课程报告安排

- 介绍一篇论文，可以是
  - 发表于知名会议/期刊、关于生成模型的论文（推荐会议/期刊列表见第 0 讲 PPT）
  - 自己的工作但与生成模型有相关性
- 占 40% 总分
- 共78位同学选课，随机挑选分为四批作报告
  - 于第 14、15、16 周课上报告
  - 未选中课上报告的，提交录制视频（优先适用于非全日制同学）
- 扫码，登记姓名、报告论文信息
  -
- 每人报告 10 分钟，讨论 2 分钟
  - 建议参阅顶会论文报告视频，了解如何在较短时间内介绍工作

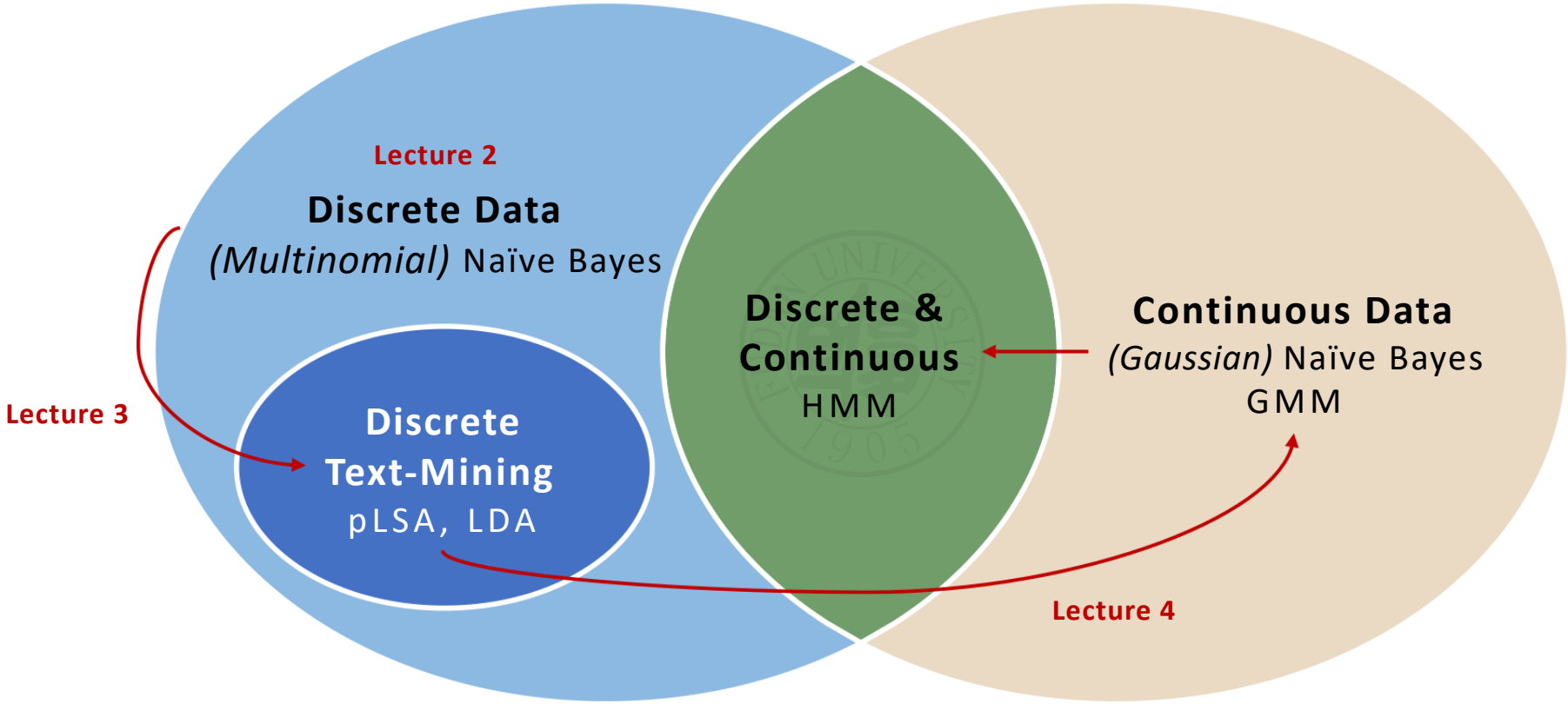
Generative Models: Fundamentals and Applications

Lecture 3: Topic Modeling

Shuigeng Zhou, Wenhui Liu
College of CSAI

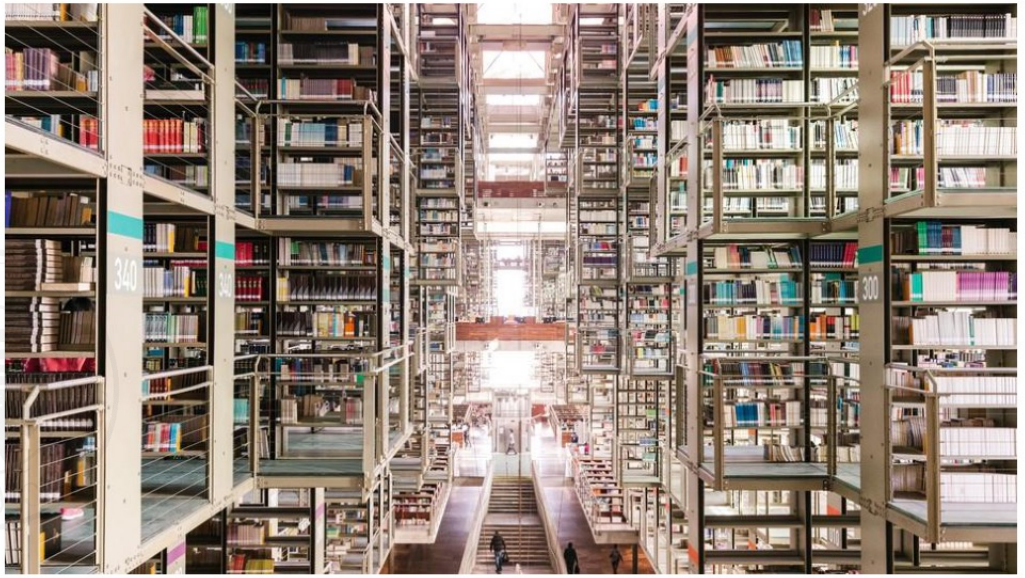
September 29, 2026

A Roadmap


Lecture 2
Discrete Data
(Multinomial) Naïve Bayes
Lecture 3
Discrete
Text-Mining
pLSA, LDA
Discrete &
Continuous
HMM
Continuous Data
(Gaussian) Naïve Bayes
GMM
Lecture 4

主题模型

- 如何从海量文档中：
  - 将主题分门别类？
  - 描述主题随时间如何演变？
  - 快速了解主题的热点和趋势？
  - 发现主题之间的关系？
  - 理解相关信息如何影响内容？



主题模型

- 主题模型
  - 以无监督方式，从大规模语料中**自动发现潜在结构和规律**，使语料可组织、可解释、可分析的方法



主题模型

- 应用场景
  - 新闻分类
  - 趋势预测
  - 科研文献挖掘
  - 社交媒体分析



Credit: Dall-E 3

主题模型

- 语言模型 (Language Model)
  - 实现自然语言自动处理的模型
  - 目标
    - 给一篇文档分配一个概率

$$P(D) = P(w_1, w_2, w_3, \ldots, w_m)$$

    - 假设“文档是**随机生成**的”，再用概率模型去描述它
  - 两种描述方式：
    - 下一词预测
    - 主题模型

下一词预测

- 语言可视作一个局部依赖序列
  - 每个词的概率取决于之前的上下文

$$P(w_1, w_2, \ldots, w_m) = \prod_{t=1}^{m} P(w_t \mid w_1, w_2, \ldots, w_{t-1})$$



