Mixture Language Models

ChengXiang Zhai

Department of Computer Science

University of Illinois, Urbana-Champaign

Central Questions to Ask about a LM: "ADMI"

- **Application**: Why do you need a LM? For what purpose?
  - → Evaluation metric for a LM
  - Topic mining and anlaysis
- **Data:** What kind of data do you want to model?
  - → Data set for estimation & evaluation
  - Text documents & context
- **Model:** How do you define the model?
  - → Assumptions to be made
  - Mixture of unigram LMs
- **Inference:** How do you infer/estimate the parameters?
  - → Inference/Estimation algorithm
  - EM algorithm, Bayesian

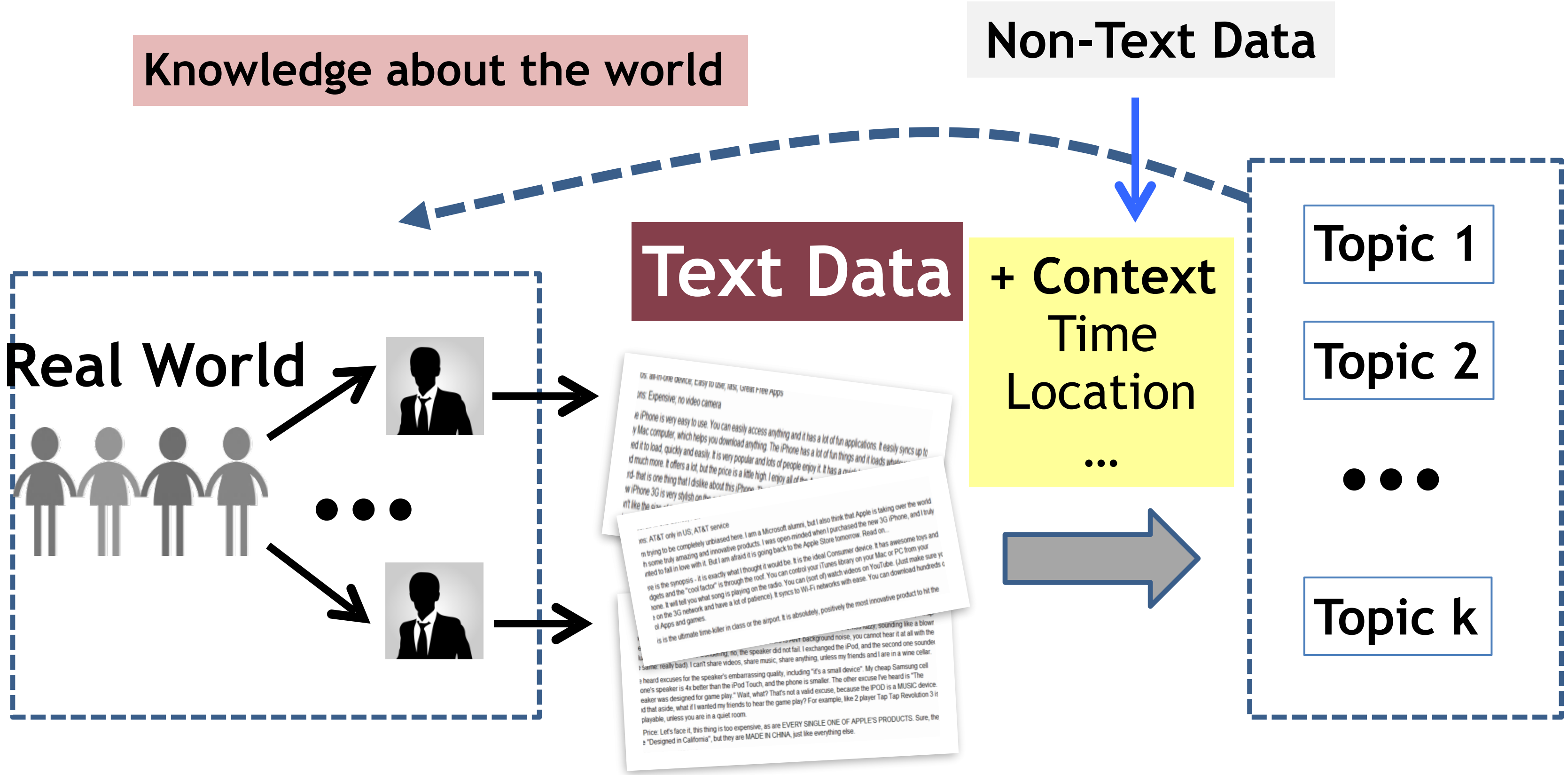
Outline

- Motivation
- Mining one topic
- Two-component mixture model
- EM algorithm
- Probabilistic Latent Semantic Analysis (PLSA)
- Extensions of PLSA

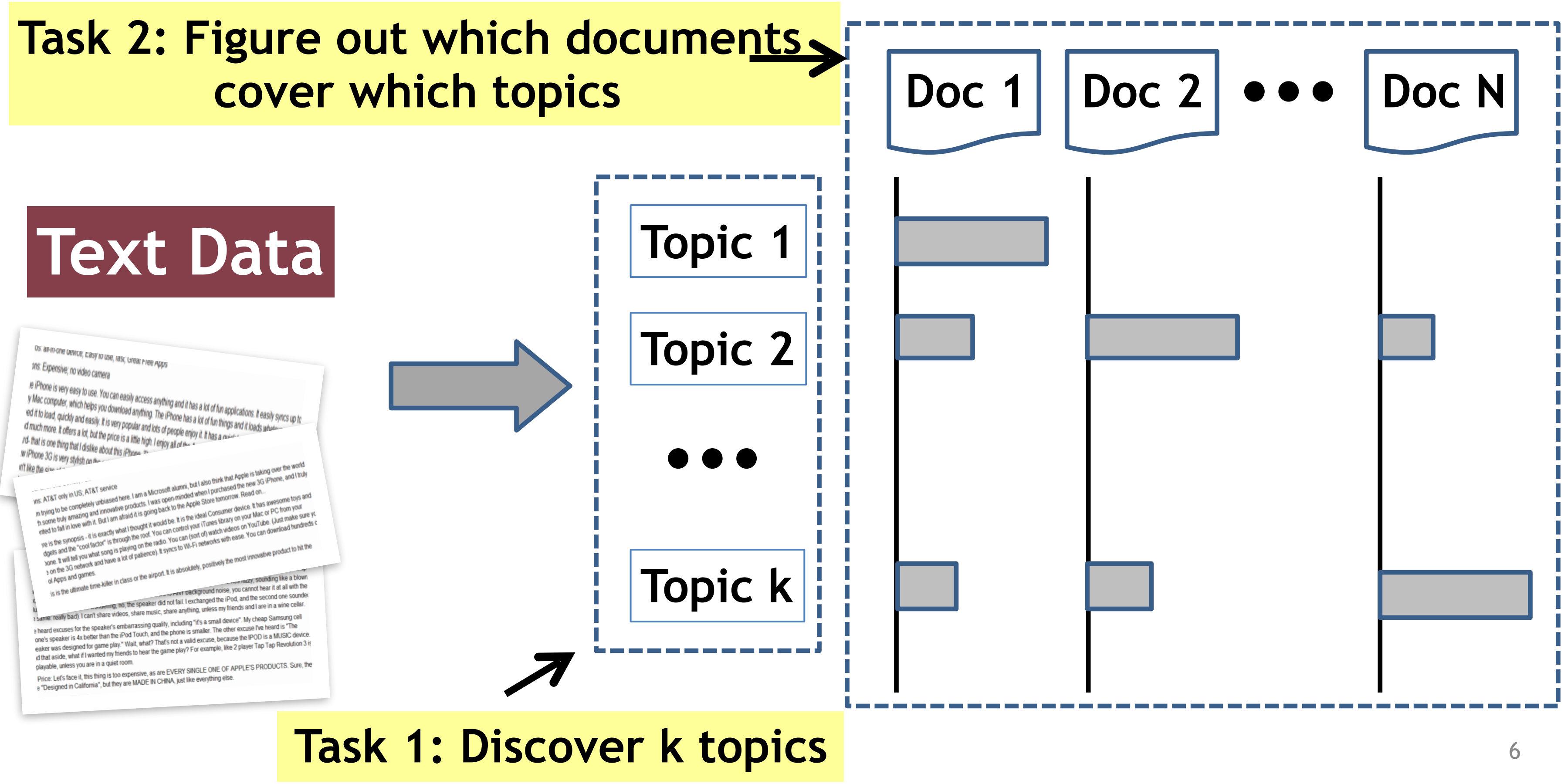
Topic Mining and Analysis: Motivation

- Topic ≈ 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



Tasks of Topic Mining and Analysis

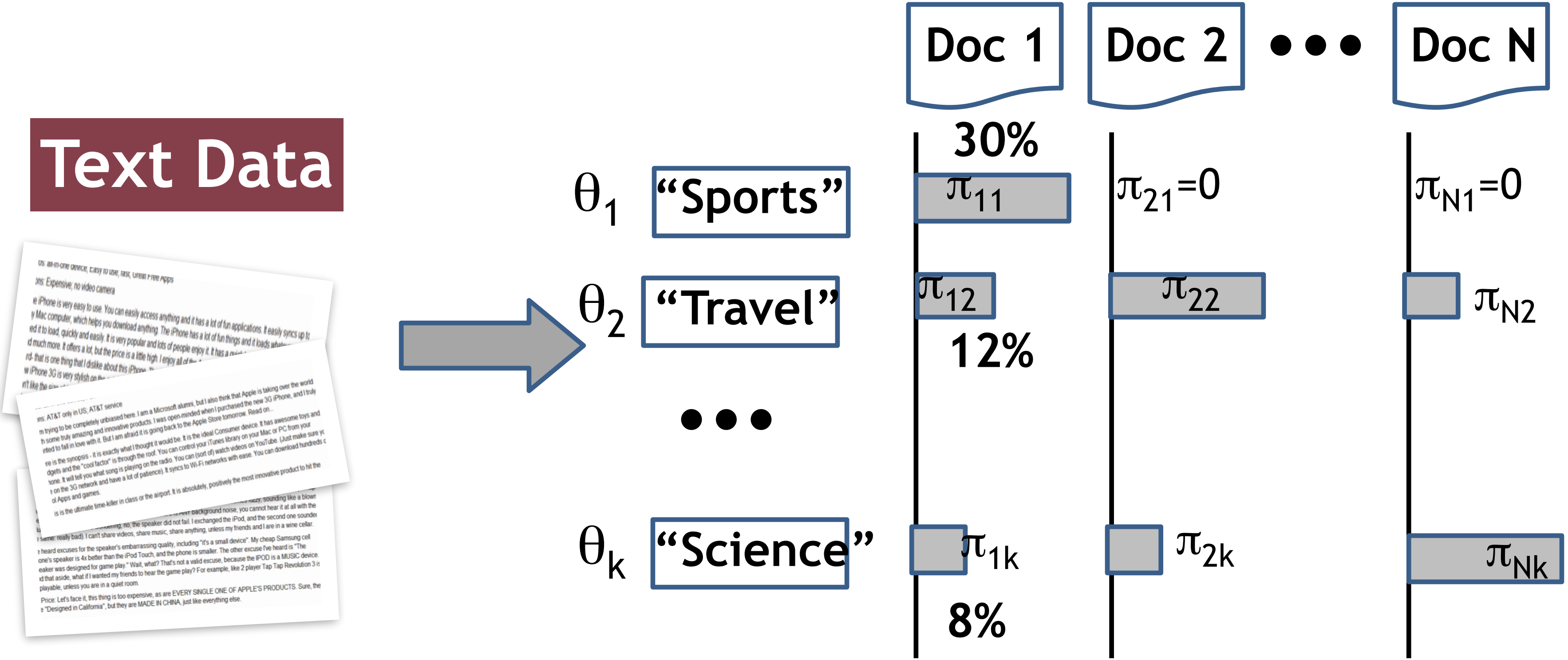


Formal Definition of Topic Mining and Analysis

- 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: $\{ \pi_{i1}, \ldots, \pi_{ik} \}$** $\sum_{j=1}^{k} \pi_{ij} = 1$
  - π_{ij} = prob. of d_i covering topic θ_j

How to define θ_i ?

Initial Idea: Topic = Term



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}

Doc d_i

θ_1 "Sports" π_{i1} count("sports", d_i)=4

θ_2 "Travel" π_{i2} count("travel", d_i) =2

…

θ_k "Science" π_{ik} count("science", d_i)=1

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

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

θ_2 "Travel" $\pi_{i2} \propto c(\text{"travel"}, d_i) = 1 > 0$

…

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

1. Need to count related words also!

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

3. Mine complicated topics?

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 covera **+ 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" | θ_2 "Travel" | • • • | θ_k "Science"

$P(w \mid \theta_1)$		$P(w \mid \theta_2)$		$P(w \mid \theta_k)$	
sports	0.02	travel	0.05	science	0.04
game	0.01	attraction	0.03	scientist	0.03
basketball	0.005	trip	0.01	spaceship	0.006
football	0.004	flight	0.004	telescope	0.004
play	0.003	hotel	0.003	genomics	0.004
star	0.003	island	0.003	star	0.002
…		…		…	
nba	0.001	culture	0.001	genetics	0.001
…		…		…	
travel	0.0005	play	0.0002	travel	0.00001
…		…		…	

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

Vocabulary Set: V={w1, w2,….}

Probabilistic Topic Mining and Analysis

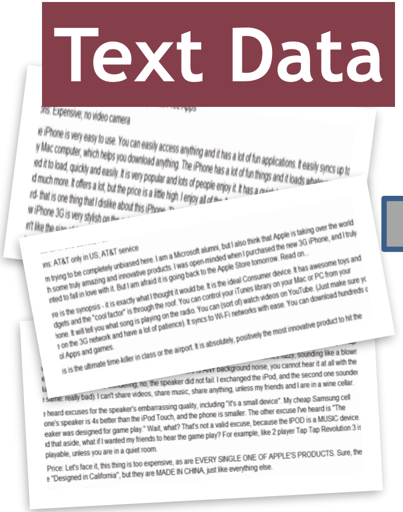
- Input
  - A **collection** of **N** text documents $C=\{d_1, \ldots, d_N\}$
  - **Vocabulary set:** $V=\{w_1, \ldots, w_M\}$
  - **Number of topics: k**
- Output
  - **k topics, each a word distribution:** { θ_1, …,

    $$\sum_{w \in V} p(w \mid \theta_i) = 1$$
  - **Coverage of topics in each d_i:** { π_{i1}, …, π

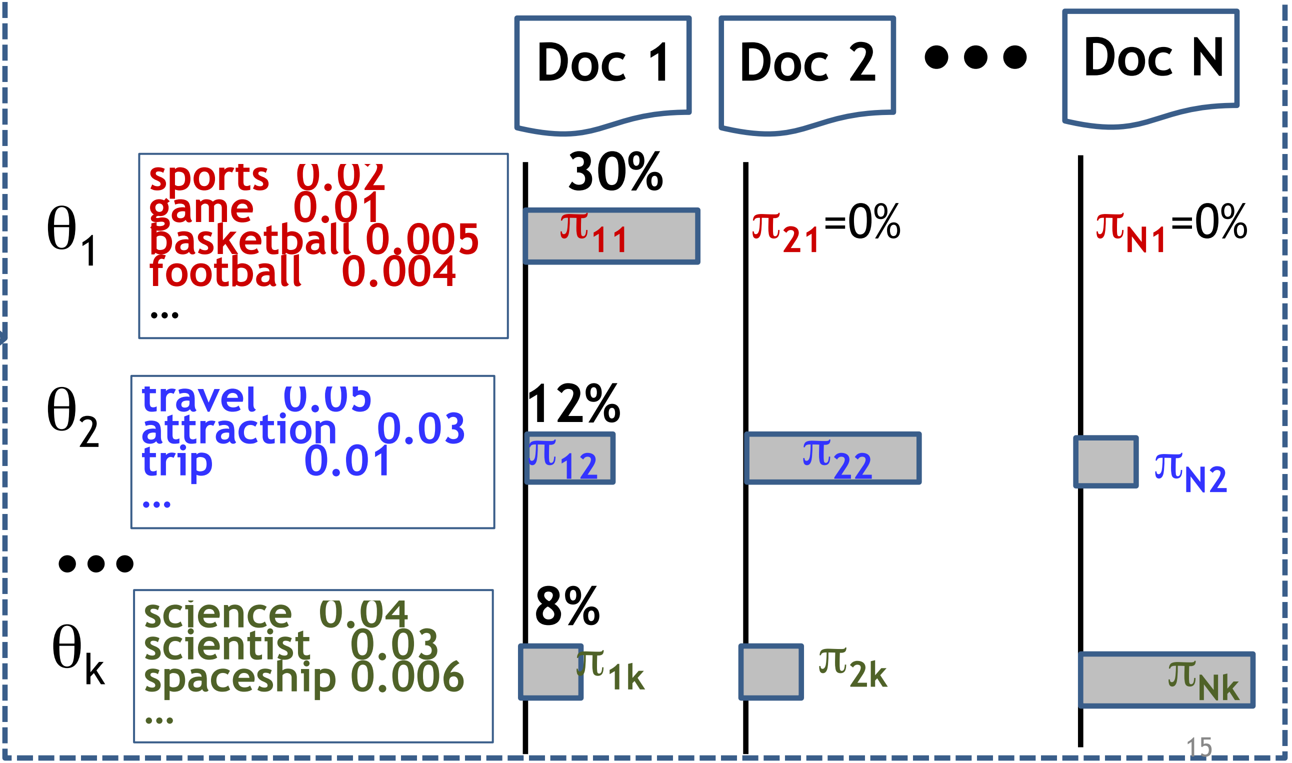
    $$\sum_{j=1}^{k} \pi_{ij} = 1$$
  - π_{ij}=prob. of d_i covering topic θ_j

