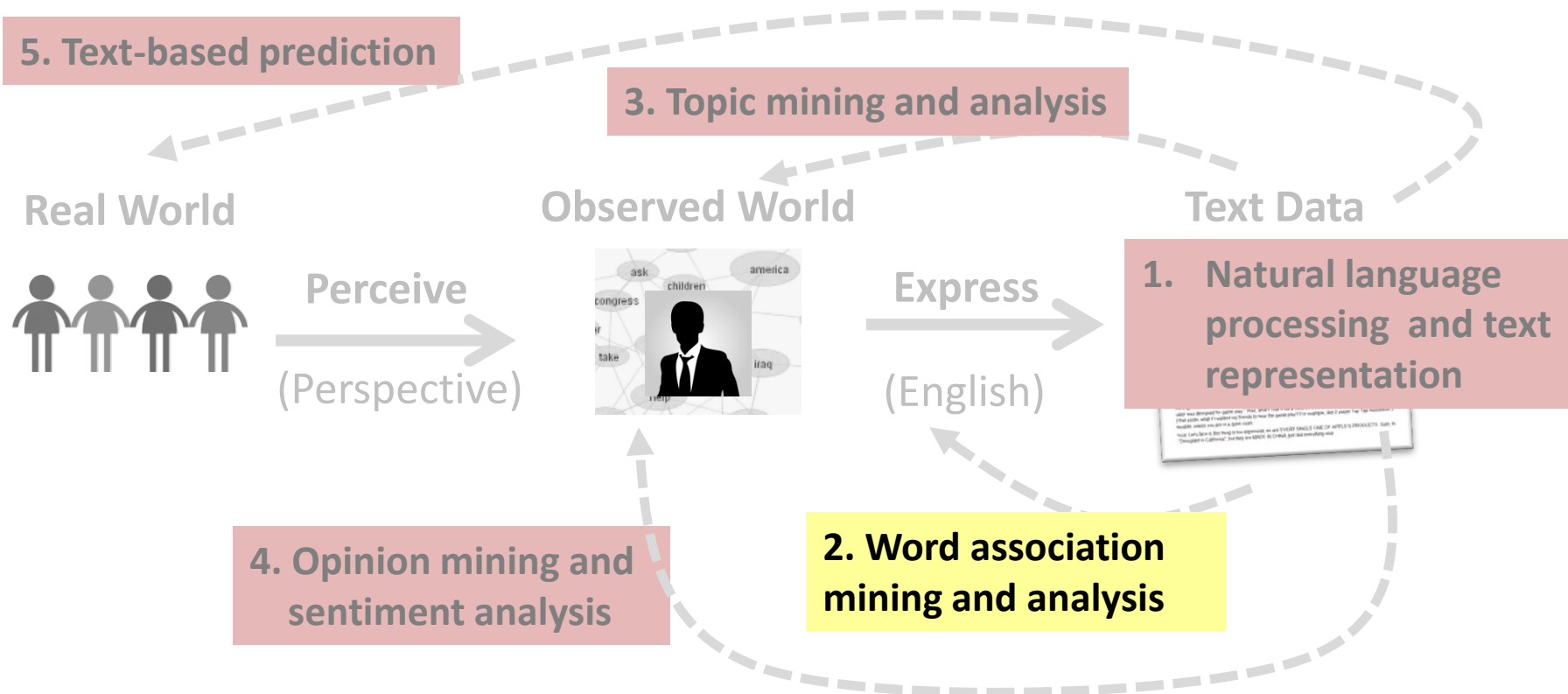


Syntagmatic Relation Discovery: Entropy

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
...				

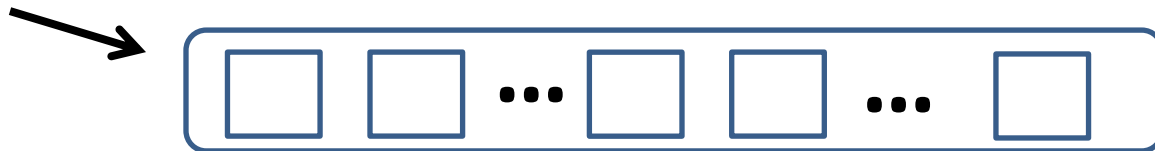
What words tend to occur
to the **left** of “**eats**”?

What words
are to the
right?

Word Prediction: Intuition

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

Text Segment (any unit, e.g., sentence, paragraph, document)



Are some words easier to predict than others?

1) $W = \text{"meat"}$

2) $W = \text{"the"}$

3) $W = \text{"unicorn"}$

Word Prediction: Formal Definition

Binary Random Variable : $X_w = \begin{cases} 1 & \text{w is present} \\ 0 & \text{w is absent} \end{cases}$
 $X_w \in \{0, 1\}$

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

The more random X_w is, the more difficult the prediction would be.

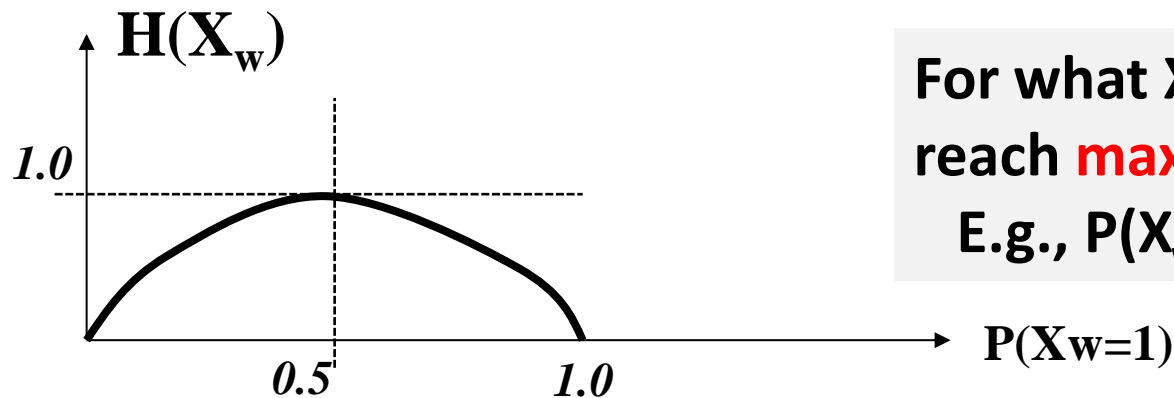
How does one quantitatively measure the “randomness” of a random variable like X_w ?

Entropy $H(X)$ Measures Randomness of X

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

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

$$= -p(X_w = 0) \log_2 p(X_w = 0) - p(X_w = 1) \log_2 p(X_w = 1) \quad \text{Define } 0 \log_2 0 = 0$$



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

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

or equivalently $P(X_w=0)$ (Why?)

