=0) = p(X_{w1}=0)$$

$$\boxed{p(X_{w1}=1, X_{w2}=1)} + p(X_{w1}=0, X_{w2}=1) = \boxed{p(X_{w2}=1)}$$

$$p(X_{w1}=1, X_{w2}=0) + p(X_{w1}=0, X_{w2}=0) = p(X_{w2}=0)$$

We only need to know $p(X_{w1}=1)$, $p(X_{w2}=1)$, and $p(X_{w1}=1, X_{w2}=1)$.

Estimation of Probabilities (Depending on the Data)

$$p(X_{w1}=1)=\frac{count(w1)}{N}$$

$$p(X_{w2}=1)=\frac{count(w2)}{N}$$

$$p(X_{w1}=1,X_{w2}=1)=\frac{count(w1,w2)}{N}$$

	W1	W2	
Segment_1	1	0	Only W1 occurred
Segment_2	1	1	Both occurred
Segment_3	1	1	Both occurred
Segment_4	0	0	Neither occurred
…			
Segment_N	0	1	Only W2 occurred

Count(w1) = total number segments that contain W1
Count(w2) = total number segments that contain W2
Count(w1, w2) = total number segments that contain both W1 and W2

Smoothing: Accommodating Zero Counts

$$p(X_{w1}=1)=\frac{\text{count}(w1)+0.5}{N+1}$$

$$p(X_{w2}=1)=\frac{\text{count}(w2)+0.5}{N+1}$$

$$p(X_{w1}=1,X_{w2}=1)=\frac{\text{count}(w1,w2)+0.25}{N+1}$$

	W1	W2
¼ PseudoSeg_1	0	0
¼ PseudoSeg_2	1	0
¼ PseudoSeg_3	0	1
¼ PseudoSeg_4	1	1

Segment_1	1	0
…		
Segment_N	0	1

Actually observed data

<u>Smoothing</u>: Add pseudo data so that no event has zero counts (pretend we observed extra data)

Summary of Syntagmatic Relation Discovery

- **Syntagmatic relation** can be discovered by **measuring correlations** between occurrences of two words.
- Three concepts from Information Theory:
  - **Entropy** H(X): measures the uncertainty of a random variable X
  - **Conditional entropy** H(X|Y): entropy of X given we know Y
  - **Mutual information** I(X;Y): entropy reduction of X (or Y) due to knowing Y (or X)
- **Mutual information** provides a principled way for **discovering syntagmatic relations**.