The Computation Task

INPUT: C, k, V



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



Generative Model for Text Mining

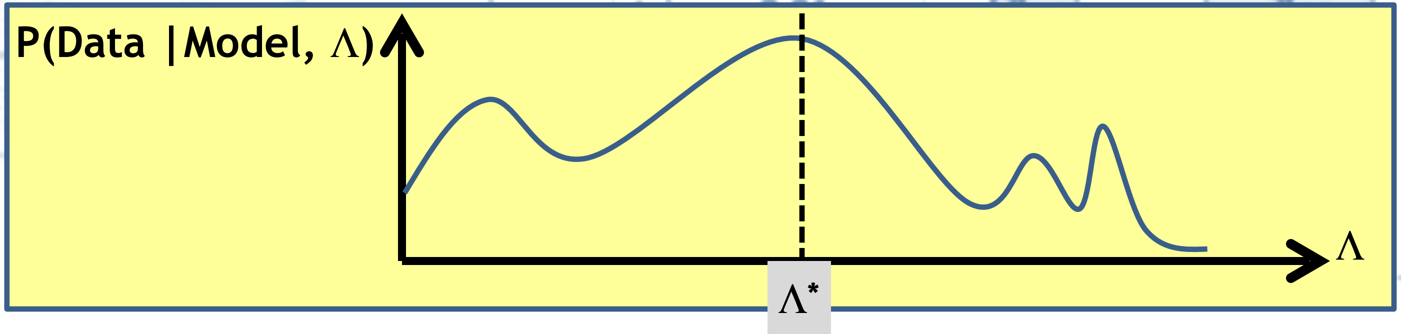
Modeling of Data Generation: P(Data | Model, Λ)

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

How many parameters in total?

Parameter Estimation/ Inferences

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



General Ideas of Generative Models for Text Mining

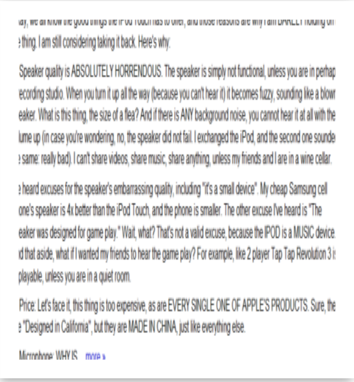
- **Model data generation: P(Data |Model, Λ)**
- **Infer the most likely parameter values Λ^*** given a particular data set: Λ^* **= argmax $_\Lambda$ p(Data| Model, Λ)**
- <u>**Take Λ^* as the "knowledge"**</u> to be mined for the text mining problem
- **Adjust** the design of the model to discover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

Simplest Case of Topic Model: Mining One Topic

INPUT: C={d}, V

OUTPUT: { θ}

Text Data

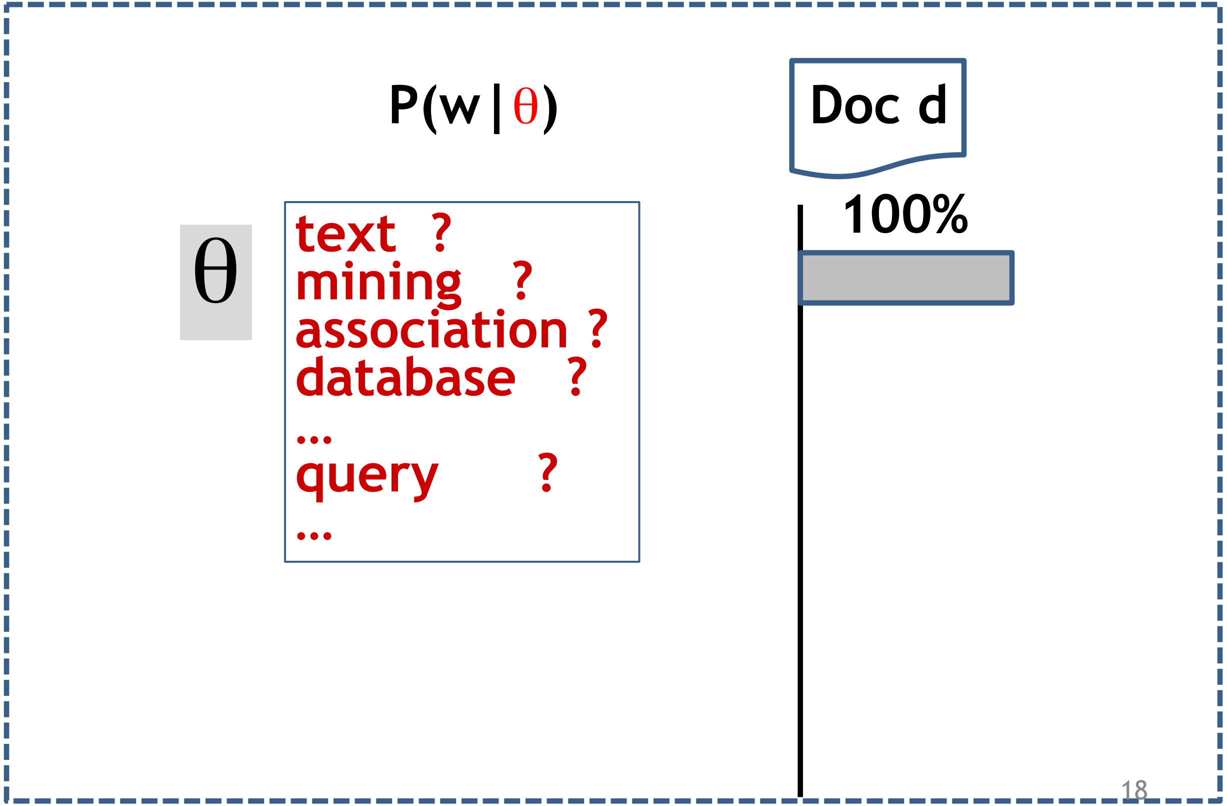

ay, we all know the good things the iPod Touch has to offer, and those reasons are why I am BARELY holding on thing. I am still considering taking it back. Here's why.

Speaker quality is ABSOLUTELY HORRENDOUS. The speaker is simply not functional, unless you are in perhap ecording studio. When you turn it up all the way (because you can't hear it) it becomes fuzzy, sounding like a blown eaker. What is this thing, the size of a flea? And if there is ANY background noise, you cannot hear it at all with the lume up (in case you're wondering, no, the speaker did not fail. I exchanged the iPod, and the second one sounde same: really bad). I can't share videos, share music, share anything, unless my friends and I are in a wine cellar.

heard excuses for the speaker's embarrassing quality, including "it's a small device". My cheap Samsung cell one's speaker is 4x better than the iPod Touch, and the phone is smaller. The other excuse I've heard is "The eaker was designed for game play." Wait, what? That's not a valid excuse, because the IPOD is a MUSIC device. d that aside, what if I wanted my friends to hear the game play? For example, like 2 player Tap Tap Revolution 3 i playable, unless you are in a quiet room.

Price: Let's face it, this thing is too expensive, as are EVERY SINGLE ONE OF APPLE'S PRODUCTS. Sure, the "Designed in California", but they are MADE IN CHINA, just like everything else.

Microphone: WHY IS... more »



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, …, M; $\theta_1+\dots+\theta_M=1$

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

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

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

Normalized Counts

What Does the Topic Look Like?

d

Text mining paper

p(w| θ)

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?

Factoring out Background Words

p(w| θ)

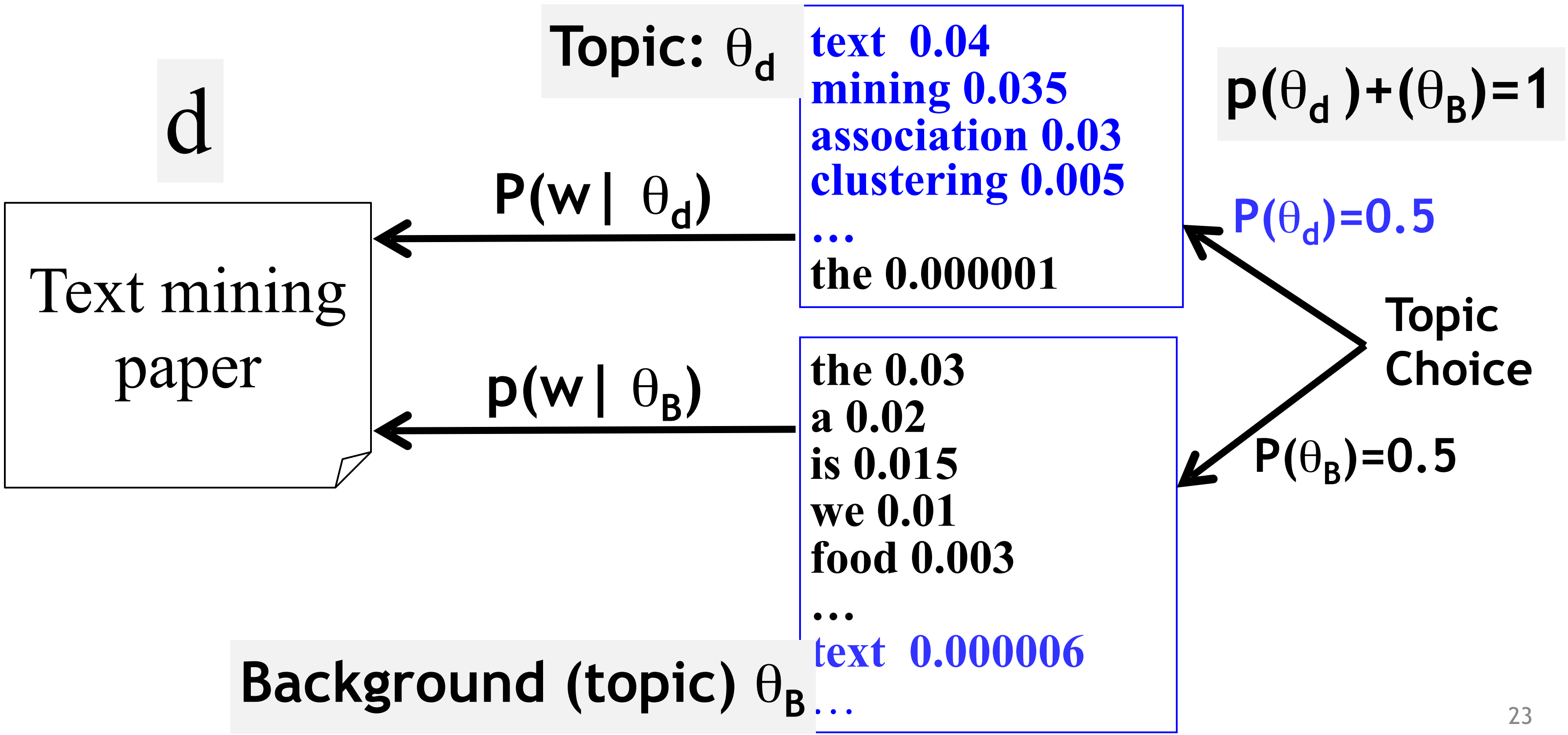
d

Text mining paper

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

How can we get rid of these common words?

Generate d Using Two Word Distributions



What's the probability of observing a word w?

Topic: θ_d

text 0.04
mining 0.035

$p(\theta_d)+(\theta_B)=1$

d

=0.5

Topic

hoice

.5

"the"

?

?

$P(\text{"the"})=p(\theta_d)p(\text{"the"}|\theta_d) + p(\theta_B)p(\text{"the"}|\theta_B)$

$= 0.5*0.000001+0.5*0.03$

$P(\text{"text"})=p(\theta_d)p(\text{"text"}|\theta_d) + p(\theta_B)p(\text{"text"}|\theta_B)$

$= 0.5*0.04+0.5*0.000006$

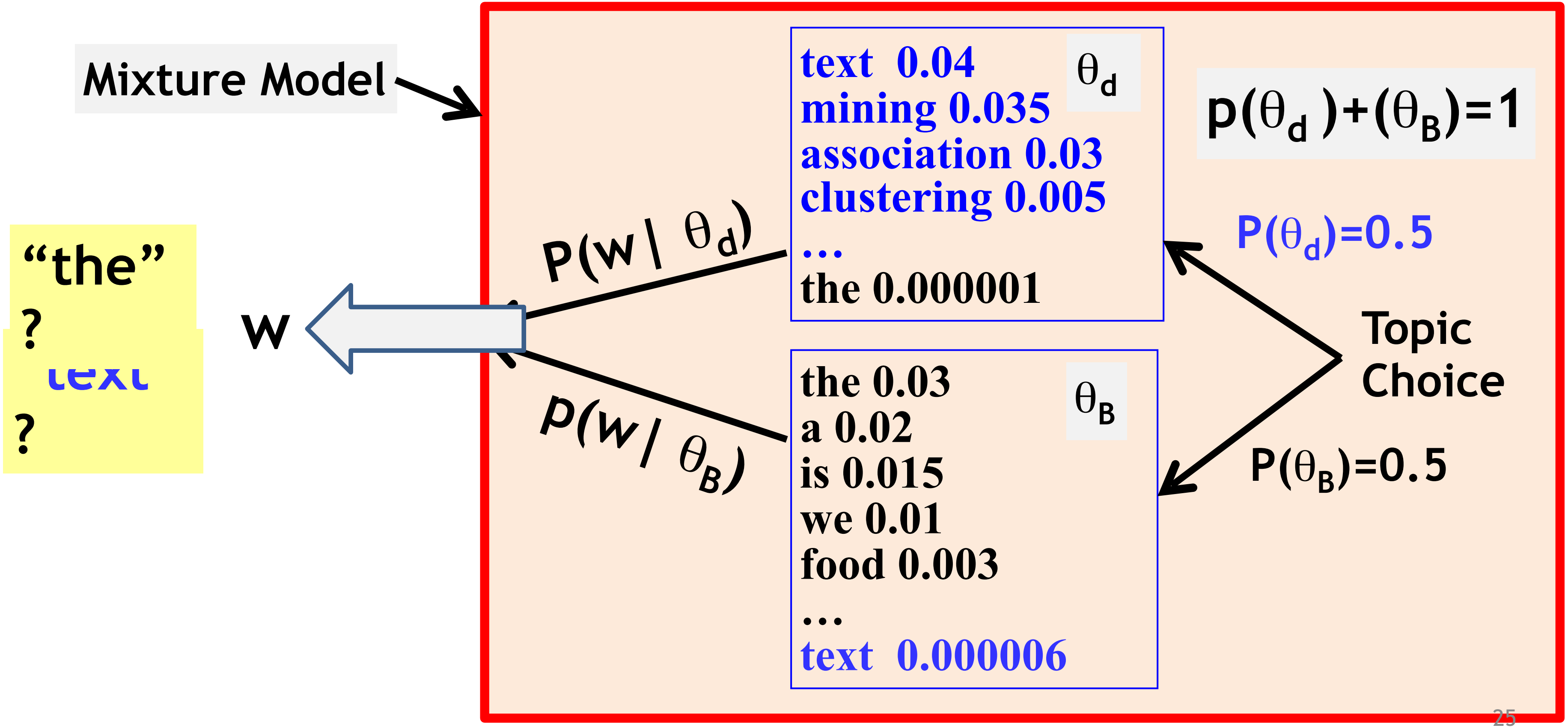
...

text 0.000006

...