Entropy $H(X)$: Coin Tossing

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

X_{coin} : tossing a coin

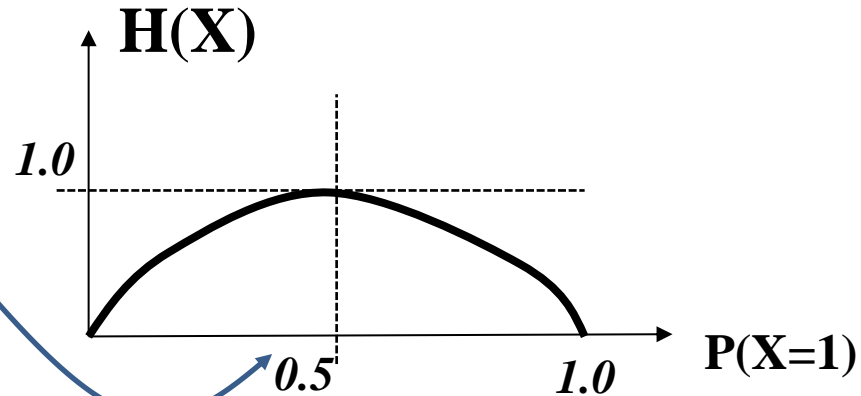
$$X_{\text{coin}} = \begin{cases} 1 & \text{Head} \\ 0 & \text{Tail} \end{cases}$$

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) $W = \text{"meat"}$

2) $W = \text{"the"}$

3) $W = \text{"unicorn"}$

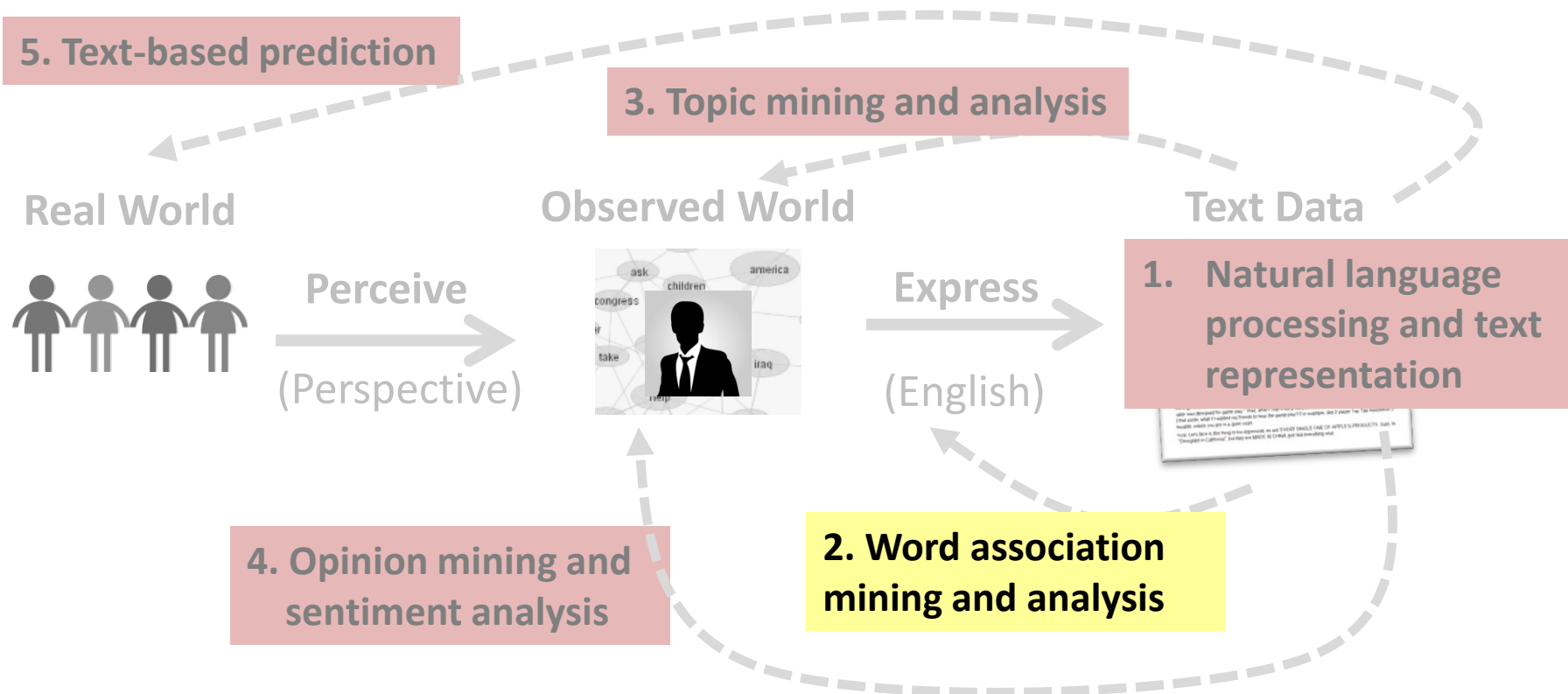
Which is **high/low**? $H(X_{\text{meat}})$, $H(X_{\text{the}})$, or $H(X_{\text{unicorn}})$?

$H(X_{\text{the}}) \approx 0 \rightarrow$ no uncertainty since $p(X_{\text{the}}=1) \approx 1$

High entropy words are harder to predict!

Syntagmatic Relation Discovery: Conditional Entropy

Syntagmatic Relation Discovery: Conditional Entropy



What If We Know More About a Text Segment?

Prediction question: Is “**meat**” present (or absent) in this segment?



Does presence of “**eats**” help predict the presence of “**meat**”?

Does it **reduce** the uncertainty about “meat”, i.e., $H(X_{\text{meat}})$?

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

Conditional Entropy

Know nothing about the segment

Know “eats” is present ($X_{eats} = 1$)

$$p(X_{meat} = 1) \quad \text{-----} \rightarrow \quad p(X_{meat} = 1 \mid X_{eats} = 1)$$

$$p(X_{meat} = 0) \quad \text{-----} \rightarrow \quad p(X_{meat} = 0 \mid X_{eats} = 1)$$

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



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

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

Conditional Entropy: Complete Definition

$$\begin{aligned} H(X_{meat} / X_{eats}) &= \sum_{u \in \{0,1\}} [p(X_{eats} = u) H(X_{meat} | X_{eats} = u)] \\ &= \sum_{u \in \{0,1\}} [p(X_{eats} = u) \sum_{v \in \{0,1\}} [-p(X_{meat} = v | X_{eats} = u) \log_2 p(X_{meat} = v | X_{eats} = u)]] \end{aligned}$$

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

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

Conditional Entropy to Capture Syntagmatic Relation

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

$$H(X_{meat} | X_{meat}) = ?$$

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, $H(X_{meat})$?