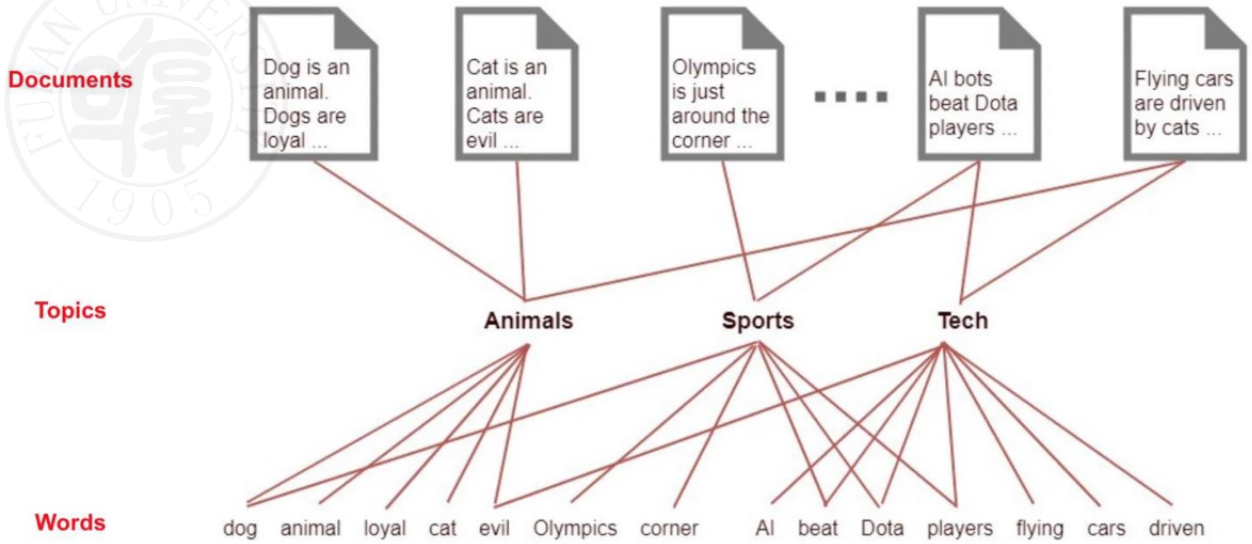
  - 例如
    - N-gram、RNN语言模型、Transformer (GPT)

主题模型

- 语言可视作主题的混合
  - 文档对应一批主题，单词采集自主题的词分布
  - 词的生成依赖主题

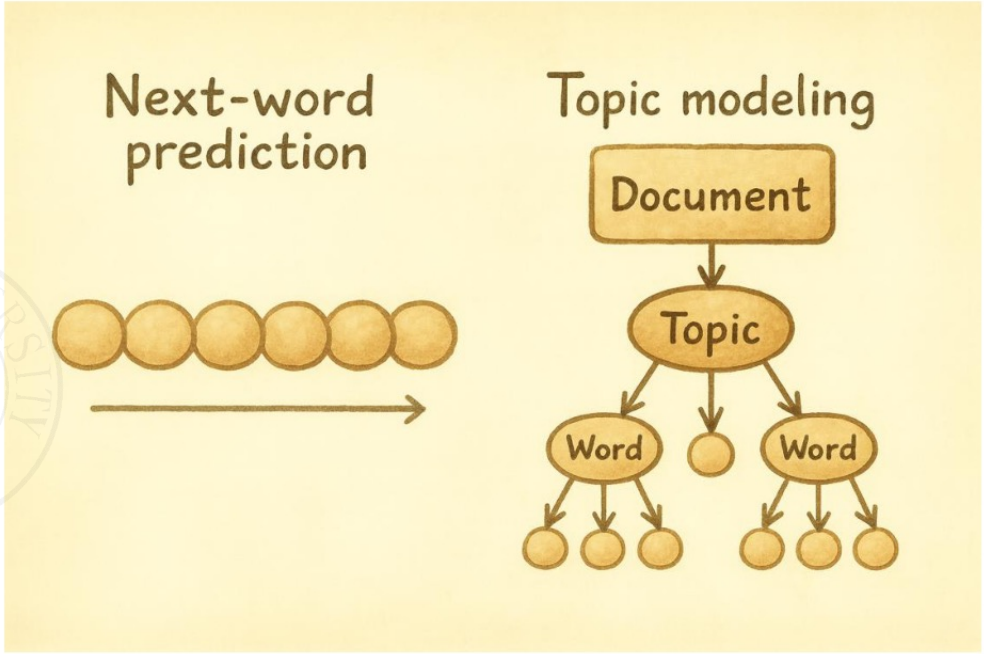
$$P(D) = \prod_{t=1}^{m} \sum_{z} P(z \mid d)\, P(w_t \mid z)$$

  - 例如
    - pLSA、LDA



下一词预测 vs. 主题模型

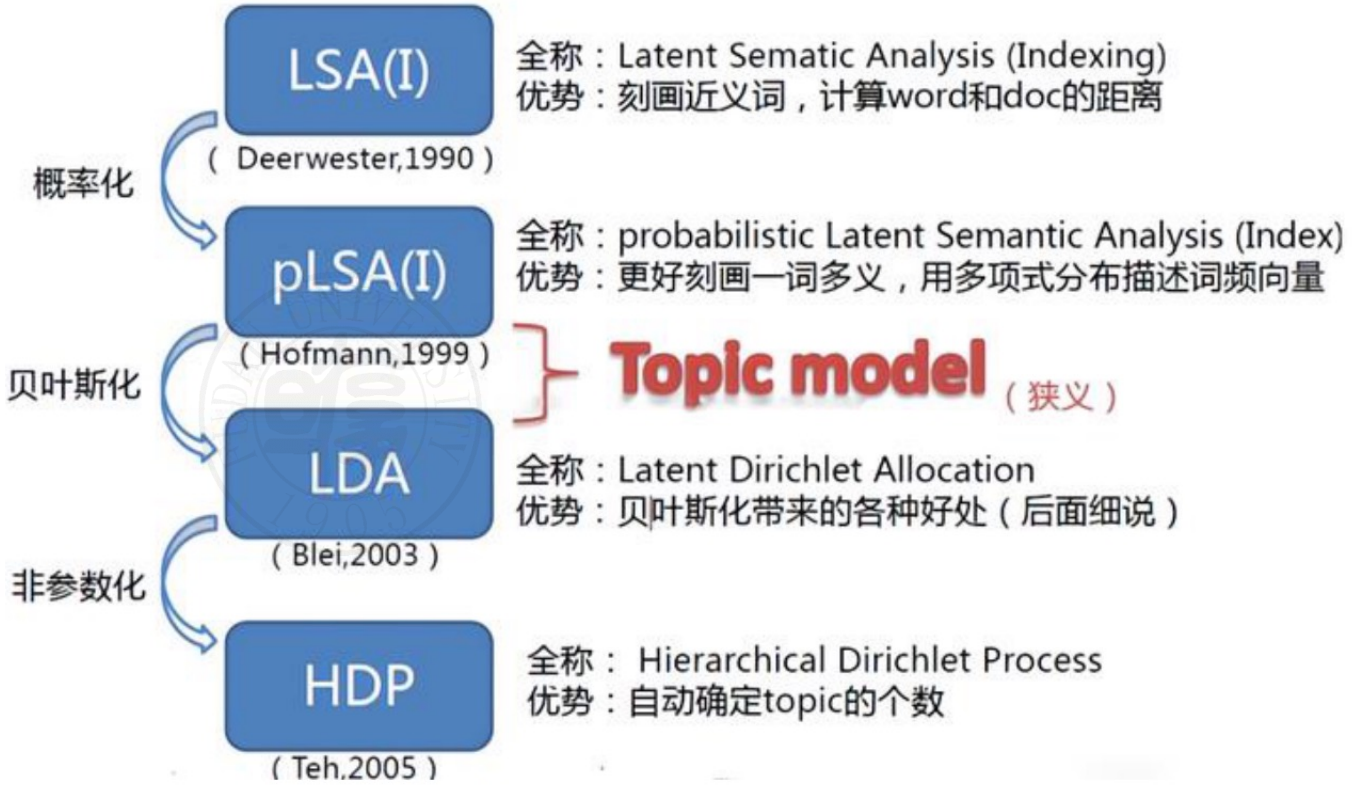
- 下一词预测
  - 强调局部上下文
  - 逐词生成——“写句子”
  - 句子生成、机器翻译、聊天机器人
- 主题模型
  - 强调**全局语义结构**
  - **从主题分布生成——“列提纲”**
  - 文档聚类、主题发现、趋势分析