Summary of Word Association Mining

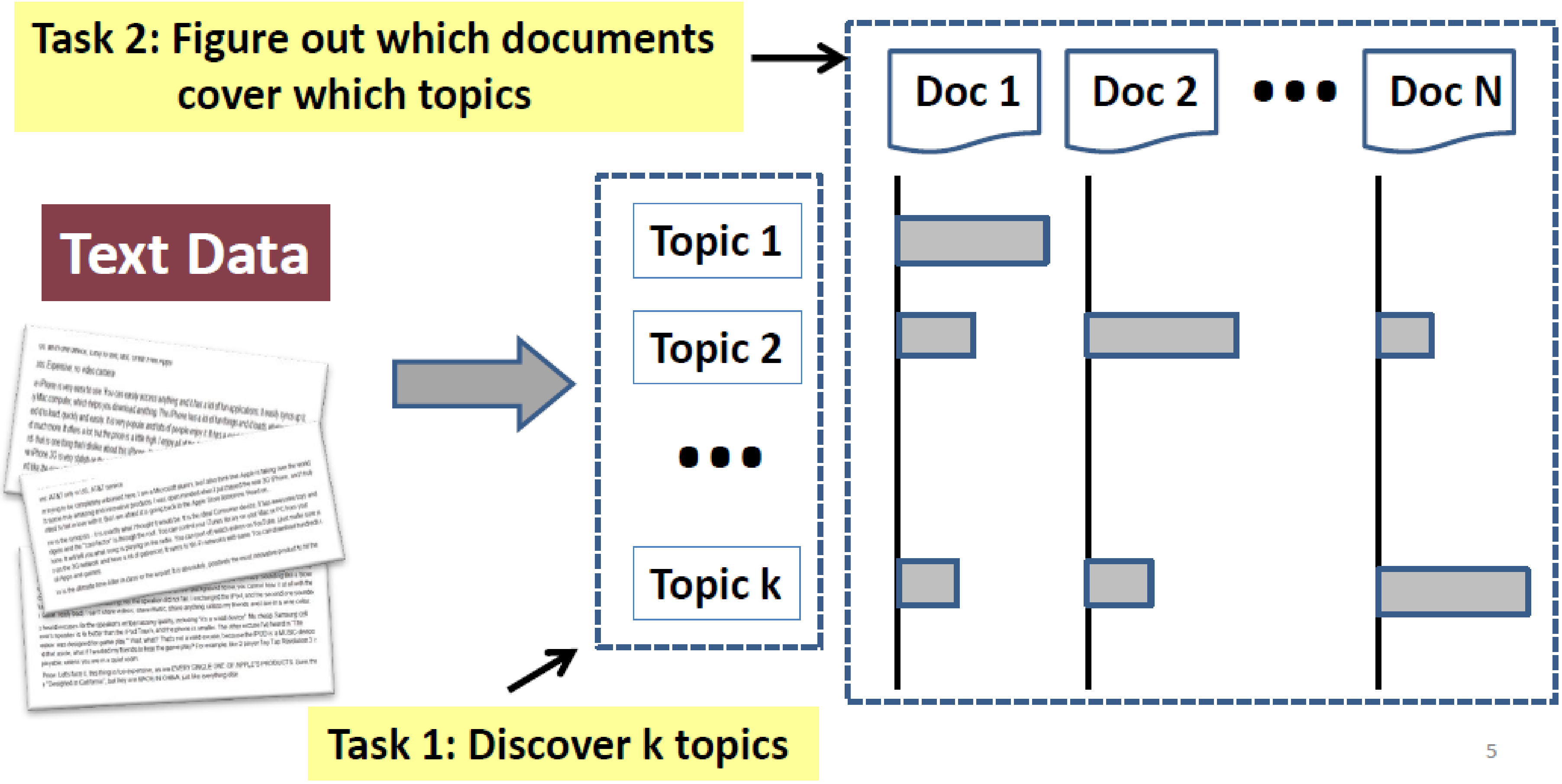
- Two basic associations: **paradigmatic and syntagmatic**
  - Generally applicable to any items in any language (e.g., phrases or entities as units)
- Pure statistical approaches are available for discovering both (can be combined to perform joint analysis).
  - Generally applicable to any text with no human effort
  - Different ways to define "context" and "segment" lead to interesting variations of applications
- Discovered associations can support many other applications.

Topic Mining and Analysis: Motivation and Task Definition

Topic Mining and Analysis: Motivation

- Topic ≈ **main idea** discussed in text data; **knowledge** about the world
  - Theme/subject of a discussion or conversation
  - Different granularities (e.g., topic of a sentence, an article, etc.)
- Many applications require discovery of topics in text
  - Summaries from Twitter
  - Current research topics in data mining; their difference from 5 years ago?
  - Likes and dislikes about iPhone 6
  - Major topics debated in 2012 presidential election

Tasks of Topic Mining and Analysis



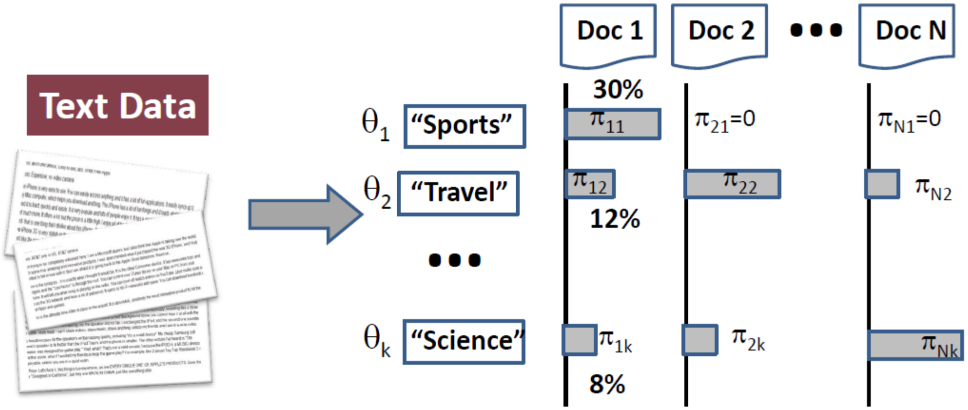
Formal Definition of Topic Mining and Analysis