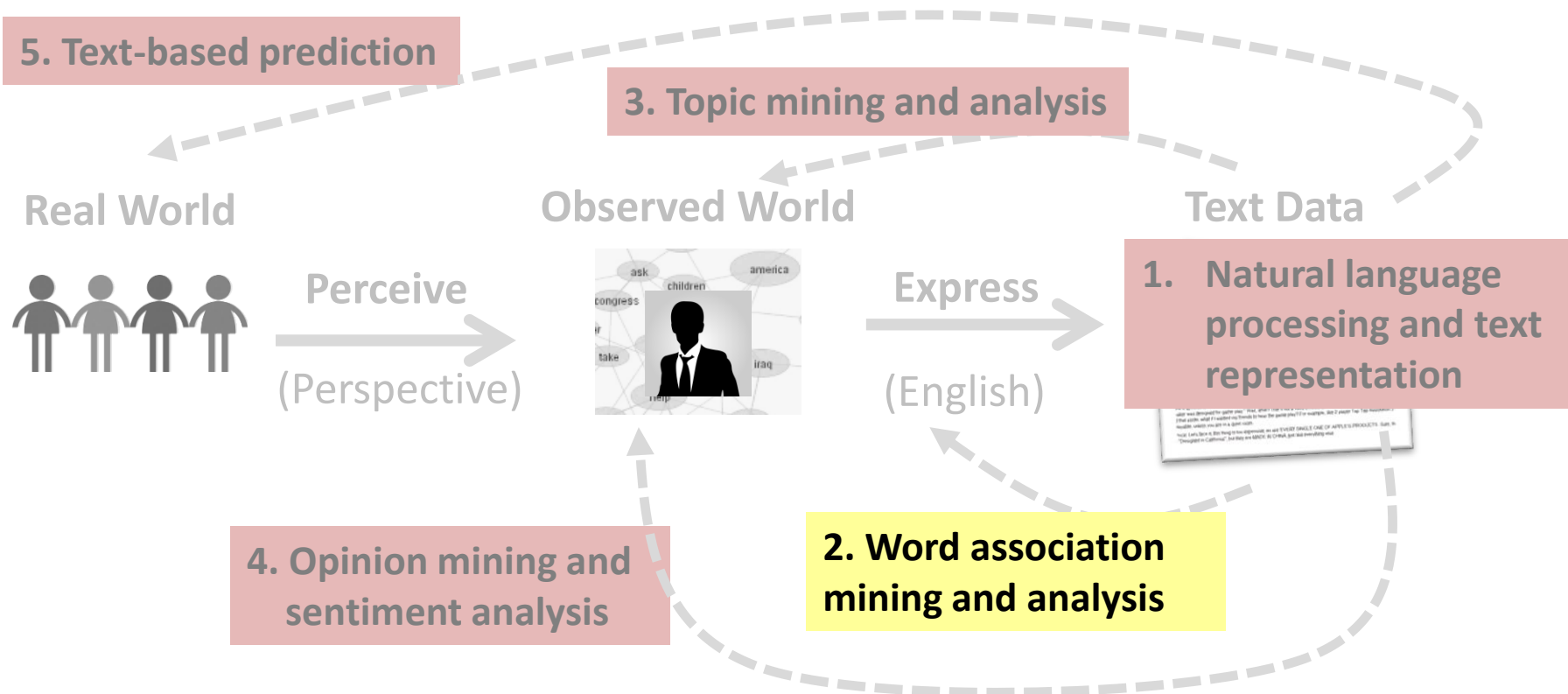
Conditional Entropy for Mining Syntagmatic Relations

- For each word W_1
 - For every other word W_2 , compute conditional entropy $H(X_{W_1} | X_{W_2})$
 - Sort all the candidate words in ascending order of $H(X_{W_1} | X_{W_2})$
 - Take the top-ranked candidate words as words that have potential syntagmatic relations with W_1
 - Need to use a threshold for each W_1
- However, while $H(X_{W_1} | X_{W_2})$ and $H(X_{W_1} | X_{W_3})$ are comparable, $H(X_{W_1} | X_{W_2})$ and $H(X_{W_3} | X_{W_2})$ aren't!

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

Syntagmatic Relation Discovery: Mutual Information

Syntagmatic Relation Discovery: Mutual Information



Mutual Information $I(X;Y)$: Measuring Entropy Reduction

How much reduction in the entropy of X can we obtain by knowing Y ?

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

When we fix X to rank different Y s, $I(X;Y)$ and $H(X|Y)$ give the same order but $I(X;Y)$ allows us to compare different (X,Y) pairs.

Mutual Information $I(X;Y)$ for Syntagmatic Relation Mining

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

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

Which **words** have high mutual information with “**eats**”?

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

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

Rewriting Mutual Information (MI) Using KL-divergence

The observed 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 expected 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 from the expected distribution under the independence assumption. The larger the divergence is, the higher the MI would be.

Probabilities Involved in Mutual Information

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

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:

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



Both w1 & w2 occur



Only w1 occurs



Only w2 occurs



None of them occurs

Relations Between Different Probabilities

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:

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

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 w_1 :

$$p(X_{w_1}=1) + p(X_{w_1}=0) = 1$$

Presence & absence of w_2 :

$$p(X_{w_2}=1) + p(X_{w_2}=0) = 1$$

Co-occurrences of w_1 and w_2 :

$$p(X_{w_1}=1, X_{w_2}=1) + p(X_{w_1}=1, X_{w_2}=0) + p(X_{w_1}=0, X_{w_2}=1) + p(X_{w_1}=0, X_{w_2}=0) = 1$$

$$p(X_{w_1}=1, X_{w_2}=1) + p(X_{w_1}=1, X_{w_2}=0) = p(X_{w_1}=1)$$

$$p(X_{w_1}=0, X_{w_2}=1) + p(X_{w_1}=0, X_{w_2}=0) = p(X_{w_1}=0)$$

$$p(X_{w_1}=1, X_{w_2}=1) + p(X_{w_1}=0, X_{w_2}=1) = p(X_{w_2}=1)$$

$$p(X_{w_1}=1, X_{w_2}=0) + p(X_{w_1}=0, X_{w_2}=0) = p(X_{w_2}=0)$$

We only need to know $p(X_{w_1}=1)$, $p(X_{w_2}=1)$, and $p(X_{w_1}=1, X_{w_2}=1)$.

Estimation of Probabilities (Depending on the Data)

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

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

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

	<u>W1</u>	<u>W2</u>	
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

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

Segment_1	1	0
...		
Segment_N	0	1

Actually observed 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

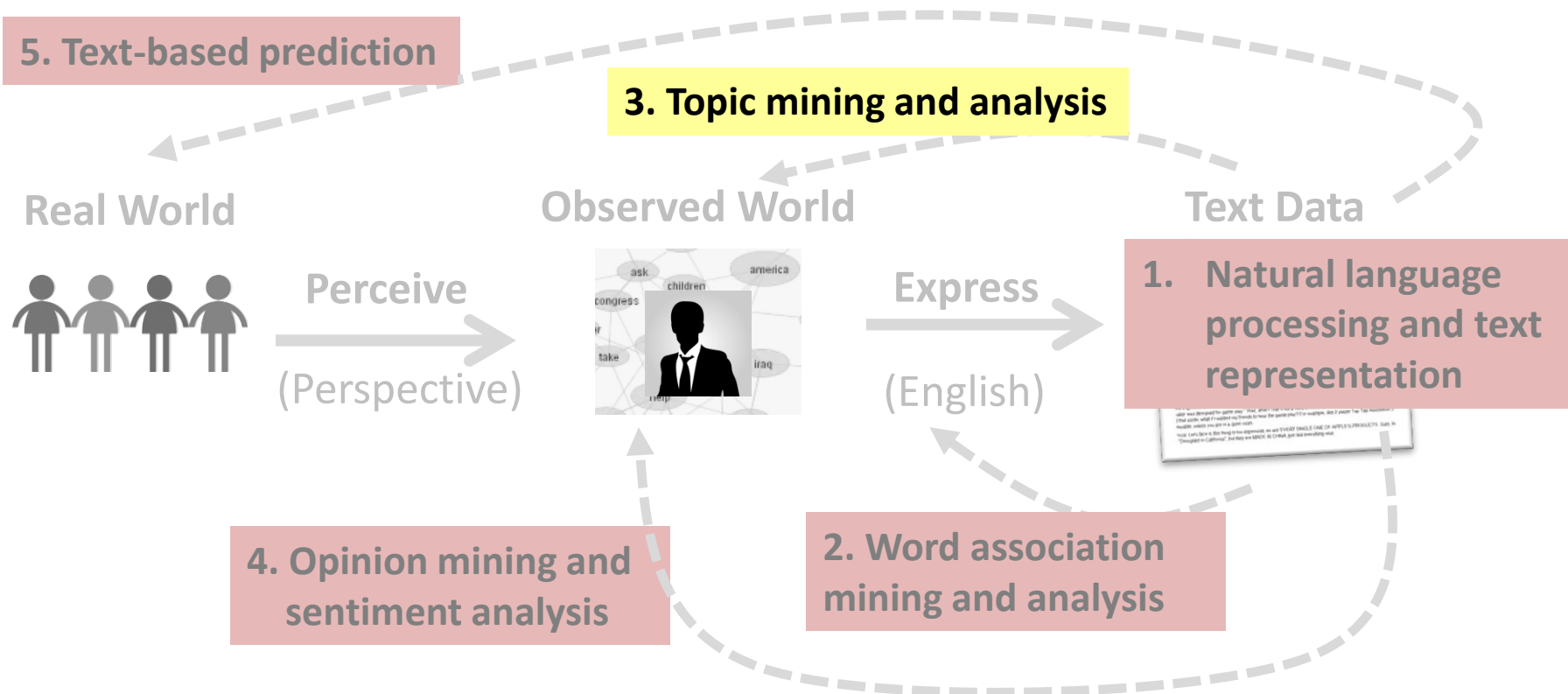
- Two basic associations: paradigmatic and syntagmatic
 - Generally applicable to any items in any language (e.g., phrases or entities as units)
- Pure statistical approaches are available for discovering both (can be combined to perform joint analysis).
 - Generally applicable to any text with no human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

