

# An Introduction to Latent Dirichlet Allocation(LDA)

Tutorial

Rong YE

# CONTENTS

---

- PART I: Prerequisite
- PART II: LDA model
- PART III: LDA parameter inference
- PART IV: Language model evaluation
- PART V: Applications

# Prerequisite

Part I

- Conjugacy distribution
- Dirichlet distribution
- Statistical Language Model
- Topic models

# Bayesian inference and conjugacy distribution

---

- Bayes' theorem

$$p(A | B) = \frac{p(B | A) \cdot p(A)}{p(B)} \propto p(B | A) \cdot p(A)$$

$$\textit{posterior} \propto \textit{prior} \times \textit{likelihood}$$

- Conjugacy distribution:

- Posterior is in the same probability distribution family as the prior.
- Conjugacy prior (given likelihood...)

# Example - Binomial & Beta

---

- Given likelihood of a binomial distribution

$$p(X = k | \theta, n) = \binom{n}{k} \theta^k (1 - \theta)^{n-k} \propto \theta^k (1 - \theta)^{n-k}$$

- Consider a *beta* distribution

- Form:  $p(\theta | \alpha, \beta) = \text{const} \times \theta^{\alpha-1} (1 - \theta)^{\beta-1}$

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

- Update  $\theta$  -- posterior

$$p(\theta | X = k, n) \propto p(X = k | \theta, n) \times p(\theta)$$

$$= \theta^{k+\alpha-1} (1 - \theta)^{n-k+\beta-1}$$

$$= \text{Beta}(k + \alpha, n - k + \beta)$$

# Multinomial case

---

- Multinomial likelihood

$$p(X_1 = x_1, X_2 = x_2, \dots, X_k = x_k \mid \vec{\theta}, n) \propto \prod_{i=1..k} \theta_i^{x_i}$$

- What about prior?

- $p(\vec{\theta})$  should in form  $\prod_{i=1}^k \theta_i^{???}$

- Dirichlet Distribution!

$$p(\vec{\theta} \mid \vec{\alpha}) = \frac{1}{\Delta(\vec{\alpha})} \prod_{i=1}^k \theta_i^{\alpha_i - 1} \quad \Delta(\vec{\alpha}) = \frac{\Gamma(\sum_{k=1}^K \alpha_k)}{\prod_{k=1}^K \Gamma(\alpha_k)} = \text{const.}$$

# Statistical Language Model

---

- Text(sequence of words) representation

$$p(W) = p(w_1, w_2, \dots, w_n) = p(w_1) p(w_2 | w_1) p(w_3 | w_2, w_1) \dots p(w_n | w_{n-1}, w_{n-2} \dots w_1)$$

$$p(\text{I am a student}) = p(\text{I}) p(\text{am} | \text{I}) p(\text{a} | \text{am}, \text{I}) p(\text{student} | \text{a}, \text{am}, \text{I})$$

- Markov property

- Bi-gram:  $p(W) = p(w_1) p(w_2 | w_1) p(w_3 | w_2) \dots p(w_n | w_{n-1})$

$$p(\text{I am a student}) = p(\text{I}) p(\text{am} | \text{I}) p(\text{a} | \text{am}) p(\text{student} | \text{a})$$

- Tri-gram, N-gram

- Independence - **Unigram** model

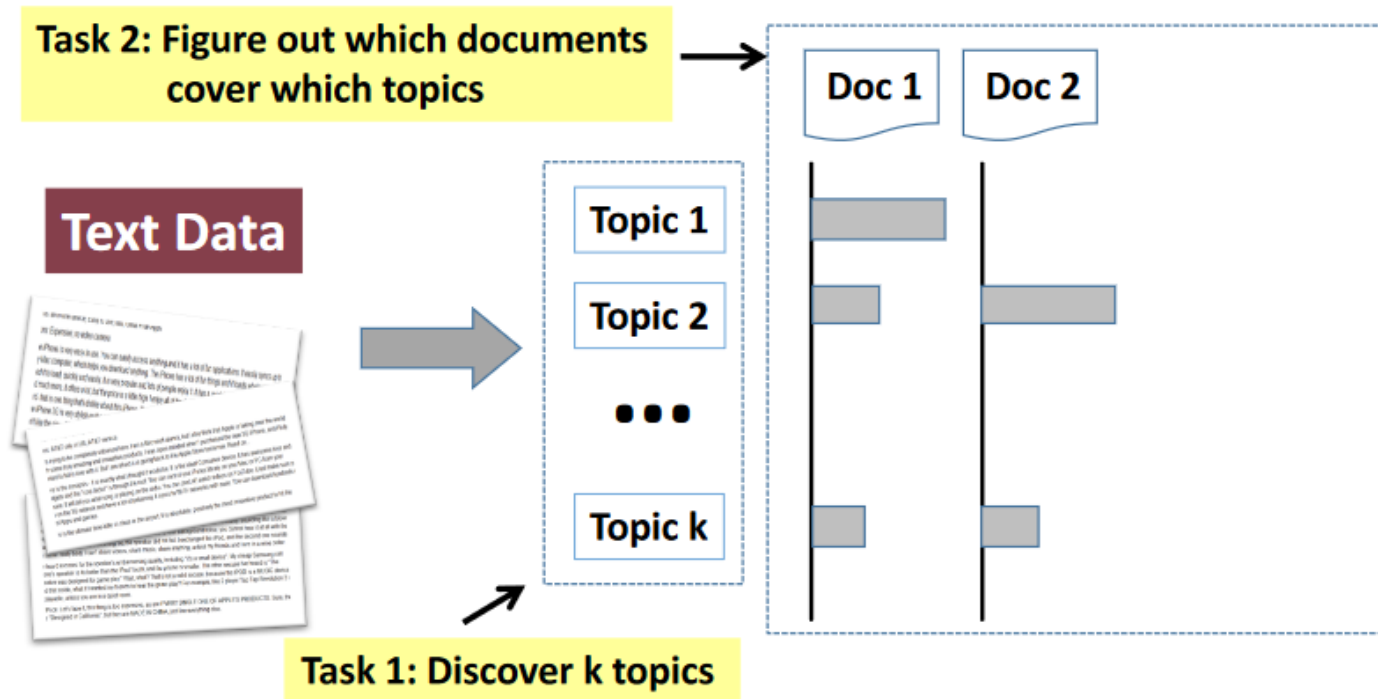
- $p(W) = p(w_1) p(w_2) \dots p(w_N)$

- **Bag-of-words:**  $\vec{n} = (n_1, n_2 \dots n_V)$

$$p(W) = p(\vec{n}) = \text{multi}(\vec{n} | \vec{p}, N)$$

# Topic models - intro

- Care about content: Topic model



- Variable about topic:  $z$

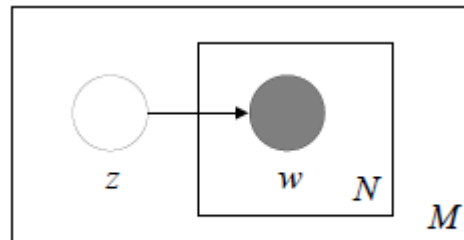


# Topic model1 – Mixture of Unigram

---

- Care about content: Topic model
- Mixture of unigram
  - Each document is generated by first choosing a topic  $z$  and then generating  $N$  words independently.
- Mixture of several topics:

$$p(W) = \sum_z p(z) \cdot p(W | z) = \sum_z p(z) \cdot \prod_i p(w_i | z)$$



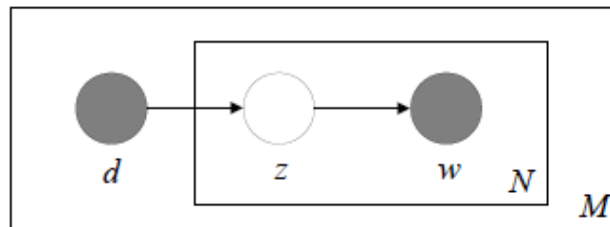
(b) mixture of unigrams

# Topic model2 - pLSA

---

- Probabilistic Latent Semantic Analysis(pLSA)
  - Also, Probabilistic Latent Semantic Indexing(pLSI)
  - K topics, V words, M documents
  - Given document  $d$ ,  $p(w | d) = \prod_i p(w_i | d)$