Credit: Dall-E 3

主题模型

- 演化
  - 代数分解
  - -> 概率化
  - -> 贝叶斯化
  - -> 非参数化

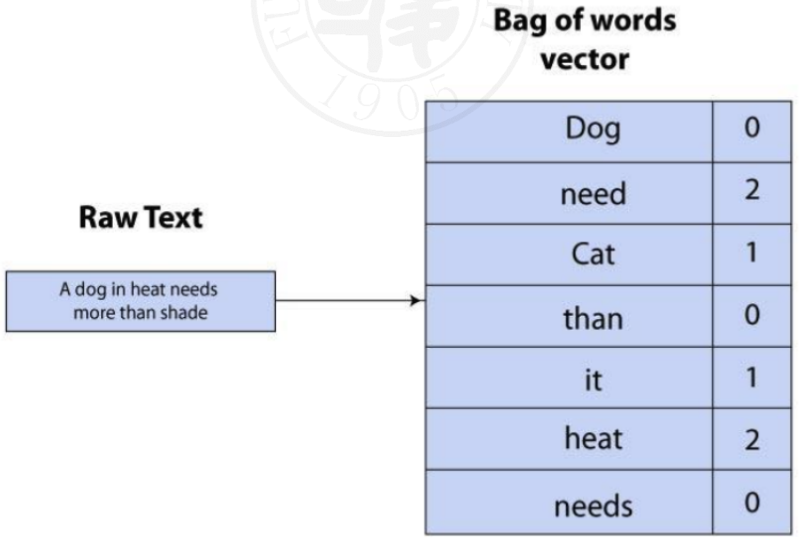


Outline

- 单词向量空间和话题向量空间
- 期望最大化算法 (EM)
- 概率潜在语义分析(pLSA)
- 潜在狄利克雷分布(LDA)

单词向量空间(word vector space)

- 基本假设
  - 将单词的出现情况表示为一个向量，用向量来表示文档的语义内容
    - 向量的每一个维度代表一个单词，向量的维度是语料库中所有可能的单词数量
    - 维度的数值代表对应单词出现在该文档中的频数或权值



单词向量空间

- 单词向量空间
  - 文档集合中的每一篇文档都可以用一个向量来表示
  - 所有可能的向量组成了一个向量空间
  - 向量空间的度量表示文本之间的“语义相似度”
    - 内积

$$x_i \cdot x_j$$

    - 标准化内积（余弦相似度）

$$\frac{x_i \cdot x_j}{\|x_i\| \cdot \|x_j\|}$$

单词向量空间

- 定义
  - 给定N个文本的集合$D = \{d_1, d_2, \dots, d_N\}$，以及M个单词的集合$W = \{w_1, w_2, \dots, w_M\}$
  - 将单词在文本中的出现情况用单词-文本矩阵（word-document matrix）表示，记作X
    - x_{ij}：单词w_i在文本d_j中出现的频数或权值
    - 单词-文本矩阵的第j列向量 x_j表示文本d_j

$$x_j = \begin{bmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{Mj} \end{bmatrix}, \quad j = 1, 2, \cdots, n$$

单词向量空间

- 实例：9个文档，11个单词

1. The Neatest Little **Guide** to **Stock Market Investing**
2. **Investing** For **Dummies**, 4th Edition
3. The Little **Book** of Common Sense **investing**: The Only Way to Guarantee Your Fair Share of **Stock Market** Returns
4. The Little **Book** of **Value Investing**
5. **Value Investing**: From Graham to Buffett and Beyond
6. **Rich Dad's Guide** to **Investing**: What the **Rich** Invest in, That the Poor and the Middle Class Do Not!
7. **Investing** in **Real Estate**, 5th Edition
8. **Stock Investing** For **Dummies**
9. **Rich Dad**'s Advisors: The ABC's of **Real Estate Investing**: The Secrets of Finding Hidden Profits Most Investors Miss

单词向量空间

- 单词-文本矩阵
  - 统计单词w_i在文本d_j中出现的频数
  - 列-文档；行-单词
  - 往往是一个稀疏矩阵

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

单词向量空间

- 单词-文本矩阵
  - 统计单词w_i在文本d_j中出现的权值
    - 单词频率-逆文本频率（TF-IDF）

$$\mathrm{TFIDF}_{ij} = \frac{\mathrm{tf}_{ij}}{\mathrm{tf}_{*j}} \log \frac{\mathrm{Df}}{\mathrm{df}_i}$$

      - tf_{ij}：单词w_i在文本d_j中出现的频数
      - tf_{*j}：文本d_j出现的所有单词的总频数
      - df_i：含有单词w_i的文本数
      - Df：文本集合中的文本数量

单词向量空间

- 优点
  - 模型简单
  - 计算效率高
- 局限
  - 内积相似度未必能够准确表达两个文本的语义相似度
    - 一词多义性 (polysemy)
    - 多词一义性 (synonymy)

单词向量空间

- 例：如右图
  - 一词多义性(polysemy)
    - 文本 d_1与d_2相似度并不高，但文本内容相似
  - 多词一义性(synonymy)
    - 文本 d_3与d_4相似度高，但文本内容不相似

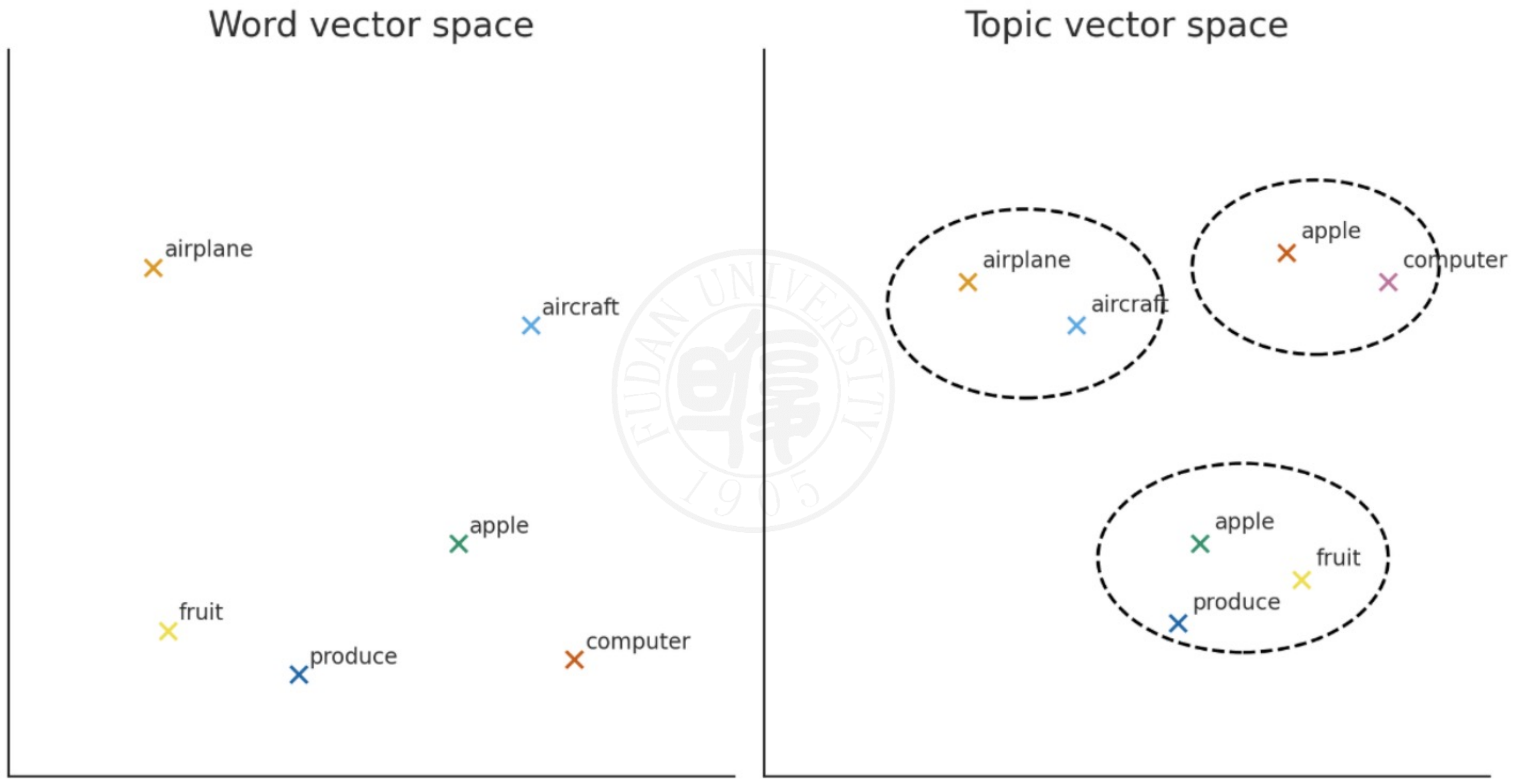
	d_1	d_2	d_3	d_4
airplane	2			
aircraft		2		
computer			1	
apple			2	3
fruit				1
produce	1	2	2	1