Additional Reading

- Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press. Cambridge, MA: May 1999. (Chapter 5 on collocations)
- Chengxiang Zhai, Exploiting context to identify lexical atoms: A statistical view of linguistic context. Proceedings of the International and Interdisciplinary Conference on Modelling and Using Context (CONTEXT-97), Rio de Janeiro, Brzil, Feb. 4-6, 1997. pp. 119-129.
- Shan Jiang and ChengXiang Zhai, Random walks on adjacency graphs for mining lexical relations from big text data. Proceedings of IEEE BigData Conference 2014, pp. 549-554.

Topic Mining and Analysis: Motivation and Task Definition

Topic Mining and Analysis: Motivation and Task Definition



Topic Mining and Analysis: Motivation

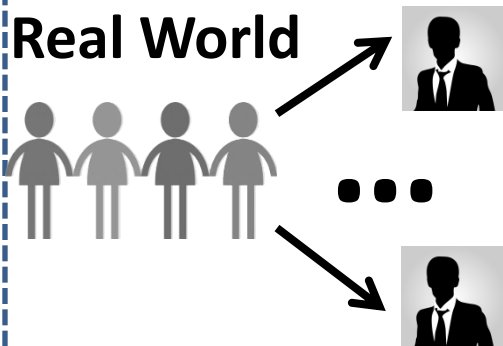
- Topic \approx main idea discussed in text data
 - Theme/subject of a discussion or conversation
 - Different granularities (e.g., topic of a sentence, an article, etc.)
- Many applications require discovery of topics in text
 - What are Twitter users talking about today?
 - What are the current research topics in data mining? How are they different from those 5 years ago?
 - What do people like about the iPhone 6? What do they dislike?
 - What were the major topics debated in 2012 presidential election?

Topics As Knowledge About the World

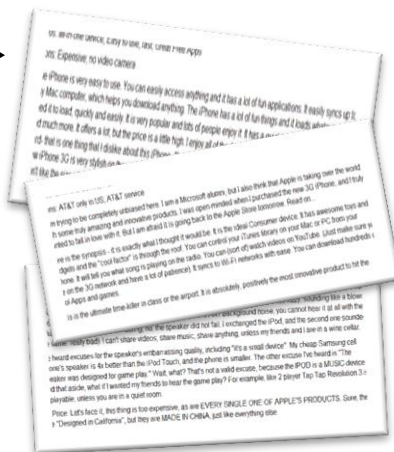
Knowledge about the world

Non-Text Data

Real World



Text Data



+ Context
Time
Location
...

Topic 1

Topic 2

...

Topic k

Tasks of Topic Mining and Analysis

Task 2: Figure out which documents cover which topics

Text Data



Topic 1

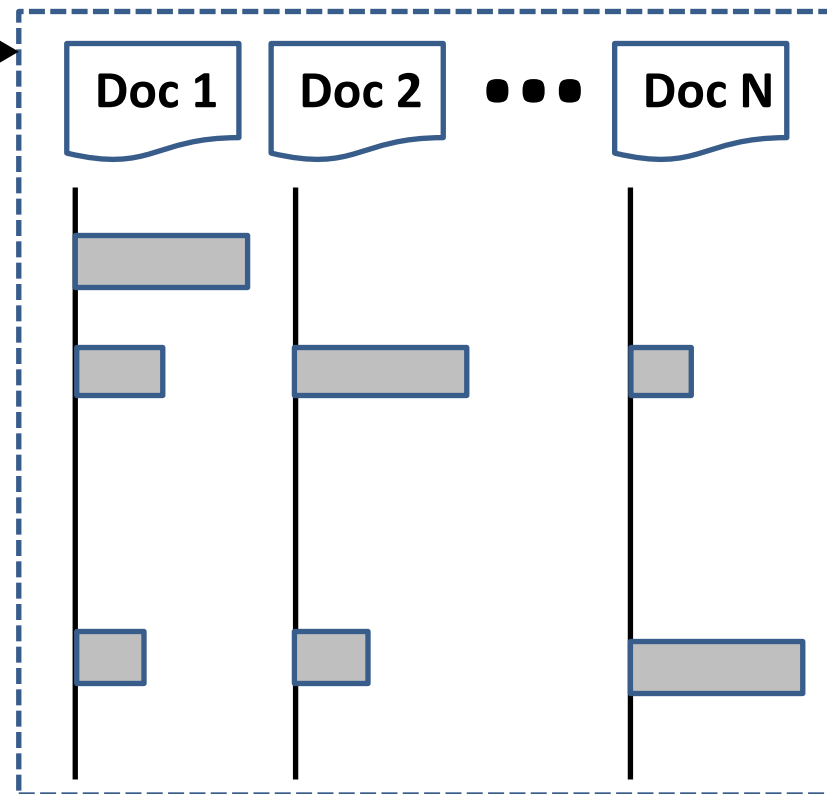
Topic 2

...

Topic k



Task 1: Discover k topics



Formal Definition of Topic Mining and Analysis

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

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

How to define θ_i ?