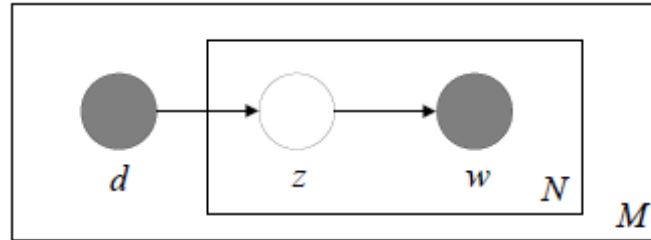
$$= \prod_i \sum_{z=1}^K p(w_i | z) p(z | d)$$



(c) pLSI/aspect model

# Topic model2 - pLSA

---



(c) pLSI/aspect model

- Observable variables:  $d, w$

$$p(w, d) = \sum_{d_j} p(d_j) \prod_i p(w_i | d_j)$$
$$= \sum_{d_j} p(d_j) \prod_i \sum_{z=1}^K p(w_i | z) p(z | d_j)$$

- Drawbacks
  - $d$ : a *r.v.*, probability?
  - $(KV + KM)$  Parameters – overfit

# Latent Dirichlet Allocation (LDA)

Part II

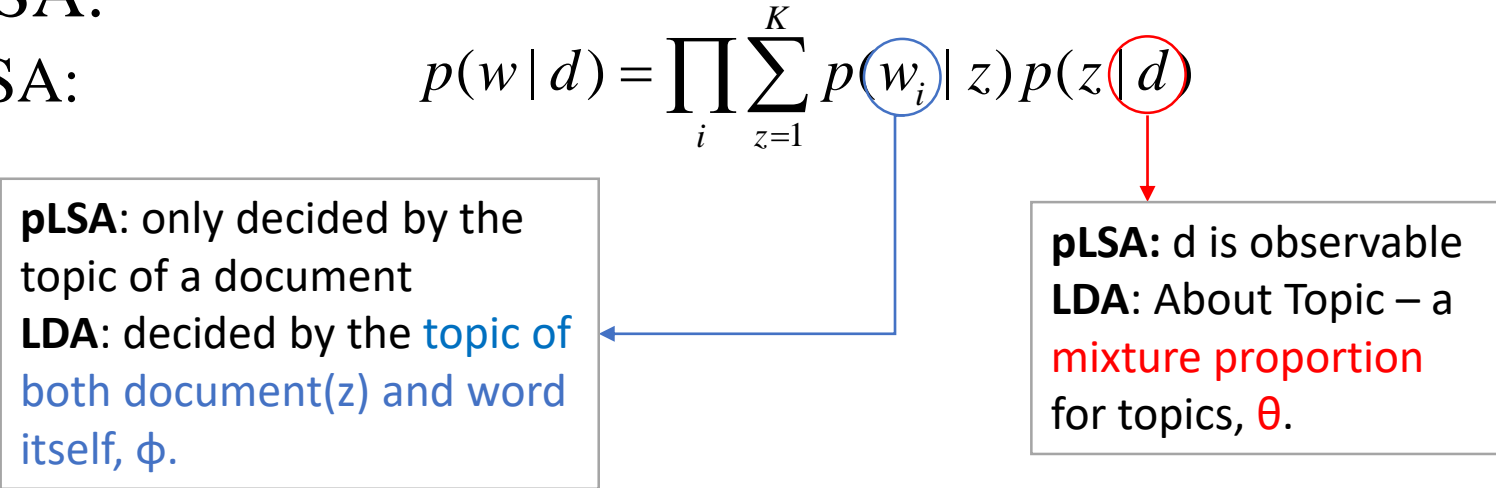
- LDA – intro
- Priors
- Generation procedure
- Bayes' net
- Likelihoods

# LDA - a Bayesian version of PLSA

---

- LDA vs. pLSA:

- Recall pLSA:

$$p(w | d) = \prod_i \sum_{z=1}^K p(w_i | z) p(z | d)$$


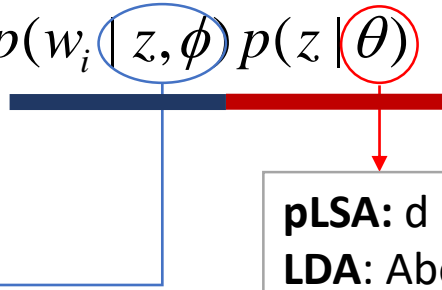
**pLSA:** only decided by the topic of a document  
**LDA:** decided by the **topic of both document(z) and word itself,  $\phi$ .**

**pLSA:** d is observable  
**LDA:** About Topic – a **mixture proportion** for topics,  $\theta$ .

# Assumptions of LDA model

- LDA vs. pLSA:

- LDA :

$$p(w|\phi, \theta) = \prod_i \sum_{z=1}^K p(w_i | z, \phi) p(z | \theta)$$


**pLSA:** only decided by the topic of a document  
**LDA:** decided by the **topic of both document(z) and word itself,  $\phi$ .**

**pLSA:** d is observable  
**LDA:** About Topic – a **mixture proportion** for topics,  $\theta$ .

- $\theta$ :

- 文档A = 0.5 人工智能+ 0.3 名人+ 0.2 娱乐,  $\theta_A = (0.5, 0.3, 0.2)$
    - 文档B = 0.3 人工智能+ 0.1 名人+ 0.5 娱乐,  $\theta_B = (0.3, 0.1, 0.5)$

