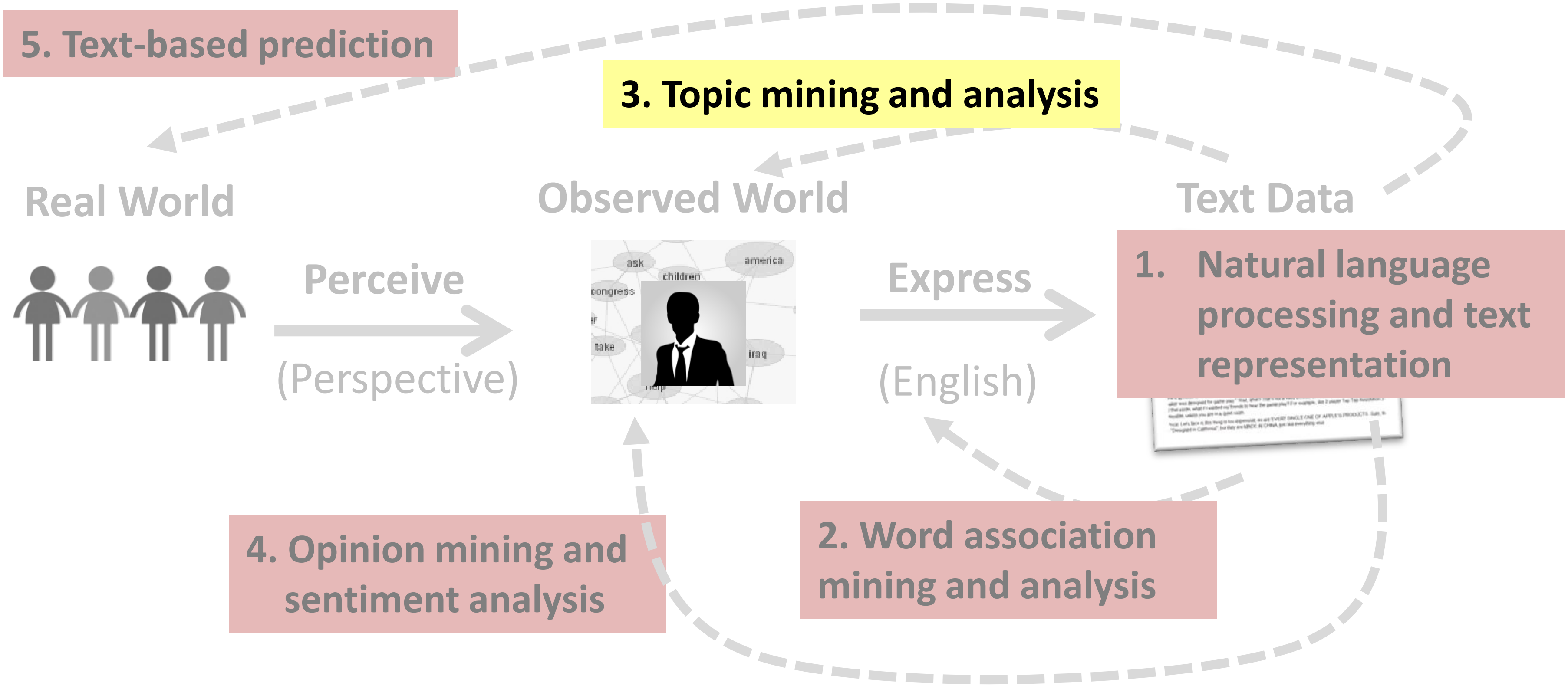
Latent Dirichlet Allocation (LDA)

ChengXiang "Cheng" Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign

Latent Dirichlet Allocation (LDA)



Extensions of PLSA

- PLSA with prior knowledge ➔ User-controlled PLSA
- PLSA as a generative model ➔ Latent Dirichlet Allocation

隐式狄利克雷分布

PLSA with Prior Knowledge

PLSA是完全基于data的,做最大似然估计

- 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 bad tags.
- We can incorporate such knowledge as priors of PLSA model

Maximum a Posteriori (MAP) Estimate

最大后验
估计

$$\Lambda^* = \arg\max_{\Lambda} \underbrace{p(\Lambda)}_{\text{先验}} \underbrace{p(Data \mid \Lambda)}_{\text{基于data}}$$

- We may use $p(\Lambda)$ to encode all kinds of preferences and constraints, e.g.,
  - $p(\Lambda)>0$ if and only if one topic is precisely "background": $p(w|\theta_B)$
  - $p(\Lambda)>0$ if and only if for a particular doc d, $\pi_{d,3}=0$ and $\pi_{d,1}=1/2$
  - $p(\Lambda)$ 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 $p(w|\theta_i)$

共轭先验

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

battery	0.5
life	0.5

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

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

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

相较于之前修改的部分

Pseudo counts of w from prior θ'

← 反映先验知识

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

Sum of all pseudo counts

What if μ=0? What if μ=+∞?

用来控制两者强弱 $\mu\in[0,+\infty)$

去除先验

完全为先验控制.

