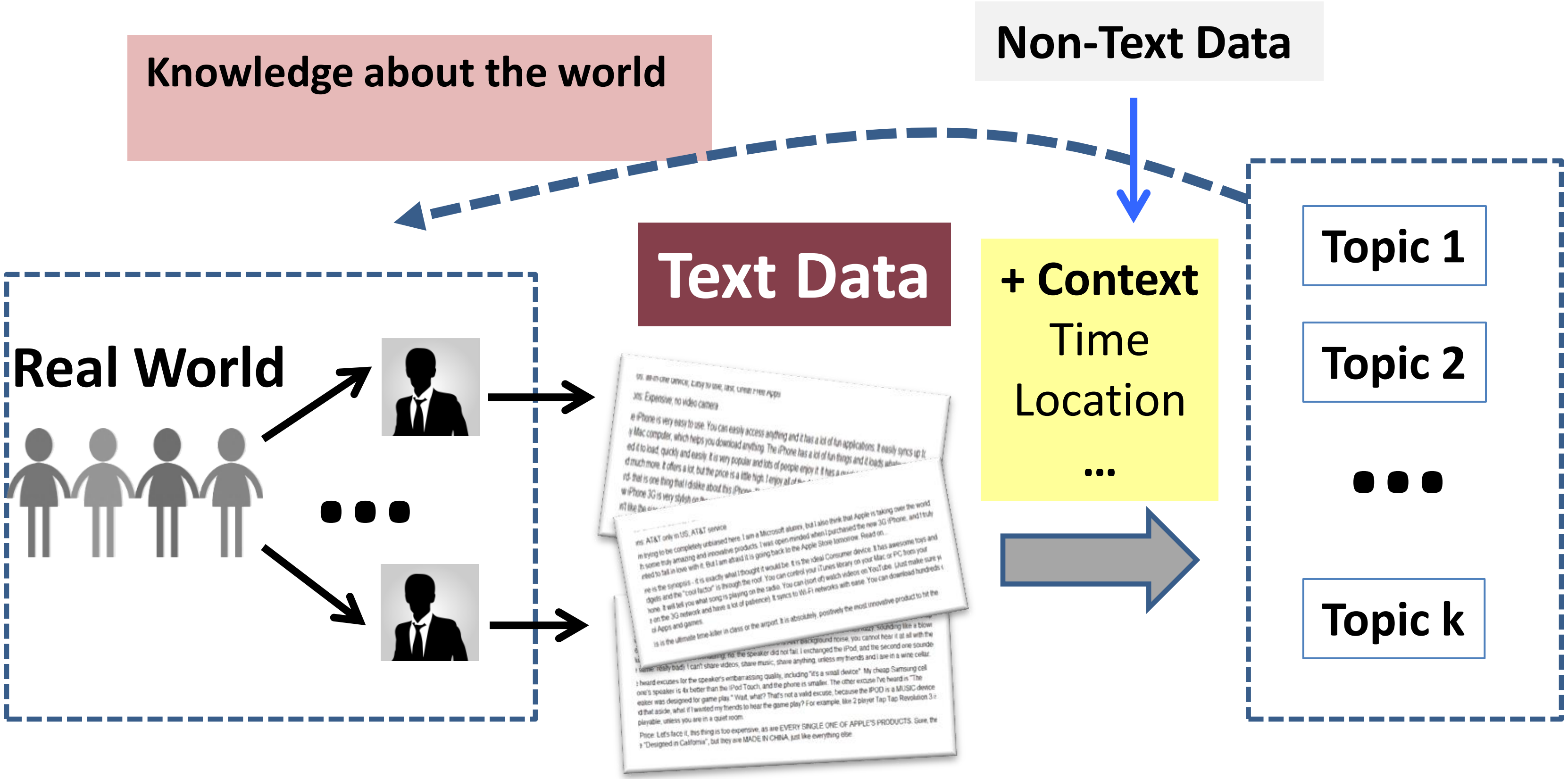
Topic Mining and Analysis: Motivation

- 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



Knowledge about the world

Non-Text Data

Real World

Text Data

+ Context
Time
Location
…

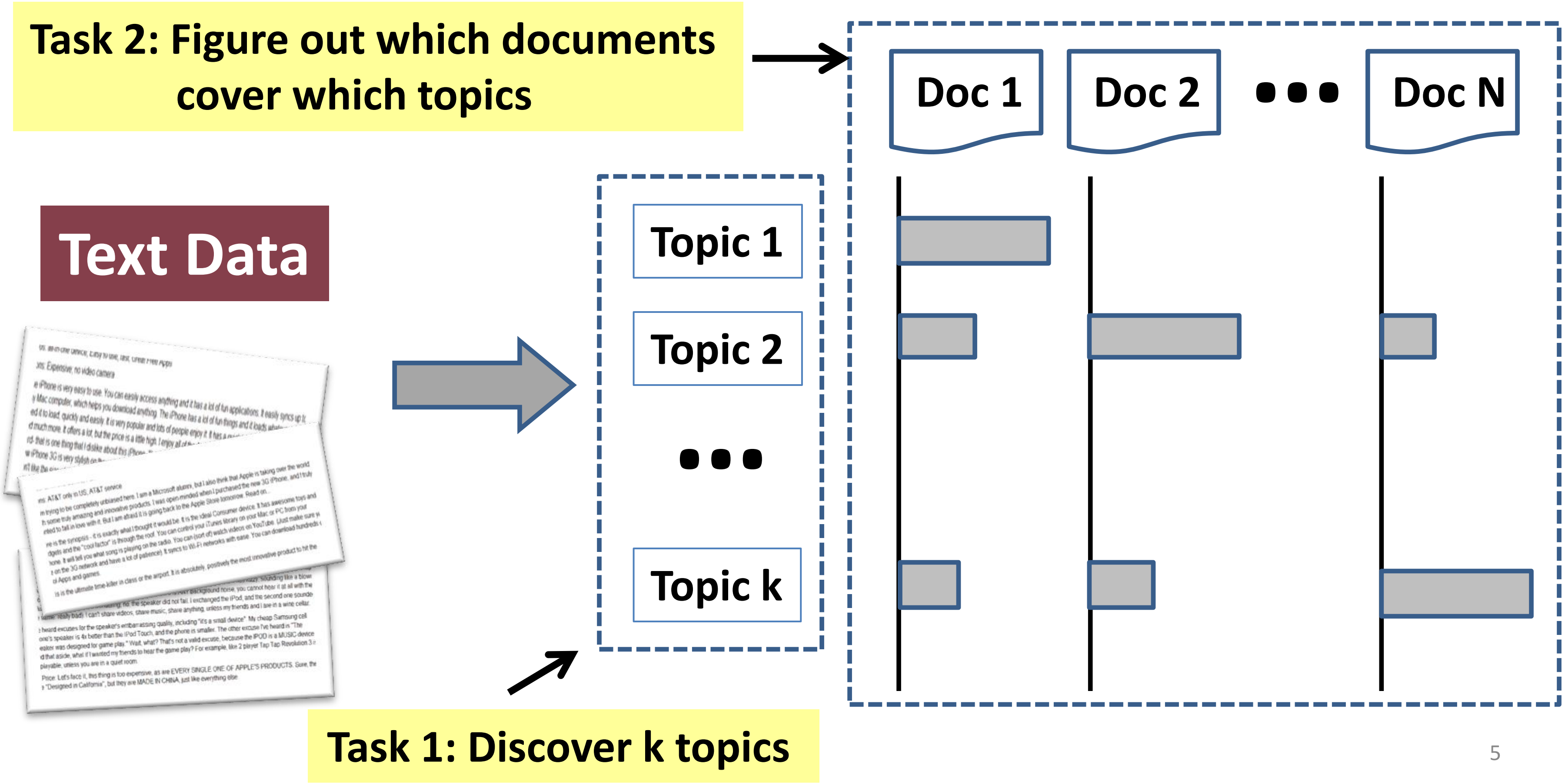
Topic 1

Topic 2

…

Topic k

Tasks of Topic Mining and Analysis



Task 2: Figure out which documents cover which topics

Text Data

Task 1: Discover k topics

Formal Definition of Topic Mining and Analysis

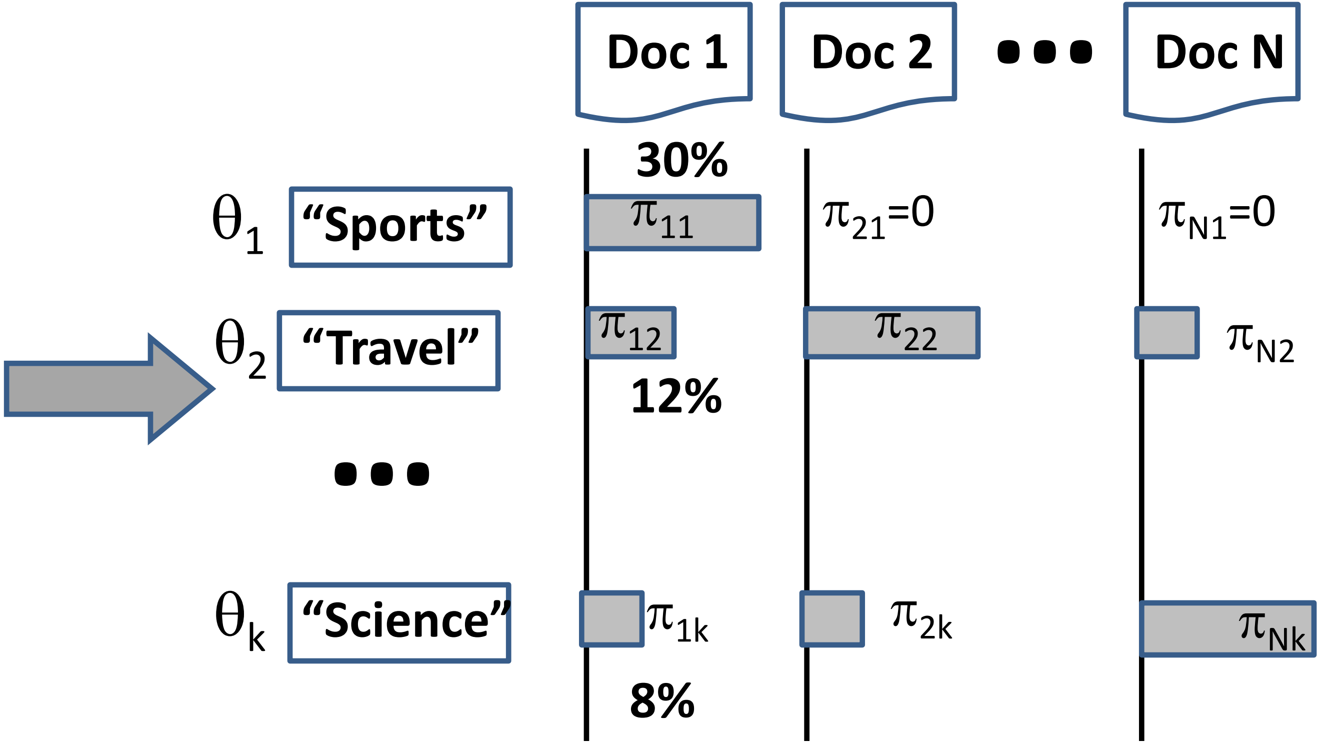
- Input
  - A **collection** of **N** text documents **$C=\{d_1, \ldots, d_N\}$**
  - **Number of topics**: **k**
- Output
  - **k topics**: **$\{ \theta_1, \ldots, \theta_k \}$**
  - **Coverage of topics in each d_i**: **$\{ \pi_{i1}, \ldots, \pi_{ik} \}$**
  - π_{ij} = prob. of d_i covering topic θ_j

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

How to define θ_i ?