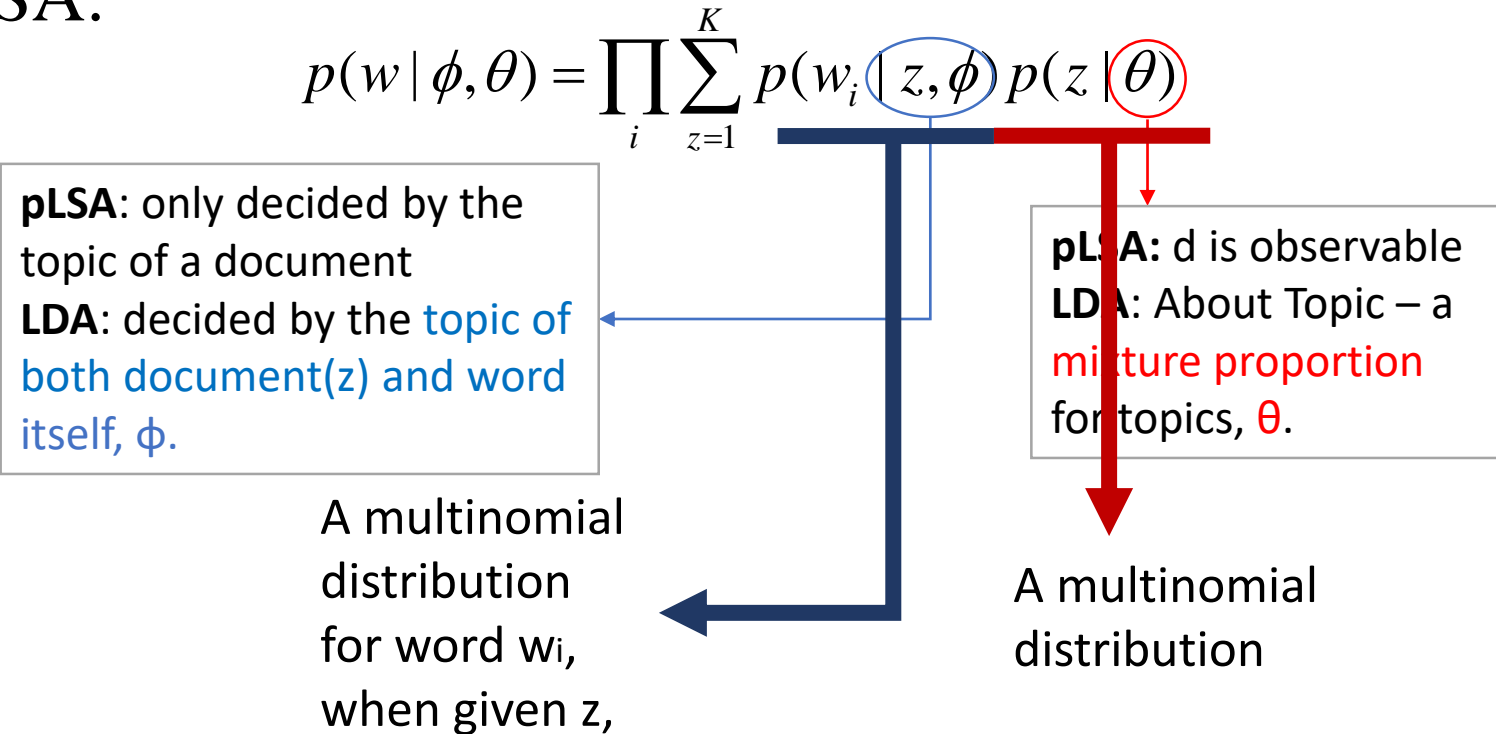
- $\phi$ :

- 人工智能= 0.8 “BAT” + 0.2 “马云” + 0.0 “杨超越”,  $\phi_{\text{人工智能}} = (0.8, 0.2, 0.0)$
    - 名人 = 0.1 “BAT” + 0.6 “马云” + 0.3 “杨超越”,  $\phi_{\text{名人}} = (0.1, 0.6, 0.3)$
    - 娱乐 = 0.1 “BAT” + 0.1 “马云” + 0.8 “杨超越”,  $\phi_{\text{娱乐}} = (0.1, 0.1, 0.8)$

# Assumptions of LDA model

- LDA vs. pLSA:

- LDA :

$$p(w | \phi, \theta) = \prod_i \sum_{z=1}^K p(w_i | z, \phi) p(z | \theta)$$


**pLSA:** only decided by the topic of a document  
**LDA:** decided by the **topic of both document(z) and word itself,  $\phi$ .**

A multinomial distribution for word  $w_i$ , when given  $z$ ,

**pLSA:**  $d$  is observable  
**LDA:** About Topic – a **mixture proportion** for topics,  $\theta$ .

A multinomial distribution

- Use Dirichlet distribution for the priors of  $\theta, \phi$

# Priors of LDA

---

- Use Dirichlet distribution for the priors of  $\theta, \phi$

- $\Theta$  : a topic mixture proportion for document,

$M \times K$  matrix

$$\theta_m = (\theta_{m1}, \theta_{m2} \dots \theta_{mK}) \quad \theta_1, \theta_2 \dots \theta_M \stackrel{iid}{\sim} \text{Dir}(\vec{\alpha}), \alpha \in \mathbb{R}^K$$

- $\Phi$  : a word mixture proportion for topic,

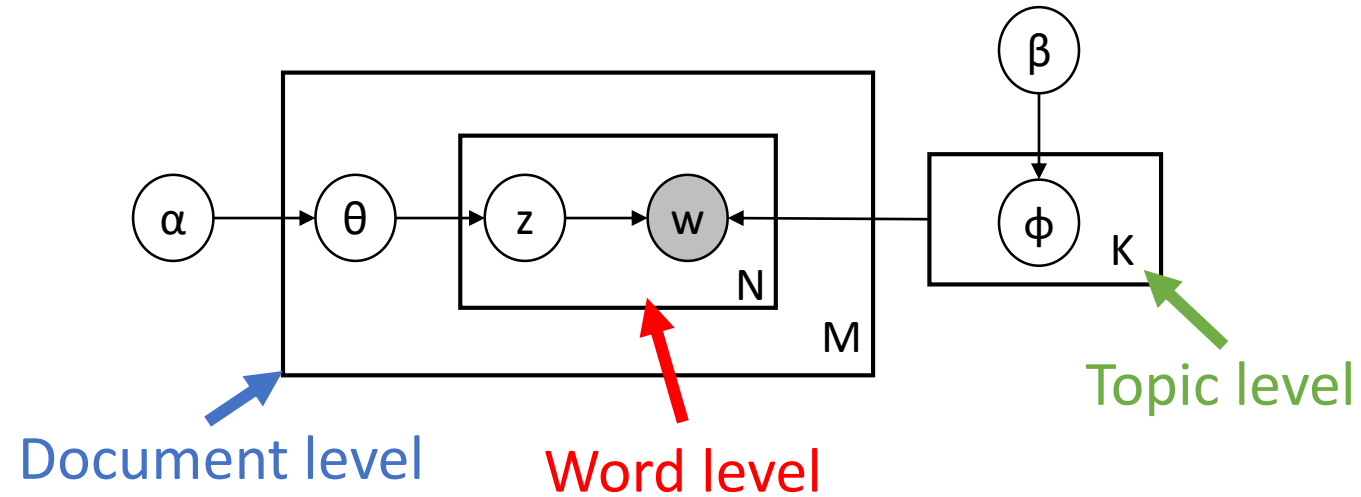
$K \times V$  matrix

$$\phi_k = (\phi_{k1}, \phi_{k2} \dots \phi_{kN}) \quad \phi_1, \phi_2 \dots \phi_K \stackrel{iid}{\sim} \text{Dir}(\vec{\beta}), \beta \in \mathbb{R}^V$$



# LDA – a generative model

- Bayes' net representation



# LDA – a generative model

---

- Generation procedure
  - Topic level  $\rightarrow$  word level  $\leftarrow$  document level

---

LDA generation procedure

---

**Topic level:**

Sample topic  $\phi_k \sim \text{Dir}(\beta), \beta \in \mathbb{R}^V, k = 1, 2 \dots K$

**Document level:**

for each document  $m=1, 2 \dots M$ :