话题向量空间(topic vector space)

- 两个文本的语义相似度可以体现在两者的**话题相似度**上
  - 一个文本一般含有若干个话题。 如果两个文本的话题相似，那么两者的语义应该也相似

- 话题可以由若干个语义相关的单词表示
  - 同义词（如“airplane”与“aircraft"）可以表示同一个话题
  - 多义词（如“apple"）可以表示不同的话题

单词 vs. 话题向量空间



Credit: Dall-E 3

话题向量空间

- 基本概念
  - 给定一个文本，用话题空间的一个向量表示该文本
    - 向量的每一维度对应一个话题
    - 其数值为该话题在该文本中出现的权值
  - 用两个向量的内积或标准化内积表示对应的两个文本的语义相似度
- 话题的个数通常远小于单词的个数
  - 语义抽象、降低维度

话题向量空间

- 定义
  - 给定N个文本的集合$D = \{d_1, d_2, \dots, d_N\}$，以及$M$个单词的集合$W = \{w_1, w_2, \dots, w_M\}$
  - 假设所有文本共含有K个话题
  - 假设每个话题由一个定义在单词集合W上的M维向量表示，称为话题向量，即

$$t_k = \begin{bmatrix} t_{1k} \\ t_{2k} \\ \vdots \\ t_{Mk} \end{bmatrix}, \qquad k = 1,2, \dots, K$$

  - K个话题向量张成一个话题向量空间T

话题向量空间

- 单词-话题矩阵（word-topic matrix）
  - K个话题向量组成一个矩阵

$$T = [t_1, t_2, \ldots, t_K] = \begin{bmatrix} t_{11} & t_{12} & \cdots & t_{1K} \\ t_{21} & t_{22} & \cdots & t_{2K} \\ \vdots & \vdots & \vdots & \vdots \\ t_{M1} & t_{M2} & \cdots & t_{MK} \end{bmatrix}$$

话题向量空间

- 文本d_j在话题向量空间T的表示
  - 文本d_j在单词向量空间中表示为向量x_j
  - 将x_j投影到话题向量空间T中，得到在话题向量空间的向量y_j
  - y_j是一个K维向量， 其表达式为

$$y_j = \begin{bmatrix} y_{1j} \\ t_{2j} \\ \vdots \\ t_{Kj} \end{bmatrix}, \qquad j = 1,2,\dots,N$$

话题向量空间

- 话题-文本矩阵
  - 文本集D在话题向量空间T的表示

$$Y = [y_1, y_2, \cdots, y_N] = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1N} \\ y_{21} & y_{22} & \cdots & y_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ y_{K1} & y_{K2} & \cdots & y_{KN} \end{bmatrix}$$

话题向量空间

- 三个矩阵的关系
  - 单词-文本矩阵=单词-话题矩阵×话题-文本矩阵

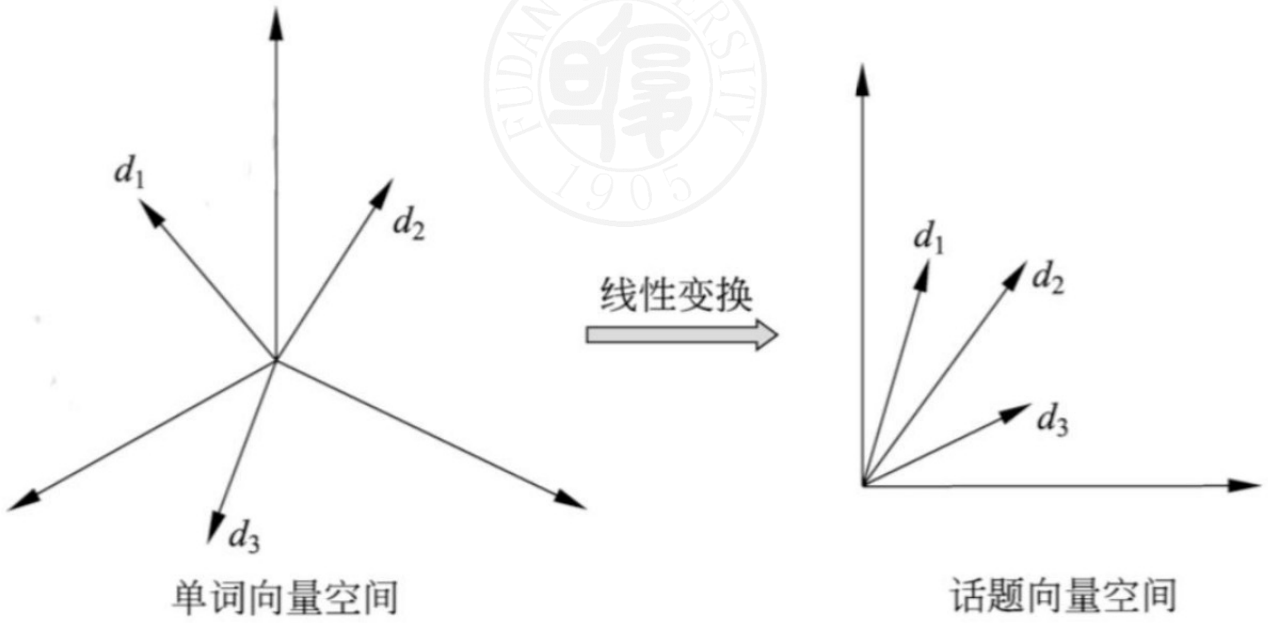
$$X_{MN} = T_{MK} \times Y_{KN}$$

  - 文档x_j表示成K个话题向量以y_j为系数的线性组合

$$x_j = y_{1j}t_1 + y_{2j}t_2 + \cdots + y_{Kj}t_K$$

话题向量空间

- 潜在语义分析（LSA），直观上是
  - 将文本在单词向量空间的表示，通过线性变换，转换为在话题向量空间中的表示

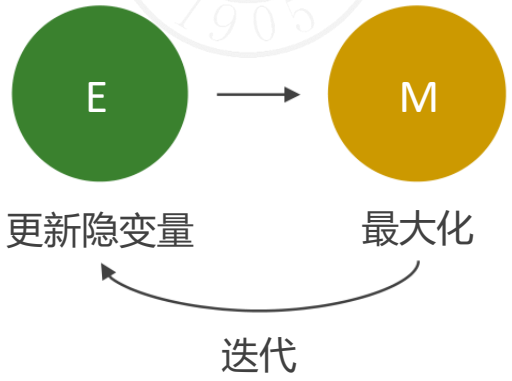


Outline

- 单词向量空间和话题向量空间
- 期望最大化算法 (EM)
- 概率潜在语义分析(pLSA)
- 潜在狄利克雷分布(LDA)