Topic Mining and Analysis: Term as Topic

Formal Definition of Topic Mining and Analysis

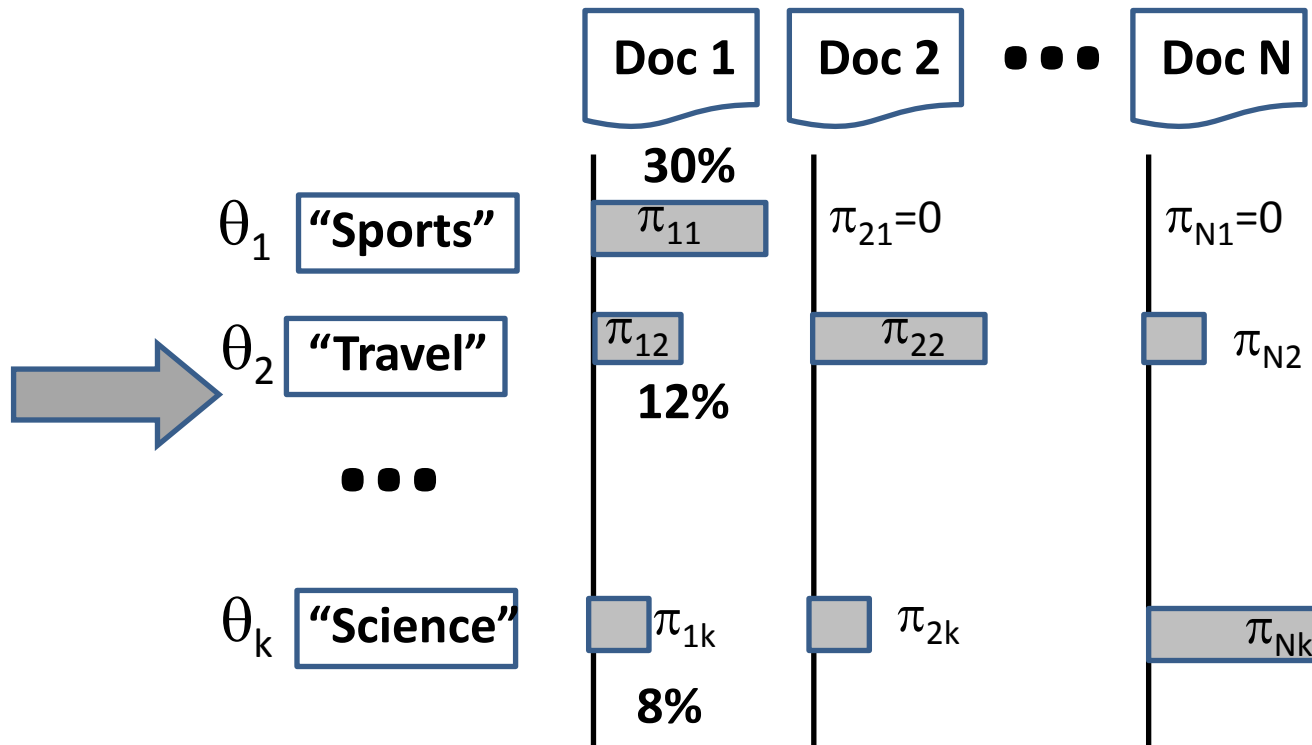
- Input
 - A **collection** of **N** text documents **$C=\{d_1, \dots, d_N\}$**
 - **Number of topics: k**
- Output
 - **k topics: $\{\theta_1, \dots, \theta_k\}$**
 - **Coverage of topics in each d_i : $\{\pi_{i1}, \dots, \pi_{ik}\}$**
 - π_{ij} =prob. of d_i covering topic θ_j

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

How to define θ_i ?

Initial Idea: Topic = Term

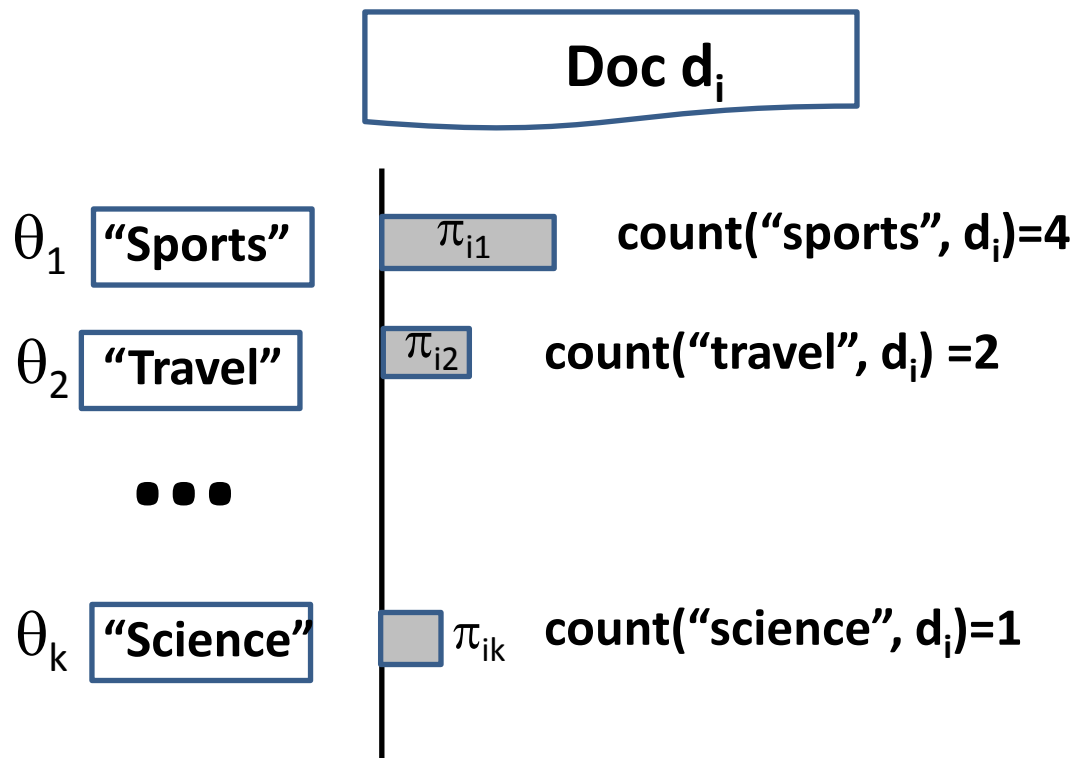
Text Data



Mining k Topical Terms from Collection C

- Parse text in C to obtain candidate terms (e.g., term = word).
- Design a scoring function to measure how good each term is as a topic.
 - Favor a representative term (high frequency is favored)
 - Avoid words that are too frequent (e.g., “the”, “a”).
 - TF-IDF weighting from retrieval can be very useful.
 - Domain-specific heuristics are possible (e.g., favor title words, hashtags in tweets).
- Pick k terms with the highest scores but try to minimize redundancy.
 - If multiple terms are very similar or closely related, pick only one of them and ignore others.

Computing Topic Coverage: π_{ij}



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

How Well Does This Approach Work?

Doc d_i

Cavaliers vs. Golden State Warriors: NBA playoff finals ...
basketball game ... **travel** to Cleveland ... **star** ...

θ_1 **"Sports"**

$$\pi_{i1} \propto c(\text{"sports"}, d_i) = 0$$

1. Need to count
related words also!

θ_2 **"Travel"**

$$\pi_{i2} \propto c(\text{"travel"}, d_i) = 1 > 0$$

...

2. "Star" can be ambiguous (e.g., star in the sky).

θ_k **"Science"**

$$\pi_{ik} \propto c(\text{"science"}, d_i) = 0$$

3. Mine complicated topics?

Problems with “Term as Topic”

- Lack of expressive power
 - Can only represent simple/general topics
 - Can’t represent complicated topics
- Incompleteness in vocabulary coverage
 - Can’t capture variations of vocabulary (e.g., related words)
- Word sense ambiguity
 - A topical term or related term can be ambiguous (e.g., basketball star vs. star in the sky)

Topic Mining and Analysis: Probabilistic Topic Models

5. Text-based prediction

3. Topic mining and analysis

1. Natural language processing and text representation

4. Opinion mining and sentiment analysis

2. Word association mining and analysis

Problems with “Term as Topic”

- Lack of expressive power → **Topic = {Multiple Words}**
 - Can only represent simple/general topics
 - Can't represent complicated topics
- Incompleteness in vocabulary coverage **+ weights on words**
 - Can't capture variations of vocabulary (e.g., related words)
- Word sense ambiguity → **Split an ambiguous word**
 - A topical term or related term can be ambiguous (e.g., basketball star vs. star in the sky)

A probabilistic topic model can do all these!