Sample mixture proportion  $\theta_m \sim \text{Dir}(\alpha), \alpha \in \mathbb{R}^K$

Sample document length  $N_m \sim \text{poisson}(\xi)$

**Word level:**

For word  $n = 1 \dots N_m$ :

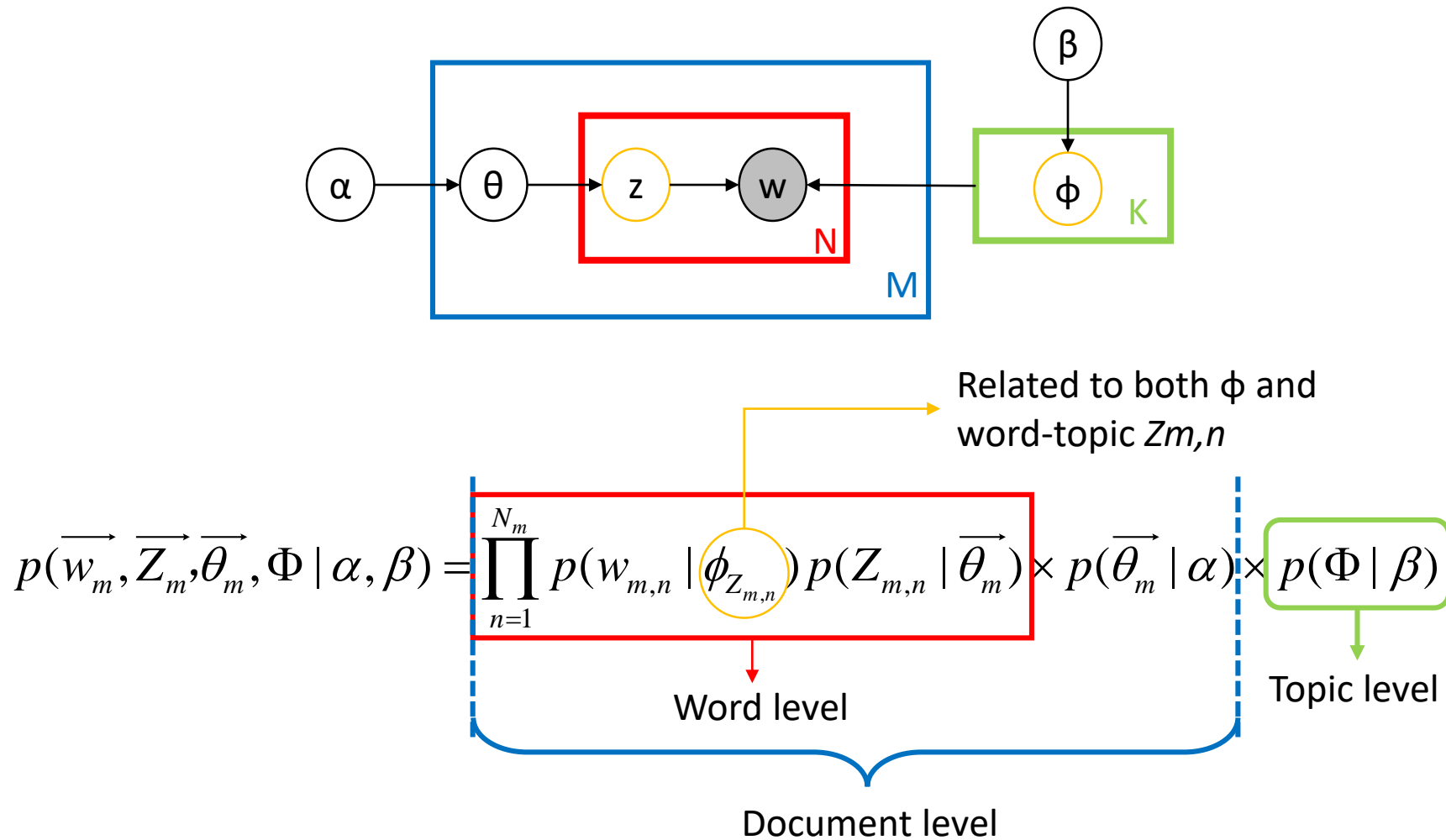
Sample topic index  $Z_{m,n} \sim \text{Multi}(\theta_m)$

Sample term for word  $W_{m,n} \sim \text{Multi}(\phi_{Z_{m,n}})$

Related to both  $\phi$  and  $Z_{m,n}$

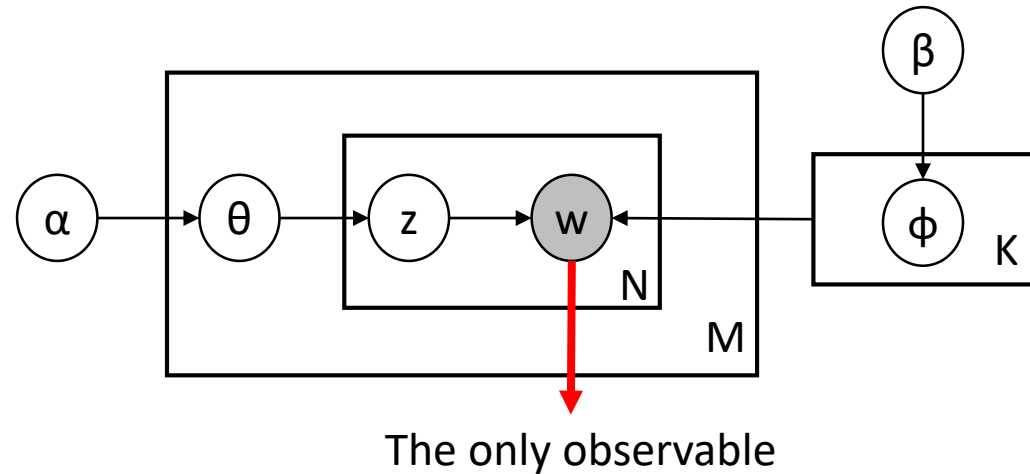
# LDA – Likelihood

- For document  $m$ , joint distribution:



# LDA – Likelihood

- Likelihood for documents



$$p(\overrightarrow{w_m} | \alpha, \beta) = \iint p(\overrightarrow{\theta_m} | \alpha) \times p(\Phi | \beta) \cdot \prod_{n=1}^{N_m} \sum_{z_{m,n}=1}^K p(w_{m,n} | \phi_{z_{m,n}}) p(z_{m,n} | \overrightarrow{\theta_m}) d\overrightarrow{\theta_m} d\Phi$$

$$p(W | \alpha, \beta) = \prod_{m=1}^M p(\overrightarrow{w_m} | \alpha, \beta)$$

# Parameter inference

Part III

- Parameters –intro
- Method 1:  
Variational inference
- Method 2:  
Via Gibbs sampling

# Parameters inference - intro

---

- Parameters we want to know:
  - $\theta$ : a topic mixture proportion for document
  - $\phi$ : a word mixture proportion for topic

- Maximize a posterior (MAP):

$$p(\underbrace{Z, \theta, \Phi}_{\substack{\downarrow \\ \text{All the hiddens}}} \mid \alpha, \beta, W) = \frac{p(W, Z, \theta, \Phi \mid \alpha, \beta)}{p(W \mid \alpha, \beta)}$$