Initial Idea: Topic = Term

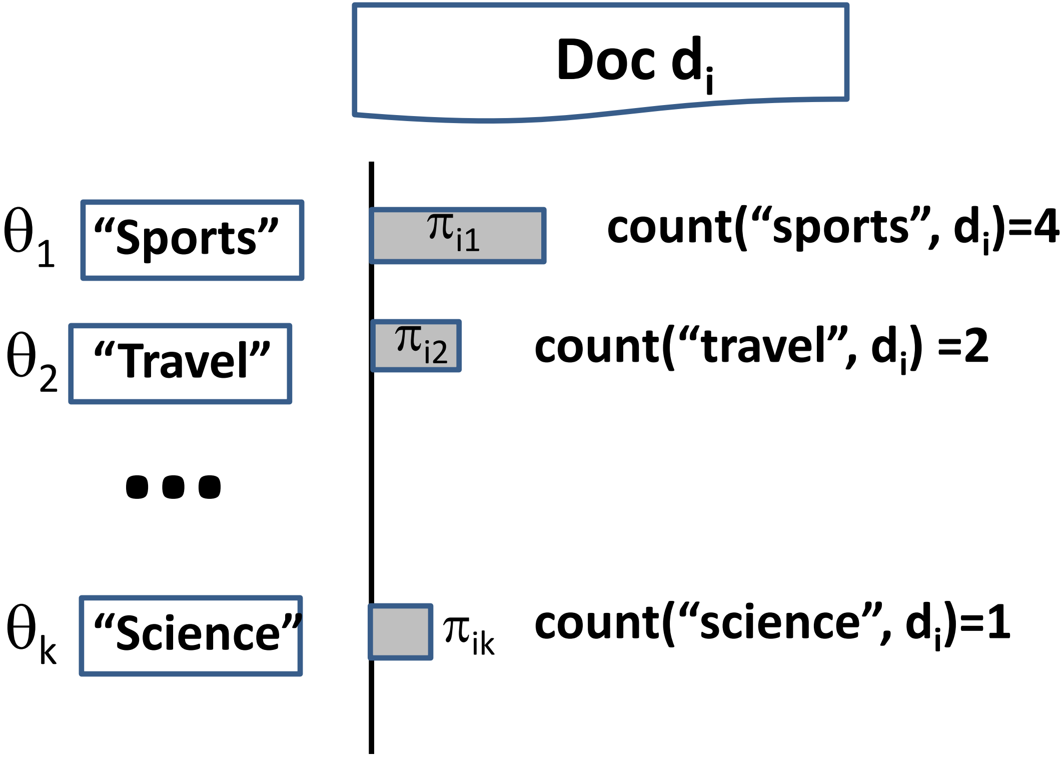
Text Data



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}



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

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

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
  - Can only represent simple/general topics
  - Can’t represent complicated topics

➔ Topic = {Multiple Words}

- Incompleteness in vocabulary coverage
  - Can’t capture variations of vocabulary (e.g., related words)

+ weights on words

- Word sense ambiguity
  - A topical term or related term can be ambiguous (e.g., basketball star vs. star in the sky)

➔ Split an ambiguous word

A probabilistic topic model can do all these!

Improved Idea: Topic = Word Distribution

θ_1 "Sports"

P(w|θ_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|θ_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|θ_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
…	

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

Vocabulary Set: V={w1, w2,….}

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

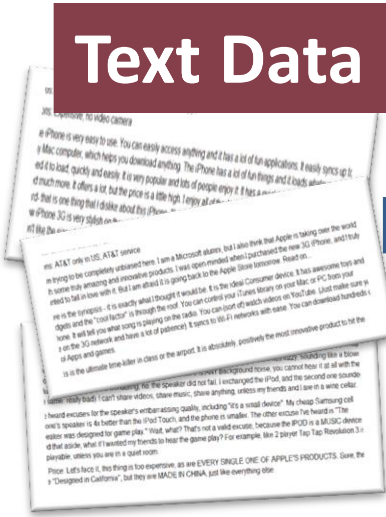
$$\sum_{w \in V} p(w \mid \theta_i) = 1$$

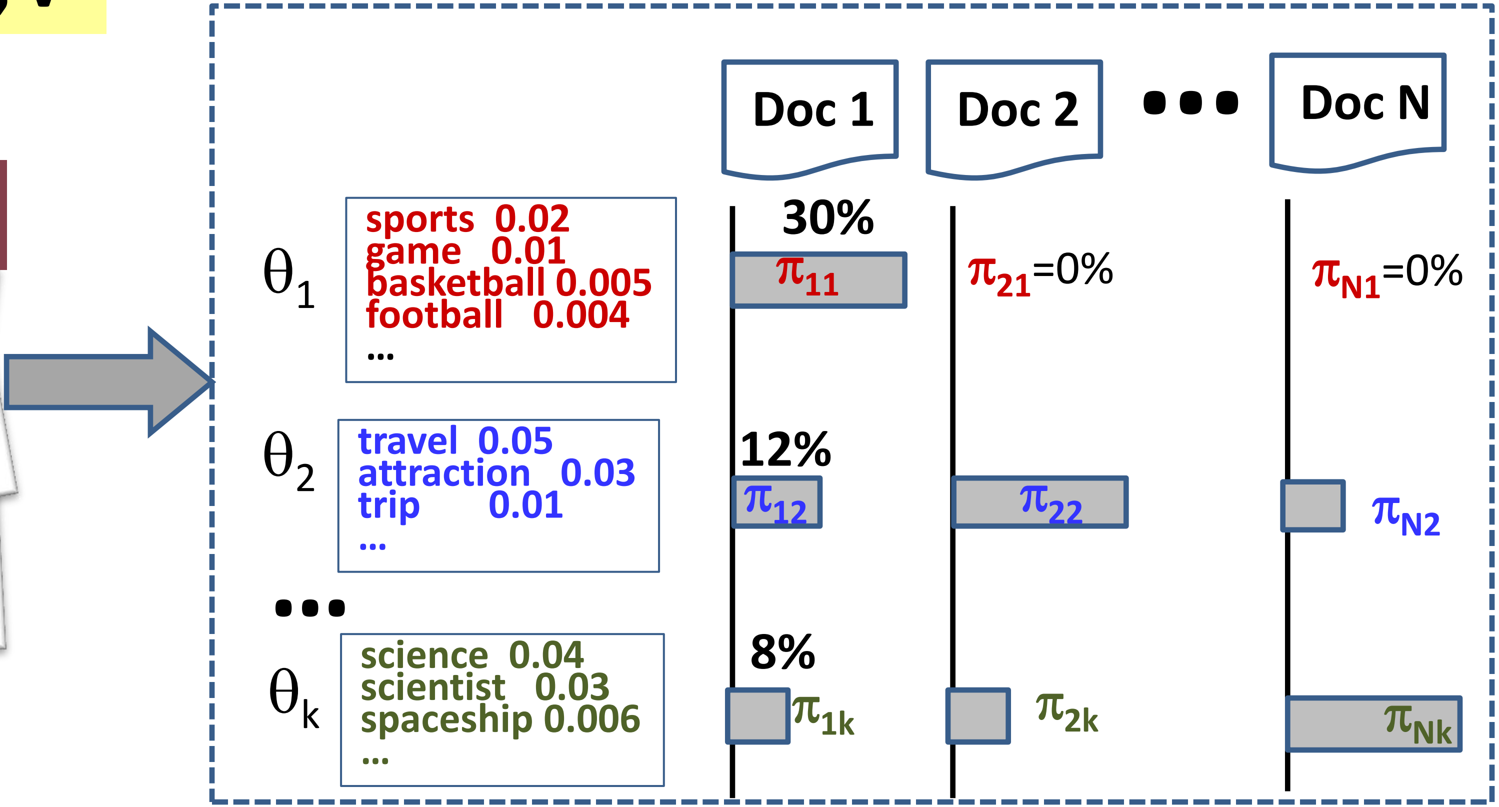
$$\sum_{j=1}^{k} \pi_{ij} = 1$$

The Computation Task

INPUT: C, k, V

OUTPUT: { θ_1, …, θ_k }, { π_{i1}, …, π_{ik} }





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
…

Generative Model for Text Mining

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

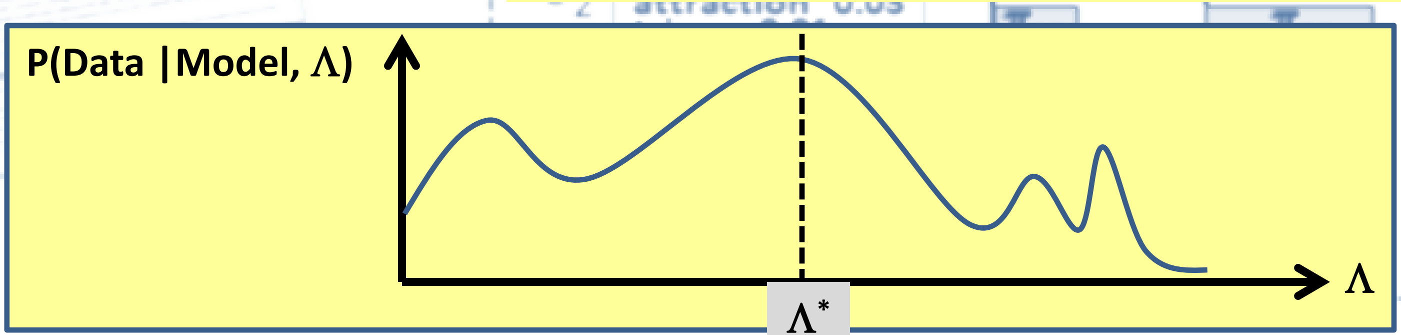
Λ=({ θ_1, …, θ_k }, { π_{11}, …, π_{1k} }, …, { π_{N1}, …, π_{Nk} })

Text Data

How many parameters in total?