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: $V=\{w_1, w_2, \dots\}$

Probabilistic Topic Mining and Analysis

- Input

- A collection of **N** text documents **$C=\{d_1, \dots, d_N\}$**
- Vocabulary set: **$V=\{w_1, \dots, w_M\}$**
- Number of topics: **k**

- Output

- k topics, each a word distribution: **$\{ \theta_1, \dots, \theta_k \}$**
- Coverage of topics in each d_i : **$\{ \pi_{i1}, \dots, \pi_{ik} \}$**
- π_{ij} =prob. of d_i covering topic θ_j

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

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

The Computation Task

INPUT: C, k, V

OUTPUT: $\{ \theta_1, \dots, \theta_k \}, \{ \pi_{i1}, \dots, \pi_{ik} \}$

Text Data

θ_1

sports 0.02
game 0.01
basketball 0.005
football 0.004
...

θ_2

travel 0.05
attraction 0.03
trip 0.01
...

...

θ_k

science 0.04
scientist 0.03
spaceship 0.006
...

Doc 1

30%

π_{11}

Doc 2

$\pi_{21}=0\%$

...

Doc N

$\pi_{N1}=0\%$

12%

π_{12}

π_{22}

π_{N2}

8%

π_{1k}

π_{2k}

π_{Nk}

Generative Model for Text Mining

Modeling of Data Generation: $P(\text{Data} \mid \text{Model}, \Lambda)$

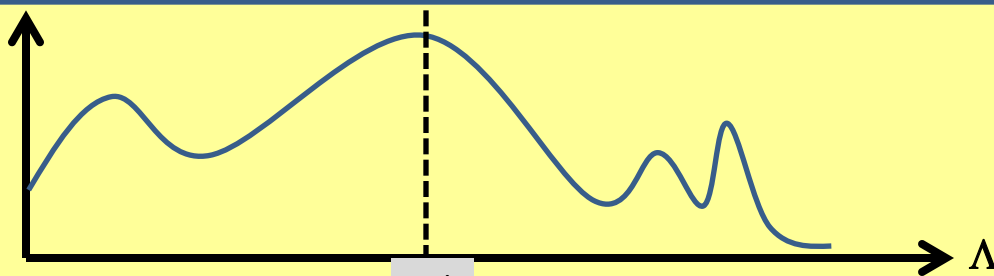
$$\Lambda = (\{ \theta_1, \dots, \theta_k \}, \{ \pi_{11}, \dots, \pi_{1k} \}, \dots, \{ \pi_{N1}, \dots, \pi_{Nk} \})$$

How many parameters in total?

Parameter Estimation/ Inferences

$$\Lambda^* = \operatorname{argmax}_{\Lambda} p(\text{Data} \mid \text{Model}, \Lambda)$$

$P(\text{Data} \mid \text{Model}, \Lambda)$



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

$$\Lambda = (\{ \theta_1, \dots, \theta_k \}, \{ \pi_{11}, \dots, \pi_{1k} \}, \dots, \{ \pi_{N1}, \dots, \pi_{Nk} \})$$

$$\forall j \in [1, k], \sum_{w \in V} p(w | \theta_j) = 1$$

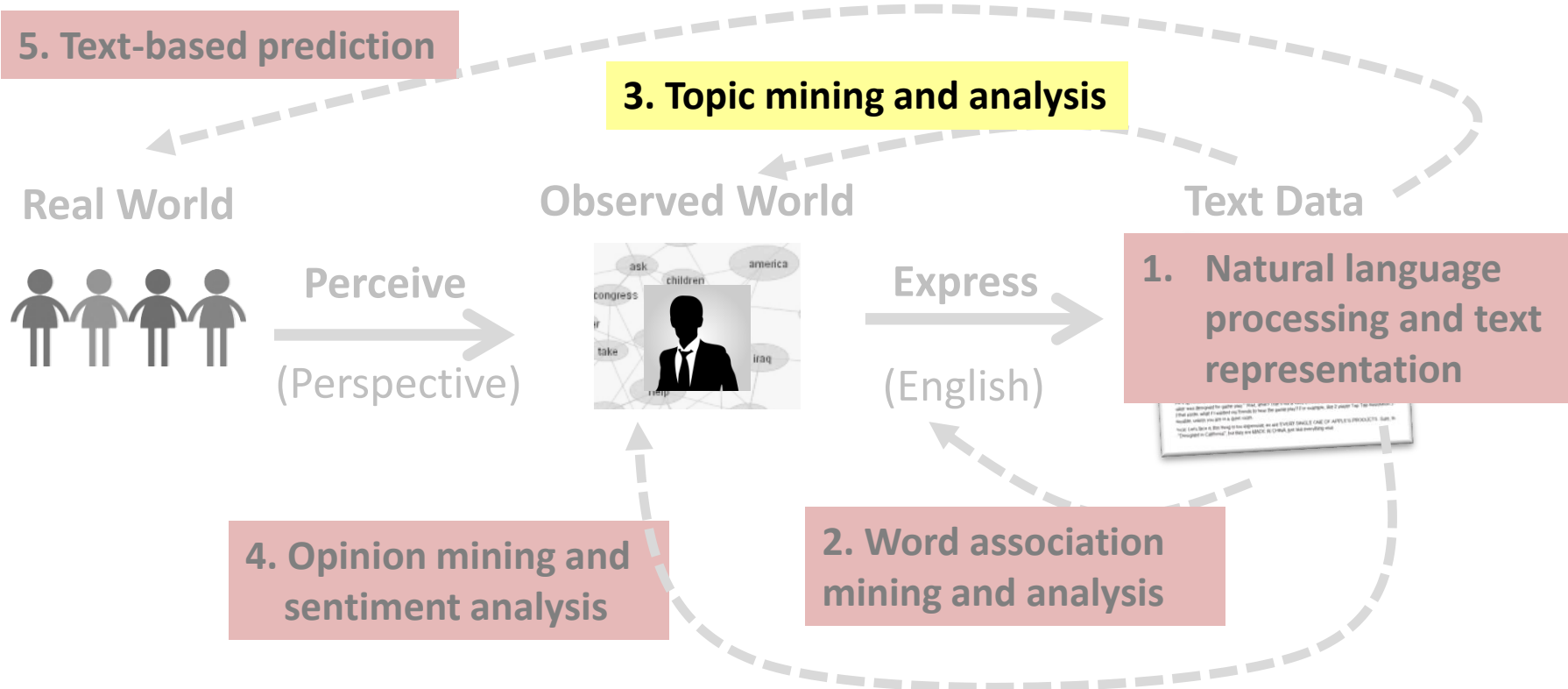
$$\forall i \in [1, N], \sum_{j=1}^k \pi_{ij} = 1$$

Summary (cont.)