Background (topic) θ_B

The Idea of a Mixture Model



As a Generative Model…

text 0.04
mining 0.035
association 0.03
clustering 0.005

θ_d

$p(\theta_d)+(\theta_B)=1$

w

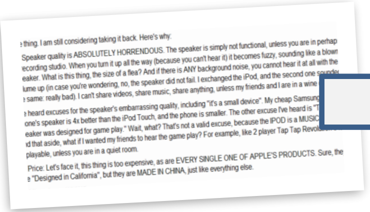
Formally defines the following generative model:

$$p(w)=p(\theta_d)p(w|\theta_d) + p(\theta_B)p(w|\theta_B)$$

Estimate of the model "discovers"

two topics + topic coverage

What if $p(\theta_d)=1$ or $p(\theta_B)=1$?


thing. I am still considering taking it back. Here's why:

Speaker quality is ABSOLUTELY HORRENDOUS. The speaker is simply not functional, unless you are in perhap recording studio. When you turn it up all the way (because you can't hear it) it becomes fuzzy, sounding like a blown eaker. What is this thing, the size of a flea? And if there is ANY background noise, you cannot hear it at all with the lume up (in case you're wondering, no, the speaker did not fail. I exchanged the iPod, and the second one sounde same: really bad). I can't share videos, share music, share anything, unless my friends and I are in a wine

heard excuses for the speaker's embarrassing quality, including "it's a small device". My cheap Samsung one's speaker is 4x better than the iPod Touch, and the phone is smaller. The other excuse I've heard is "T eaker was designed for game play." Wait, what? That's not a valid excuse, because the IPOD is a MUSIC d that aside, what if I wanted my friends to hear the game play? For example, like 2 player Tap Tap Revol playable, unless you are in a quiet room.

Price: Let's face it, this thing is too expensive, as are EVERY SINGLE ONE OF APPLE'S PRODUCTS. Sure, the "Designed in California", but they are MADE IN CHINA, just like everything else.

Mixture of Two Unigram Language Models

- **Data**: Document d
- Mixture **Model**: **parameters** $\Lambda=(\{p(w|\theta_d)\}, \{p(w|\theta_B)\}, p(\theta_B), p(\theta_d))$
  - Two unigram LMs: θ_d **(the topic of d);** θ_B **(background topic)**
  - Mixing weight (topic choice): $\mathbf{p(\theta_d)+p(\theta_B)=1}$
- **Likelihood** function:

$$p(d|\Lambda) = \prod_{i=1}^{|d|} p(x_i|\Lambda) = \prod_{i=1}^{|d|}[p(\theta_d)p(x_i|\theta_d) + p(\theta_B)p(x_i|\theta_B)]$$

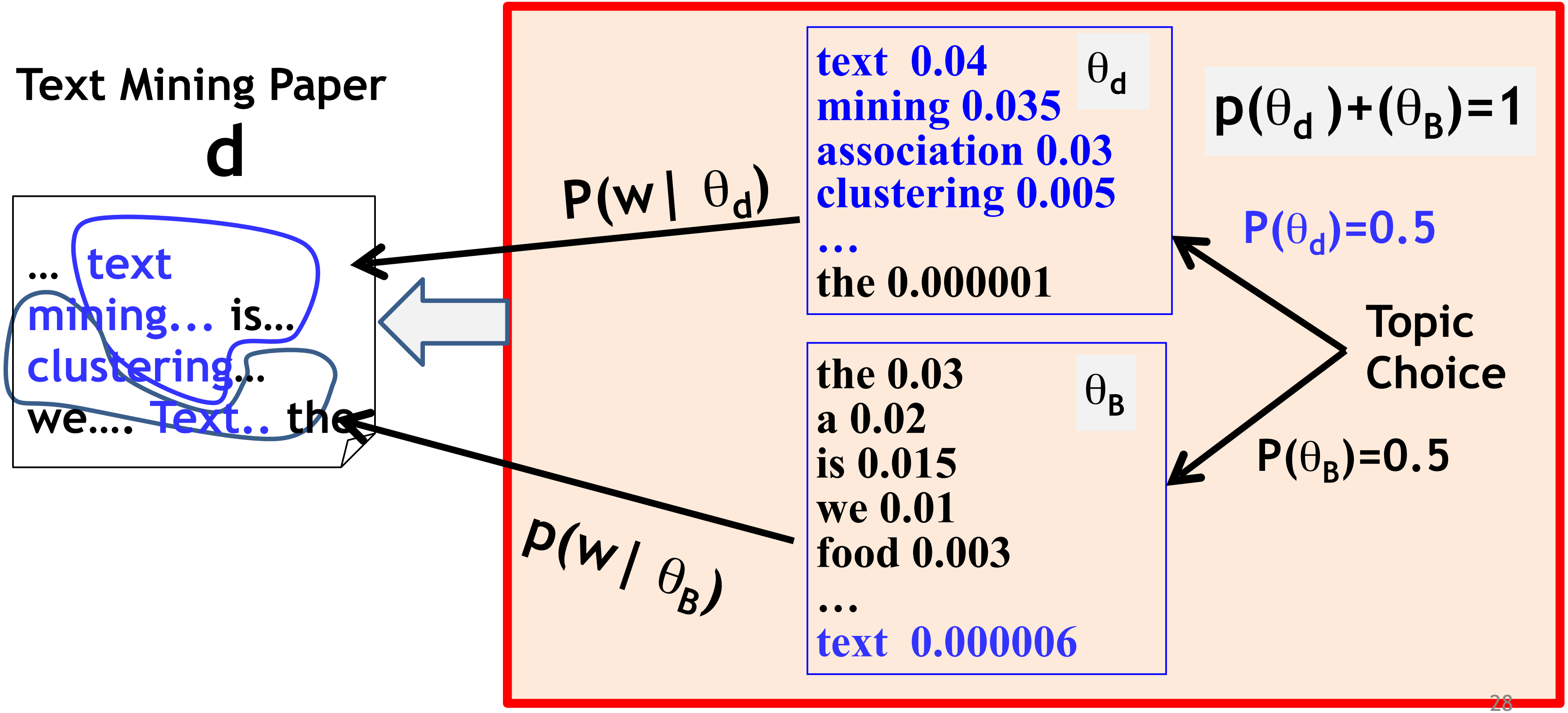
$$= \prod_{i=1}^{M}[p(\theta_d)p(w_i|\theta_d) + p(\theta_B)p(w_i|\theta_B)]^{c(w,d)}$$

- ML **Estimate:**

$$\Lambda^* = \arg\max_{\Lambda} p(d|\Lambda)$$

Subject to $\sum_{i=1}^{M} p(w_i|\theta_d) = \sum_{i=1}^{M} p(w_i|\theta_B) = 1 \qquad p(\theta_d) + p(\theta_B) = 1$

Back to Factoring out Background Words



Estimation of One Topic: $P(w|\theta_d)$

Adjust θ_d to maximize $p(d|\Lambda)$
(all other parameters are known)

Would the ML estimate <u>demote</u> background words in θ_d ?

d

… text
mining… is…
clustering…
we…. Text.. the

θ_d

text ?
mining ?
association ?
clustering ?
…
the ?

θ_B

the 0.03
a 0.02
is 0.015
we 0.01
food 0.003
…
text 0.000006

$p(\theta_d)+(\theta_B)=1$

$P(\theta_d)=0.5$

Topic Choice

$P(\theta_B)=0.5$

Behavior of a Mixture Model

d = text the

Likelihood:

$P(\text{"text"}) = p(\theta_d)p(\text{"text"}|\theta_d) + p(\theta_B)p(\text{"text"}|\theta_B)$

$= 0.5*p(\text{"text"}|\theta_d) + 0.5*0.1$

$P(\text{"the"}) = 0.5*p(\text{"the"}|\theta_d) + 0.5*0.9$

$p(d|\Lambda) = p(\text{"text"}|\Lambda)\, p(\text{"the"}|\Lambda)$

$= [0.5*p(\text{"text"}|\theta_d) + 0.5*0.1] \times [0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$

text ? the ?	θ_d

$P(\theta_d)=0.5$

$P(\theta_B)=0.5$

the 0.9 text 0.1	θ_B

How can we set $p(\text{"text"}|\theta_d)$ & $p(\text{"text"}|\theta_d)$ to maximize it?

Note that $p(\text{"text"}|\theta_d) + p(\text{"the"}|\theta_d) = 1$

"Collaboration" and "Competition" of θ_d and θ_B

$p(d|\Lambda)=p(\text{"text"}|\Lambda)\ p(\text{"the"}|\Lambda)$

$= [0.5*p(\text{"text"}|\theta_d) + 0.5*0.1] \times$

$[0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$

Note that $p(\text{"text"}|\theta_d) + p(\text{"the"}|\theta_d) = 1$

If , then reaches maximum when

$0.5*p(\text{"text"}|\theta_d) + 0.5*0.1 = 0.5*p(\text{"the"}|\theta_d) + 0.5*0.9$

→ $p(\text{"text"}|\theta_d)=0.9 \quad >> \quad p(\text{"the"}|\theta_d) = 0.1$!

d = text the

text ?
the ?
θ_d

$P(\theta_d)=0.5$

$P(\theta_B)=0.5$

the 0.9
text 0.1
θ_B

Behavior 1: if $p(w1|\theta_B) > p(w2|\theta_B)$, then $p(w1|\theta_d) < p(w2|\theta_d)$

Response to Data Frequency

d = text the

$p(d|\Lambda) = [0.5*p(\text{"text"}|\theta_d) + 0.5*0.1]$
$\times [0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$

→ $p(\text{"text"}|\theta_d)=0.9$ >> $p(\text{"the"}|\theta_d)=0.1$!

d' = text the the the the …the

$p(d'|\Lambda) = [0.5*p(\text{"text"}|\theta_d) + 0.5*0.1]$
$\times [0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$
$\times [0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$ …
$\times [0.5*p(\text{"the"}|\theta_d) + 0.5*0.9]$

What if we increase $p(\theta_B)$?

What's the optimal solution now? $p(\text{"the"}|\theta_d) > 0.1$? or $p(\text{"the"}|\theta_d) < 0.1$?

Behavior 2: high frequency words get higher $p(w|\theta_d)$

Estimation of One Topic: $P(w|\theta_d)$

How to set θ_d to maximize $p(d|\Lambda)$?
(all other parameters are known)

d

... text
mining... is...
clustering...
we.... Text.. the

θ_d
text ?
mining ?
association ?
clustering ?
…
the ?

$p(\theta_d)+(\theta_B)=1$

$P(\theta_d)=0.5$

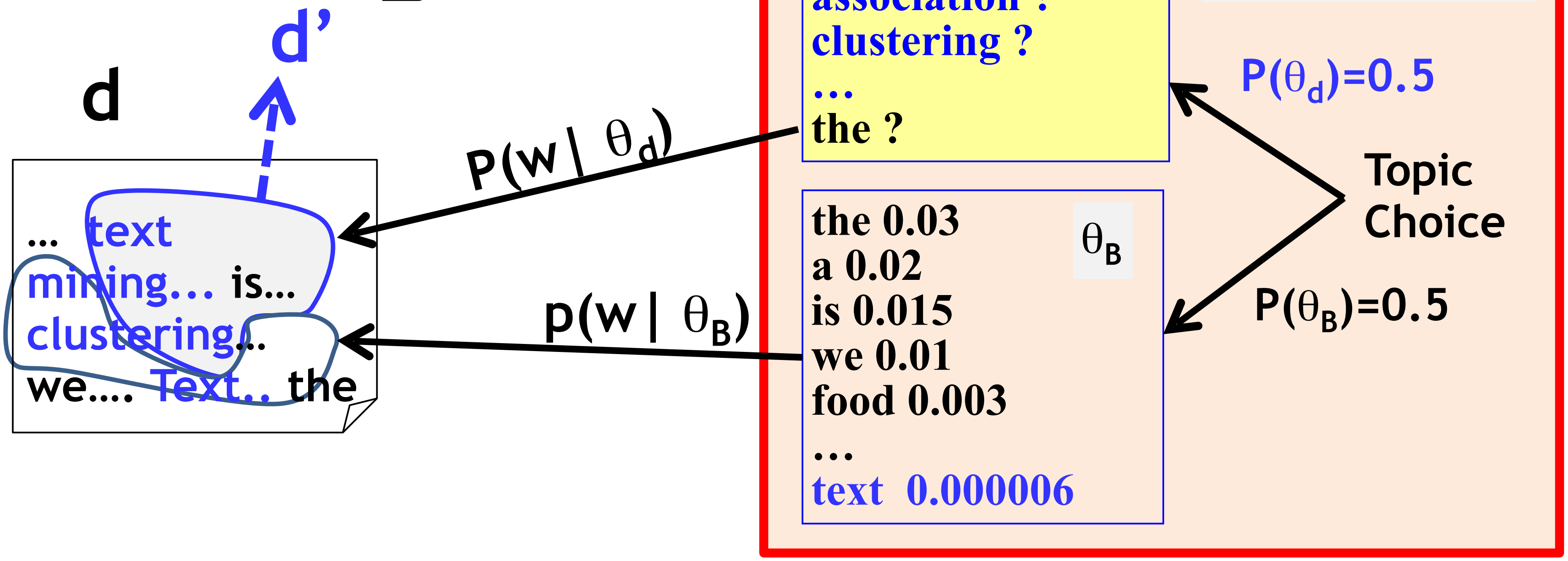
Topic Choice

$P(\theta_B)=0.5$

θ_B
the 0.03
a 0.02
is 0.015
we 0.01
food 0.003
…
text 0.000006

If we know which word is from which distribution…

$$p(w_i \mid \theta_d) = \frac{c(w_i, d')}{\sum_{w' \in V} c(w', d')}$$



Given all the parameters, infer the distribution a word is from…

Is “text” more likely from θ_d or θ_B?

From θ_d (Z=0)?

$p(\theta_d)p(\text{“text”} | \theta_d)$

From θ_B (Z=1)?