- Input
  - A **collection** of **N** text documents $C=\{d_1, \ldots, d_N\}$
  - **Number of topics**: **k**
- Output
  - **k topics**: $\{ \theta_1, \ldots, \theta_k \}$
  - **Coverage of topics in each** d_i: $\{ \pi_{i1}, \ldots, \pi_{ik} \}$
  - π_{ij} = prob. of d_i covering topic θ_j

$$\sum_{j=1}^{k} \pi_{ij} = 1$$

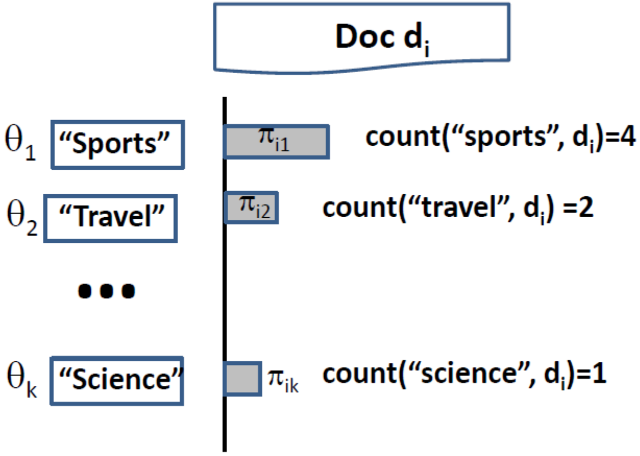
How to define θ_i ?

Topic = Term



Drawbacks:

- Lack of **expressive power**
  - Can represent simple/general topics, but not complicated topics
- **Incompleteness** in vocabulary coverage
  - Can't capture variations of vocabulary (e.g., related words)
- Word sense **ambiguity**
  - of the topical term (e.g., basketball star vs. star in the sky)



$$\pi_{ij} = \frac{\text{count}(\theta_j, d_i)}{\sum_{L=1}^{k} \text{count}(\theta_L, d_i)}$$

Topic Mining and Analysis: Probabilistic Topic Models

Improved Idea: Topic = Word Distribution

θ_1 **"Sports"**

$P(w|\theta_1)$

sports	0.02
game	0.01
basketball	0.005
football	0.004
play	0.003
star	0.003
…	
nba	0.001
…	
travel	0.0005
…	

θ_2 **"Travel"**

$P(w|\theta_2)$

travel	0.05
attraction	0.03
trip	0.01
flight	0.004
hotel	0.003
island	0.003
…	
culture	0.001
…	
play	0.0002
…	

…

θ_k **"Science"**

$P(w|\theta_k)$

science	0.04
scientist	0.03
spaceship	0.006
telescope	0.004
genomics	0.004
star	0.002
…	
genetics	0.001
…	
travel	0.00001
…	

$$\sum_{w \in V} p(w|\theta_i) = 1$$

Vocabulary Set: V={w1, w2,….}

4

Resolving drawbacks of topic = term:

- Lack of expressive power (no <u>complicated topics</u>) - **Topic = {Multiple Words}**
- Incompleteness in vocabulary coverage (no <u>related words</u>) – introduce **weights on words**
- Word sense ambiguity (<u>star</u>) - **split an ambiguous word** (into different distributions)

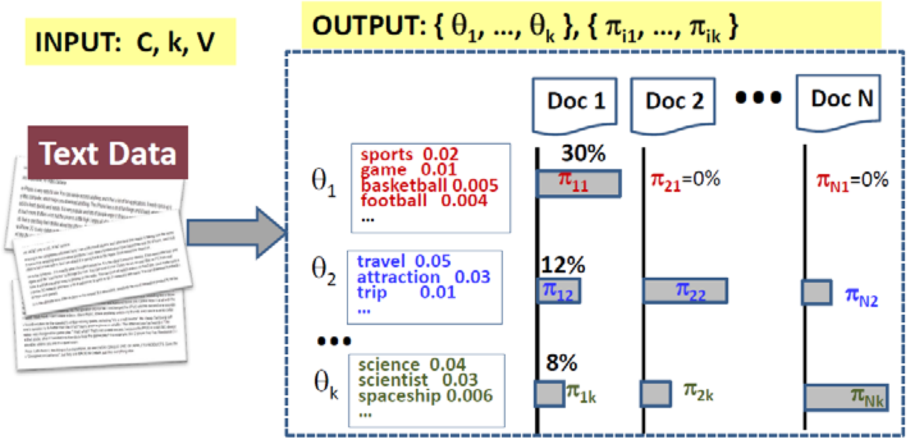
A **probabilistic topic model** can do all of these!