期望最大化算法

- 概率模型内部存在隐变量，不能直接用MLE估计参数
- 期望最大化算法（Exception Maximization Algorithm，EM）
  - 一种启发式算法，用于对含有**隐变量的概率模型**参数做极大似然估计
  - 通过迭代隐变量、参数，逼近最优解



期望最大化算法

- 实例：三硬币模型
  - 硬币A、B、C，正面概率π，p，q
  - 先抛硬币A，如果A是正面则需要抛硬币B，否则就抛硬币C。如果B或C是正面结果为1，否则结果为0
  - 独立进行N次试验。取N = 10，得到的观测结果：

    1,1,0,1,0,0,1,0,1,1
  - 问题：如何估算π，p，q？

期望最大化算法

- 实例：三硬币模型
  - 问题：如何估算π，p，q？
    - 用向量θ来表示整个模型中的未知参数π,p和q
    - 用Y表示观测变量，第i次观测结果的值记作y_i
    - 用Z表示隐含变量，记录在试验中抛掷A硬币的结果，第i次观测结果的值记作z_i

$$
\begin{aligned}
P(Y|\theta) &= \sum_Z P(Y,Z|\theta) \\
&= \sum_Z P(Z|\theta)P(Y|Z,\theta) \\
&= \pi p^y(1-p)^{1-y} + (1-\pi)q^y(1-q)^{1-y}
\end{aligned}
$$

期望最大化算法

- 实例：三硬币模型
  - 问题：只看结果，不看中间过程，估算π，p，q？
    - 极大似然估计

$$\max_{\theta} L(\theta) = \max_{\theta} \log \prod_{i=1}^{N} \mathrm{P}(y_i|\theta)$$

$$= \max_{\theta} \sum_{i=1}^{N} \log[\pi p^{y_i}(1-p)^{1-y_i} + (1-\pi) q^{y_i}(1-q)^{1-y_i}]$$

    - 直接求解析解比较困难，故用迭代法进行求解。

期望最大化算法

- 不完全数据 $P(Y|\theta)$
  - 对数似然函数

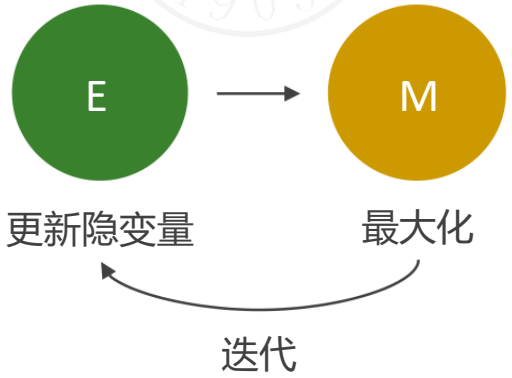
$$L(\theta) = \log \prod_{i=1}^{N} P(y_i|\theta) = \log \prod_{i=1}^{N} \sum_{z_i} P(y_i, z_i|\theta) = \sum_{i=1}^{N} \log \sum_{z_i} P(y_i, z_i|\theta)$$

- 完全数据 $P(Y, Z \mid \theta)$
  - 对数似然函数

$$L_c(\theta) = \log \prod_{i=1}^{N} P(y_i, z_i|\theta) = \sum_{i=1}^{N} \log P(y_i, z_i|\theta)$$

期望最大化算法

- EM 算法：迭代显式估计隐变量 Z
  - 令θ^t是参数θ在第t次迭代计算得到的估计值
  - 期望（E步）：根据θ^t计算此时的Z值，把不完整的似然补全成一个完整数据问题
  - 最大化（M步）：再代入Z，求第$t+1$次迭代的参数估计值θ^{t+1}



期望最大化算法

- EM算法
  - E步
    - 希望求 $\log P(Y,Z|\theta)$ ——不能直接计算， Z没有观测到
    - 转而求$\log P(Y,Z|\theta)$关于$P(Z|Y,\theta^t)$的期望，即关于隐变量的后验分布下的期望

$$Q(\theta,\theta^t) = E_Z[\log P(Y,Z|\theta) \mid Y,\theta^t]$$
$$= \sum_Z P(Z|Y,\theta^t) \log P(Y,Z|\theta)$$

  - Q函数：完全数据的对数似然$\log P(Y,Z|\theta)$关于在给定观测数据Y和当前参数θ^t下对未观测数据Z的条件概率分布$P(Z|Y,\theta^t)$的期望

期望最大化算法

- EM算法
  - M步
    - 求极大，即极大化Q函数得到参数的新估计值θ^{t+1}
      - 极大似然估计MLE

$$\theta^{t+1} = \text{argmax}_\theta \, Q(\theta, \theta^t)$$

      - 最大后验概率估计MAP

$$\theta^{t+1} = \text{argmax}_\theta \, Q(\theta, \theta^t) + \log P(\theta)$$

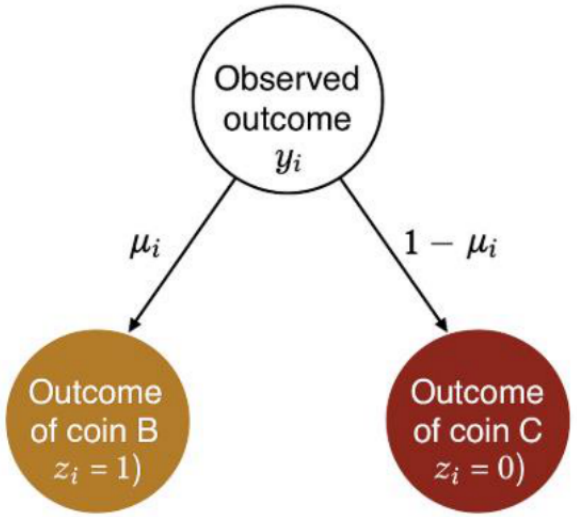
期望最大化算法

- 实例：三硬币模型
  - 输入：观测结果1,1,0,1,0,0,1,0,1,1
  - 模型参数取初值$\theta^{(0)} = \left(\pi^{(0)}, p^{(0)}, q^{(0)}\right) = (0.5,0.5,0.5)$
  - E步：

$$Q(\theta, \theta^t) = \sum_Z P(Z|Y, \theta^t) \log P(Y, Z|\theta)$$

    - 假设观测数据y_i是由抛硬币B得到的概率为μ_i

$$\begin{aligned}\mu_i &= P(z_i = 1|y_i, \theta) \\ &= \frac{P(y_i, z_i = 1|\theta)}{P(y_i, z_i = 1|\theta) + P(y_i, z_i = 0|\theta)} \\ &= \frac{\pi p^{y_i}(1-p)^{1-y_i}}{\pi p^{y_i}(1-p)^{1-y_i} + (1-\pi)q^{y_i}(1-q)^{1-y_i}}\end{aligned}$$