$p(\theta_B)p(\text{“text”} | \theta_B)$

$P(w| \theta_d)$

	θ_d
text 0.04	
mining 0.035	
association 0.03	
clustering 0.005	
…	
the 0.000001	

$p(w| \theta_B)$

	θ_B
the 0.03	
a 0.02	
is 0.015	
we 0.01	
food 0.003	
…	
text 0.000006	

$p(\theta_d)+p(\theta_B)=1$

$P(\theta_d)=0.5$

Topic Choice

$P(\theta_B)=0.5$

$$p(z=0 | w=\text{"text"}) = \frac{p(\theta_d)p(\text{"text"} | \theta_d)}{p(\theta_d)p(\text{"text"} | \theta_d) + p(\theta_B)p(\text{"text"} | \theta_B)}$$

The Expectation-Maximization (EM) Algorithm

Hidden Variable:
$z \in \{0, 1\}$

	z
the	**1**
paper	**1**
presents	**1**
a	**1**
text	**0**
mining	**0**
algorithm	**0**
for	**1**
clustering	**0**
...	**...**

Initialize $p(w|\theta_d)$ with random values.
Then iteratively improve it using E-step & M-step.
Stop when likelihood doesn't change.

$$p^{(n)}(z=0|w) = \frac{p(\theta_d)p^{(n)}(w|\theta_d)}{p(\theta_d)p^{(n)}(w|\theta_d) + p(\theta_B)p(w|\theta_B)}$$

E-step

How likely w is from θ_d

$$p^{(n+1)}(w|\theta_d) = \frac{c(w,d)p^{(n)}(z=0|w)}{\sum_{w' \in V} c(w',d)p^{(n)}(z=0|w')}$$

M-step

EM Computation in Action

E-step $$p^{(n)}(z=0|w)=\frac{p(\theta_d)p^{(n)}(w|\theta_d)}{p(\theta_d)p^{(n)}(w|\theta_d)+p(\theta_B)p(w|\theta_B)}$$

M-step $$p^{(n+1)}(w|\theta_d)=\frac{c(w,d)p^{(n)}(z=0|w)}{\sum_{w'\in V}c(w',d)p^{(n)}(z=0|w')}$$

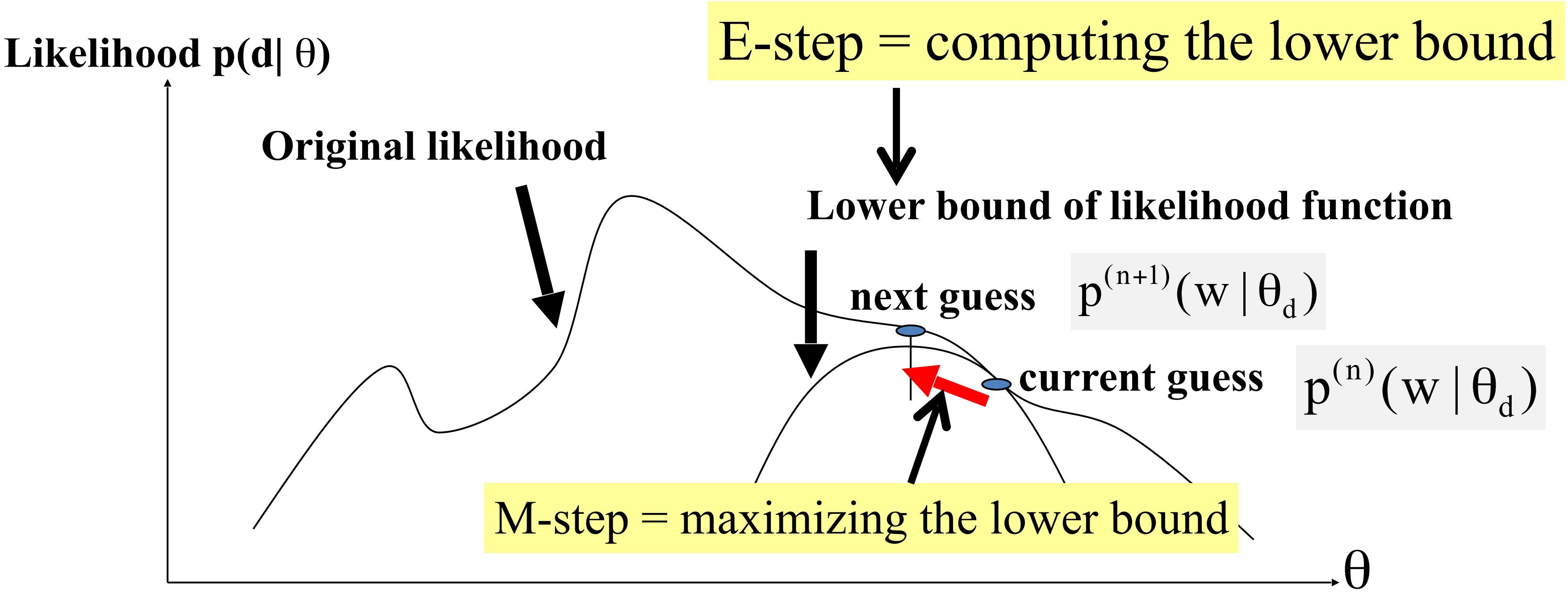
Assume $p(\theta_d)=p(\theta_B)=0.5$ and $p(w|\theta_B)$ is known

Word	#	$p(w\|\theta_B)$	Iteration 1		Iteration 2		Iteration 3	
			$P(w\|\theta)$	$p(z=0\|w)$	$P(w\|\theta)$	$P(z=0\|w)$	$P(w\|\theta)$	$P(z=0\|w)$
The	4	0.5	**0.25**	0.33	**0.20**	0.29	**0.18**	0.26
Paper	2	0.3	**0.25**	0.45	**0.14**	0.32	**0.10**	0.25
Text	4	0.1	**0.25**	0.71	**0.44**	0.81	**0.50**	0.93
Mining	2	0.1	**0.25**	0.71	**0.22**	0.69	**0.22**	0.69
Log-Likelihood			-16.96		-16.13		-16.02	

Likelihood increasing

“By products”: Are they also useful?

EM As Hill-Climbing → Converge to Local Maximum



A General Introduction to EM

Data: X (observed) + H(hidden) Parameter: θ

"Incomplete" likelihood: $L(\theta) = \log p(X|\theta)$

"Complete" likelihood: $Lc(\theta) = \log p(X,H|\theta)$

EM tries to iteratively maximize the incomplete likelihood:

Starting with an initial guess $\theta^{(0)}$,

1. E-step: compute the expectation of the complete likelihood

$$Q(\theta;\theta^{(n-1)}) = E_{\theta^{(n-1)}}[L_c(\theta)\,|\,X] = \sum_{h_i} p(H=h_i\,|\,X,\theta^{(n-1)})\log P(X,h_i)$$

2. M-step: compute $\theta^{(n)}$ by maximizing the Q-function

$$\theta^{(n)} = \arg\max_{\theta} Q(\theta;\theta^{(n-1)}) = \arg\max_{\theta} \sum_{h_i} p(H=h_i\,|\,X,\theta^{(n-1)})\log P(X,h_i)$$

Convergence Guarantee

Goal: maximizing "Incomplete" likelihood: $L(\theta) = \log p(X|\theta)$

I.e., choosing $\theta^{(n)}$, so that $L(\theta^{(n)}) - L(\theta^{(n-1)}) \geq 0$

Note that, since $p(X,H|\theta) = p(H|X,\theta)\, P(X|\theta)$, $L(\theta) = Lc(\theta) - \log p(H|X,\theta)$

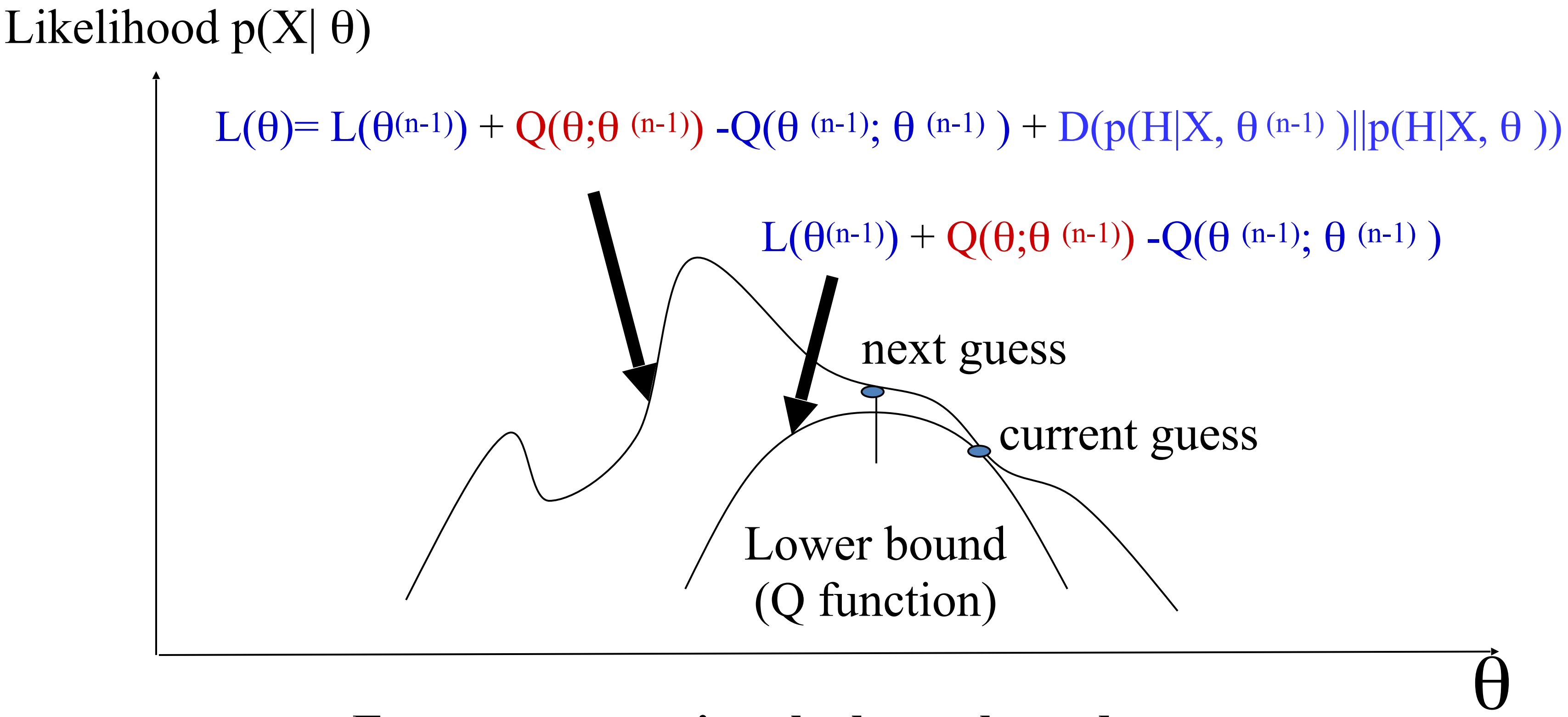
$$L(\theta^{(n)}) - L(\theta^{(n-1)}) = Lc(\theta^{(n)}) - Lc(\theta^{(n-1)}) + \log [p(H|X,\theta^{(n-1)}) / p(H|X,\theta^{(n)})]$$

Taking <u>expectation w.r.t. $p(H|X, \theta^{(n-1)})$,</u>

$$\underbrace{L(\theta^{(n)}) - L(\theta^{(n-1)})}_{\text{Doesn't contain H}} = \underbrace{Q(\theta^{(n)}; \theta^{(n-1)})}_{\text{EM chooses } \theta^{(n)} \text{ to maximize Q}} - Q(\theta^{(n-1)}; \theta^{(n-1)}) + \underbrace{D(p(H|X,\theta^{(n-1)}) \| p(H|X,\theta^{(n)}))}_{\text{KL-divergence, always non-negative}}$$

$$\boxed{\text{Therefore, } \quad L(\theta^{(n)}) \geq L(\theta^{(n-1)})!}$$

EM as Hill-Climbing: converging to a local maximum



E-step = computing the lower bound
M-step = maximizing the lower bound

Document as a Sample of Mixed Topics

Topic θ_1

government 0.3
response 0.2
...

Topic θ_2

city 0.2
new 0.1
orleans 0.05
...

. . .

Topic θ_k

donate 0.1
relief 0.05
help 0.02
...

Background θ_B

the 0.04
a 0.03
...

Blog article about "Hurricane Katrina"

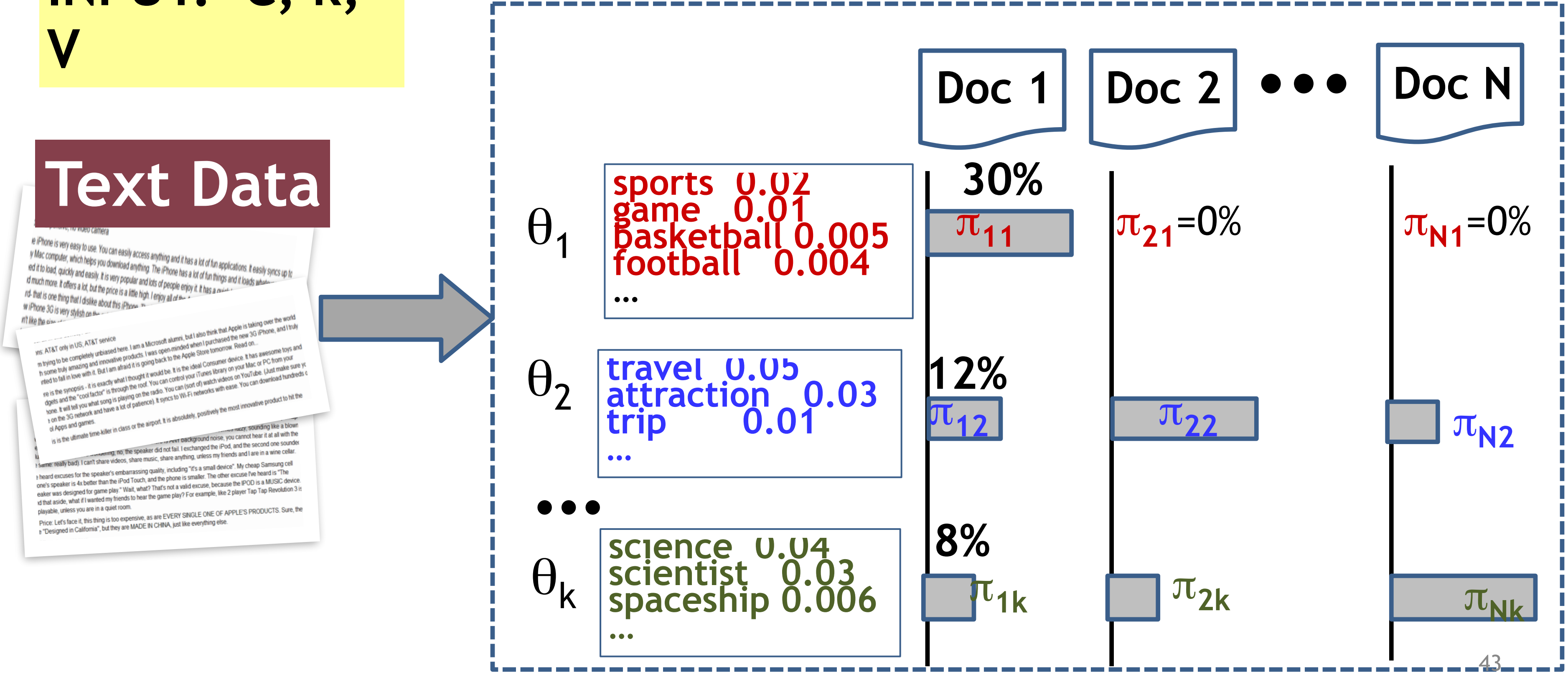
[Criticism of government response to the hurricane primarily consisted of criticism of its response to the approach of the storm and its aftermath, specifically in the delayed response] to the [flooding of New Orleans. … 80% of the 1.3 million residents of the greater New Orleans metropolitan area evacuated] …[Over seventy countries pledged monetary donations or other assistance]. …