- Intractable ☹

# Method 1: Variational inference

---

- Idea: find parameters  $\Lambda$  of a family of distributions  $q(.|\Lambda)$  on the latent variables (usually exponential family)

- 

$$\Lambda^* = \arg \min_{\Lambda} D_{KL}(q(\theta, z, \phi | \Lambda) || p(\theta, z, \phi | w, \alpha, \beta))$$

- It can be proved that

$$\Lambda^* = \arg \max_{\Lambda} E_q[\log p(\theta, z, \phi, W | \alpha, \beta) - \log q(\theta, z, \phi | \Lambda)]$$

- More detailed: *Blei et.al.(2003)*

# Method 2: Via Gibbs sampling

---

- Idea: Consider  $p(\vec{z} | \vec{w})$ , and adjust word-count/topic counts etc. according to  $\vec{z}$  follows  $p(\vec{z} | \vec{w})$ .
- Steps (general):
  - Initialization
  - Gibbs sampling and adjust word-counts/topic-counts etc.
  - Compute  $\hat{\theta}$  and  $\hat{\phi}$  according to hyperparameters and word-counts/topic-counts



# Gibbs sampling algorithm for LDA

---

- Joint

$$p(\vec{w}, \vec{z} | \vec{\alpha}, \vec{\beta}) = \prod_{k=1}^K \frac{\Delta(\vec{n}_k + \vec{\beta})}{\Delta(\vec{\beta})} \cdot \prod_{m=1}^M \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}$$

- Conditional (eq.79):

$$P(z_i = k | \vec{z}_{-i}, \vec{w}) \propto \frac{n_{k,-i}^{(t)} + \beta_t}{\sum_{t=1}^V n_{k,-i}^{(t)} + \beta_t} \cdot \frac{n_{m,-i}^{(k)} + \alpha_k}{[\sum_{k=1}^K n_{m,-i}^{(k)} + \alpha_k] - 1}$$

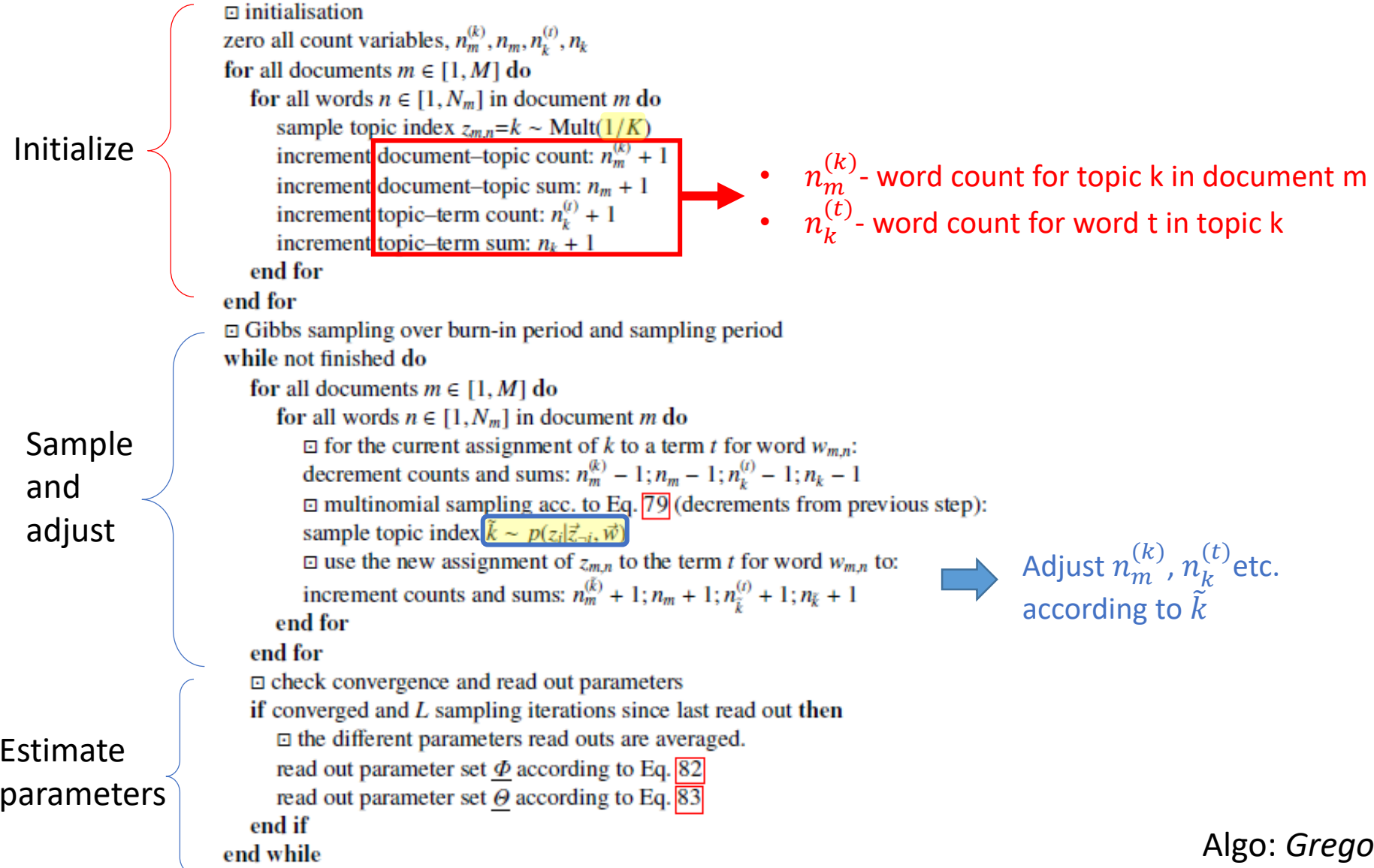
- Estimation (eq.82,83):

$$\varphi_{kt} = \frac{n_{k,-i}^{(t)} + \beta_t}{\sum_{t=1}^V n_{k,-i}^{(t)} + \beta_t}$$

$$\theta_{mk} = \frac{n_{m,-i}^{(k)} + \alpha_k}{\sum_{k=1}^K n_{m,-i}^{(k)} + \alpha_k}$$

- Detailed derivation: *Gregor(2006)*

# Gibbs sampling algorithm for LDA



# Model evaluation

Part IV

- Perplexity
- Example: Perplexity for N-gram model
- Perplexity for LDA

# Perplexity – how good is our model

---

- Definition:

- In NLP, perplexity is a measure of the ability of a model to generalize to **unseen data (test set)**.

- For a sentence:

$$\text{Perplexity}(W) = p(w_1 w_2 \dots w_N)^{-\frac{1}{N}}$$

- For a corpus:

$$\text{Perplexity}(D_{test}) = \exp \left( - \frac{\sum_{d=1}^{|D_{test}|} \log P(\vec{w}_d)}{\sum_{d=1}^{|D_{test}|} N_d} \right)$$

# Example: Perplexity in N-gram model

- Perplexity: The lower the better.