Probabilistic Topic Mining and Analysis

- Input
  - A **collection** of **N** text documents **C={d_1, …, d_N}**
  - **Vocabulary set: V={w_1, …, w_M}**
  - **Number of topics**: **k**
- Output
  - **k topics, each a word distribution: { θ_1, …, θ_k }**
  - **Coverage of topics in each d_i: { π_{i1}, …, π_{ik} }**
  - π_{ij}=prob. of d_i covering topic θ_j

$$\sum_{w \in V} p(w \mid \theta_i) = 1$$

$$\sum_{j=1}^{k} \pi_{ij} = 1$$



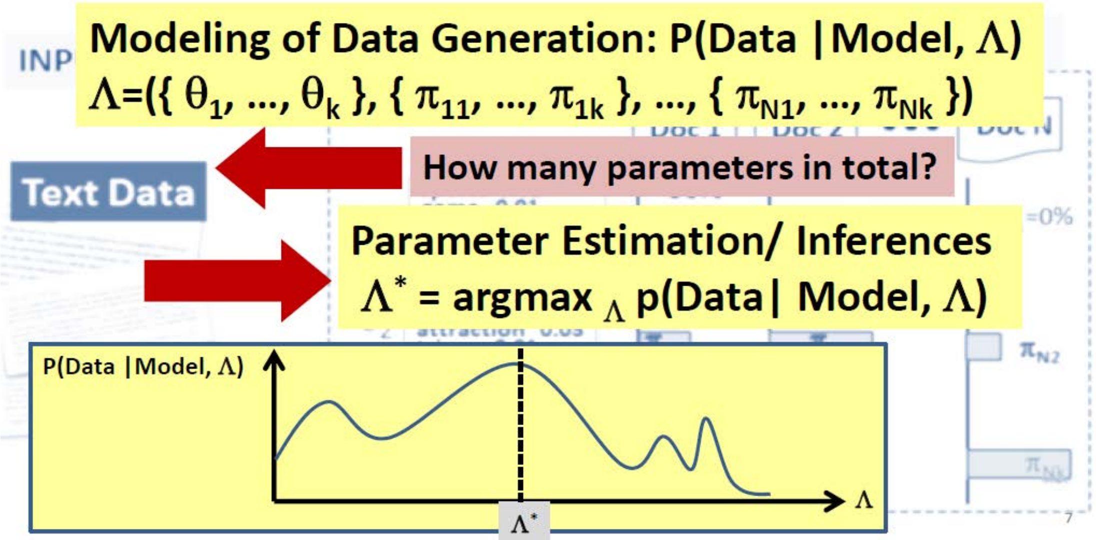
Generative Model for Text Mining

Summary

- **Topic** represented as **word distribution**
  - Multiple words: allow for describing a complicated topic
  - Weights on words: model subtle semantic variations of a topic
- Task of topic mining and analysis
  - Input: collection **C**, number of topics **k**, vocabulary set **V**
  - Output: a **set of topics**, each a word distribution; coverage of all **topics in each document**

Λ=({ θ_1, …, θ_k }, { π_{11}, …, π_{1k} }, …, { π_{N1}, …, π_{Nk} })

$$\forall j \in [1,k], \sum_{w \in V} p(w \mid \theta_j) = 1 \qquad \forall i \in [1,N], \sum_{j=1}^{k} \pi_{ij} = 1$$



- **Generative model** for text mining
  - **Model data generation** with a prob. model: **P(Data |Model, Λ)**
  - **Infer the most likely parameter values** Λ^* given a particular data set: Λ^* = **argmax** $_\Lambda$ **p(Data| Model, Λ)**
  - **Take Λ^* as the "knowledge"** to be mined for the text mining problem
  - **Adjust** the design of the model to discover different knowledge

Probabilistic Topic Models:
Overview of Statistical Language Models

Statistical Language Model (SLM)