Many applications are possible if we can "decode" the topics in text…

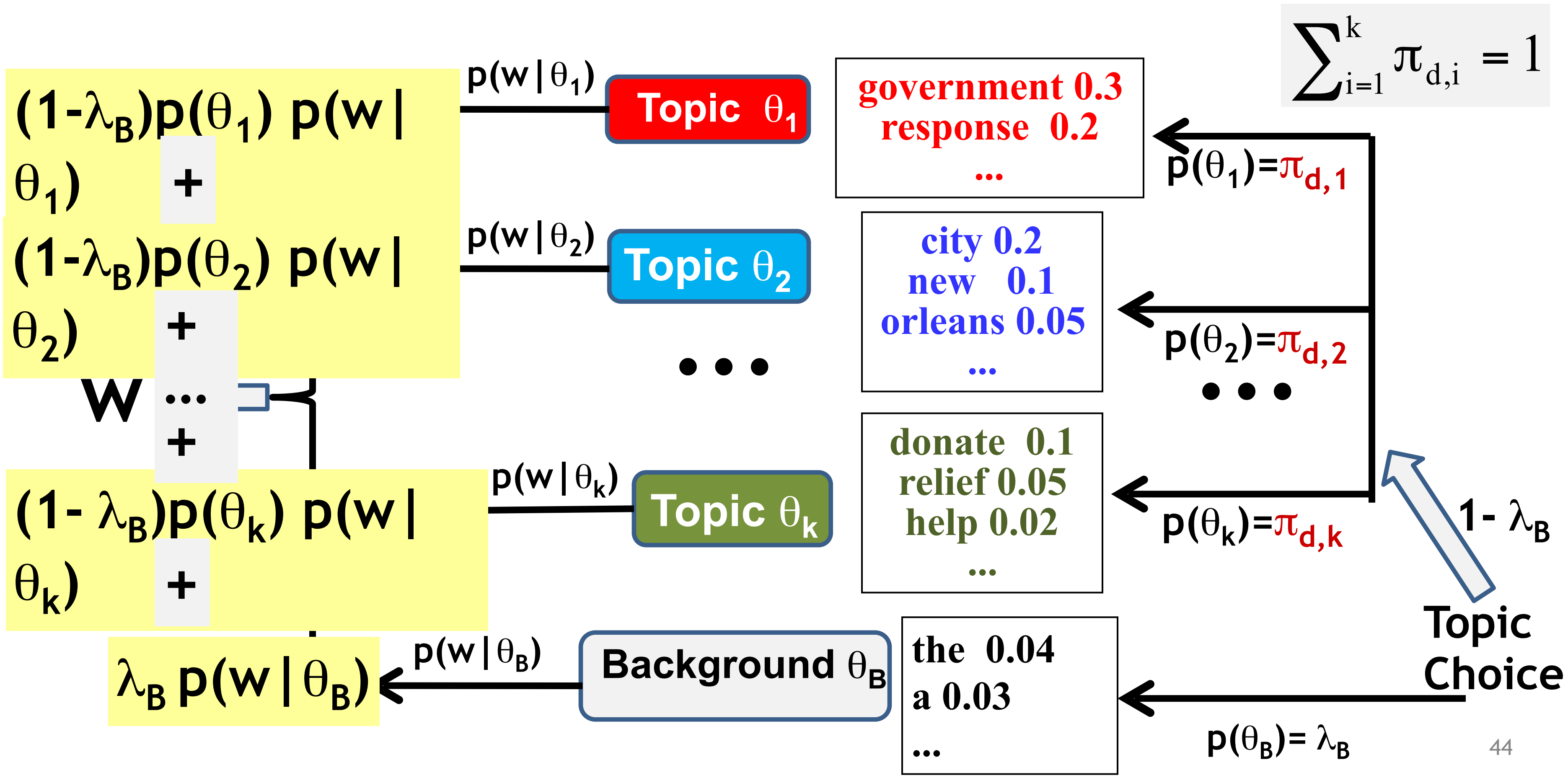
Mining Multiple Topics from Text

INPUT: C, k, V

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



Generating Text with Multiple Topics: p(w)=?



Probabilistic Latent Semantic Analysis (PLSA)

Percentage of background words (known)

Background LM (known)

Coverage of topic θ_j in doc d

Prob. of word w in topic θ_j

$$p_d(w) = \lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)$$

$$\log p(d) = \sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)]$$

$$\log p(C|\Lambda) = \sum_{d\in C}\sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)]$$

Unknown Parameters: $\Lambda=(\{\pi_{d,j}\}, \{\theta_j\})$, j=1, …, k

How many unknown parameters are there in total?

ML Parameter Estimation

$$p_d(w) = \lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)$$

$$\log p(d) = \sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)]$$

$$\log p(C|\Lambda) = \sum_{d\in C}\sum_{w\in V} c(w,d)\log[\lambda_B p(w|\theta_B) + (1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}\, p(w|\theta_j)]$$

Constrained Optimization: $\Lambda^* = \arg\max_\Lambda \mathrm{p}(\mathrm{C}|\Lambda)$

$$\forall j\in[1,k], \sum_{i=1}^{M} p(w_i|\theta_j) = 1 \qquad \forall d\in C, \sum_{j=1}^{k}\pi_{d,j} = 1$$

EM Algorithm for PLSA: E-Step

Hidden Variable (=topic indicator): $z_{d,w} \in$ {B, 1, 2, …, k}

Probability that **w in doc d** is generated from **topic** θ_j

Use of Bayes Rule

$$p(z_{d,w} = j) = \frac{\pi_{d,j}^{(n)} p^{(n)}(w \mid \theta_j)}{\sum_{j'=1}^{k} \pi_{d,j'}^{(n)} p^{(n)}(w \mid \theta_{j'})}$$

$$p(z_{d,w} = B) = \frac{\lambda_B p(w \mid \theta_B)}{\lambda_B p(w \mid \theta_B) + (1 - \lambda_B) \sum_{j=1}^{k} \pi_{d,j}^{(n)} p^{(n)}(w \mid \theta_j)}$$

Probability that **w in doc d** is generated from **background** θ_B

EM Algorithm for PLSA: M-Step

Hidden Variable (=topic indicator): $z_{d,w} \in \{B, 1, 2, \ldots, k\}$

sed on “allocated” word counts to topic θ_j

Re-estimated **probability** of **doc d** covering **topic** θ_j

$$\pi_{d,j}^{(n+1)} = \frac{\sum_{w \in V} \boxed{c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}}{\sum_{j'} \sum_{w \in V} c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j')}$$

$$p^{(n+1)}(w \mid \theta_j) = \frac{\sum_{d \in C} \boxed{c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}}{\sum_{w' \in V} \sum_{d \in C} c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)}$$

Re-estimated **probability** of **word w** for **topic** θ_j

Computation of the EM Algorithm

- Initialize all unknown parameters randomly
- Repeat until likelihood converges
  - E-step
  - M-step

$$p(z_{d,w} = j) \propto \pi_{d,j}^{(n)} p^{(n)}(w \mid \theta_j)$$

$$\sum_{j=1}^{k} p(z_{d,w} = j) = 1$$

$$p(z_{d,w} = B) \propto \lambda_B p(w \mid \theta_B) \leftarrow$$

What's the normalizer for this one?

$$\pi_{d,j}^{(n+1)} \propto \sum_{w \in V} c(w,d)(1 - p(z_{d,w} = B))p(z_{d,w} = j)$$

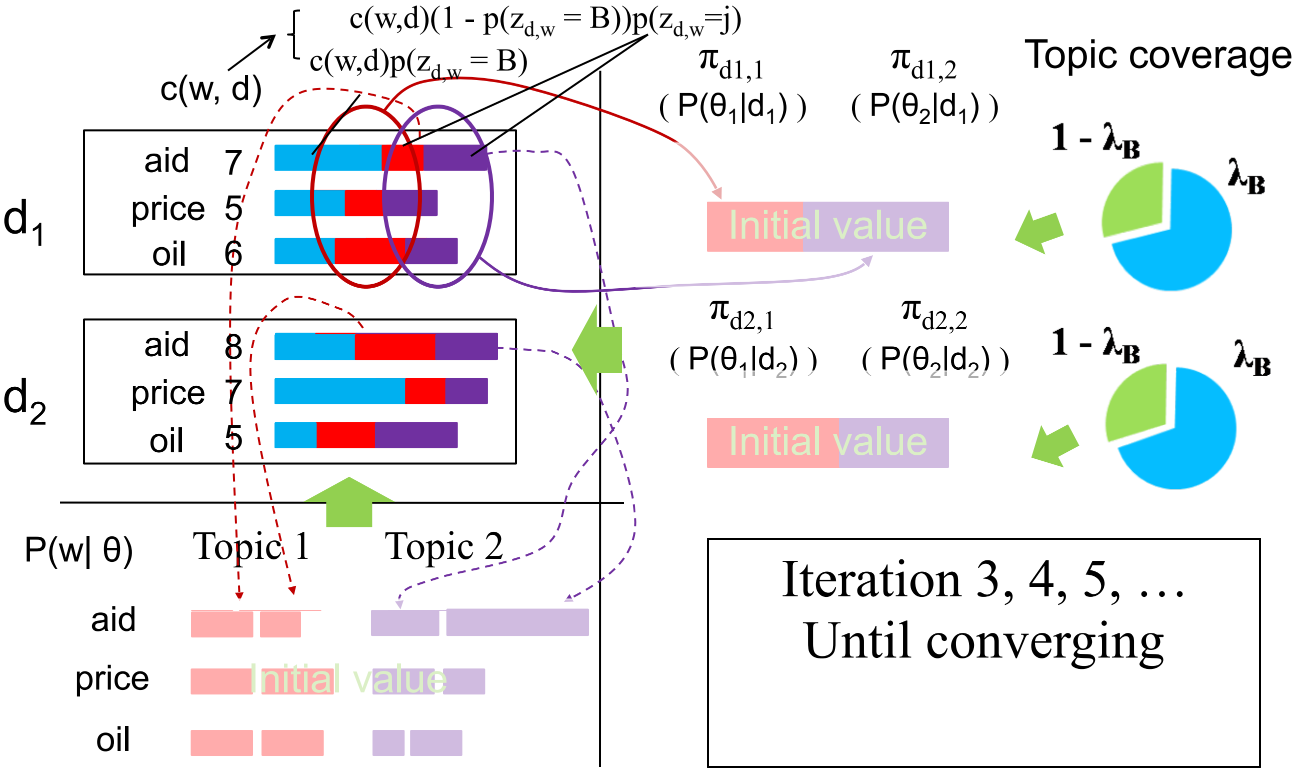
$$\forall d \in C, \sum_{j=1}^{k} \pi_{d,j} = 1$$

$$p^{(n+1)}(w \mid \theta_j) \propto \sum_{d \in C} c(w,d)(1 - p(z_{d,w} = B))p(z_{d,w} = j)$$

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

In general, accumulate counts, and then normalize

Illustration of EM Algorithm for PLSA



Iteration 3, 4, 5, …
Until converging

Applications of Mixture Models for Text Mining

Likelihood:

$$p(d \mid \theta_1 \oplus \theta_2) = \prod_{w \in V} [\lambda p(w \mid \theta_1) + (1-\lambda) p(w \mid \theta_2)]^{c(w,d)}$$

$$\log p(d \mid \theta_1 \oplus \theta_2) = \sum_{w \in V} c(w,d) \log[\lambda p(w \mid \theta_1) + (1-\lambda) p(w \mid \theta_2)]$$

Application Scenarios:

- $p(w|\theta_1)$ & $p(w|\theta_2)$ are known; estimate λ

The doc is about text mining and food nutrition, how much percent is about text mining?

- $p(w|\theta_1)$ & λ are known; estimate $p(w|\theta_2)$

30% of the doc is about text mining, what's the rest about?

- $p(w|\theta_1)$ is known; estimate λ & $p(w|\theta_2)$

The doc is about text mining, is it also about some other topic, and if so to what extent?

- λ is known; estimate $p(w|\theta_1)$& $p(w|\theta_2)$

30% of the doc is about one topic and 70% is about another, what are these two topics?

- Estimate λ, $p(w|\theta_1)$, $p(w|\theta_2)$

The doc is about two subtopics, find out what these two subtopics are and to what extent the doc covers each.

Use PLSA for Text Mining

- PLSA would be able to generate
  - Topic coverage in each document: $p(\pi_d = j)$
  - Word distribution for each topic: $p(w|\theta_j)$
  - Topic assignment at the word level for each document
  - The number of topics must be given in advance
- These probabilities can be used in many different ways
  - θ_j naturally serves as a word cluster
  - $\pi_{d,j}$ can be used for document clustering $j^* = \arg\max_j \pi_{d,j}$
  - Contextual text mining: Make these parameters conditioned on context, e.g.,
    - $p(\theta_j |time)$, from which we can compute/plot $p(time| \theta_j)$
    - $p(\theta_j |location)$, from which we can compute/plot $p(loc| \theta_j)$

How to Help Users Interpret a Topic Model? [Mei et al. 07b]

- Use top words
  - automatic, but hard to make sense

Term, relevance, weight, feedback

- Human generated labels
  - Make sense, bu[illegible]cale up

Retrieval Models

term	0.16
relevance	0.08
weight	0.07
feedback	0.04
independence	0.03
model	0.[illegible]3
frequent	0.02
probabilistic	0.02
document	0.02
…	

?

insulin
foraging
foragers
collected
grains
loads
collection
nectar
…

Question: Can we automatically generate understandable labels for topics?

What is a Good Label?

Retrieval models

term	0.1599
relevance	0.0752
weight	0.0660
feedback	0.0372
independence	0.0311
model	0.0310
frequent	0.0233
probabilistic	0.0188
document	0.0173
…	

A topic from
[Mei & Zhai 06b]

- Semantically close (relevance)
- Understandable – phrases?
- High coverage inside topic
- Discriminative across topics
- …

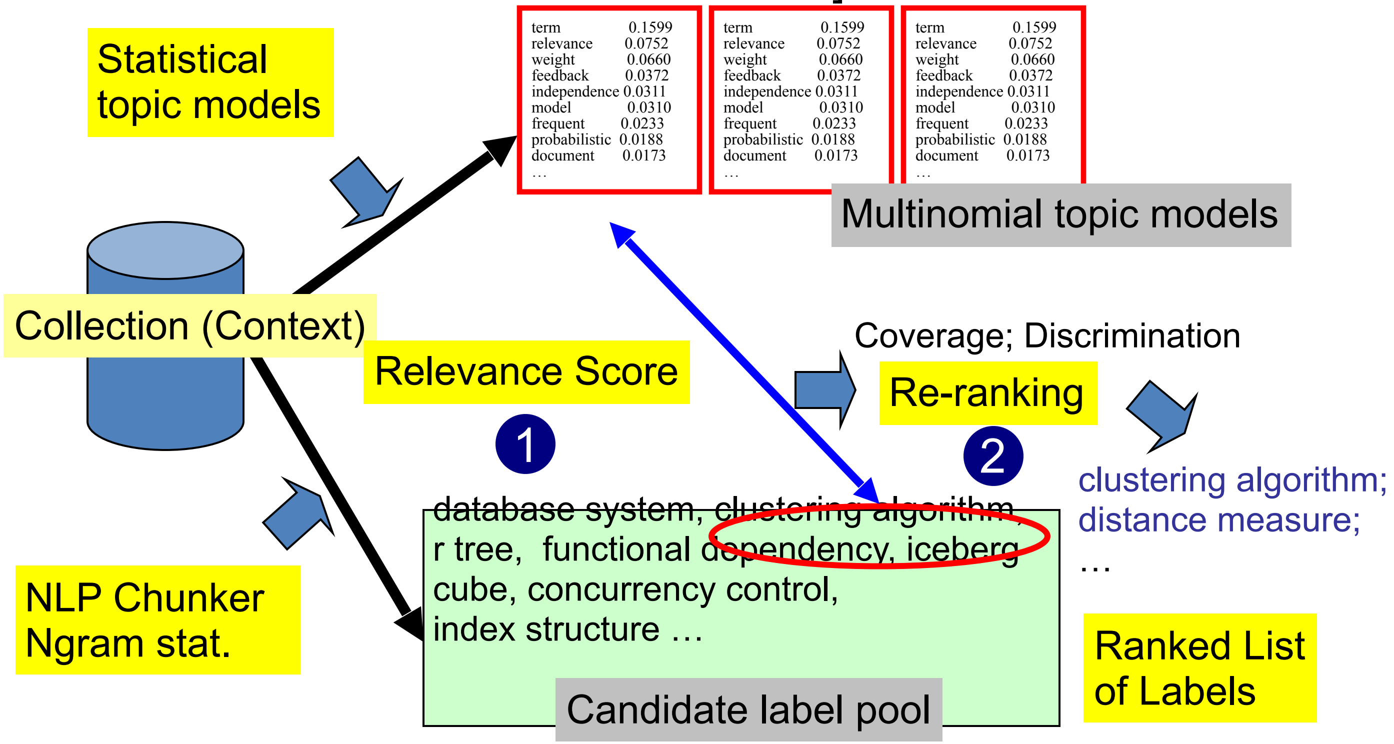
~~iPod Nano~~

~~じょうほうけんさく~~

~~Pseudo-feedback~~

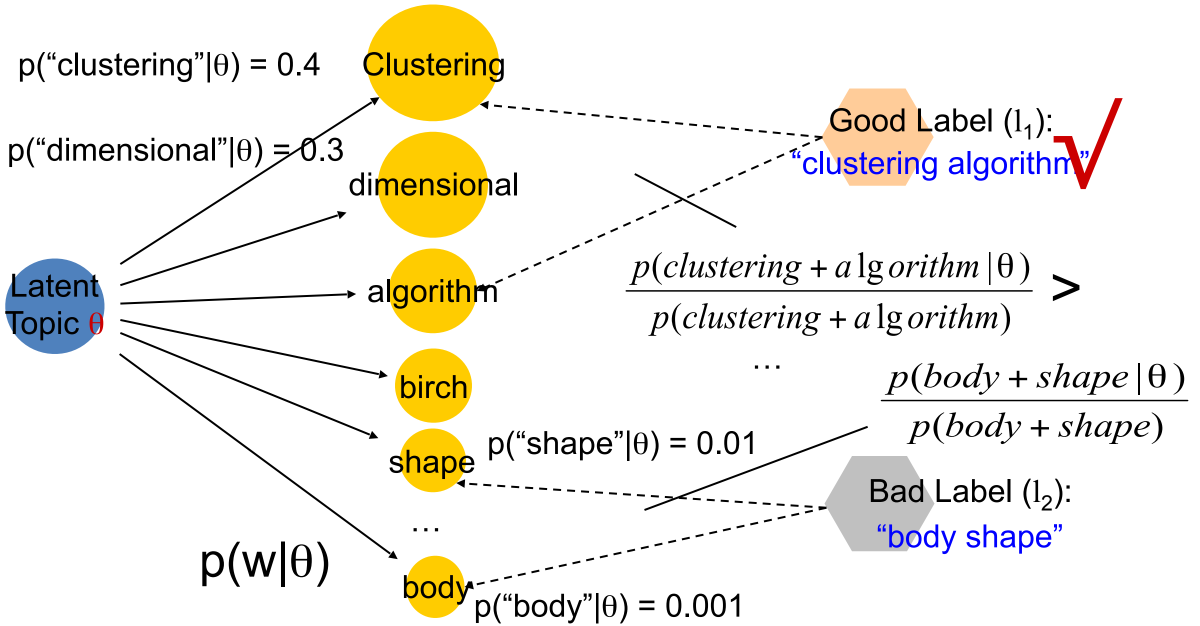
~~Information Retrieval~~

Automatic Labeling of Topics [Mei et al. 07b]