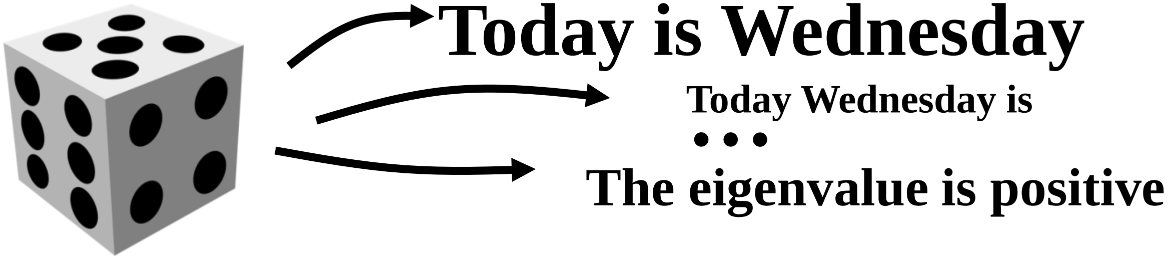
Parameter Estimation/ Inferences

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



What Is a Statistical Language Model (LM)?

- 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!
- Can also be regarded as a probabilistic mechanism for “generating” text – thus also called a “generative” model



Today is Wednesday

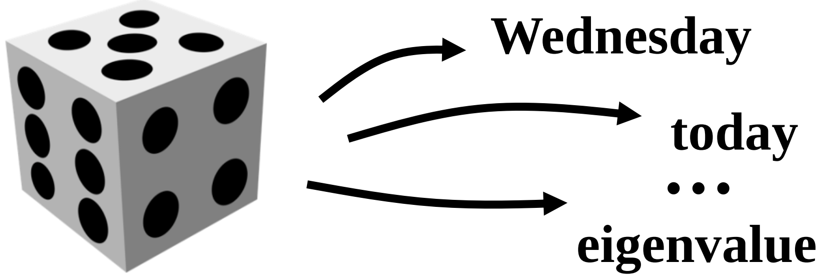
Today Wednesday is

…

The eigenvalue is positive

The Simplest Language Model: Unigram LM

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

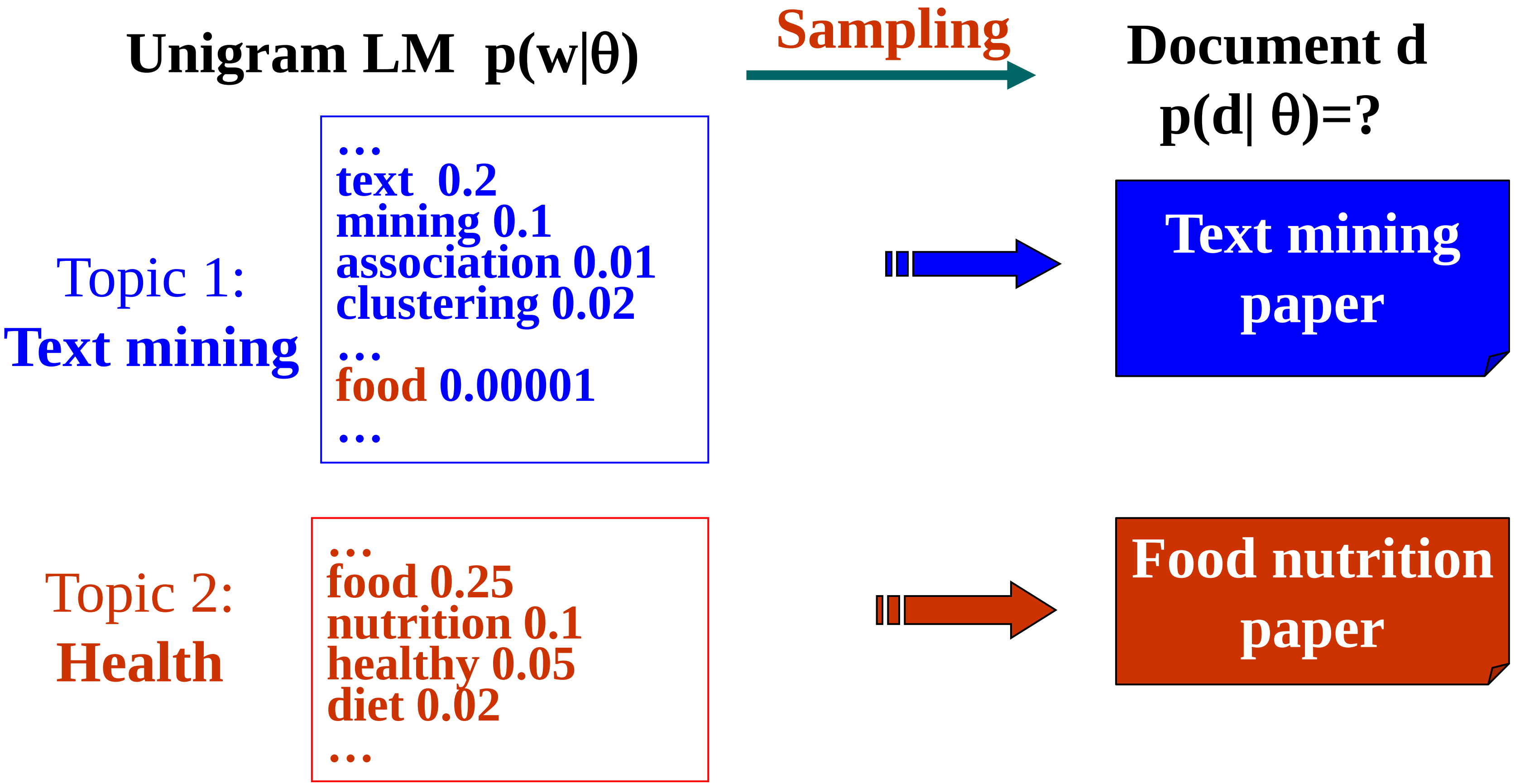


p("today is Wed")

= p("today")p("is")p("Wed")

= 0.0002 × 0.001 × 0.000015

Text Generation with Unigram LM



Unigram LM p(w|θ)

Sampling

Document d

p(d| θ)=?

Topic 1:

Text mining

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

Text mining paper

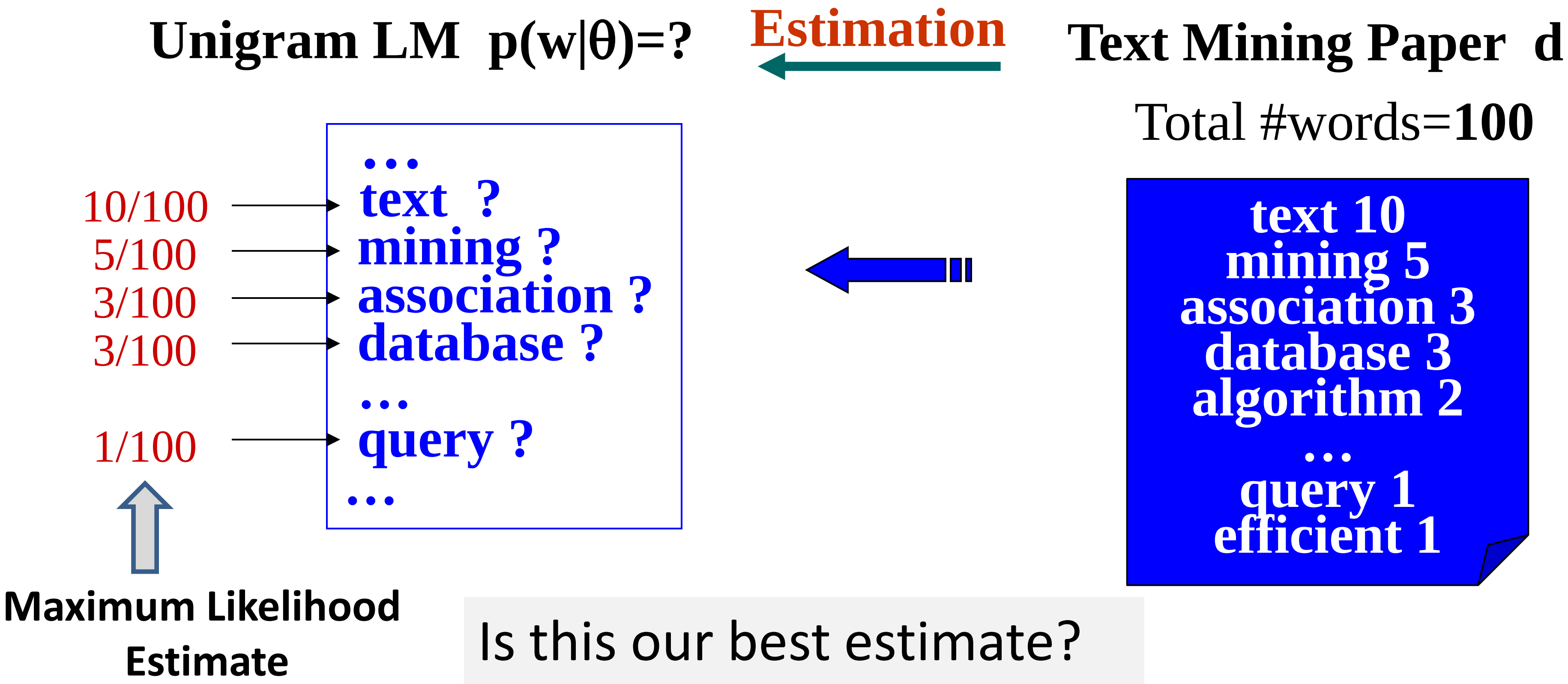
Topic 2:

Health

…
food 0.25
nutrition 0.1
healthy 0.05
diet 0.02
…

Food nutrition paper

Estimation of Unigram LM



Unigram LM p(w|θ)=? **Estimation** **Text Mining Paper d**

Total #words=**100**

	…
10/100	text ?
5/100	mining ?
3/100	association ?
3/100	database ?
	…
1/100	query ?
	…

text 10
mining 5
association 3
database 3
algorithm 2
…
query 1
efficient 1

Maximum Likelihood Estimate

Is this our best estimate?