- Training 38 million words, test 1.5 million words, Wall Street Journal (*Stanford cs124*)
$$P(w_i | w_{i-1}) = \frac{\text{count}(w_{i-1}, w_i)}{\text{count}(w_{i-1})}$$

N-gram Order	Unigram	Bigram	Trigram
Perplexity	962	170	109

## Unigram

Months the my and issue of year foreign new exchange's september were recession ex-  
change new endorsed a acquire to six executives

## Bigram

Last December through the way to preserve the Hudson corporation N. B. E. C. Taylor  
would seem to complete the major central planners one point five percent of U. S. E. has  
already old M. X. corporation of living on information such as more frequently fishing to  
keep her

## Trigram

They also point to ninety nine point six billion dollars from two hundred four oh six three  
percent of the rates of interest stores as Mexico and Brazil on market conditions

# Perplexity for LDA

---

- LDA (for given topic number = K)

$$P(\vec{w_d}) = \prod_{n=1}^{N_d} \left( \sum_{k=1}^K P(w_n | z_n = k) \cdot P(z_n = k | d) \right)$$

$$= \prod_{t=1}^V \left( \sum_{k=1}^K \theta_{d,k} \cdot \varphi_{k,t} \right)^{n_d^{(t)}}$$

$$\log P(\vec{w_d}) = \sum_{t=1}^V n_d^{(t)} \log \left( \sum_{k=1}^K \theta_{d,k} \cdot \varphi_{k,t} \right)$$

Via Gibbs

Not related  
to test  
corpus

- Topic number chosen & perplexity
  - Usually choose  $K = \operatorname{argmin} \operatorname{perplexity}(D_{\text{test}})$

# Applications & extensions

Part V

- Document modeling
- Document classification
- Collaborative filtering
- Extensions

# Document modeling

---

- (Blei, 2003) on AP corpus: 16,333 newswire articles

“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI



# Document modeling

- (Blei, 2003) on AP corpus: 16,333 newswire articles

“Arts”	“Budgets”	“Children”	“Education”
NEW FILM SHOW	MILLION TAX PROGRAM	CHILDREN WOMEN PEOPLE	SCHOOL STUDENTS SCHOOLS

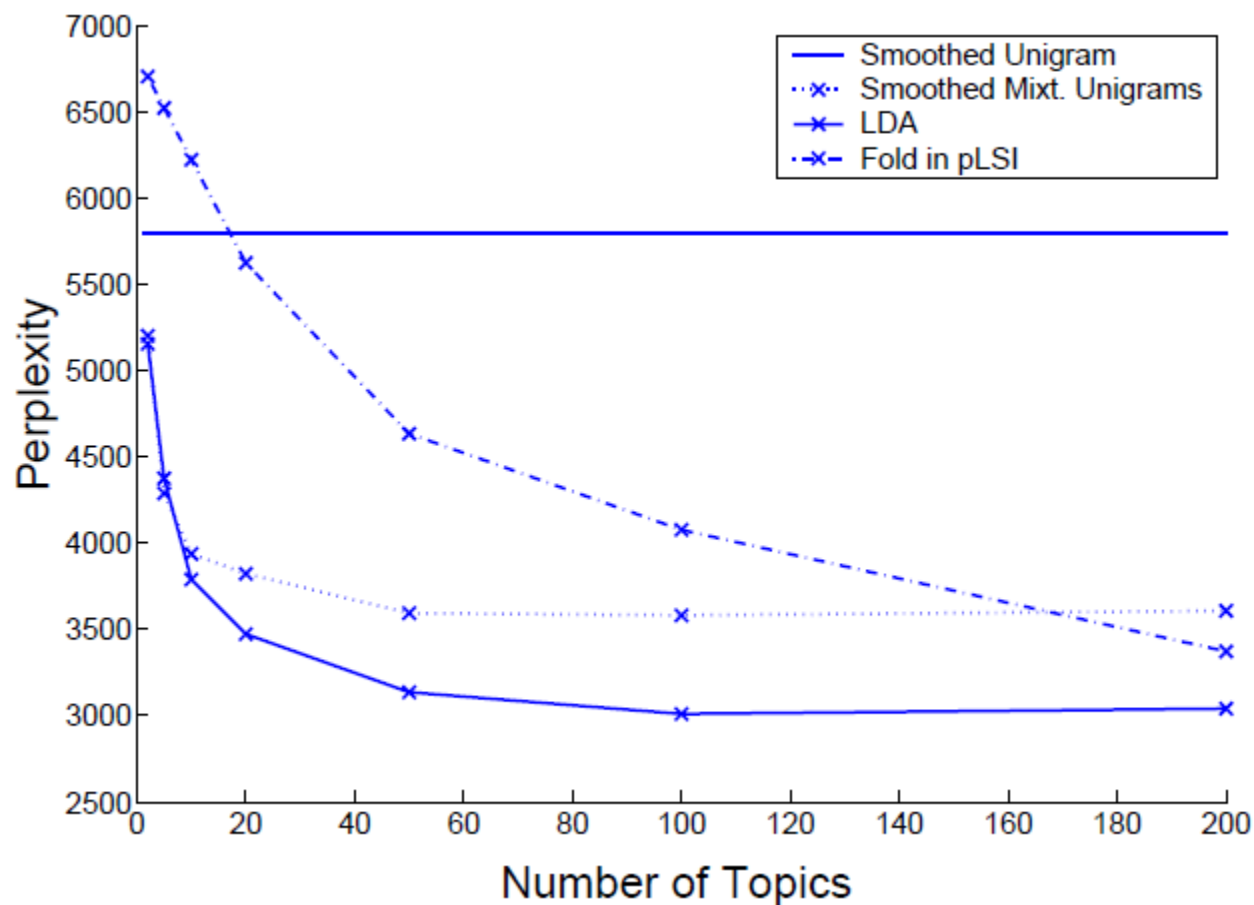
The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

Figure 8: An example article from the AP corpus. Each color codes a different factor from which the word is putatively generated.

# Document modeling

---

- Perplexity vs topic number (Blei, 2003)



# Document classification

- binary classification, Reuters-21578 dataset (Blei, 2003)
  - 8000 documents
  - Low-dimensional representations from LDA(50 topic) for document vs. word features
- When proportion of training data is low...