目标：$P(z_i|y_i, \theta)$

期望最大化算法

- 实例：三硬币模型
  - E步

$$\mu_i^{(t+1)} = \frac{\pi^{(t)}(p^{(t)})^{y_i}(1-p^{(t)})^{1-y_i}}{\pi^{(t)}(p^{(t)})^{y_i}(1-p^{(t)})^{1-y_i}+(1-\pi^{(t)})(q^{(t)})^{y_i}(1-q^{(t)})^{1-y_i}}$$

$$Q(\theta,\theta^t) = \sum_Z P(Z|Y,\theta^t)\log P(Y,Z|\theta)$$

$$= \sum_{i=1}^{N}\{\mu_i^{(t+1)}\log[\pi p^{y_i}(1-p)^{1-y_i}]$$

$$+(1-\mu_i^{(t+1)})\log[(1-\pi)q^{y_i}(1-q)^{1-y_i}]\}$$

期望最大化算法

- 实例：三硬币模型
  - M步：求函数$Q(\theta, \theta^t)$关于θ的偏导为0

$$\pi^{(t+1)} = \frac{1}{N}\sum_{i=1}^{N}\mu_i^{(t+1)}$$

$$p^{(t+1)} = \frac{\sum_{i=1}^{N}\mu_i^{(t+1)}y_i}{\sum_{i=1}^{N}\mu_i^{(t+1)}}$$

$$q^{(t+1)} = \frac{\sum_{i=1}^{N}\left(1-\mu_i^{(t+1)}\right)y_i}{\sum_{i=1}^{N}\left(1-\mu_i^{(t+1)}\right)}$$

期望最大化算法

- 极大似然估计：不完全数据

$$L(\theta) = \log \mathrm{P}(\mathrm{Y}|\theta) = \log \sum_Z P(Y, Z|\theta)$$

- EM算法：Q函数（完全数据）

$$Q(\theta, \theta^t) = \sum_Z P(Z|Y, \theta^t) \log P(Y, Z|\theta)$$

- 问题：为什么EM算法能近似实现对观测数据的极大似然估计？
  - EM算法能够单调增加观测数据的对数似然函数值

$$L(\theta^{t+1}) \geq L(\theta^t)$$

期望最大化算法

- 证明
  - 令$q_i(z_i)$是关于z_i的任意概率分布函数

$$
\begin{aligned}
L(\theta) &= \sum_{i=1}^{N} \log\left[\sum_{z_i} P(y_i, z_i|\theta)\right] \\
&= \sum_{i=1}^{N} \log\left[\sum_{z_i} q_i(z_i) \frac{P(y_i, z_i|\theta)}{q_i(z_i)}\right] \\
&\geq \sum_{i=1}^{N} \sum_{z_i} q_i(z_i) \log\left[\frac{P(y_i, z_i|\theta)}{q_i(z_i)}\right]
\end{aligned}
$$

Jensen不等式

$$f\left(\sum_{i=1}^{M} \lambda_i x_i\right) \leq \sum_{i=1}^{M} \lambda_i f(x_i)$$

  - 定义对数似然函数下界：函数$Q(\theta, q)$

$$
\begin{aligned}
Q(\theta, q) &= \sum_{i=1}^{N} \sum_{z_i} q_i(z_i) \log\left[\frac{P(y_i, z_i|\theta)}{q_i(z_i)}\right] \\
&= \sum_{i=1}^{N} \left[\mathrm{E}_{q_i} \log P(y_i, z_i|\theta) + H(q_i)\right]
\end{aligned}
$$

第i个数据的下界函数$L(\theta, q_i)$

期望最大化算法

- 证明
  - 定义第i个数据的下界函数$L(\theta, q_i)$

$$
\begin{aligned}
L(\boldsymbol{\theta}, q_i) &= \sum_{\mathbf{z}_i} q_i(\mathbf{z}_i) \log \frac{p(\mathbf{y}_i, \mathbf{z}_i|\boldsymbol{\theta})}{q_i(\mathbf{z}_i)} \\
&= \sum_{\mathbf{z}_i} q_i(\mathbf{z}_i) \log \frac{p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta}) p(\mathbf{y}_i|\boldsymbol{\theta})}{q_i(\mathbf{z}_i)} \\
&= \sum_{\mathbf{z}_i} q_i(\mathbf{z}_i) \log \frac{p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta})}{q_i(\mathbf{z}_i)} + \sum_{\mathbf{z}_i} q_i(\mathbf{z}_i) \log p(\mathbf{y}_i|\boldsymbol{\theta}) \\
&= -\mathbb{KL}\left(q_i(\mathbf{z}_i) || p(\mathbf{z}_i|\mathbf{y}_i, \boldsymbol{\theta})\right) + \log p(\mathbf{y}_i|\boldsymbol{\theta})
\end{aligned}
$$

期望最大化算法

- 证明
  - 定义第i个数据的下界函数$L(\theta, q_i)$
    - 当$q_i(z_i) = P(z_i|y_i, \theta)$，KL散度项为0，函数$L(\theta, q_i)$取最大值

$$L(\theta, q_i) = \log P(y_i|\theta)$$

      - 令θ^t是第t次迭代参数的估计值
      - E步：隐变量条件分布 $q_i^t(z_i) = P(z_i|y_i, \theta^t)$

$$L(\theta^t, q_i^t) = \log P(y_i|\theta^t)$$

$$\Rightarrow \quad L(\theta^t) = \sum_{i=1}^{N} \log P(y_i|\theta^t) = \sum_{i=1}^{N} L\left(\theta^t, q_i^t\right) = Q(\theta^t, q^t)$$

期望最大化算法

- 证明
  - M步：估计函数$Q(\theta, q^t)$
    - 对数似然函数下界为

$$Q(\theta, q^t) = \sum_{i=1}^{N} \left[\mathrm{E}_{q_i^t} \log P(y_i, z_i|\theta) + H(q_i^t) \right]$$
$$= Q(\theta, \theta^t) + \sum_{i=1}^{N} H(q_i^t)$$