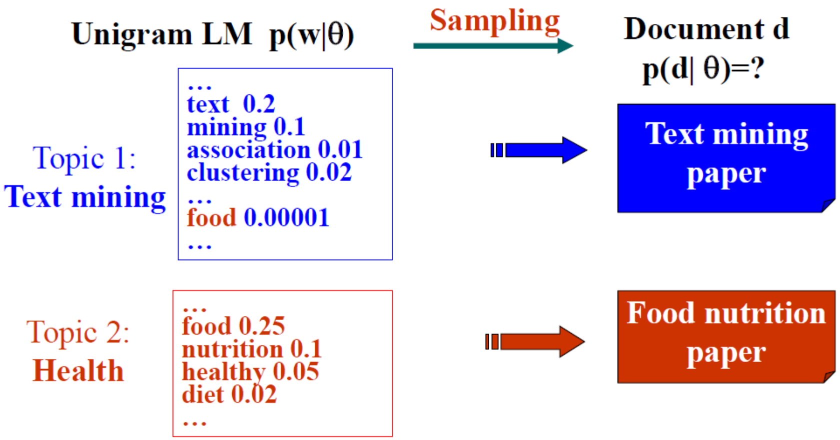
- A probability distribution over word sequences
  - p("*Today is Wednesday*") ≈ 0.001
  - p("*Today Wednesday is*") ≈ 0.0000000000001
  - p("*The eigenvalue is positive*") ≈ 0.00001
- Context-dependent!
- Probabilistic mechanism for "generating" text = **"generative" model**

The **Simplest** LM: **Unigram** LM

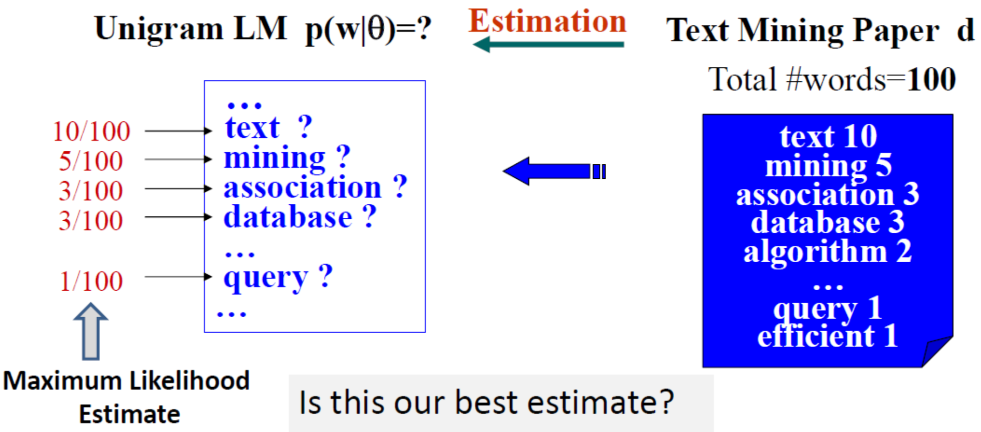
- Each word generated INDEPENDENTLY
- $p(w_1\, w_2\, \ldots\, w_n) = p(w_1)p(w_2)\ldots p(w_n)$
- Parameters: $\{p(w_i)\}$ $p(w_1)+\ldots+p(w_N)=1$ (N is voc. size)
- Text = sample drawn according to this word distribution

p("today is Wed") = p("today")p("is")p("Wed") =
= 0.0002 × 0.001 × 0.000015

Text Generation with Unigram LM



Estimation of Unigram LM



Is this our best estimate?
How do we define “best”?