How do we define "best"?

Maximum Likelihood vs. Bayesian

- Maximum likelihood estimation
  - "Best" means "data likelihood reaches maximum"

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

  - Problem: Small sample

- Bayesian estimation:

Bayes Rule

$$p(X \mid Y) = \frac{p(Y \mid X)p(X)}{p(Y)}$$

  - "Best" means being consistent with our "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

Bayesian inference: f(θ)=?

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

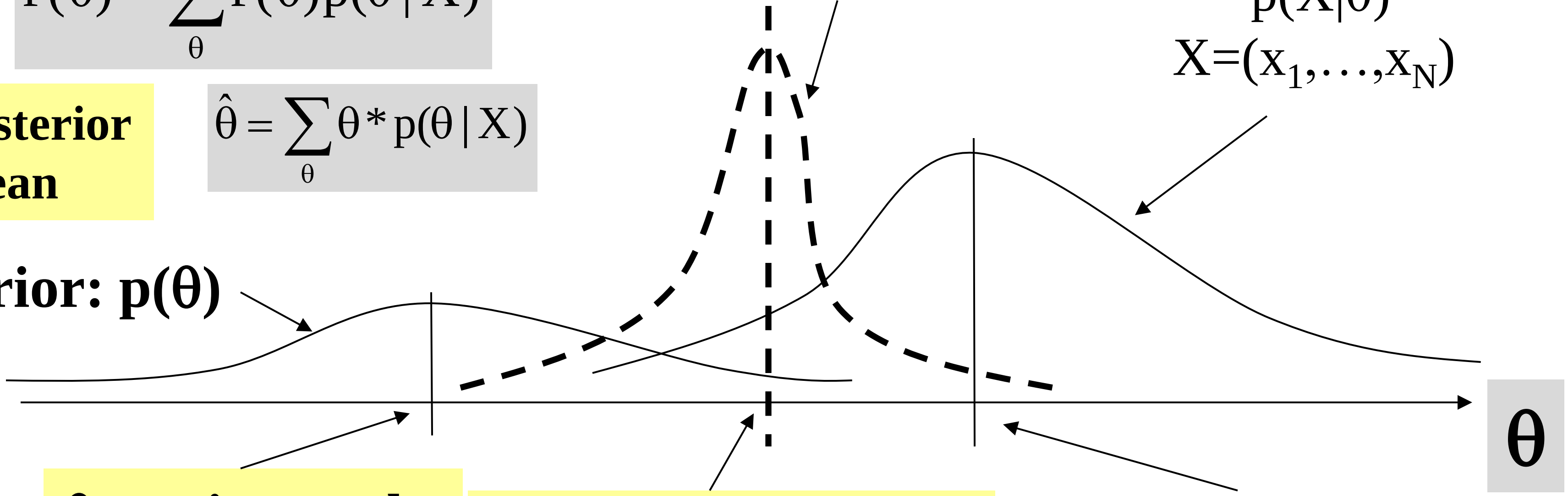
Posterior Mean

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

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

Likelihood:
$p(X|\theta)$
$X=(x_1,\ldots,x_N)$

Prior: $p(\theta)$



θ_0: prior mode

θ_1: posterior mode

θ_{ml}: ML estimate

Simplest Case of Topic Model: Mining One Topic

INPUT: C={d}, V

OUTPUT: { θ}

Text Data

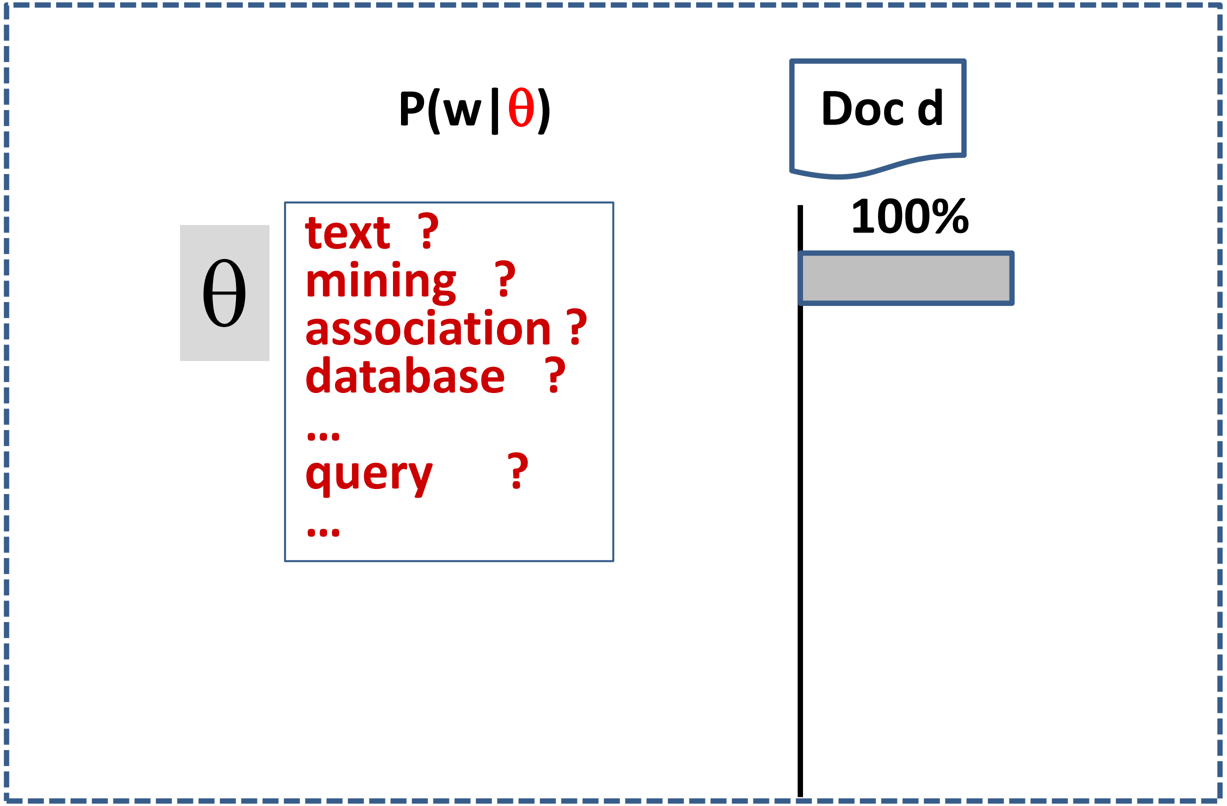
ay, we all know the good things the iPod Touch has to offer, and those reasons are why I am BARELY holding on
e thing. I am still considering taking it back. Here's why.

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 blow
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
e same: really bad). I can't share videos, share music, share anything, unless my friends and I are in a wine cellar.

e 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
e "Designed in California", but they are MADE IN CHINA, just like everything else.

Microphone: WHY IS more »



Language Model Setup

- **Data**: Document d= $x_1\ x_2 \ldots x_{|d|}$, $x_i \in V=\{w_1 ,\ldots, w_M\}$ is a word

- **Model**: Unigram LM θ(=topic) : $\{\theta_i=p(w_i\,|\theta)\}$, 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

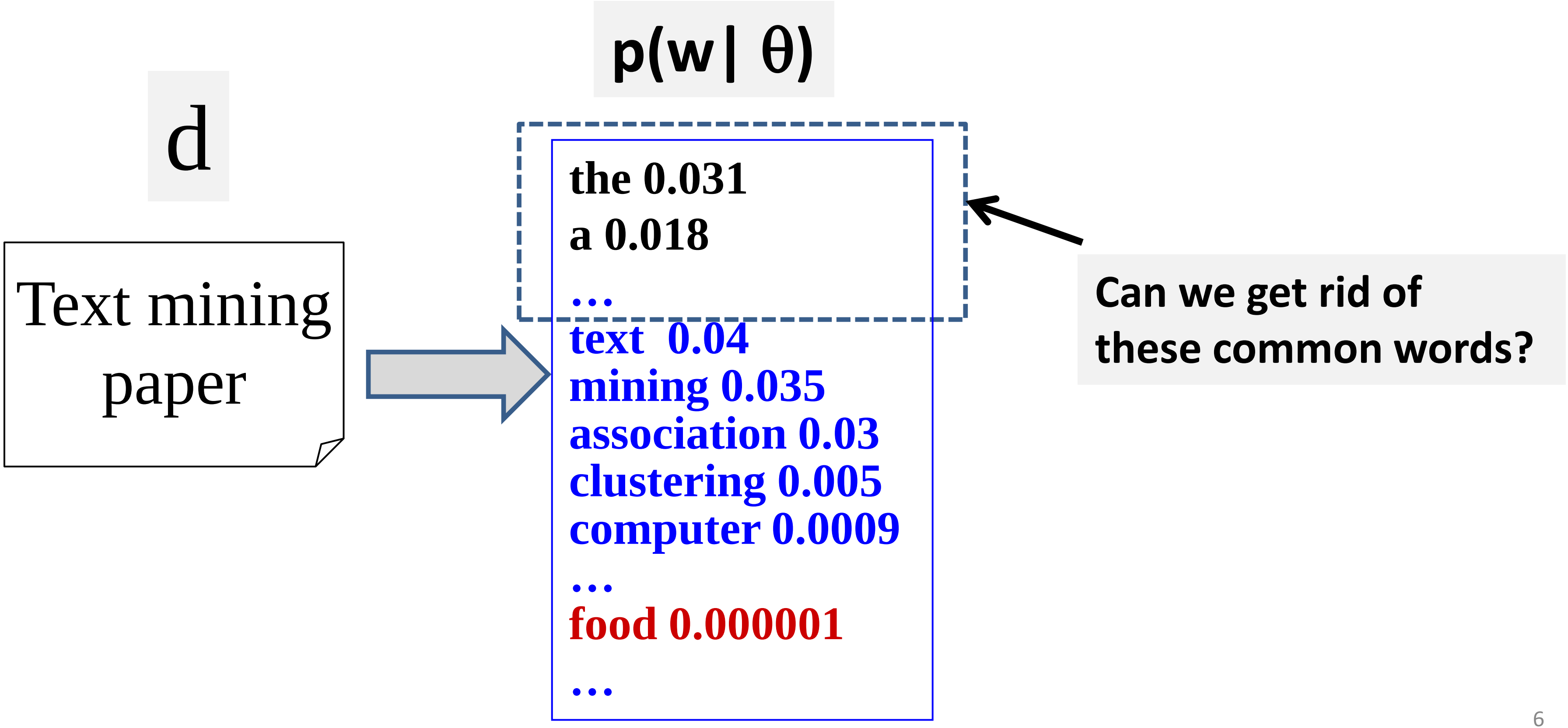
Lagrange function: $$f(q|d) = \sum_{i=1}^{M} c(w_i,d)\log q_i + /\,(\sum_{i=1}^{M} q_i - 1)$$