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

Deficiency of PLSA

- Not a generative model
  - Can't compute probability of a new document
  - Heuristic workaround is possible, though
- Many parameters ➔ high complexity of models
  - Many local maxima
  - Prone to overfitting

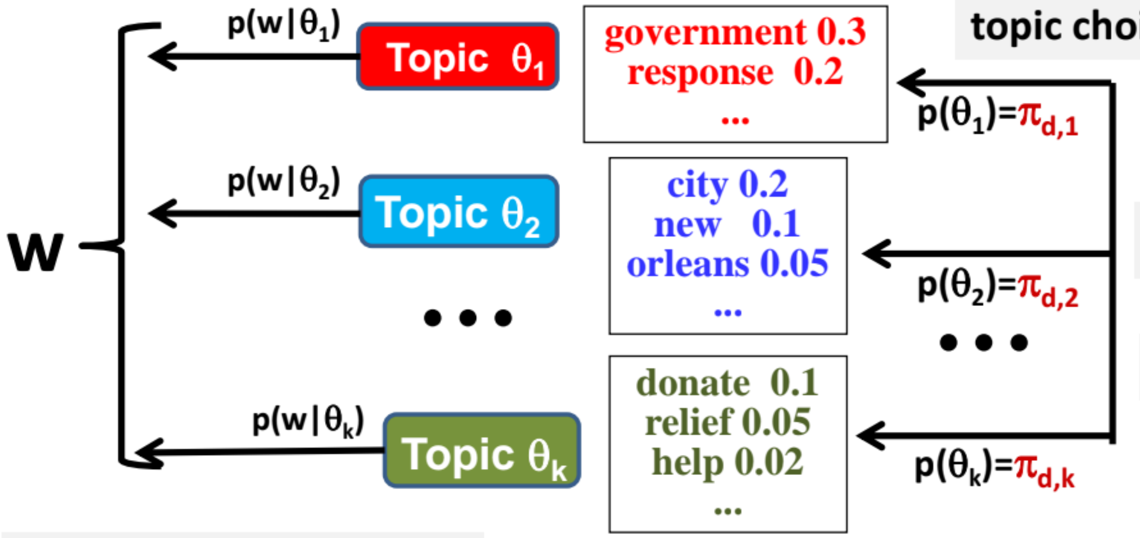
缺陷

- Not necessarily a problem for text mining (only interested in fitting the "training" documents)

Latent Dirichlet Allocation (LDA)

- Make PLSA a generative model by imposing a Dirichlet prior on the model parameters ➔
  - LDA = Bayesian version of PLSA
  - Parameters are regularized
- Can achieve the same goal as PLSA for text mining purposes
  - Topic coverage and topic word distributions can be inferred using Bayesian inference

PLSA → LDA 使用2个Dirichlet分布



Both word distributions and topic choices are free in PLSA

$\vec{\pi}_d = (\pi_{d,1}, \ldots, \pi_{d,k})$

$p(\vec{\pi}_d) = \text{Dirichlet}(\vec{\alpha})$

$\vec{\alpha} = (\alpha_1, \ldots, \alpha_k), \alpha_i > 0$

$\vec{\theta}_i = (p(w_1 | \theta_i), \ldots, p(w_M | \theta_i))$

$p(\vec{\theta}_i) = \text{Dirichlet}(\vec{\beta})$

$\vec{\beta} = (\beta_1, \ldots, \beta_M), \beta_i > 0$

LDA imposes a prior on both

Likelihood Functions for PLSA vs. LDA

PLSA

$$p_d(w \mid \{\theta_j\},\{\pi_{d,j}\}) = \underline{\sum_{j=1}^{k} \pi_{d,j}\, p(w \mid \theta_j)}$$

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

$$\log p(C \mid \{\theta_j\},\{\pi_{d,j}\}) = \sum_{d \in C} \log p(d \mid \{\theta_j\},\{\pi_{d,j}\})$$

LDA

$$p_d(w \mid \{\theta_j\},\{\pi_{d,j}\}) = \underline{\sum_{j=1}^{k} \pi_{d,j}\, p(w \mid \theta_j)}$$

作了积分

$$\log p(d \mid \vec{\alpha},\{\theta_j\}) = \int \boxed{\sum_{w \in V} c(w,d) \log[\sum_{j=1}^{k} \pi_{d,j}\, p(w \mid \theta_j)]}\, \underline{p(\vec{\pi}_d \mid \vec{\alpha})\, d\vec{\pi}_d}$$

$$\log p(C \mid \vec{\alpha},\vec{\beta}) = \int \sum_{d \in C} \log p(d \mid \vec{\alpha},\{\theta_j\}) \underline{\prod_{j=1}^{k} p(\theta_j \mid \vec{\beta})\, d\theta_1 \ldots d\theta_k}$$

Core assumption in all topic models

从多个单词分布中生成单词的概率

PLSA component

Added by LDA

Parameter Estimation and Inferences in LDA

- Parameters can be estimated using ML estimator

$$(\hat{\vec{\alpha}}, \hat{\vec{\beta}}) = \arg\max_{\vec{\alpha}, \vec{\beta}} \log p(C \mid \vec{\alpha}, \vec{\beta})$$

How many parameters in LDA vs. PLSA?

- However, $\{\theta_j\}$ **and** $\{\pi_{d,j}\}$ must now be computed using posterior inference
  - Computationally intractable
  - Must resort to approximate inference
  - Many different inference methods are available

Summary of Probabilistic Topic Models

- Probabilistic topic models provide a general principled way of mining and analyzing topics in text with many applications
- Basic task setup:
  - Input: Text data
  - Output: k topics + proportions of these topics covered in each document
- PLSA is the basic topic model, often adequate for most applications
- LDA improves over PLSA by imposing priors
  - Theoretically more appealing
  - Practically, LDA and PLSA perform similarly for many tasks

Suggested Readings

- Blei, D. 2012. “Probabilistic Topic Models.” *Communications of the ACM* 55 (4): 77–84. doi: 10.1145/2133806.2133826.
- Qiaozhu Mei, Xuehua Shen, and ChengXiang Zhai. “Automatic Labeling of Multinomial Topic Models.” *Proceedings of ACM KDD 2007*, pp. 490-499, DOI=10.1145/1281192.1281246.
- Yue Lu, Qiaozhu Mei, and Chengxiang Zhai. 2011. Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA. *Information Retrieval*, 14, 2 (April 2011), 178-203. DOI=10.1007/s10791-010-9141-9.