Maximum Likelihood vs. Bayesian

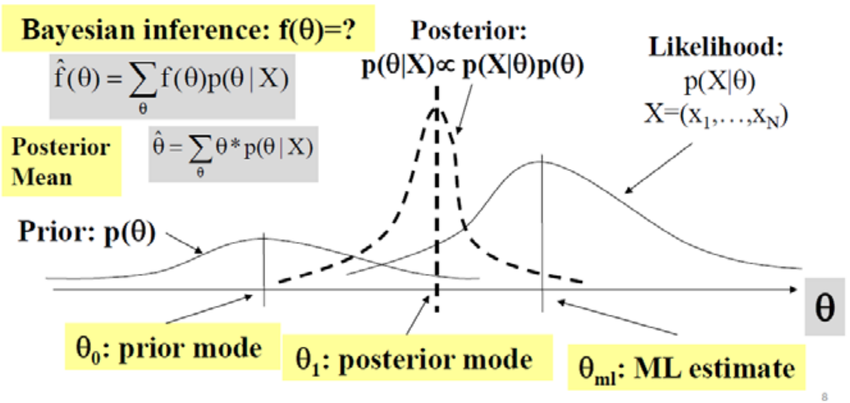
- Maximum likelihood estimation
  - "Best" – maximum data likelihood $\hat{\theta} = \arg\max_{\theta} P(X \mid \theta)$
  - Problem: Small sample
- Bayesian estimation: **Bayes Rule** $p(X \mid Y) = \frac{p(Y \mid X)p(X)}{p(Y)}$
  - "Best" - consistent with "prior" knowledge and explaining data well

$$\hat{\theta} = \arg\max_{\theta} P(\theta \mid X) = \arg\max_{\theta} P(X \mid \theta)P(\theta)$$

  - Problem: How to define prior?

Maximum a Posteriori (MAP) estimate

Illustration of Bayesian Estimation



Summary

- **Language Model** = probability distribution over text = generative model for text data
- **Unigram** Language Model = **word distribution**
- **Likelihood** function: **p(X|θ)**
  - **Given θ** ➔ which X has a higher likelihood?
  - **Given X** ➔ which θ maximizes p(X| θ)? **[ML estimate]**
- **Bayesian** estimation/inference
  - Must define a **prior: p(θ)**
  - **Posterior** distribution**: p(θ|X)∝ p(X|θ)p(θ)**

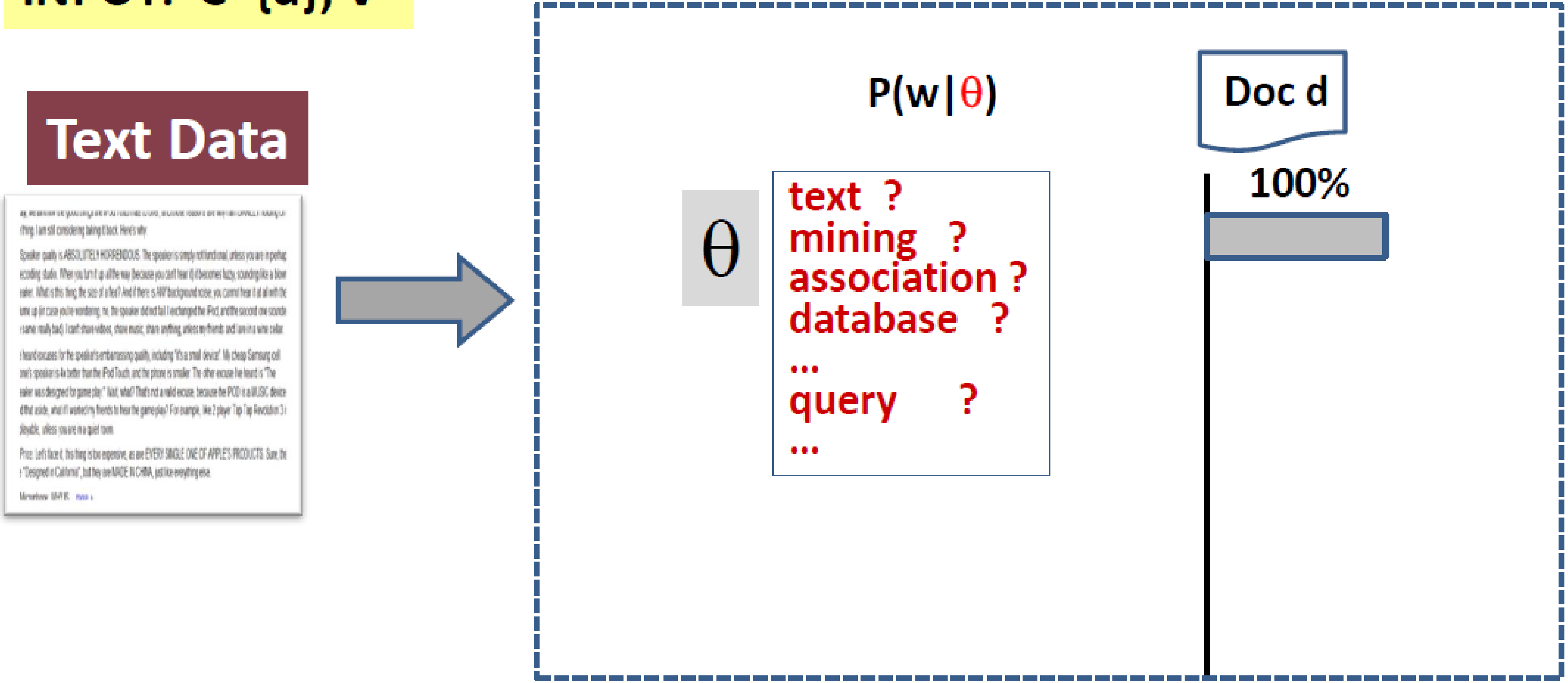
  ➔ Allows for inferring **any "derived value" from θ!**