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

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

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

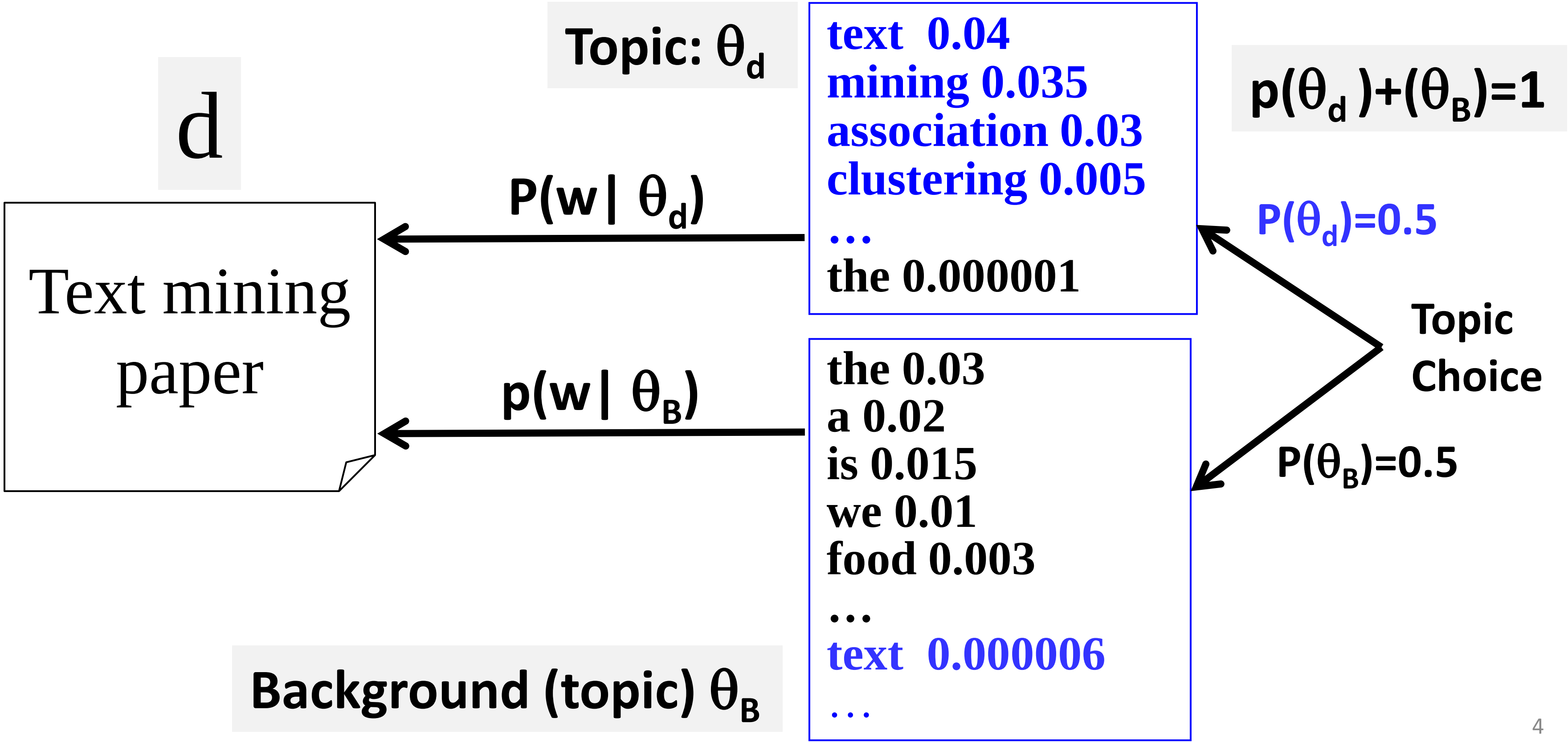
…

food 0.000001

…

Can we get rid of these common words?

Generate d Using Two Word Distributions



d

Text mining paper

Topic: θ_d

$P(w| \theta_d)$

text 0.04
mining 0.035
association 0.03
clustering 0.005
…
the 0.000001

$p(w| \theta_B)$

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

Background (topic) θ_B

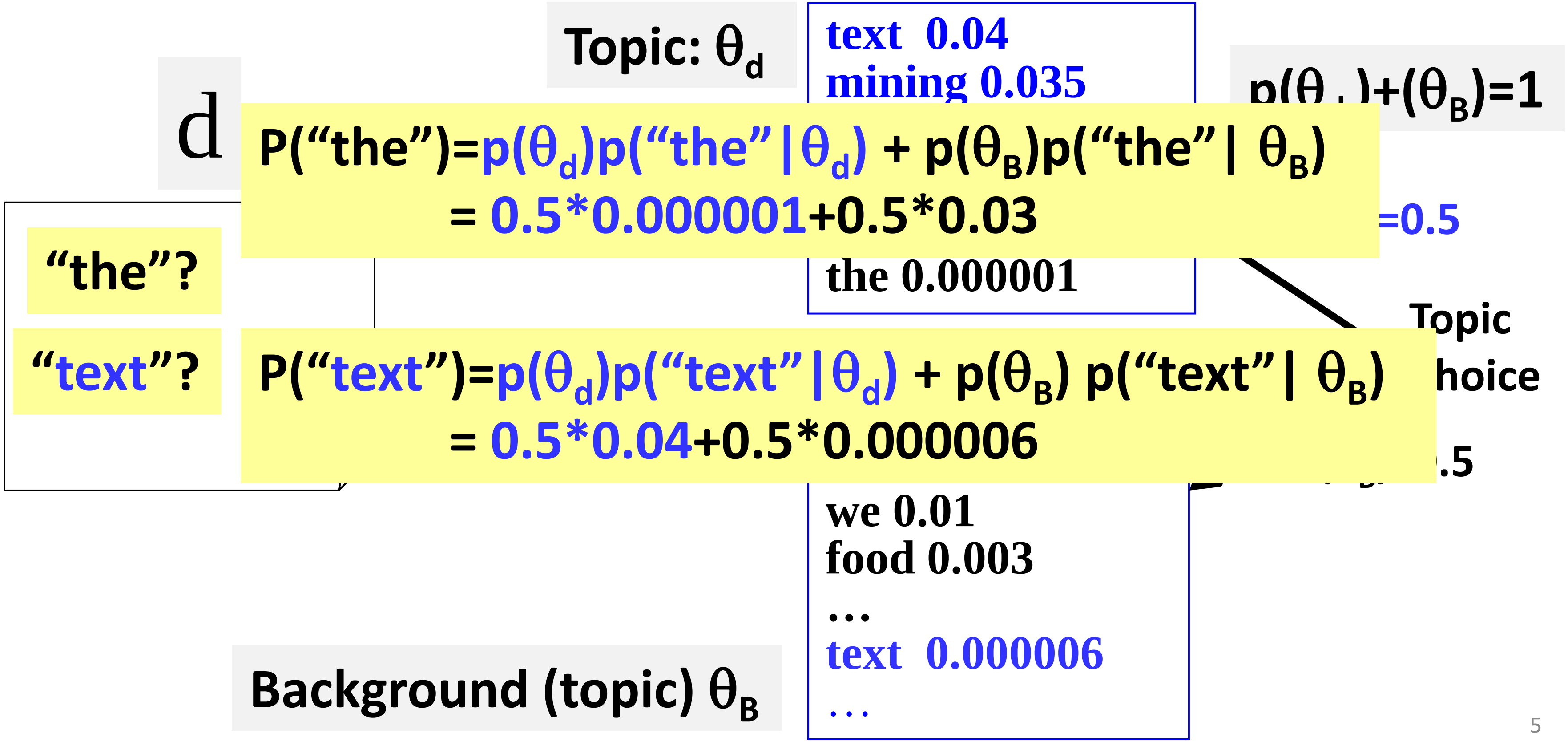
$p(\theta_d)+(\theta_B)=1$

$P(\theta_d)=0.5$

Topic Choice

$P(\theta_B)=0.5$

What’s the probability of observing a word w?



Topic: θ_d

text 0.04
mining 0.035
the 0.000001

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

$=0.5$

Topic
Choice

.5

d

“the”?

“text”?

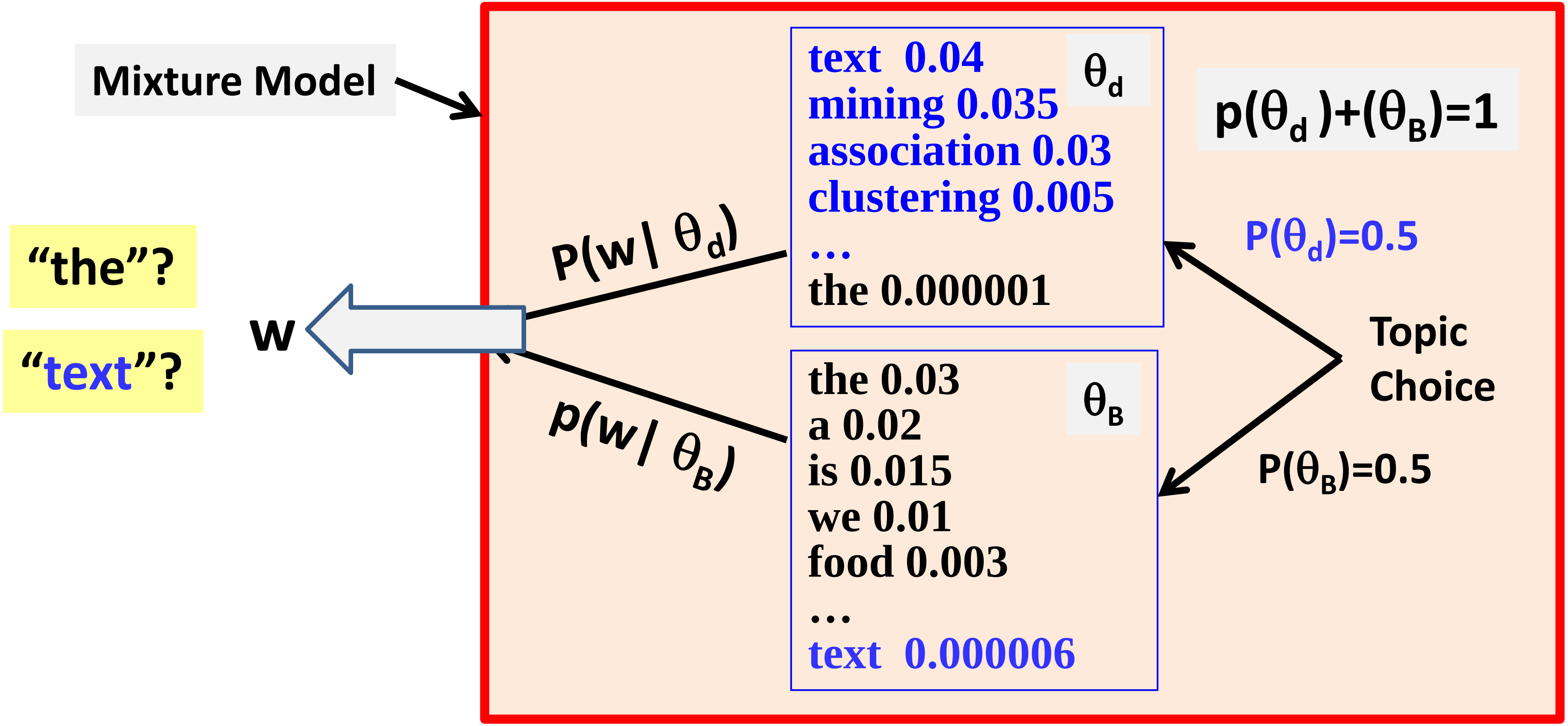
$P(\text{“the”})=p(\theta_d)p(\text{“the”}|\theta_d) + p(\theta_B)p(\text{“the”}|\ \theta_B)$
$= 0.5*0.000001+0.5*0.03$