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

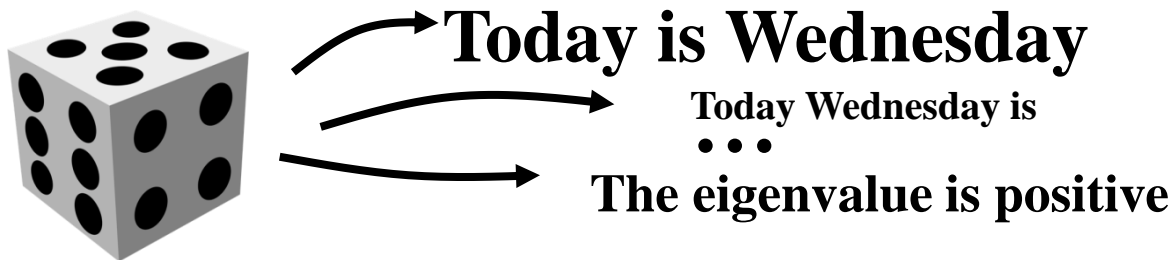
Probabilistic Topic Models: Overview of Statistical Language Models

Probabilistic Topic Models: Overview of Statistical Language Models



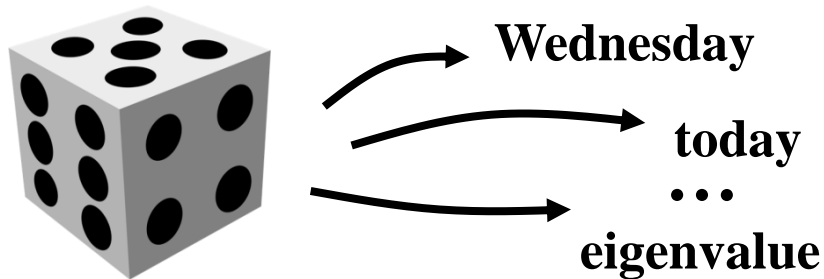
What Is a Statistical Language Model (LM)?

- A probability distribution over word sequences
 - $p(\textit{“Today is Wednesday”}) \approx 0.001$
 - $p(\textit{“Today Wednesday is”}) \approx 0.0000000000000001$
 - $p(\textit{“The eigenvalue is positive”}) \approx 0.000001$
- Context-dependent!
- Can also be regarded as a probabilistic mechanism for “generating” text – thus also called a “generative” model



The Simplest Language Model: Unigram LM

- Generate text by generating each word INDEPENDENTLY
- Thus, $p(w_1 w_2 \dots w_n) = p(w_1)p(w_2)\dots p(w_n)$
- Parameters: $\{p(w_i)\}$ $p(w_1) + \dots + p(w_N) = 1$ (N is voc. size)
- Text = sample drawn according to this **word distribution**



$$\begin{aligned} p(\text{"today is Wed"}) \\ &= p(\text{"today"})p(\text{"is"})p(\text{"Wed"}) \\ &= 0.0002 \times 0.001 \times 0.000015 \end{aligned}$$

Text Generation with Unigram LM

Unigram LM $p(w|\theta)$

Sampling



Document d

$p(d|\theta)=?$

Topic 1:
Text mining

...
text 0.2
mining 0.1
association 0.01
clustering 0.02
...
food 0.00001
...



**Text mining
paper**

Topic 2:
Health

...
food 0.25
nutrition 0.1
healthy 0.05
diet 0.02
...

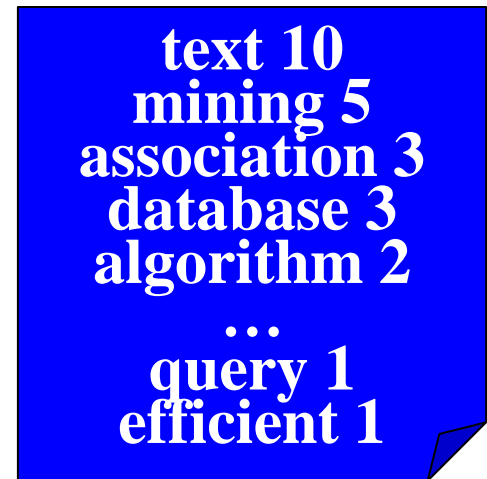
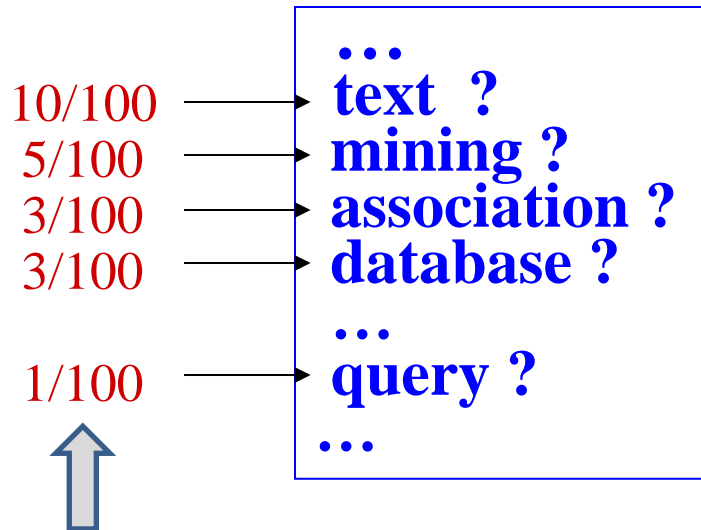


**Food nutrition
paper**

Estimation of Unigram LM

Unigram LM $p(w|\theta)=?$ **Estimation** ← Text Mining Paper d

Total #words=100



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

Maximum Likelihood vs. Bayesian

- Maximum likelihood estimation

- “Best” means “data likelihood reaches maximum”

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

- Problem: Small sample

- Bayesian estimation:

Bayes Rule

$$p(\mathbf{X} | \mathbf{Y}) = \frac{p(\mathbf{Y} | \mathbf{X})p(\mathbf{X})}{p(\mathbf{Y})}$$

- “Best” means being consistent with our “prior” knowledge and explaining data well

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

- Problem: How to define prior?



Maximum a Posteriori (MAP) estimate

Illustration of Bayesian Estimation

Bayesian inference: $f(\theta)=?$

$$\hat{f}(\theta) = \sum_{\theta} f(\theta) p(\theta | X)$$

**Posterior
Mean**

$$\hat{\theta} = \sum_{\theta} \theta^* p(\theta | X)$$

Posterior:

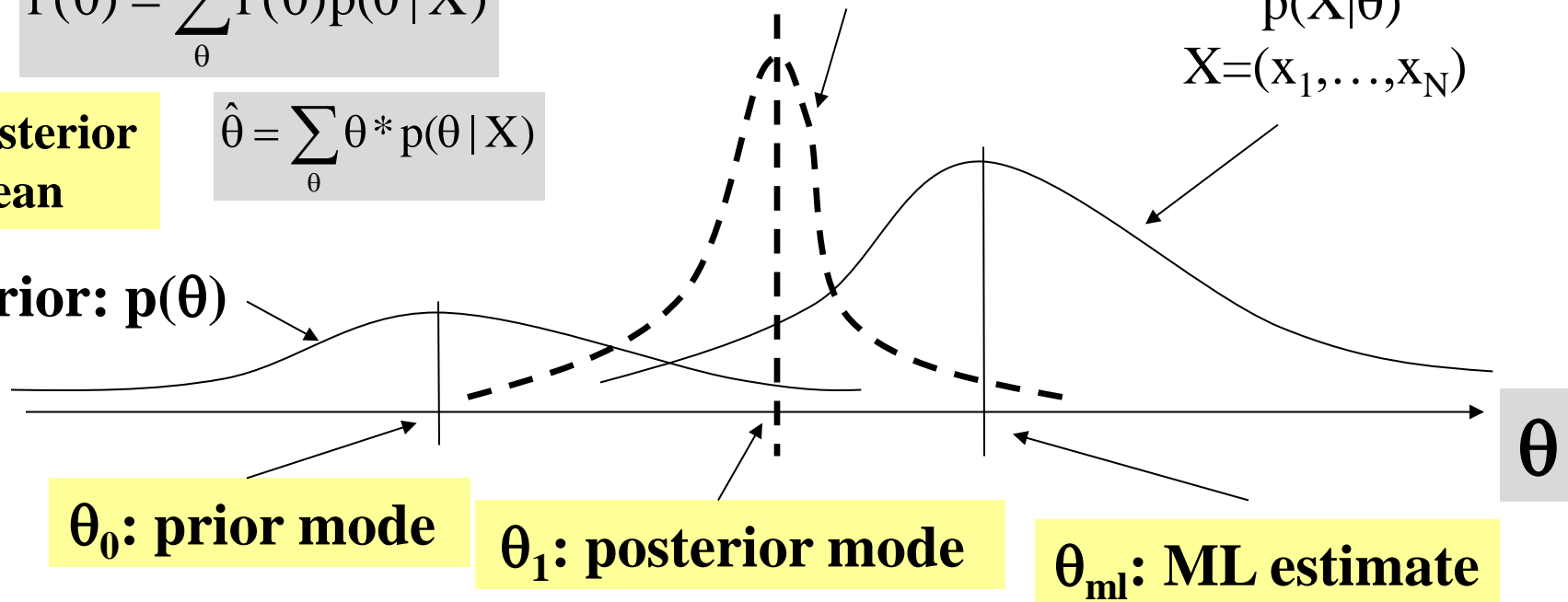
$$p(\theta|X) \propto p(X|\theta)p(\theta)$$