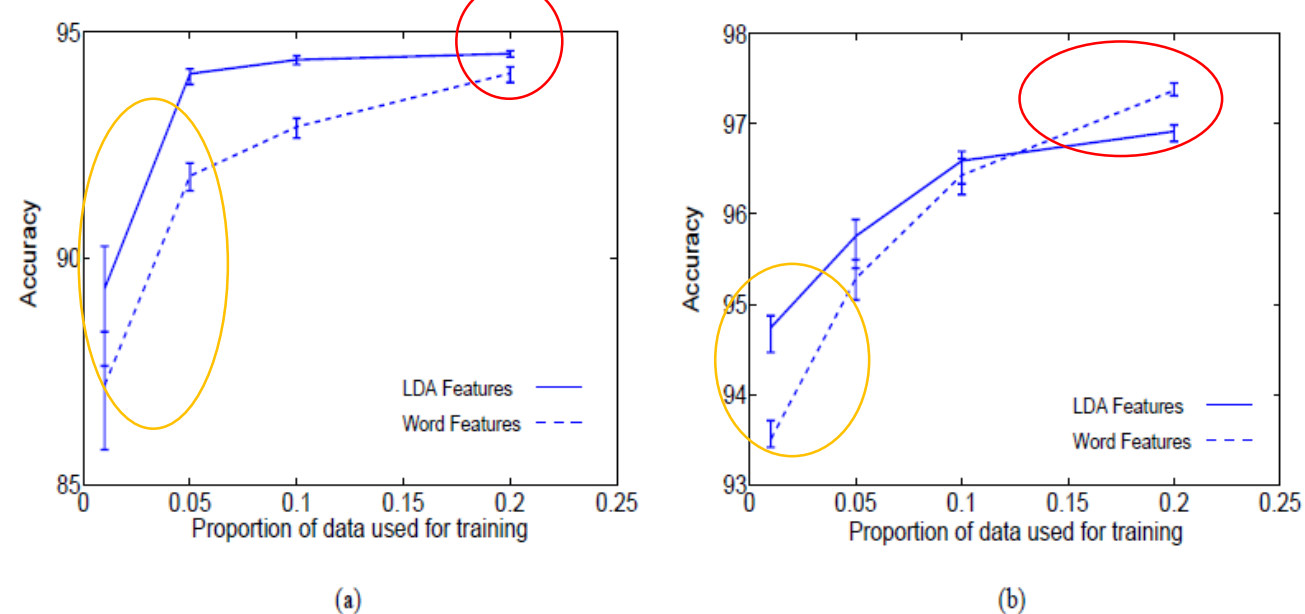
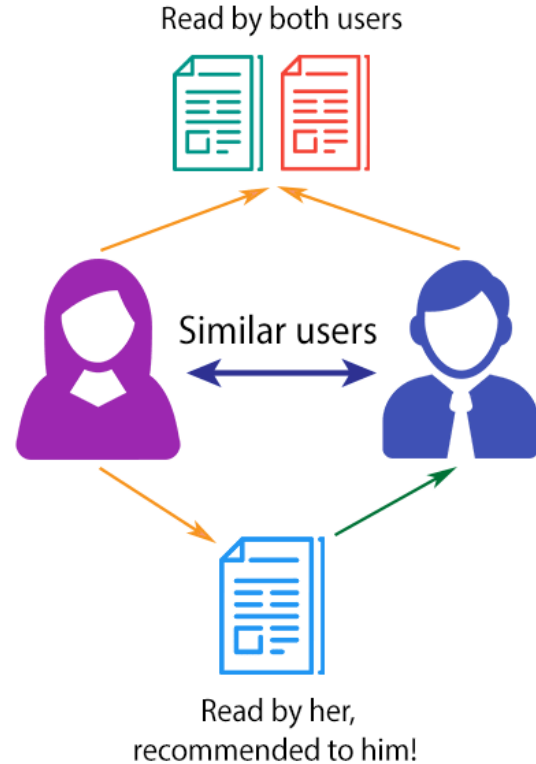


Figure 10: Classification results on two binary classification problems from the Reuters-21578 dataset for different proportions of training data. Graph (a) is EARN vs. NOT EARN. Graph (b) is GRAIN vs. NOT GRAIN.

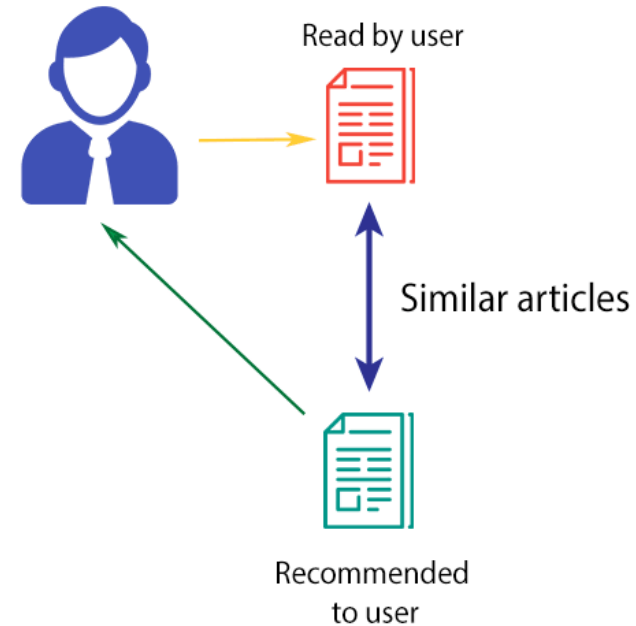
# Collaborative filtering

- Similarity  $\rightarrow$  preference

COLLABORATIVE FILTERING



CONTENT-BASED FILTERING



# Collaborative filtering

---

- User similarity  $\rightarrow$  preference
- Dataset: EachMovie (user: preferred movie choices)
- Analogy:
  - User – document
  - Movie chosen – words in document
  - 1600 movies – vocabulary
  - Movie genre – topic
    - User – genre  $\theta$ :  
Eg. Mary preference = 0.5 Romance + 0.4 comedy + 0.1 sci-fi
    - Movie–genre  $\phi$  :  
Eg. Sci-fi = 0.1 *Ready Player One* + 0.2 *Marvel's The Avengers* + 0.3 流浪地球 + ...

# Collaborative filtering

---

- 3300 training users; 390 testing users
- Predictive-Perplexity:

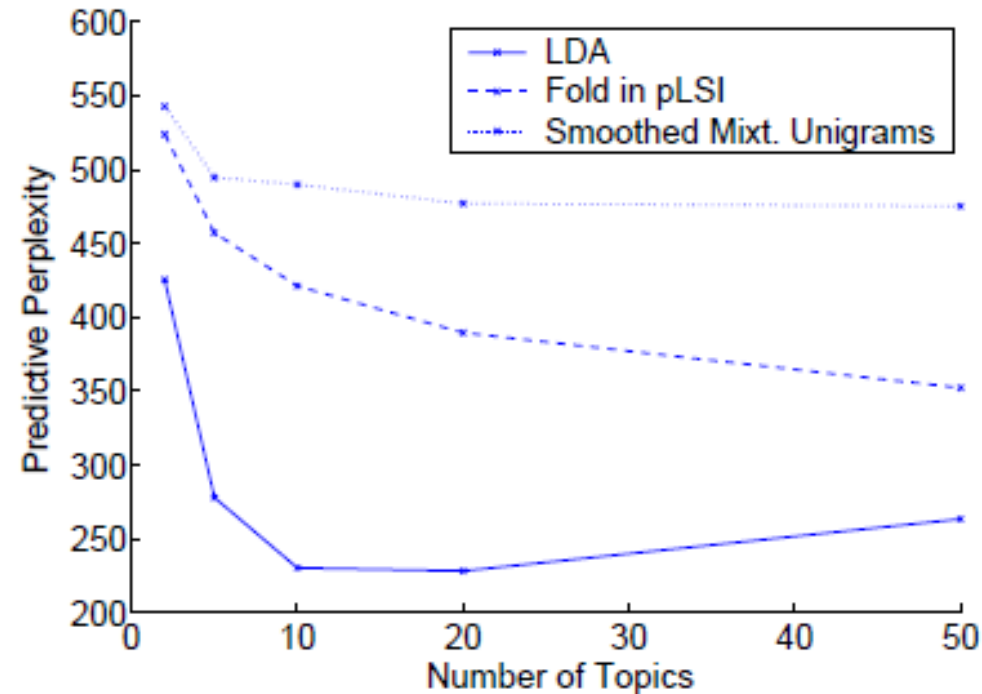


Figure 11: Results for collaborative filtering on the EachMovie data.