$P(\text{“text”})=p(\theta_d)p(\text{“text”}|\theta_d) + p(\theta_B)\ p(\text{“text”}|\ \theta_B)$
$= 0.5*0.04+0.5*0.000006$

we 0.01
food 0.003
…
text 0.000006
…

Background (topic) θ_B

The Idea of a Mixture Model



Mixture Model

"the"?

"text"?

w

$P(w|\theta_d)$

$p(w|\theta_B)$

θ_d

text	0.04
mining	0.035
association	0.03
clustering	0.005
…	
the	0.000001

θ_B

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

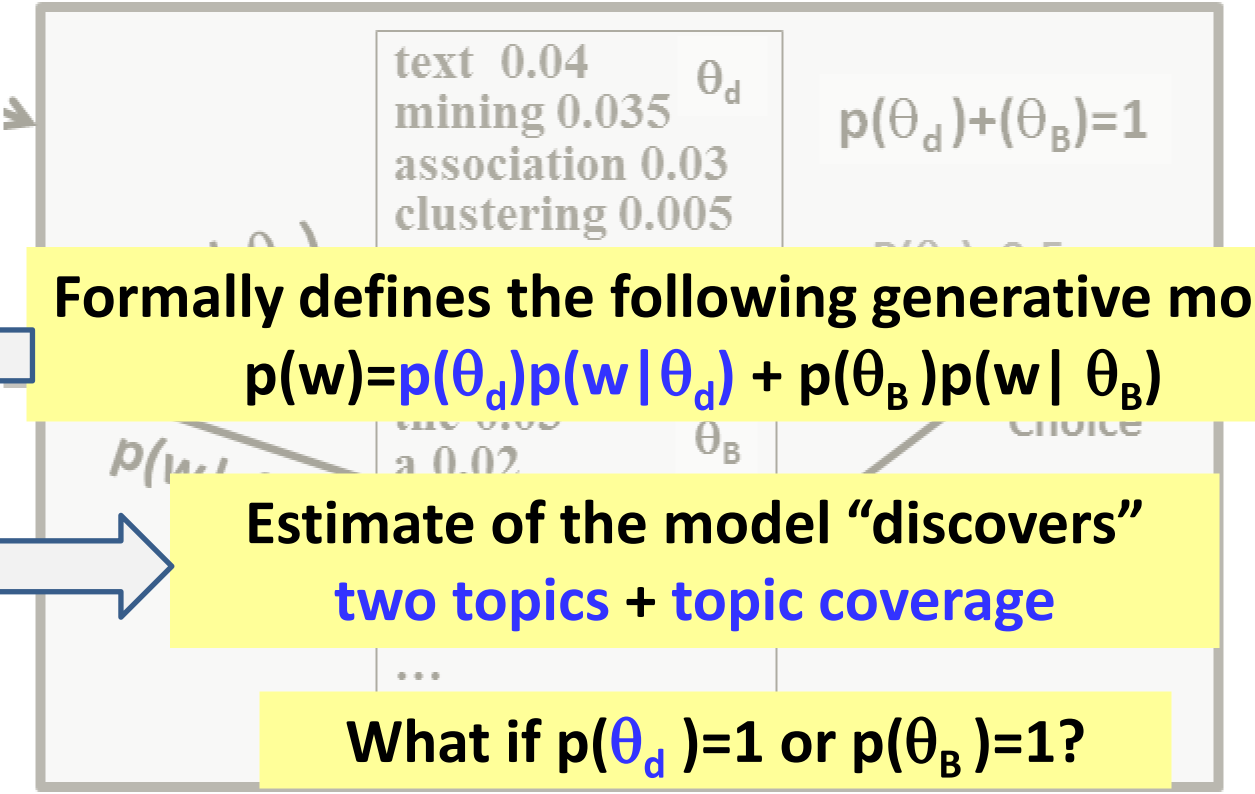
$p(\theta_d)+(\theta_B)=1$

$P(\theta_d)=0.5$

Topic Choice

$P(\theta_B)=0.5$

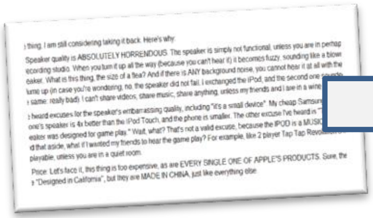
As a Generative Model…



w

Formally defines the following generative model:

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



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 sounds
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 "
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 Revolution
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.

Estimate of the model "discovers"
two topics + topic coverage

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

Mixture of Two Unigram Language Models

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

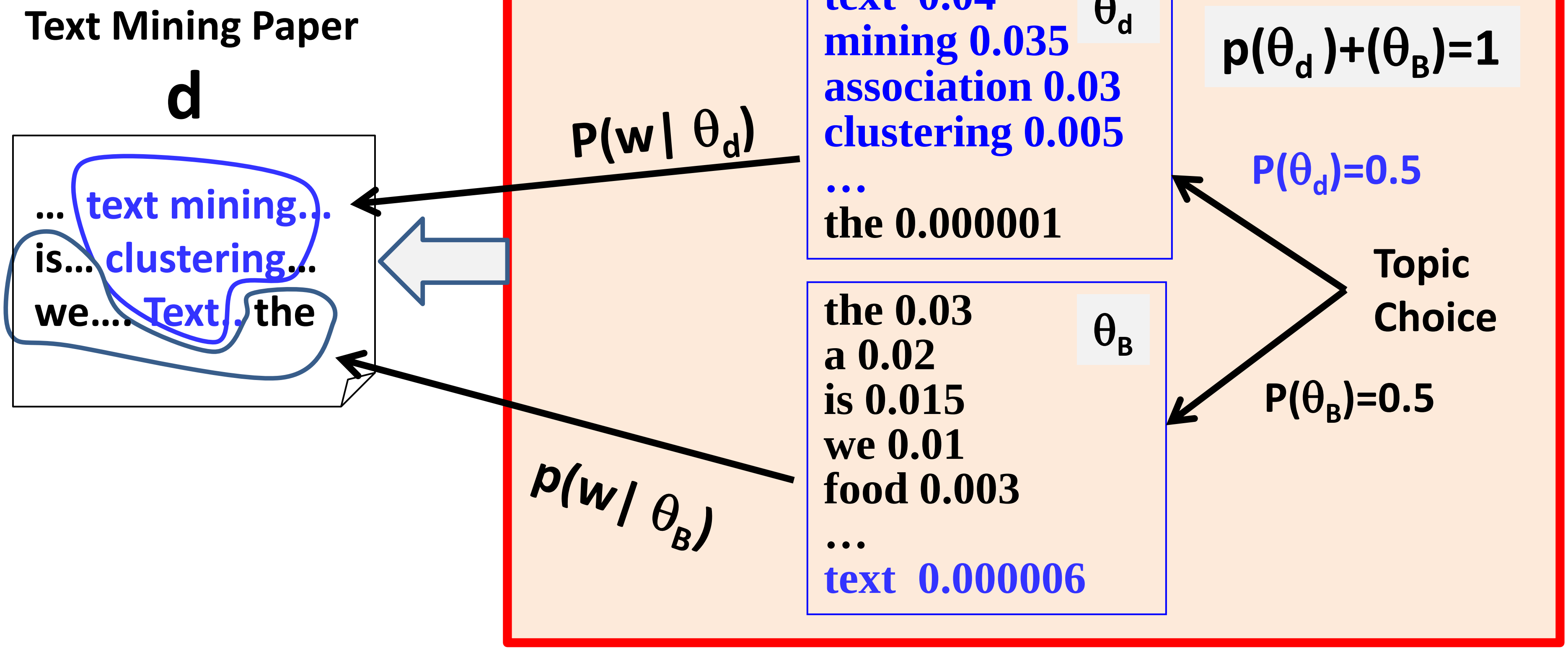
$$p(d|\Lambda) = \prod_{i=1}^{|d|} p(x_i|\Lambda) = \prod_{i=1}^{|d|} [p(\theta_d)p(x_i|\theta_d) + p(\theta_B)p(x_i|\theta_B)]$$