  - 第一项：完全数据的对数似然的期望$Q(\theta, \theta^t)$
  - 第二项：分布q_i^t的熵，与θ^t有关，与θ无关——常数

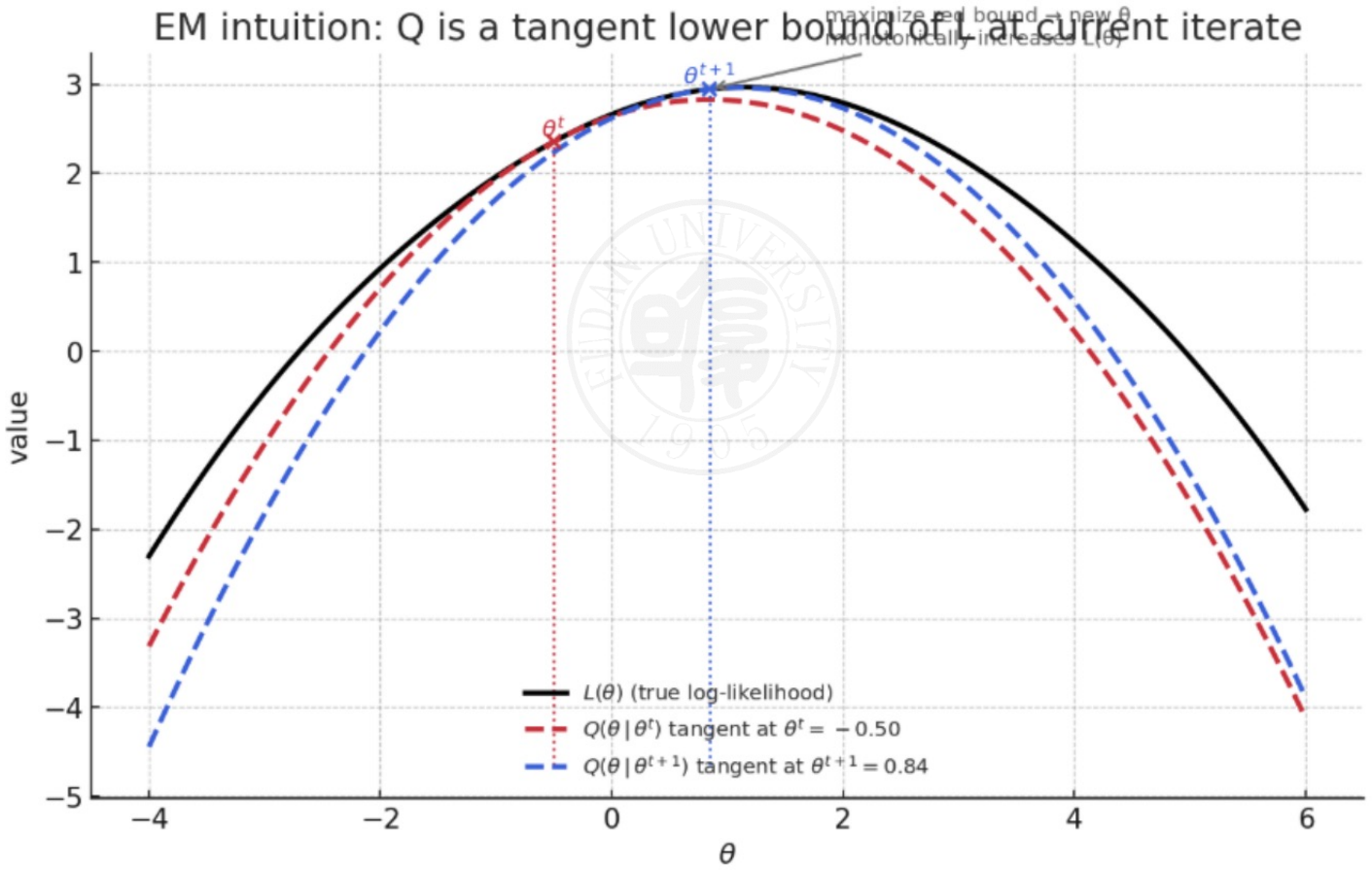
期望最大化算法

- 证明
  - 估计第t+1次迭代参数θ^{t+1}

$$\theta^{t+1} = \text{argmax}_\theta Q(\theta, \theta^t) = \text{argmax}_\theta Q(\theta, q^t)$$

$$L(\theta^{t+1}) \geq Q(\theta^{t+1}, q^t) \geq Q(\theta^t, q^t) = L(\theta^t)$$

期望最大化算法


EM intuition: Q is a tangent lower bound of L at current iterate
maximize red bound → new θ
Monotonically increases L(θ)
θ^{t+1}
θ^t
value
θ
$L(\theta)$ (true log-likelihood)
$Q(\theta\,|\,\theta^t)$ tangent at $\theta^t = -0.50$
$Q(\theta\,|\,\theta^{t+1})$ tangent at $\theta^{t+1} = 0.84$

Outline

- 单词向量空间和话题向量空间
- 期望最大化算法 (EM)
- 概率潜在语义分析(pLSA)
- 潜在狄利克雷分布(LDA)

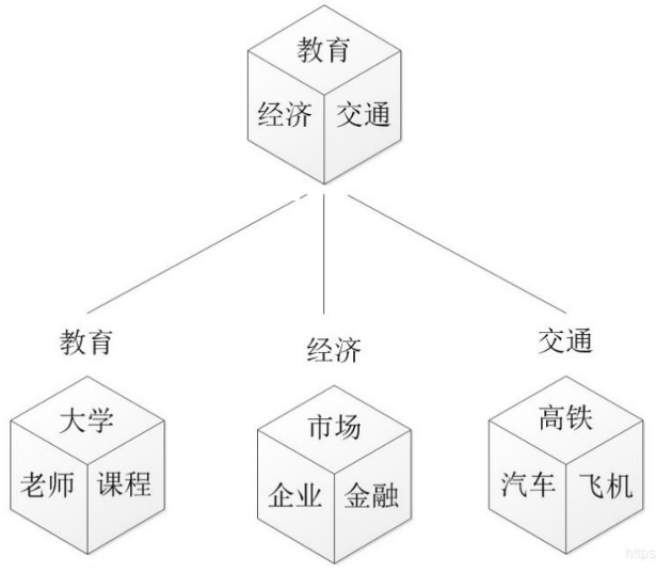
概率潜在语义分析（pLSA）

- 利用概率生成模型对文本集合**进行话题分析**的无监督学习方法

- 特点
  - 用隐变量表示话题（topic）
  - 整个模型表示文本生成话题，话题生成单词，从而得到单词-文本共现数据的过程
  - 隐含的假设每个文本由一个话题分布决定，每个话题由一个单词分布决定

概率潜在语义分析（pLSA）

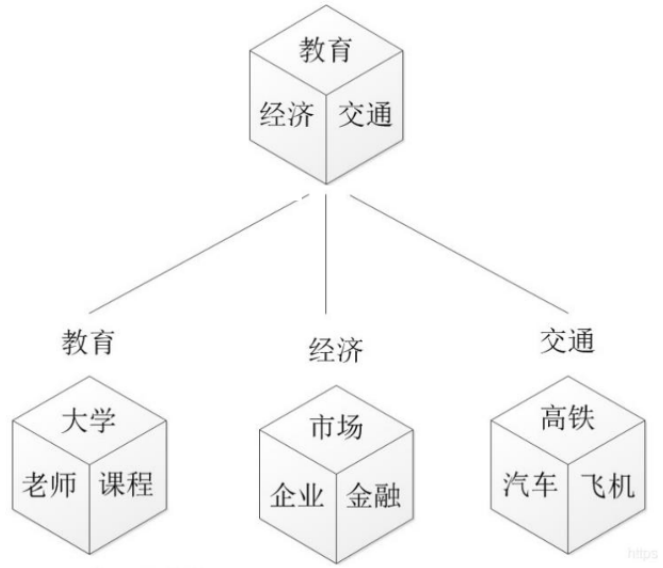
- 实例：扔骰子游戏
  - 假设一共有K个可选的话题，有M个可选的单词
    - 假设你每写一篇文档会制作一颗**K面**的“文本-话题”骰子和K个**M面**的“话题-单词”骰子
    - 每写一个单词，先扔该“文本-话题”骰子选择话题，得到话题的结果后，使用和话题结果对应的那颗“话题-单词”骰子，扔该骰子选择要写的单词



概率潜在语义分析（pLSA）

- 实例：扔骰子游戏
  - 话题分布{教育：0.5，经济：0.3，交通：0.2}
  - **教育**对应的词分布{大学：0.5，老师：0.3，课程：0.2}
  - **经济**对应的词分布{市场：0.4，企业：0.2，金融：0.4}
  - **交通**对应的词分布{高铁：0.5，汽车：0.2，飞机：0.3}