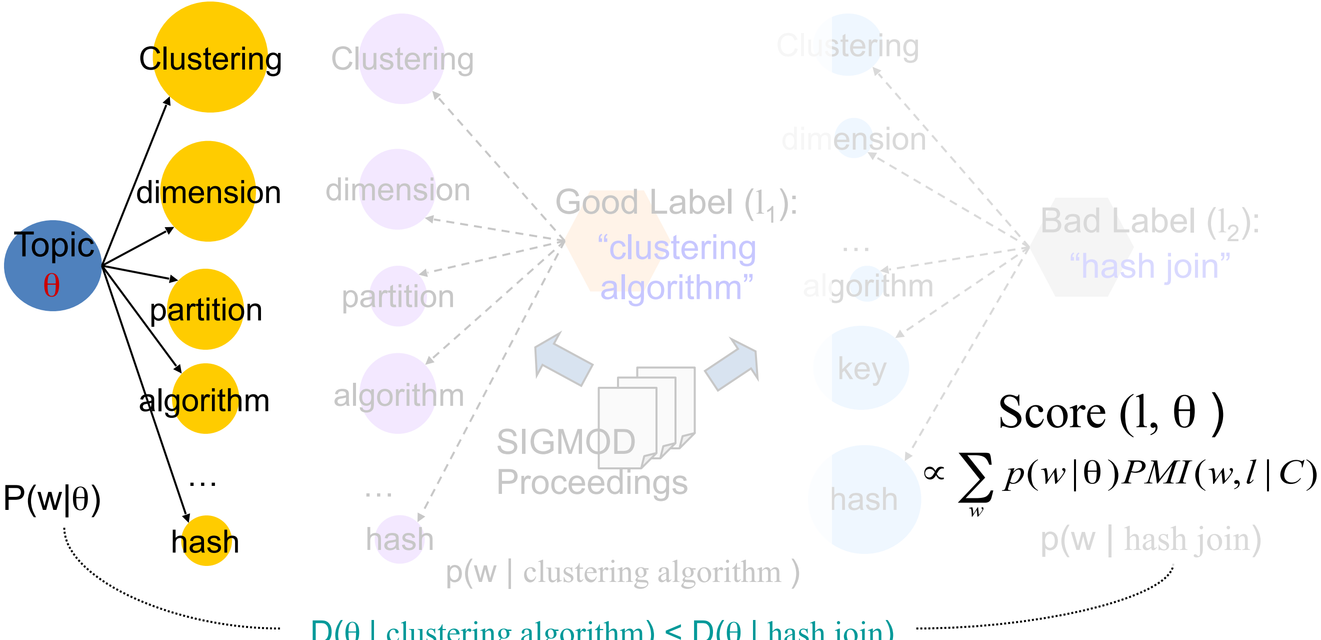
Relevance: the Zero-Order Score

- Intuition: prefer phrases well covering top words



Relevance: the First-Order Score

- Intuition: prefer phrases with similar context (distribution)



Results: Sample Topic Labels

sampling	0.06
estimation	0.04
approximate	0.04
histograms	0.03
selectivity	0.03
histogram	0.02
answers	0.02
accurate	0.02

selectivity estimation …

north	0.02
case	0.01
trial	0.01
iran	0.01
documents	0.01
walsh	0.009
reagan	0.009
charges	0.007

iran contra …

clustering algorithm
clustering structure
…

the, of, a, and, to, data,	> 0.02
…	
clustering	0.02
time	0.01
clusters	0.01
databases	0.01
large	0.01
performance	0.01
quality	0.005

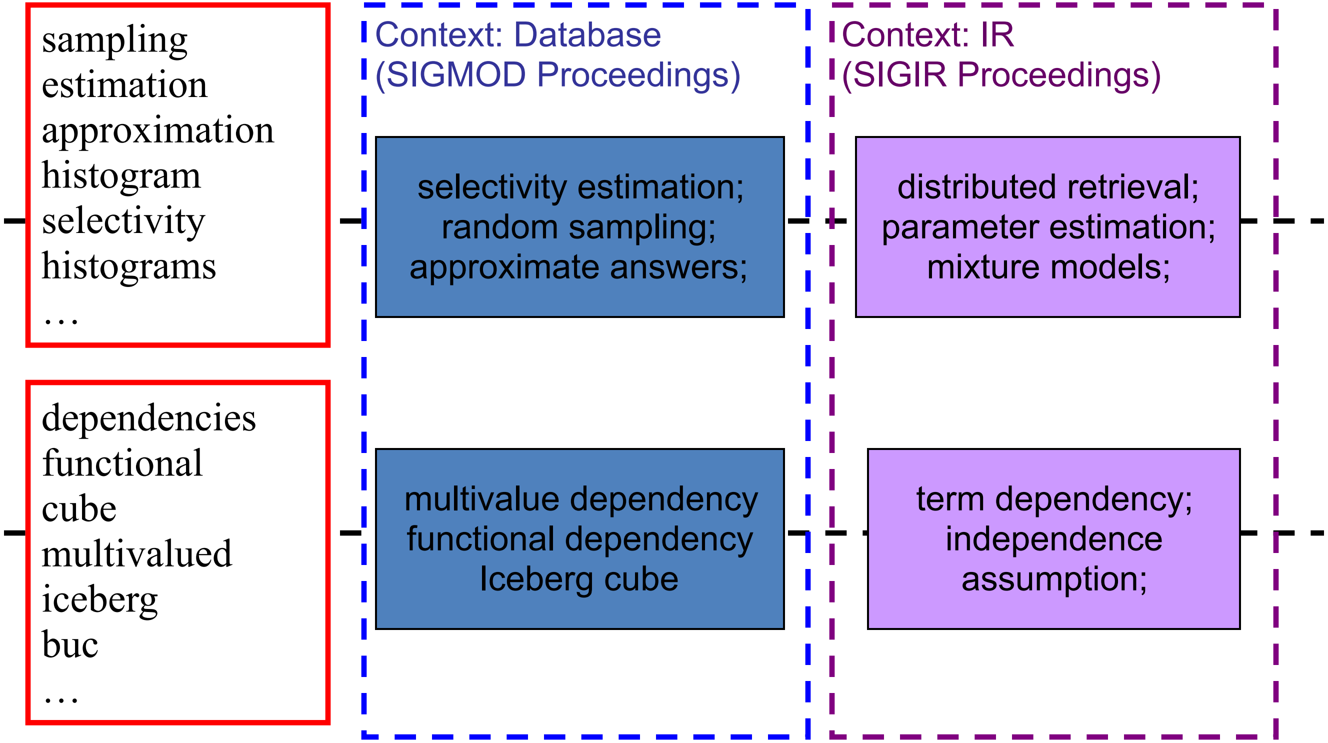
large data, data quality, high data, data application, …

r tree
b tree …

tree	0.09
trees	0.08
spatial	0.08
b	0.05
r	0.04
disk	0.02
array	0.01
cache	0.01

indexing methods

Results: Contextual-Sensitive Labeling



Extensions of PLSA

- PLSA with prior knowledge ➔ User-controlled PLSA
- PLSA for text data with context ➔ Contextualized PLSA
- PLSA as a generative model ➔ Latent Dirichlet Allocation (Bayesian inference for mixture models, covered later)

PLSA with Prior Knowledge

- Users may have expectations about which topics to analyze:
  - We expect to see “retrieval models” as a topic in IR
  - We want to see aspects such as “battery” and “memory” for opinions about a laptop
- Users may have knowledge about what topics are (or are NOT) covered in a document
  - Tags = topics ➔ A doc can only be generated using topics corresponding to the tags assigned to the document
- We can incorporate such knowledge as priors of PLSA model

Maximum a Posteriori (MAP) Estimate

$$\Lambda^* = \arg\max_{\Lambda} p(\Lambda) p(Data \mid \Lambda)$$

- We may use p(Λ) to encode all kinds of preferences and constraints, e.g.,
  - p(Λ)>0 if and only if one topic is precisely "background": p(w| θ_B)
  - p(Λ)>0 if and only if for a particular doc d, $\pi_{d,3}$=0 and $\pi_{d,1}$=1/2
  - p(Λ) favors a Λ with topics that assign high probabilities to some particular words
- The MAP estimate (with conjugate prior) can be computed using a similar EM algorithm to the ML estimate with smoothing to reflect prior preferences

EM Algorithm with Conjugate Prior on $p(w|\theta_i)$

$$p(z_{d,w}=j)=\frac{\pi_{d,j}^{(n)}p^{(n)}(w|\theta_j)}{\sum_{j'=1}^{k}\pi_{d,j'}^{(n)}p^{(n)}(w|\theta_{j'})}$$

$$p(z_{d,w}=B)=\frac{\lambda_B p(w|\theta_B)}{\lambda_B p(w|\theta_B)+(1-\lambda_B)\sum_{j=1}^{k}\pi_{d,j}^{(n)}p^{(n)}(w|\theta_j)}$$

$$\pi_{d,j}^{(n+1)}=\frac{\sum_{w\in V}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)}{\sum_{j'}\sum_{w\in V}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j')}$$

$$p^{(n+1)}(w|\theta_j)=\frac{\sum_{d\in C}c(w,d)(1-p(z_{d,w}=B))p(z_{d,w}=j)+\mu p(w|\theta'_j)}{\sum_{w'\in V}\sum_{d\in C}c(w',d)(1-p(z_{d,w'}=B))p(z_{d,w'}=j)+\mu}$$

Prior: $p(w|\theta'_j)$

battery 0.5
life 0.5

Pseudo counts of w from prior θ'

Sum of all pseudo counts

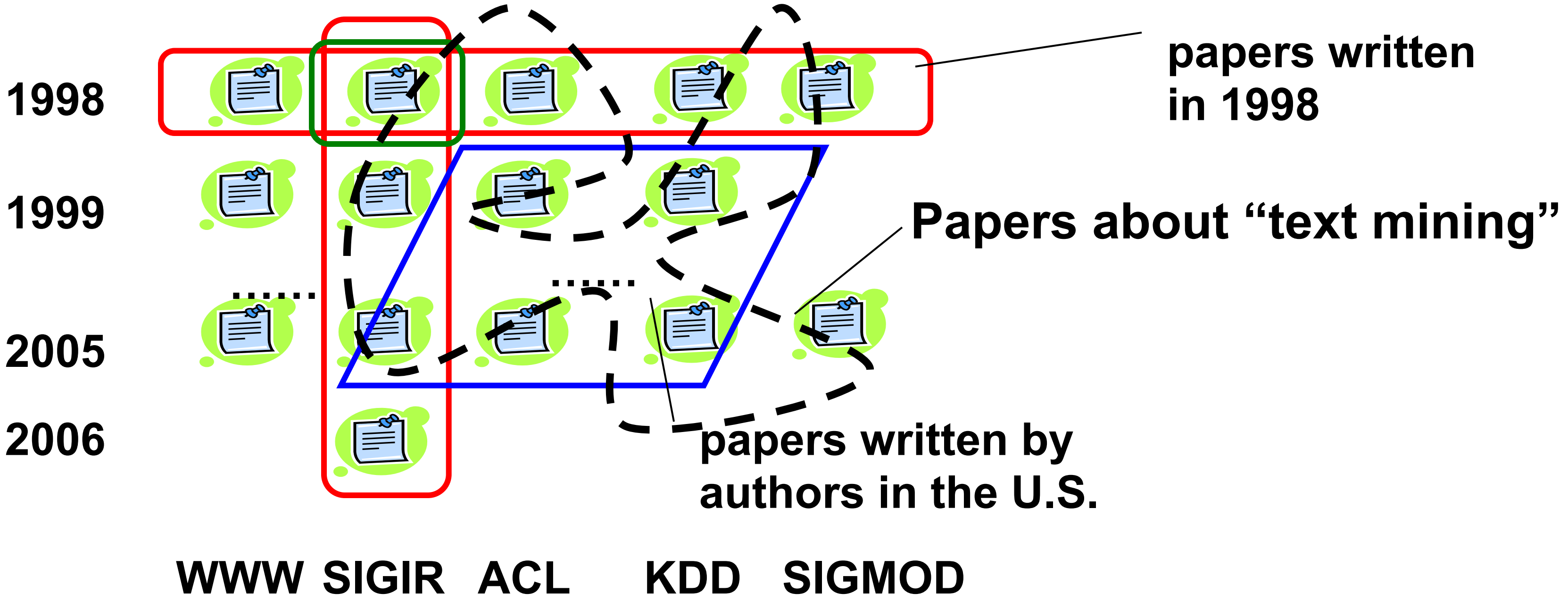
What if μ=0? What if μ=+∞?