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

- 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



Text Mining Paper

d

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

$P(w| \theta_d)$

text 0.04
mining 0.035
association 0.03
clustering 0.005
…
the 0.000001

θ_d

$p(w| \theta_B)$

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

θ_B

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

$P(\theta_d)=0.5$

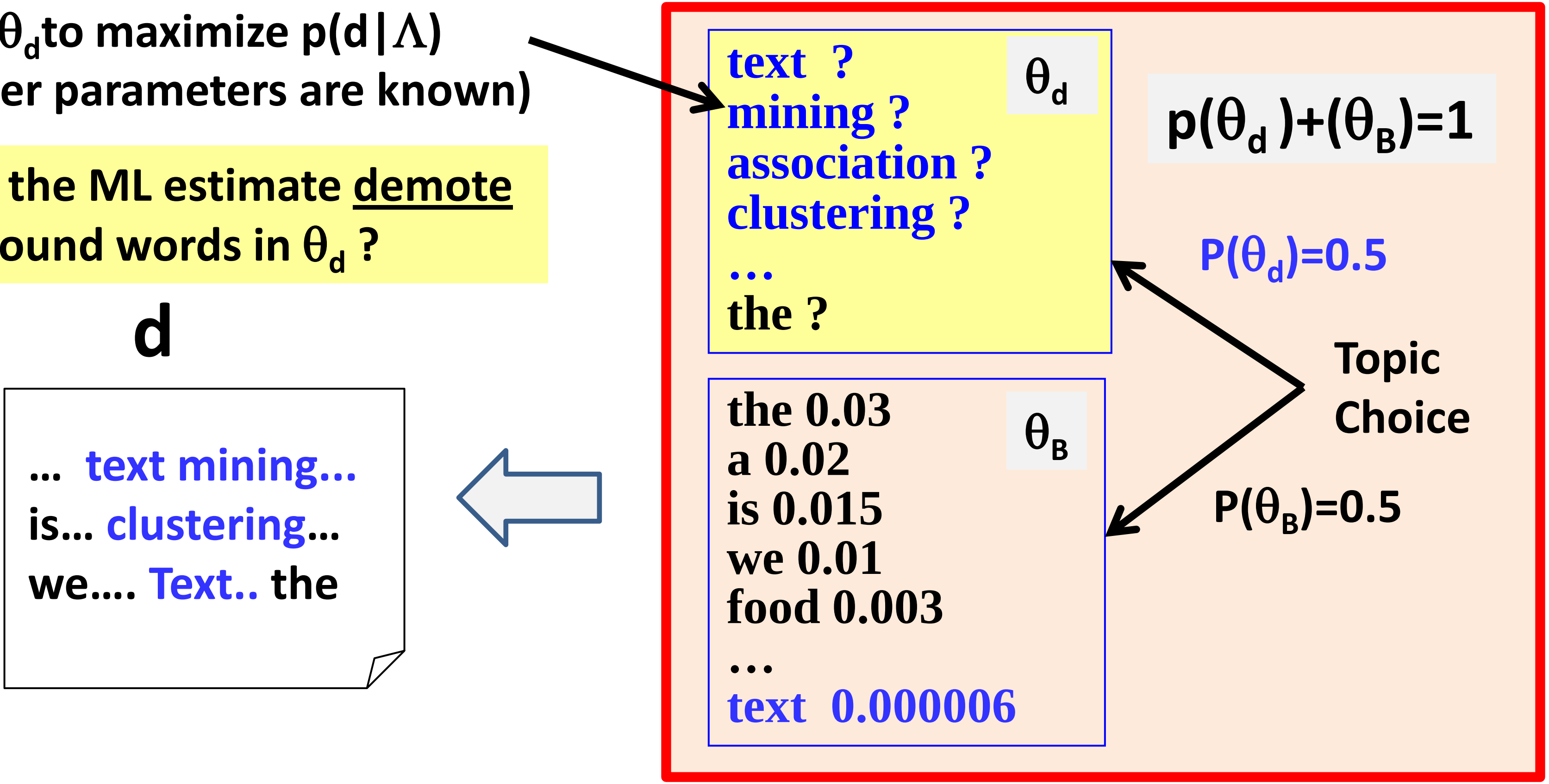
Topic Choice

$P(\theta_B)=0.5$

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

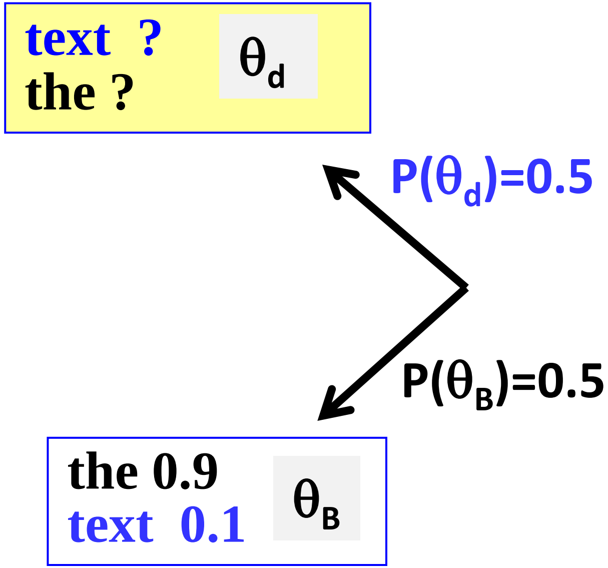
Likelihood:

$$P(\text{“text”}) = p(\theta_d)p(\text{“text”}|\theta_d) + p(\theta_B)p(\text{“text”}|\theta_B)$$
$$= 0.5 * p(\text{“text”}|\theta_d) + 0.5 * 0.1$$

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

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

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



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 $p(\text{“text”}|\theta_d) + p(\text{“the”}|\theta_d) = 1$

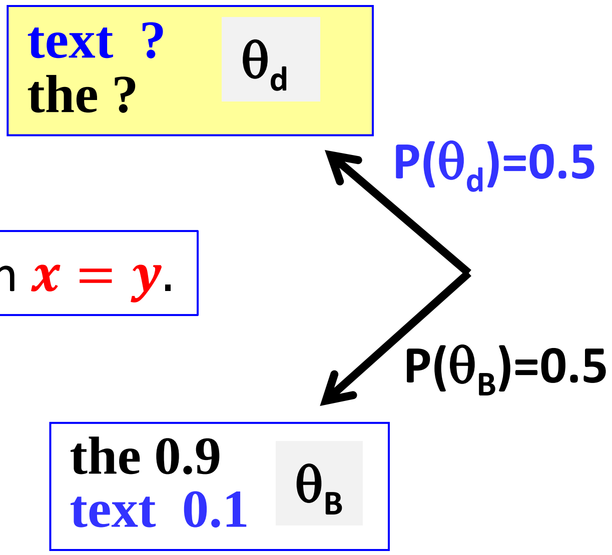
"Collaboration" and "Competition" of θ_d and θ_B

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

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

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

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

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

If $x + y = constant$, then xy reaches maximum when $x = y$.

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

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

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

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

➔ $p(\text{“text”}|\theta_d)=0.9 \quad >> \quad p(\text{“the”}|\theta_d) =0.1$!

d’ = text the
the the
the …the

$$p(d'|\Lambda) = [0.5*p(\text{“text”}|\theta_d) + 0.5*0.1]$$
$$\text{x } [0.5*p(\text{“the”}|\theta_d) + 0.5*0.9]$$
$$\text{x } [0.5*p(\text{“the”}|\theta_d) + 0.5*0.9]$$
$$\text{x } [0.5*p(\text{“the”}|\theta_d) + 0.5*0.9]$$
$$\dots$$
$$\text{x } [0.5*p(\text{“the”}|\theta_d) + 0.5*0.9]$$

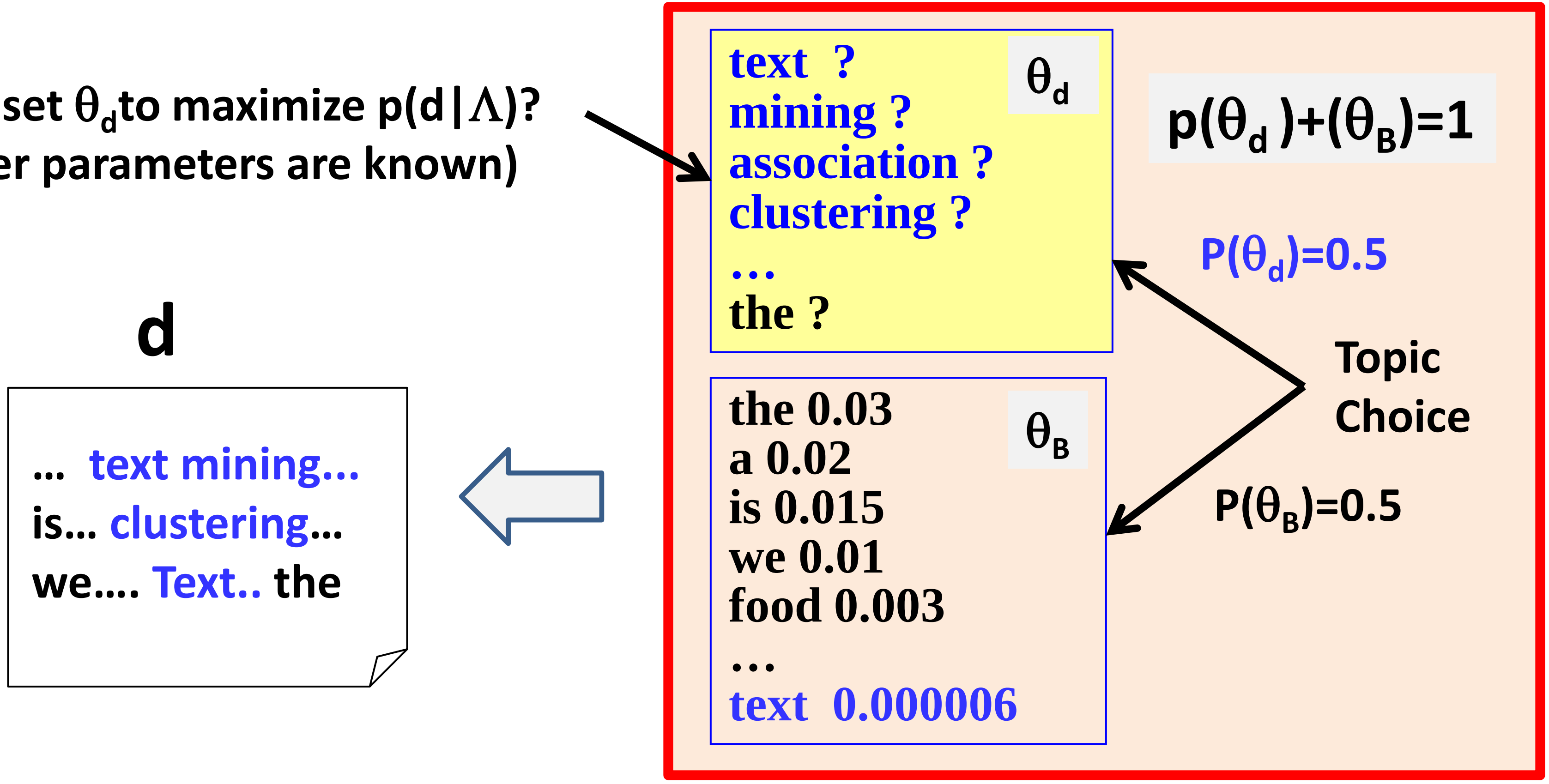
What if we increase $p(\theta_B)$?