Probabilistic Topic Models: Mining One Topic

Simplest Case of Topic Model: **Mining One Topic**

INPUT: C={d}, V

OUTPUT: { θ}



Language Model Setup

- **Data**: Document **d**= $x_1\ x_2 \ldots x_{|d|}$, $\mathbf{x_i} \in$ **V**={w_1 ,…, $\mathbf{w_M}$}is a word
- **Model**: Unigram LM θ(=topic) : {θ_i=p(w_i |θ)}, i=1, …, M; $\theta_1+\ldots+\theta_M=1$
- **Likelihood** function: $p(d|\theta) = p(x_1|\theta) \times \ldots \times p(x_{|d|}|\theta)$

$$= p(w_1|\theta)^{c(w_1,d)} \times \ldots \times p(w_M|\theta)^{c(w_M,d)}$$

$$= \prod_{i=1}^{M} p(w_i|\theta)^{c(w_i,d)} = \prod_{i=1}^{M} \theta_i^{c(w_i,d)}$$

- ML **estimate:** $(\hat{\theta}_1,\ldots,\hat{\theta}_M) = \arg\max_{\theta_1,\ldots,\theta_M} p(d|\theta) = \arg\max_{\theta_1,\ldots,\theta_M} \prod_{i=1}^{M} \theta_i^{c(w_i,d)}$

Computation of Maximum Likelihood Estimate

Maximize p(d|θ) $(\hat{\theta}_1,\ldots,\hat{\theta}_M) = \arg\max_{\theta_1,\ldots,\theta_M} p(d|\theta) = \arg\max_{\theta_1,\ldots,\theta_M} \prod_{i=1}^{M} \theta_i^{c(w_i,d)}$

Max. Log-Likelihood $(\hat{\theta}_1,\ldots,\hat{\theta}_M) = \arg\max_{\theta_1,\ldots,\theta_M} \log[p(d|\theta)] = \arg\max_{\theta_1,\ldots,\theta_M} \sum_{i=1}^{M} c(w_i,d)\log\theta_i$

Subject to constraint: $\sum_{i=1}^{M}\theta_i = 1$

Use Lagrange multiplier approach

Lagrange function: $f(\theta|d) = \sum_{i=1}^{M} c(w_i,d)\log\theta_i + \lambda(\sum_{i=1}^{M}\theta_i - 1)$

Normalized Counts

$$\frac{\partial f(\theta|d)}{\partial\theta_i} = \frac{c(w_i,d)}{\theta_i} + \lambda = 0 \rightarrow \theta_i = -\frac{c(w_i,d)}{\lambda}$$

$$\sum_{i=1}^{M} -\frac{c(w_i,d)}{\lambda} = 1 \rightarrow \lambda = -\sum_{i=1}^{N} c(w_i,d) \rightarrow \hat{\theta}_i = p(w_i|\hat{\theta}) = \frac{c(w_i,d)}{\sum_{i=1}^{M} c(w_i,d)} = \frac{c(w_i,d)}{|d|}$$

What Does the Topic Look Like?

d

Text mining paper

$p(w|\theta)$

the 0.031
a 0.018
…
text 0.04
mining 0.035
association 0.03
clustering 0.005
computer 0.0009
…
food 0.000001
…

Can we get rid of these common words?