We may also set any parameter to a constant (including 0) as needed

Contextual Text Mining: Motivation

- Text often has rich context information
  - Direct context (Meta-Data): time, location, authors, source, …
  - Indirect context (additional data related to meta-data): social network of the author, author's age, other text from the same source, etc.
  - Any related data can be regarded as context
- Context can be used to
  - Partition text data for comparative analysis
  - Provide meaning to the discovered topics

Context = Partitioning of Text



Enables discovery of knowledge associated with different context as needed

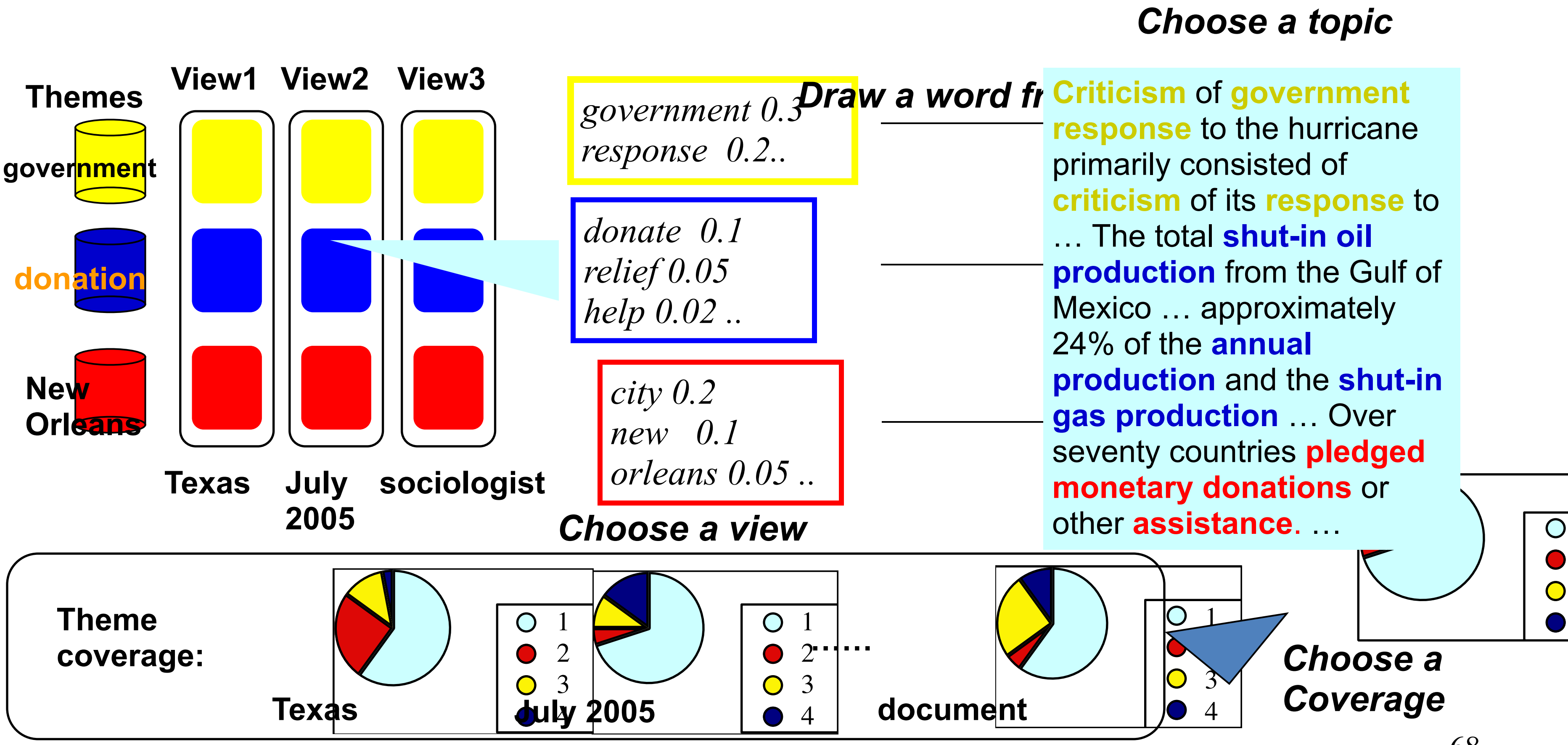
Many Interesting Questions Require Contextual Text Mining

- What topics have been gaining increasing attention recently in data mining research? (time as context)
- Is there any difference in the responses of people in different regions to the event? (location as context)
- What are the common research interests of two researchers? (authors as context)
- Is there any difference in the research topics published by authors in the USA and those outside? (author’s affiliation and location as context)
- Is there any difference in the opinions about a topic expressed on one social network and another? (social network of authors and topic as context)
- Are there topics in news data that are correlated with sudden changes in stock prices? (time series as context)
- What issues “mattered” in the 2012 presidential election? (time series as context)

Contextual Probabilistic Latent Semantic Analysis (CPLSA) [Mei & Zhai 06]

- General idea:
  - Explicitly add interesting context variables into a generative model (➔ enable discovery contextualized topics)
  - Context influences both coverage and content variation of topics
- As an extension of PLSA
  - Model the conditional likelihood of text given context
  - Assume context-dependent views of a topic
  - Assume context-dependent topic coverage
  - EM algorithm can still be used for parameter estimation
  - Estimated parameters naturally contain context variables, enabling contextual text mining

Generation Process of CPLSA



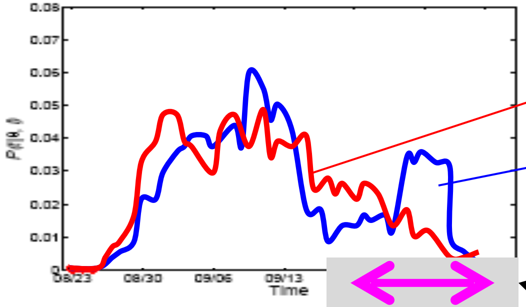
Comparing News Articles [Zhai et al. 04]
Iraq War (30 articles) vs. Afghan War (26 articles)

The common theme indicates that "United Nations" is involved in both wars

	Cluster 1	Cluster 2	Cluster 3
Common Theme	united 0.042 nations 0.04 …	killed 0.035 month 0.032 deaths 0.023 …	…
Iraq Theme	n 0.03 Weapons 0.024 Inspections 0.023 …	troops 0.016 hoon 0.015 sanches 0.012 …	…
Afghan Theme	Northern 0.04 alliance 0.04 kabul 0.03 taleban 0.025 aid 0.02	taleban 0.026 rumsfeld 0.02 hotel 0.012 front 0.011 …	…

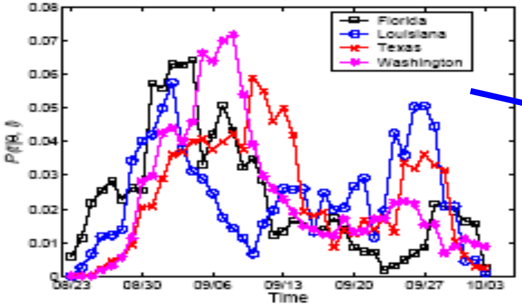
Collection-specific themes indicate different roles of "United Nations" in the two wars

Theme Life Cycles in Blog Articles About "Hurricane Katrina" [Mei et al. 06]



(a) Theme life cycles in Texas (Hurricane Katrina)

Hurricane Rita



(b) Theme "New Orleans" over states (Hurricane Katrina)

Oil Price

price 0.0772
oil 0.0643
gas 0.0454
increase 0.0210
product 0.0203
fuel 0.0188
company 0.0182
…

New Orleans

city 0.0634
orleans 0.0541
new 0.0342
louisiana 0.0235
flood 0.0227
evacuate 0.0211
storm 0.0177
…

Spatial Distribution of the Topic “Government Response” in Blog Articles About Hurricane Katrina [Mei et al. 06]

(a) Week1: 08/23-08/29

(b) Week Two: 08/30-09/05

(c) Week Three:09/06-09/12

Theme 1
Government Response
bush 0.0716374
president 0.0610942
federal 0.0514114
govern 0.0476977
fema 0.0474692
administrate 0.0233903
response 0.0208351
brown 0.0199573
blame 0.0170033
governor 0.0142153

(d) Week Four: 09/13-09/19

(e) Week Five: 09/20-09/26

Event Impact Analysis: IR Research [Mei & Zhai 06]

Topic: retrieval models

SIGIR papers

term	*0.1599*
relevance	*0.0752*
weight	*0.0660*
feedback	*0.0372*
independence	*0.0311*
model	*0.0310*
frequent	*0.0233*
probabilistic	*0.0188*
document	*0.0173*
…	

vector	*0.0514*
concept	*0.0298*
extend	*0.0297*
model	*0.0291*
space	*0.0236*
boolean	*0.0151*
function	*0.0123*
feedback	*0.0077*
…	

xml	*0.0678*
email	*0.0197*
model	*0.0191*
collect	*0.0187*
judgment	*0.0102*
rank	*0.0097*
subtopic	*0.0079*
…	

1992

Star

probabilist	*0.0778*
model	*0.0432*
logic	*0.0404*
ir	*0.0338*
boolean	*0.0281*
algebra	*0.0200*
estimate	*0.0119*
weight	*0.0111*
…	

A seminal paper [Croft & Ponte 98]

1998

year

model	*0.1687*
language	*0.0753*
estimate	*0.0520*
parameter	*0.0281*
distribution	*0.0268*
probable	*0.0205*
smooth	*0.0198*
markov	*0.0137*
likelihood	*0.0059*
…	

Topic Analysis with Network Context

- The **context** of a text article can form a **network**, e.g.,
  - Authors of research articles may form **collaboration networks**
  - Authors of social media content form **social networks**
  - Locations associated with text can be connected to form a **geographic network**
- **Benefit** of **joint analysis** of text and its network context
  - Network imposes **constraints** on topics in text (**authors connected in a network tend to write about similar topics**)
  - Text helps **characterize** the content associated with each subnetwork (e.g., difference in opinions expressed in two subnetworks?)

Network Supervised Topic Modeling: General Idea [Mei et al. 08]

- Probabilistic topic modeling as optimization: maximize likelihood

$$\Lambda^* = \arg\max_{\Lambda} p(\mathrm{TextData} \mid \Lambda)$$

- Main idea: network imposes constraints on model parameters Λ
  - The text at two adjacent nodes of the network tends to cover similar topics
  - Topic distributions are smoothed over adjacent nodes
  - Add network-induced regularizers to the likelihood objective function

Any generative model

Any network

$$\Lambda^* = \arg\max_{\Lambda} f(p(\mathrm{TextData} \mid \Lambda), r(\Lambda, \mathrm{Network}))$$

Any way to combine

Any regularizer

Instantiation: NetPLSA [Mei et al. 08]

Network-induced prior: Neighbors have similar topic distribution

Modified objective function

Text collection

PLSA log-likelihood

$$O(C,G) = (1-\lambda)\cdot\left(\sum_{d}\sum_{w} c(w,d)\log\sum_{j=1}^{k} p(\theta_j \mid d)\,p(w \mid \theta_j)\right)$$

Network graph

$$+\boxed{\lambda}\cdot\left(-\frac{1}{2}\sum_{\langle u,v\rangle\in E} \underline{w(u,v)}\,\underline{\sum_{j=1}^{k}\left(p(\theta_j \mid u)-p(\theta_j \mid v)\right)^2}\right)$$

Influence of network constraint

Weight of edge (u,v)

Quantify the difference in the topic coverage at node u and v

Mining 4 Topical Communities: Results of PLSA

Can't uncover the 4 communities (IR, DM, ML, Web)

Topic 1		Topic 2		Topic 3		Topic 4	
term	0.02	peer	0.02	visual	0.02	interface	0.02
question	0.02	patterns	0.01	analog	0.02	towards	0.02
protein	0.01	mining	0.01	neurons	0.02	browsing	0.02
training	0.01	clusters	0.01	vlsi	0.01	xml	0.01
weighting	0.01	stream	0.01	motion	0.01	generation	0.01
multiple	0.01	frequent	0.01	chip	0.01	design	0.01
recognition	0.01	e	0.01	natural	0.01	engine	0.01
relations	0.01	page	0.01	cortex	0.01	service	0.01
library	0.01	gene	0.01	spike	0.01	social	0.01

Mining 4 Topical Communities: Results of NetPLSA

Uncovers the 4 communities well

Information Retrieval		Data Mining		Machine Learning		Web	
retrieval	0.13	mining	0.11	neural	0.06	web	0.05
information	0.05	data	0.06	learning	0.02	services	0.03
document	0.03	discovery	0.03	networks	0.02	semantic	0.03
query	0.03	databases	0.02	recognition	0.02	services	0.03
text	0.03	rules	0.02	analog	0.01	peer	0.02
search	0.03	association	0.02	vlsi	0.01	ontologies	0.02
evaluation	0.02	patterns	0.02	neurons	0.01	rdf	0.02
user	0.02	frequent	0.01	gaussian	0.01	management	0.01
relevance	0.02	streams	0.01	network	0.01	ontology	0.01

Text Information Network

- In general, we can view text data that naturally "lives" in a rich information network with all other related data
- Text data can be associated with
  - Nodes of the network
  - Edges of the network
  - Paths of the network
  - Subnetworks
  - …
- Analysis of text should be using the entire network!

Text Mining for Understanding Time Series

What might have caused the stock market crash?

Sept 11 attack!

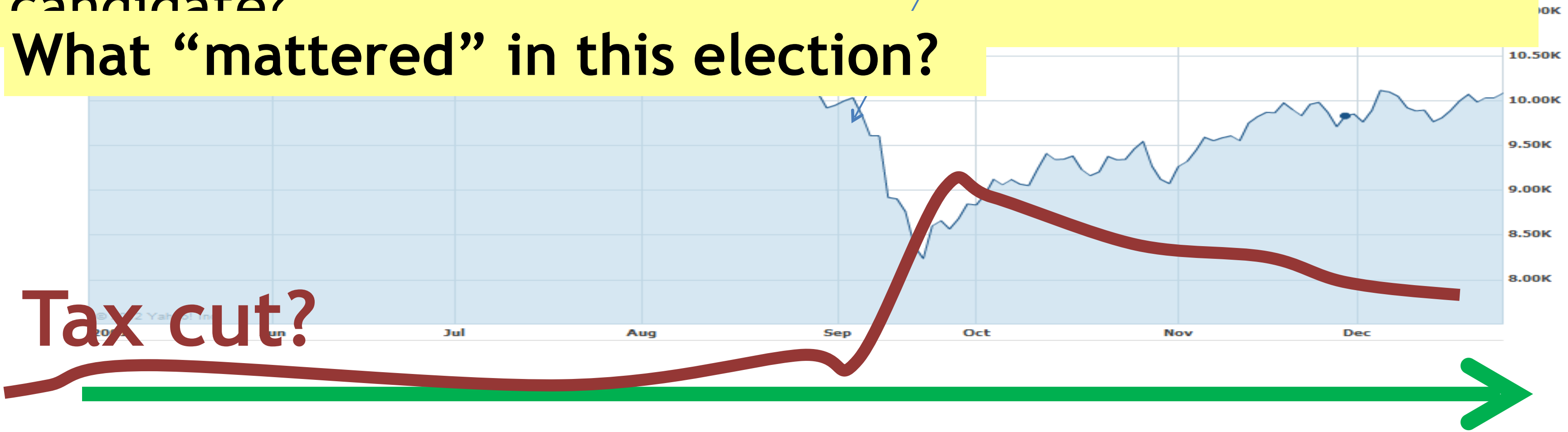
Any clues in the companion news stream?

Dow Jones Industrial Average [Source: Yahoo Finance]

Analysis of Presidential Prediction Markets

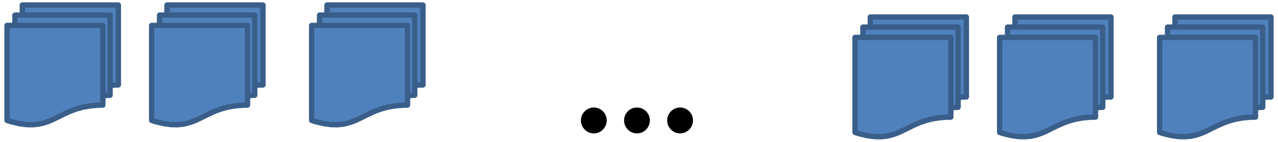
What might have caused the sudden drop of price for this candidate?

What "mattered" in this election?



Tax cut?