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: $\mathbf{P(w|\ \theta_d)}$

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 ?

θ_B

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

$\mathbf{p(\theta_d\ )+(\theta_B)=1}$

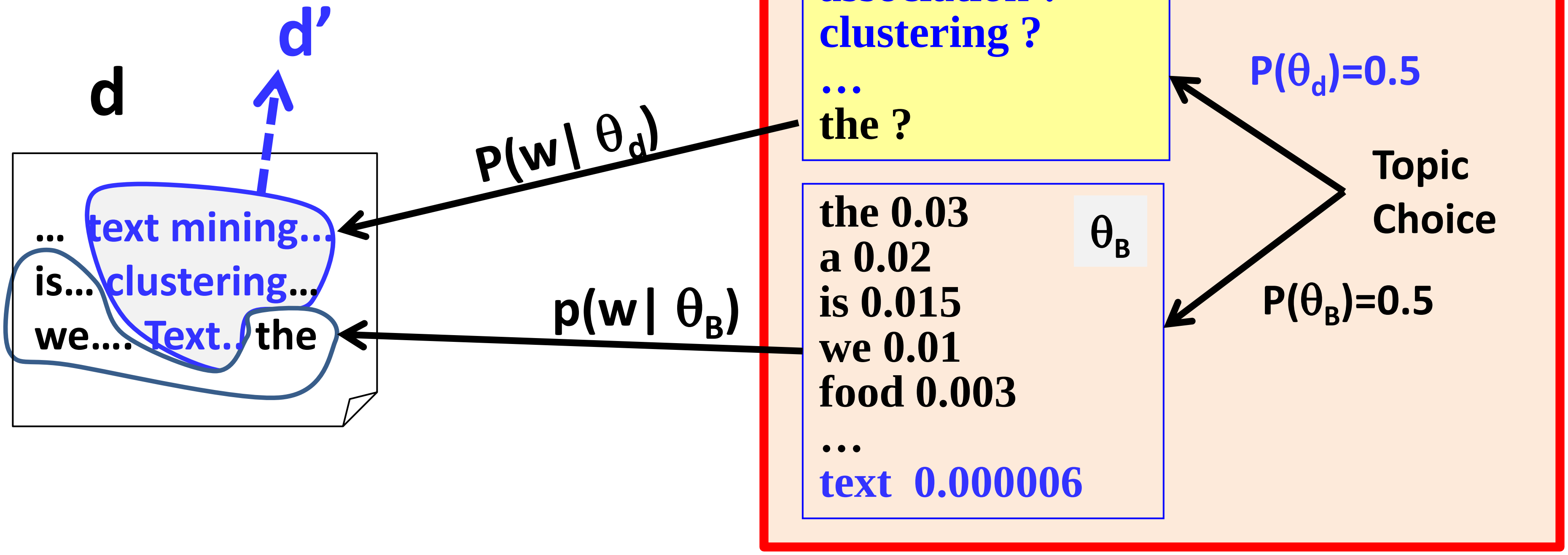
$\mathbf{P(\theta_d)=0.5}$

Topic Choice

$\mathbf{P(\theta_B)=0.5}$

If we know which word is from which distribution…

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



d'

d

… text mining…

is… clustering…

we…. Text.. the

$P(w| \theta_d)$

$p(w| \theta_B)$

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

$P(\theta_B)=0.5$

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

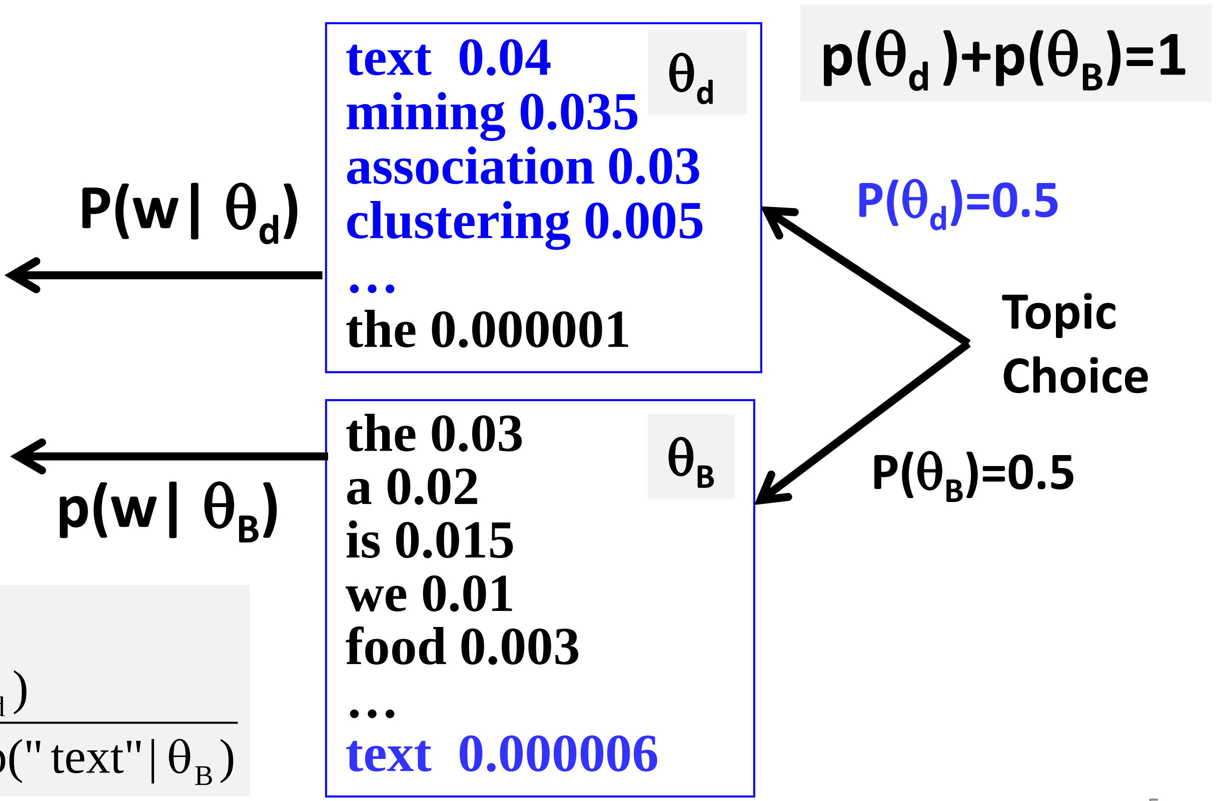
Is "text" more likely from θ_d or θ_B ?

From θ_d (Z=0)?

$p(\theta_d)p(\text{"text"}|\theta_d)$

From θ_B (Z=1)?

$p(\theta_B)p(\text{"text"}|\theta_B)$



$P(w|\theta_d)$

$p(w|\theta_B)$

θ_d

text 0.04
mining 0.035
association 0.03
clustering 0.005
…
the 0.000001

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

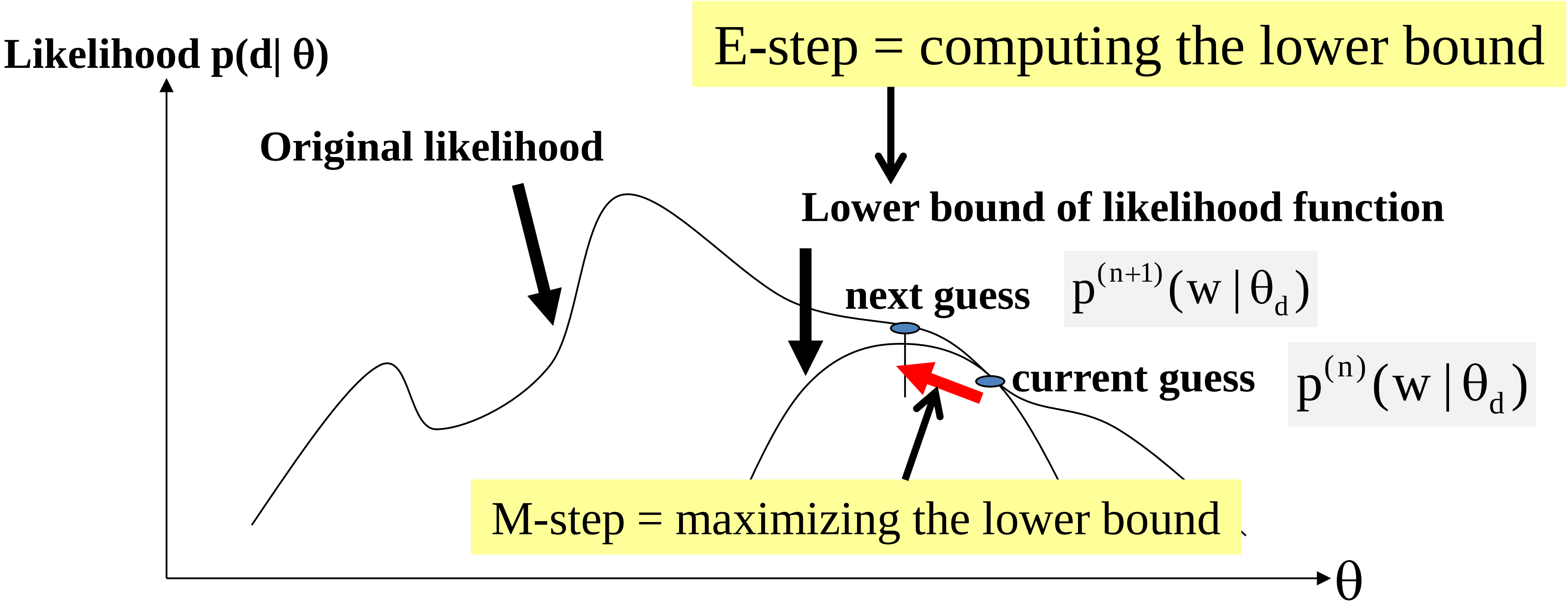
Assume
$p(\theta_d)=p(\theta_B)=0.5$
and $p(w|\theta_B)$ is known

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



Likelihood p(d| θ)

E-step = computing the lower bound

Original likelihood

Lower bound of likelihood function

next guess

$p^{(n+1)}(w \mid \theta_d)$

current guess

$p^{(n)}(w \mid \theta_d)$

M-step = maximizing the lower bound

θ