# Extensions

---

- Drawbacks of LDA model
  - Short text topics?
  - Noisy data?
  - ...
- Some extension models:
  - TwitterLDA (Zhao et.al, 2011):
    - Assumption: one topic for one tweets.
  - Labeled-LDA(Ramage et.al 2009, ACL):
    - Used in document classification
  - Hierarchically LDA(Teh et.al.2005):
    - Find K automatically
  - ...

# Reference

---

- Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.
- Heinrich G. Parameter Estimation for Text Analysis[J]. Technical Report, 2008.
- Ramage D, Hall D, Nallapati R, et al. Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora[C]//Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1. Association for Computational Linguistics, 2009: 248-256.
- Teh Y W, Jordan M I, Beal M J, et al. Sharing clusters among related groups: Hierarchical Dirichlet processes[C]//Advances in neural information processing systems. 2005: 1385-1392.
- Zhao W X, Jiang J, Weng J, et al. Comparing twitter and traditional media using topic models[C]//European Conference on Information Retrieval. Springer, Berlin, Heidelberg, 2011: 338-349.



# LAB on OJ

– Exact Inference

- <http://10.88.3.61/>
- 1003    Exact Inference

# Review on exact inference algorithm

---

- Example: MaryCall, JohnCall in Ch14
- Enumeration algorithm:
  - Step 1: Select the entries consistent with the evidence
  - Step 2: Sum out *hidden vars* to get joint of Query and evidence
  - Step 3: Normalize
- Elimination algorithm:
  - Make factors
  - Join all *factors* and eliminate all *hidden vars*.

# Review on exact inference algorithm

---

- Example: MaryCall, JohnCall in Ch14
- Enumeration algorithm:
  - Def enumeration\_ask(X, e, bn):
  - Def enumeration\_all(X, e, bn):
- Elimination algorithm:
  - Def elimination\_ask(X, e, bn):

# Problem description – sample in

---

$P(\text{Earthquake} = -)$

\*\*\*

$P(\text{Burglary} = + \mid \text{John} = +, \text{Mary} = +)$

John | Alarm

\*\*\*\*\*

0.9 +

Burglary

0.05 -

0.001

\*\*\*

\*\*\*

Mary | Alarm

Earthquake

0.7 +

0.002

0.01 -

\*\*\*

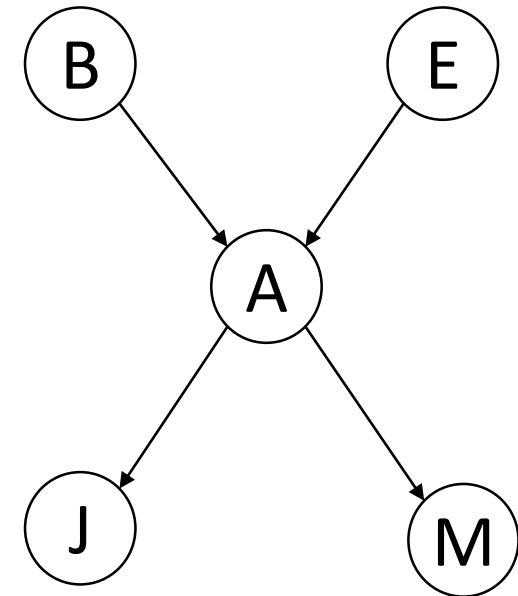
Alarm | Burglary Earthquake

0.95 + +

0.94 + -

0.29 - +

0.001 - -



# Problem description – sample out

---

probability by enumeration: 1.0

probability by elimination: 1.0

\*\*\*\*\*

probability by enumeration: 0.28

probability by elimination: 0.28

\*\*\*\*\*

# Classes

---

class BayesNet:

*used in building the BayesNet*

def \_\_init\_\_(self, node\_specs=[]):

def add(self, node\_spec):

def variable\_node(self, var):

def variable\_values(self, vars):

class BayesNode:

*used in building the BayesNode*

"""A conditional probability distribution for a boolean variable,  $P(X \mid \text{parents})$ .  
Part of a BayesNet."""

def \_\_init\_\_(self, x, parents, cpt):

def p(self, value, event):

# Classes

---

class ProbDist: *used in the computation for probability distribution*

def \_\_init\_\_(self, varname='?', freqs=None):

def **normalize**(self):

class Factor: *used in elimination algorithm*

def \_\_init\_\_(self, variables, cpt):

def **pointwise\_product**(self, other):

def **sum\_out**(self, var):

def **p**(self, e):

def **normalize**(self):

# Procedure

---

- First read the input and *build*  $bn = \text{BayesNet}()$   
by  $bn.add((node, parents, cpt))$
- Discuss (1) joint/ marginal distribution  
(2) conditional distribution
- (1): def **joint\_probability()**:  
Call *enumerate\_all()* and *elimination\_ask()* respectively
- (2): def **conditional\_probability()**:  
Call *enumerate\_ask()* and *elimination\_ask()* respectively



# pseudocode

- Enumeration algorithm

```
function ENUMERATION-ASK( $X, \mathbf{e}, bn$ ) returns a distribution over  $X$   
  inputs:  $X$ , the query variable  
            $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
            $bn$ , a Bayes net with variables  $\{X\} \cup \mathbf{E} \cup \mathbf{Y}$  /*  $\mathbf{Y} = \text{hidden variables}$  */  
  
   $Q(X) \leftarrow$  a distribution over  $X$ , initially empty  
  for each value  $x_i$  of  $X$  do  
     $Q(x_i) \leftarrow$  ENUMERATE-ALL( $bn.VARS, \mathbf{e}_{x_i}$ )  
    where  $\mathbf{e}_{x_i}$  is  $\mathbf{e}$  extended with  $X = x_i$   
  return NORMALIZE( $Q(X)$ )
```

```
function ENUMERATE-ALL( $vars, \mathbf{e}$ ) returns a real number  
  if EMPTY?( $vars$ ) then return 1.0  
   $Y \leftarrow$  FIRST( $vars$ )  
  if  $Y$  has value  $y$  in  $\mathbf{e}$   
    then return  $P(y \mid \text{parents}(Y)) \times$  ENUMERATE-ALL(REST( $vars$ ),  $\mathbf{e}$ )  
    else return  $\sum_y P(y \mid \text{parents}(Y)) \times$  ENUMERATE-ALL(REST( $vars$ ),  $\mathbf{e}_y$ )  
    where  $\mathbf{e}_y$  is  $\mathbf{e}$  extended with  $Y = y$ 
```

You can use Ynode.p()

**Figure 14.9** The enumeration algorithm for answering queries on Bayesian networks.

# pseudocode

- Elimination algorithm

```
function ELIMINATION-ASK( $X, \mathbf{e}, bn$ ) returns a distribution over  $X$   
  inputs:  $X$ , the query variable  
            $\mathbf{e}$ , observed values for variables  $\mathbf{E}$   
            $bn$ , a Bayesian network specifying joint distribution  $\mathbf{P}(X_1, \dots, X_n)$   
  
   $factors \leftarrow []$   
  for each  $var$  in ORDER( $bn.VARS$ ) do  
     $factors \leftarrow [MAKE-FACTOR(var, \mathbf{e}) | factors]$  → Actually, this is an “append” operation  
    if  $var$  is a hidden variable then  $factors \leftarrow SUM-OUT(var, factors)$   
  return NORMALIZE(POINTWISE-PRODUCT( $factors$ ))
```

**Figure 14.10** The variable elimination algorithm for inference in Bayesian networks.

How to express this statement?