Time

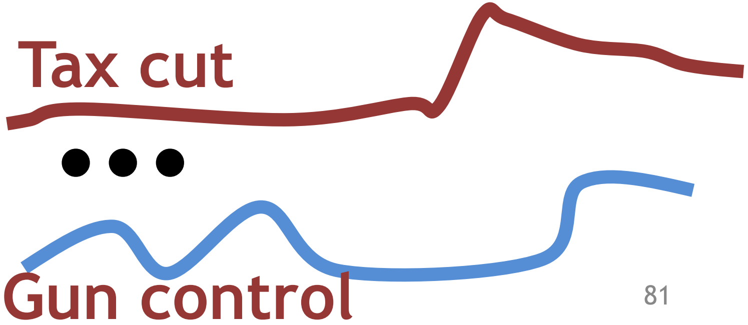


…

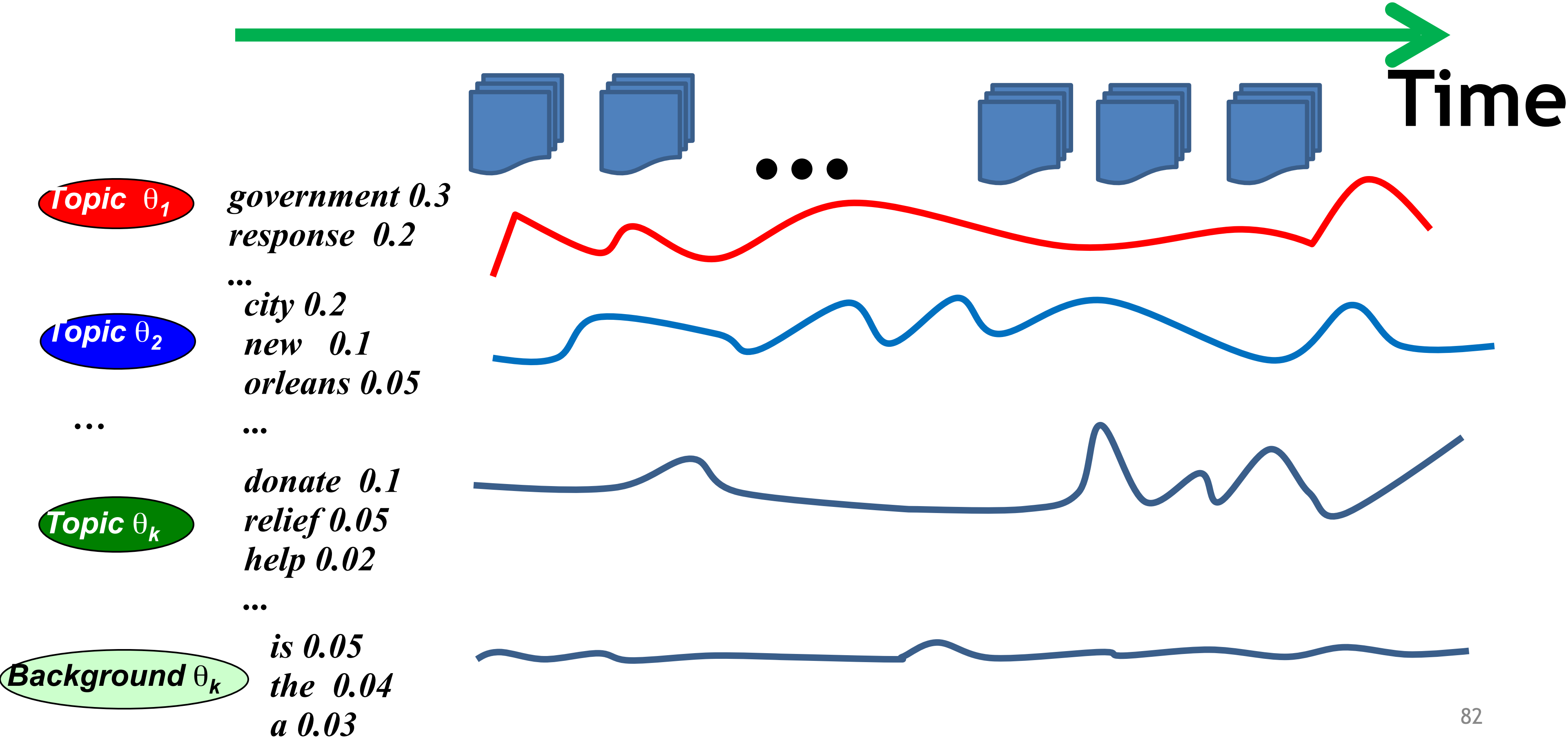
Any clues in the companion news stream?

Joint Analysis of Text and Time Series to Discover "Causal Topics"

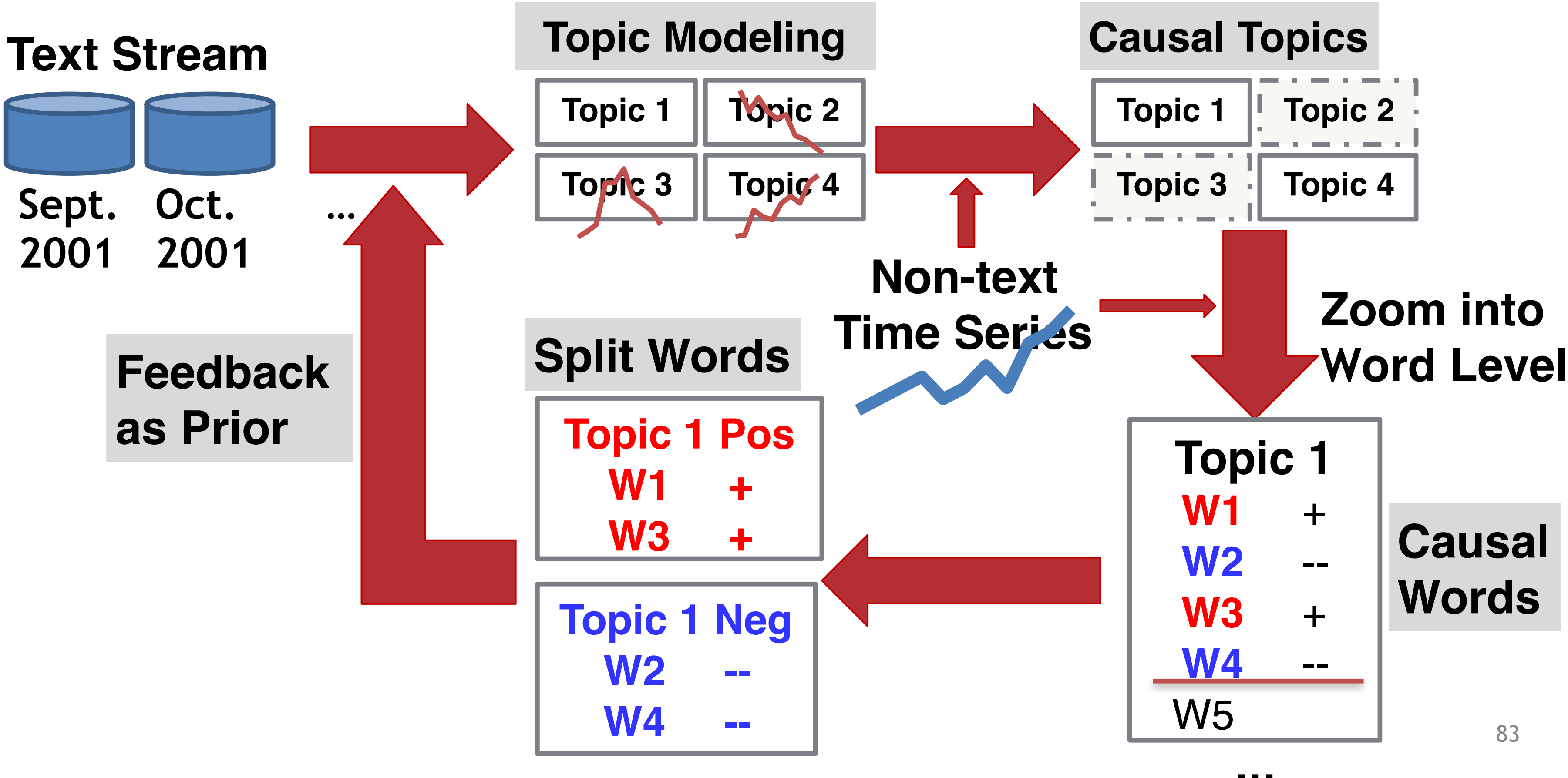
- Input:
  - Time series
  - Text data produced in a similar time period (text stream)
- Output
  - Topics whose coverage in the text stream has strong correlations with the time series ("causal" topics)



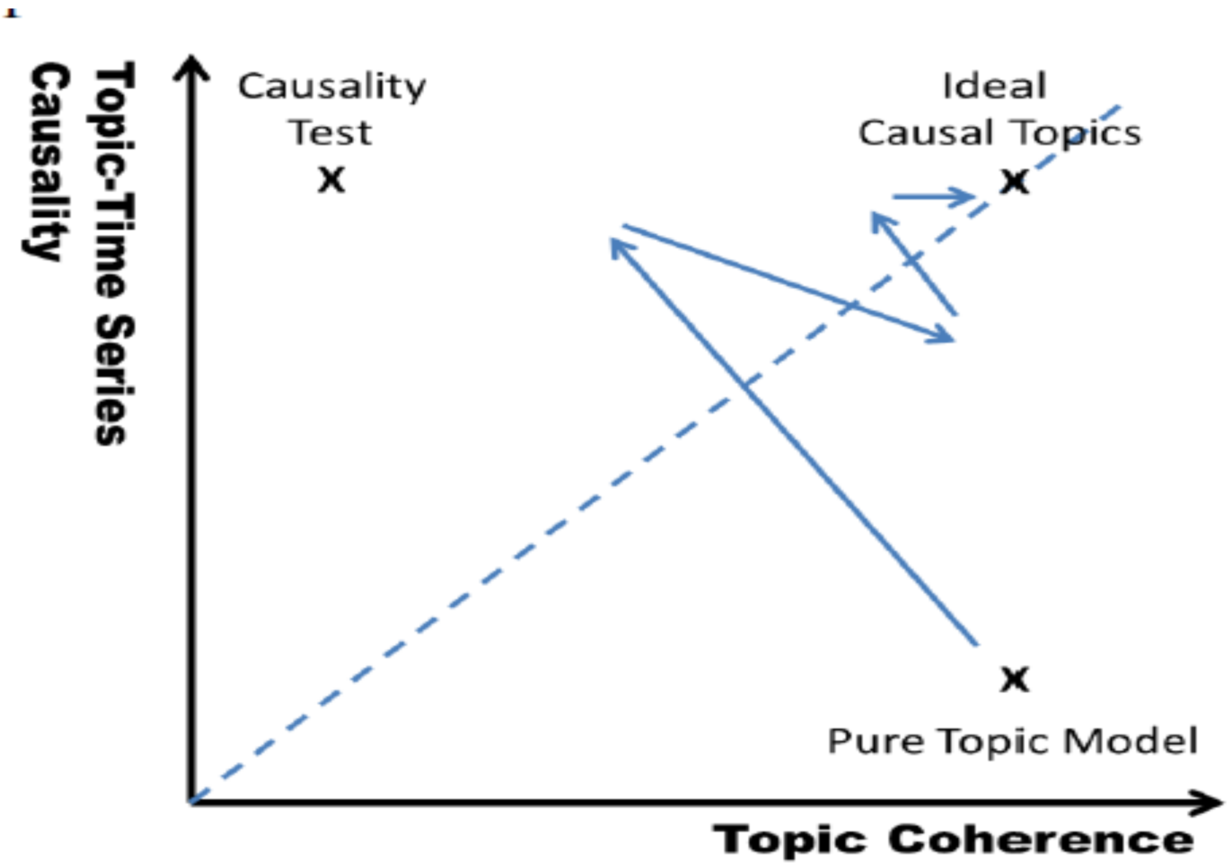
When a Topic Model Applied to Text Stream



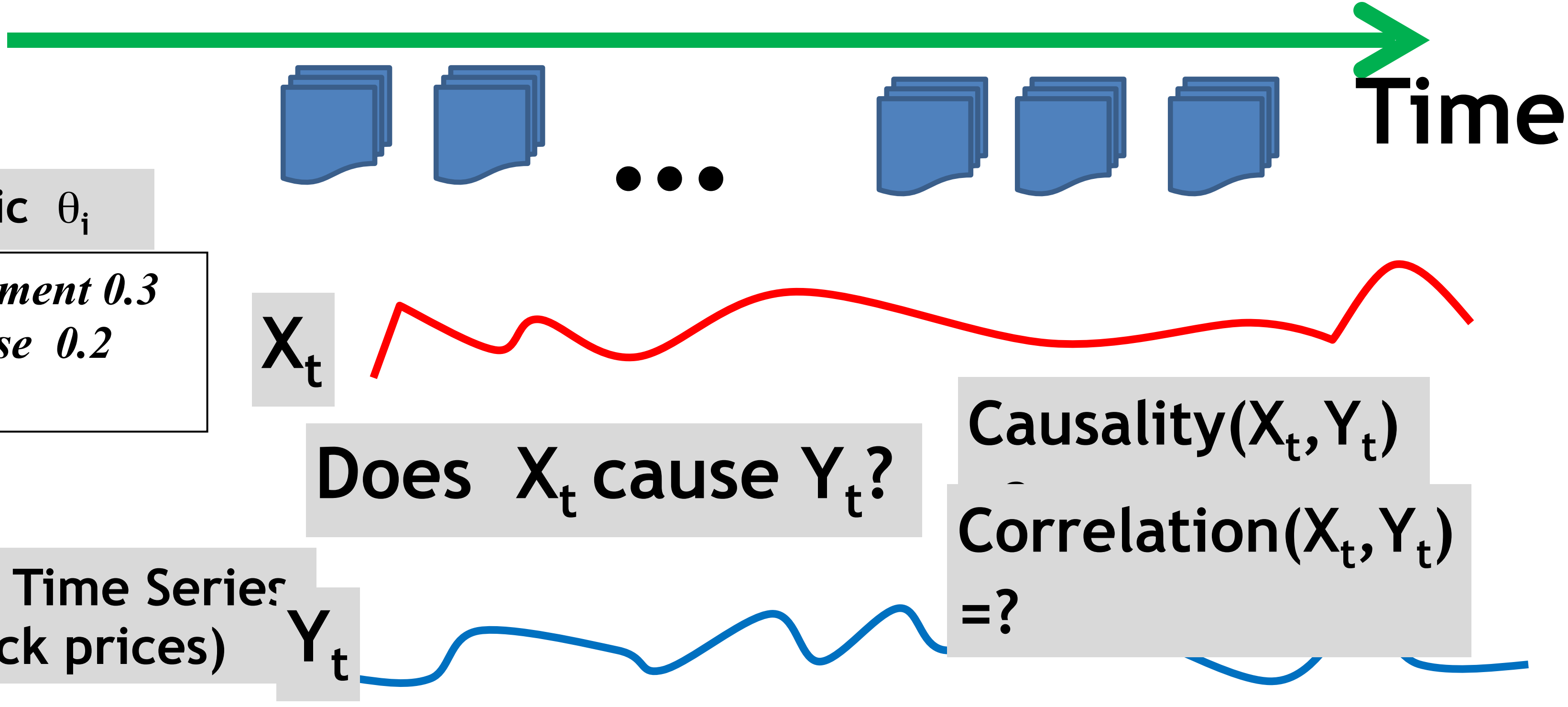
Iterative Causal Topic Modeling [Kim et al. 13]



Heuristic Optimization of Causality + Coherence



Measuring Causality (Correlation)



Topic θ_i

government 0.3
response 0.2
...

...

X_t

Does X_t cause Y_t?

Causality(X_t, Y_t)

Correlation(X_t, Y_t) =?

External Time Series (e.g. stock prices) Y_t

Granger Causality Test is often useful [Seth 07]

Topics in NY Times Correlated with Stocks [Kim et al. 13]: June 2000 ~ Dec. 2011

AAMRQ (American Airlines)	AAPL (Apple)
russia russian putin	paid notice st
europe european germany	russia russian europe
bush gore presidential	olympic games olympics
police court judge	she her ms
airlines airport air	oil ford prices
united trade terrorism	black fashion blacks
food foods cheese	**computer technology software**
nets scott basketball	**internet com web**
tennis williams open	football giants jets
awards gay boy	japan japanese plane
moss minnesota chechnya	

Topics are biased toward each time series

Major Topics in 2000 Presidential Election [Kim et al. 13]

Top Three Words in Significant Topics from NY Times
tax cut 1
screen pataki guiliani
enthusiasm door symbolic
oil energy prices
news w top
pres al vice
love tucker presented
partial **abortion** privatization
court supreme **abortion**
gun control nra

Text: NY Times (May 2000 - Oct. 2000)

Time Series: Iowa Electronic Market

http://tippie.uiowa.edu/iem/

Issues known to be important in the 2000 presidential election

What You Should Know

- What is a mixture language model, particularly mixture of unigram language model?
- What is the general form of the likelihood function of a mixture model?
- Why can the two-component mixture model with a background component language model factor out common words?
- What is PLSA and how does it work?
- How does EM algorithm work for PLSA?
- Why do we want to add a prior to PLSA and how can we modify the EM algorithm to incorporate a conjugate prior?
- Why is contextual topic modeling interesting? Can you give multiple examples of applications of contextual topic modeling?