Likelihood:

$$p(X|\theta)$$

$$X=(x_1, \dots, x_N)$$

Prior: $p(\theta)$

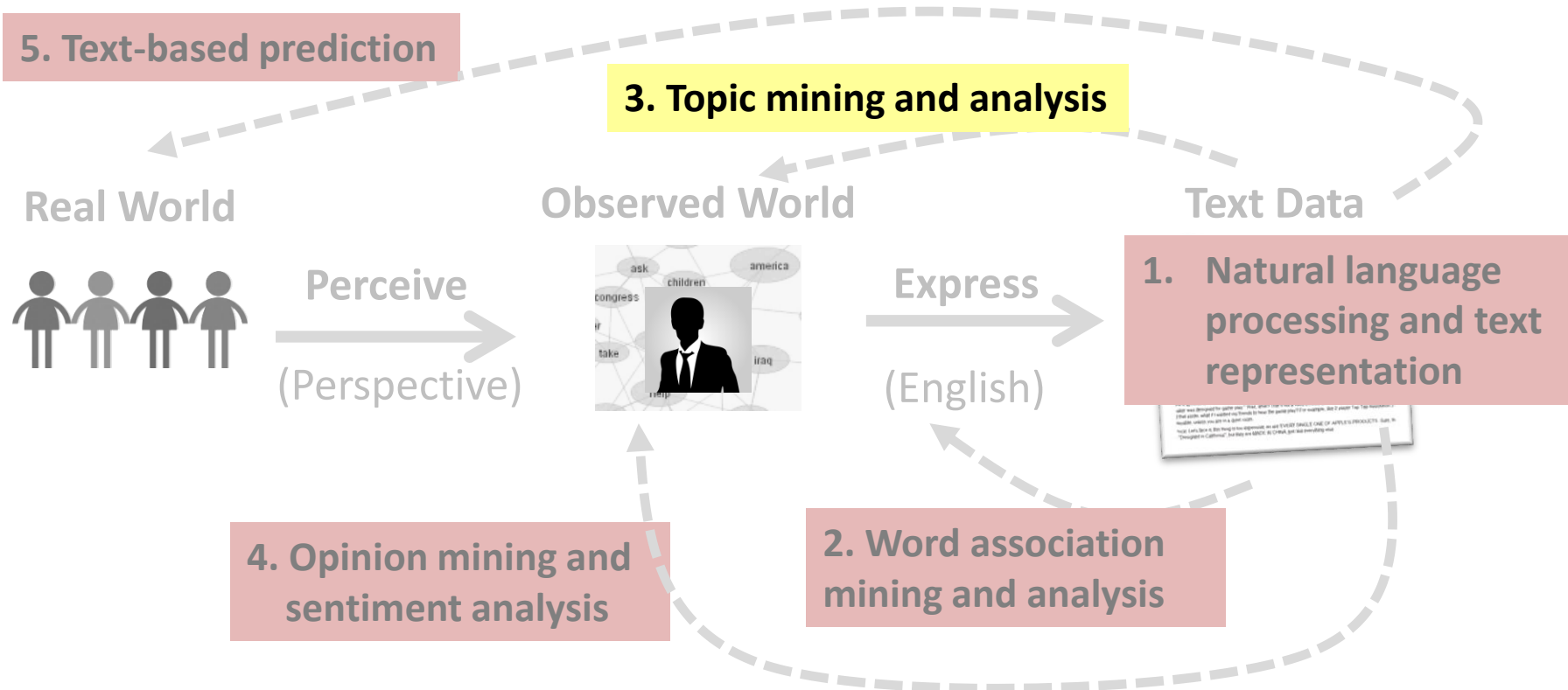


Summary

- **Language Model** = probability distribution over text = generative model for text data
- **Unigram Language Model** = **word distribution**
- **Likelihood** function: $p(\mathbf{X}|\theta)$
 - **Given $\theta \rightarrow$** which \mathbf{X} has a higher likelihood?
 - **Given $\mathbf{X} \rightarrow$** which θ maximizes $p(\mathbf{X}|\theta)$? [**ML estimate**]
- **Bayesian** estimation/inference
 - Must define a **prior**: $p(\theta)$
 - **Posterior** distribution: $p(\theta|\mathbf{X}) \propto p(\mathbf{X}|\theta)p(\theta)$
 - \rightarrow Allows for inferring any “derived value” from θ !**

Probabilistic Topic Models: Mining One Topic

Probabilistic Topic Models: Mining One Topic

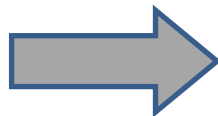


Simplest Case of Topic Model: Mining One Topic

INPUT: $C=\{d\}, V$

OUTPUT: $\{\theta\}$

Text Data



$P(w|\theta)$

θ

text ?
mining ?
association ?
database ?
...
query ?
...

Doc d

100%



Language Model Setup

- **Data:** Document $d = x_1 x_2 \dots x_{|d|}$, $x_i \in V = \{w_1, \dots, w_M\}$ is a word
- **Model:** Unigram LM θ (=topic) : $\{\theta_i = p(w_i | \theta)\}$, $i=1, \dots, M$;
 $\theta_1 + \dots + \theta_M = 1$
- **Likelihood function:** $p(d | \theta) = p(x_1 | \theta) \times \dots \times p(x_{|d|} | \theta)$
$$= p(w_1 | \theta)^{c(w_1, d)} \times \dots \times p(w_M | \theta)^{c(w_M, d)}$$
$$= \prod_{i=1}^M p(w_i | \theta)^{c(w_i, d)} = \prod_{i=1}^M \theta_i^{c(w_i, d)}$$
- **ML estimate:** $(\hat{\theta}_1, \dots, \hat{\theta}_M) = \arg \max_{\theta_1, \dots, \theta_M} p(d | \theta) = \arg \max_{\theta_1, \dots, \theta_M} \prod_{i=1}^M \theta_i^{c(w_i, d)}$

Computation of Maximum Likelihood Estimate

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

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

Subject to constraint:

$$\sum_{i=1}^M \theta_i = 1$$

Use Lagrange multiplier approach

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

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

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

**Normalized
Counts**



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?