问：生成文档{**大学模式下如何看待大学与企业的联系**}的概率？

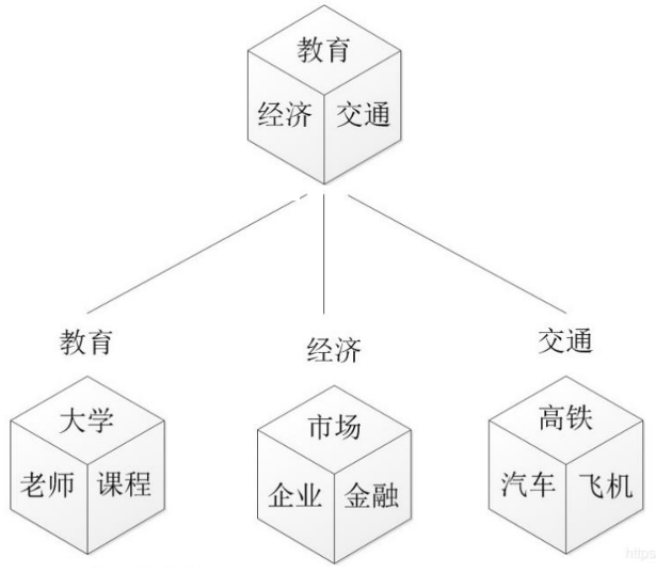


概率潜在语义分析（pLSA）

问：生成文档{**大学模式下如何看待大学与企业的联系**}的概率？

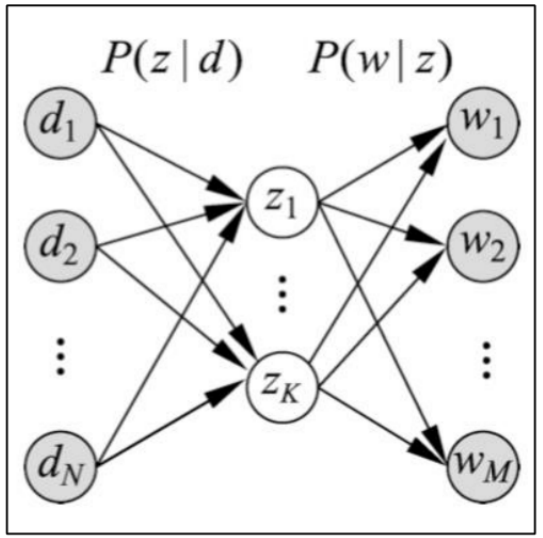
- 文档=大学大学企业

$$
\begin{aligned}
&P(\text{大学大学企业}) \\
&= P(\text{单词} = \text{大学})^2 P(\text{单词} = \text{企业}) \\
&= (P(\text{话题} = \text{教育}) P(\text{单词} = \text{大学}|\text{话题} = \text{教育}))^2 \\
&\quad \times P(\text{话题} = \text{经济}) P(\text{单词} = \text{企业}|\text{话题} = \text{经济}) \\
&= 0.00375
\end{aligned}
$$



概率潜在语义分析（pLSA）

- 真实问题及pLSA基本思想
  - 给定一个文本集合 (corpus)，如何描述文档-单词的共现关系？
  - pLSA：对文本集合，发现每个文本的话题，以及每个话题包含的单词
  - 话题是潜在的

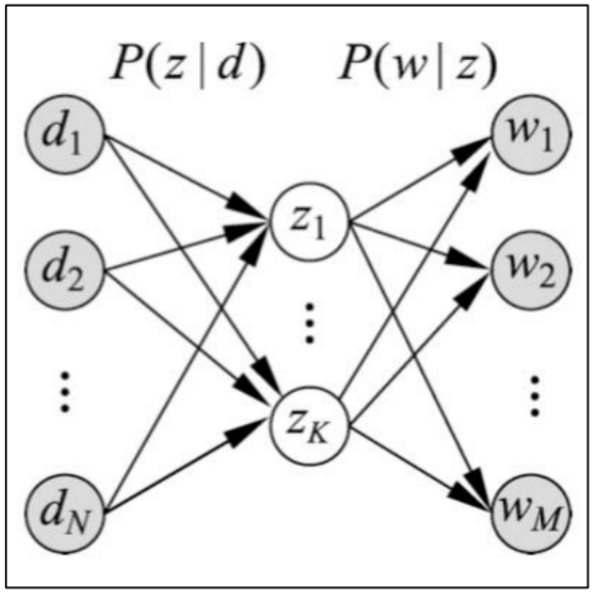


概率潜在语义分析（pLSA）

数学上，

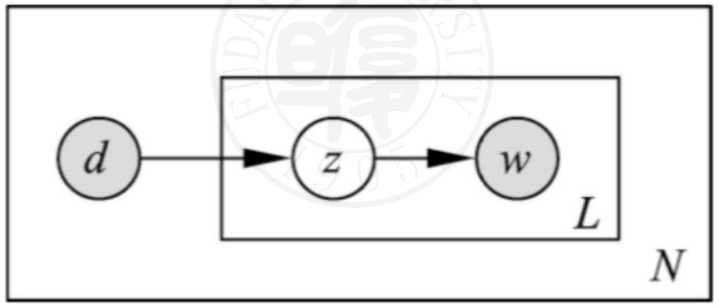
- 三类核心元素
  - 文本集$D = \{d_1, d_2, \ldots, d_N\}$
  - 单词集$W = \{w_1, w_2, \ldots, w_M\}$
  - 话题集$Z = \{z_1, z_2, \ldots, z_K\}$
- 已知
  - $P(d)$：生成文本d的概率
- 希望求解关于话题隐变量的条件分布
  - $P(z|d)$：文本的话题分布
  - $P(w|z)$：话题的单词分布
- 均为categorical分布

文本、话题、单词之间的关系



概率潜在语义分析（pLSA）

- 基本约定
  - 条件独立性假设
    - 假设在话题z给定条件下，单词w与文本d是条件独立的



  - 图模型表示的约定
    - 实心圆表示观测变量，空心圆表示隐变量，
      箭头表示概率依存关系，方框表示多次重复，方框内数字表示重复次数

概率潜在语义分析（pLSA）

- 生成模型
  - 可观测的数据：单词–文档对 (w, d) 的共现次数 $n(w, d)$
  - 整个语料库的生成概率：

$$P(X) = \prod_{(w,d)} P(w, d)^{n(w,d)}$$

    - 其中，$n(w, d)$表示单词 w在文档 d中的出现次数，总次数为 $N \times L$

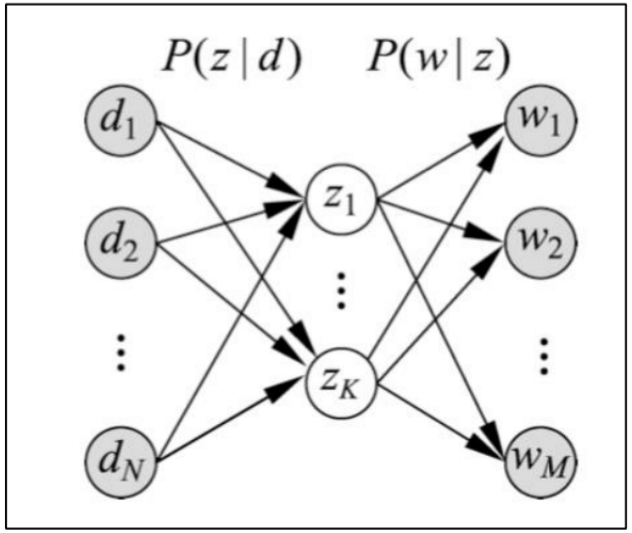
概率潜在语义分析（pLSA）

- 生成模型
  - 每个单词-文本对(w, d)的生成概率

$$
\begin{aligned}
P(w,d) &= P(d)P(w|d) \\
&= P(d)\sum_z P(w,z|d) \\
&= P(d)\sum_z P(z|d)P(w|z)
\end{aligned}
$$

概率潜在语义分析（pLSA）

- 模型参数
  - 生成模型参数
    - 定义$P(z|d),\ P(w|z)$
    - 参数个数 $O(NK+MK)$
      - 现实中$K \ll M$

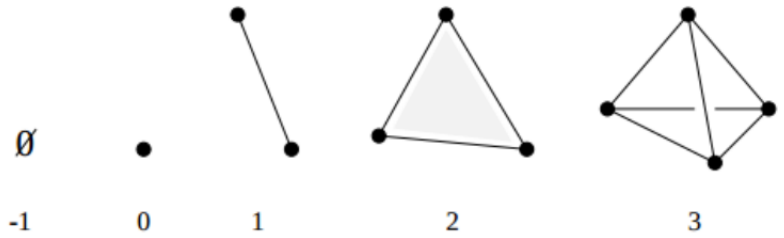


概率潜在语义分析（pLSA）

- 参数空间
  - 概率分布$P(w|d)$可以由M维空间的$(M-1)$单纯形中的点表示

$$\sum_{i=1}^{M} P(w_i|d) = 1, \quad 0 \leqslant P(w_i|d) \leqslant 1, \quad i = 1, \cdots, M$$

  - 称这$(M-1)$单纯形为单词单纯形



概率潜在语义分析（pLSA）

- 参数空间
  - 概率分布$P(w|z)$也存在于单词单纯形中，
    并且可以由$(M-1)$ 单纯形上的K个点表示

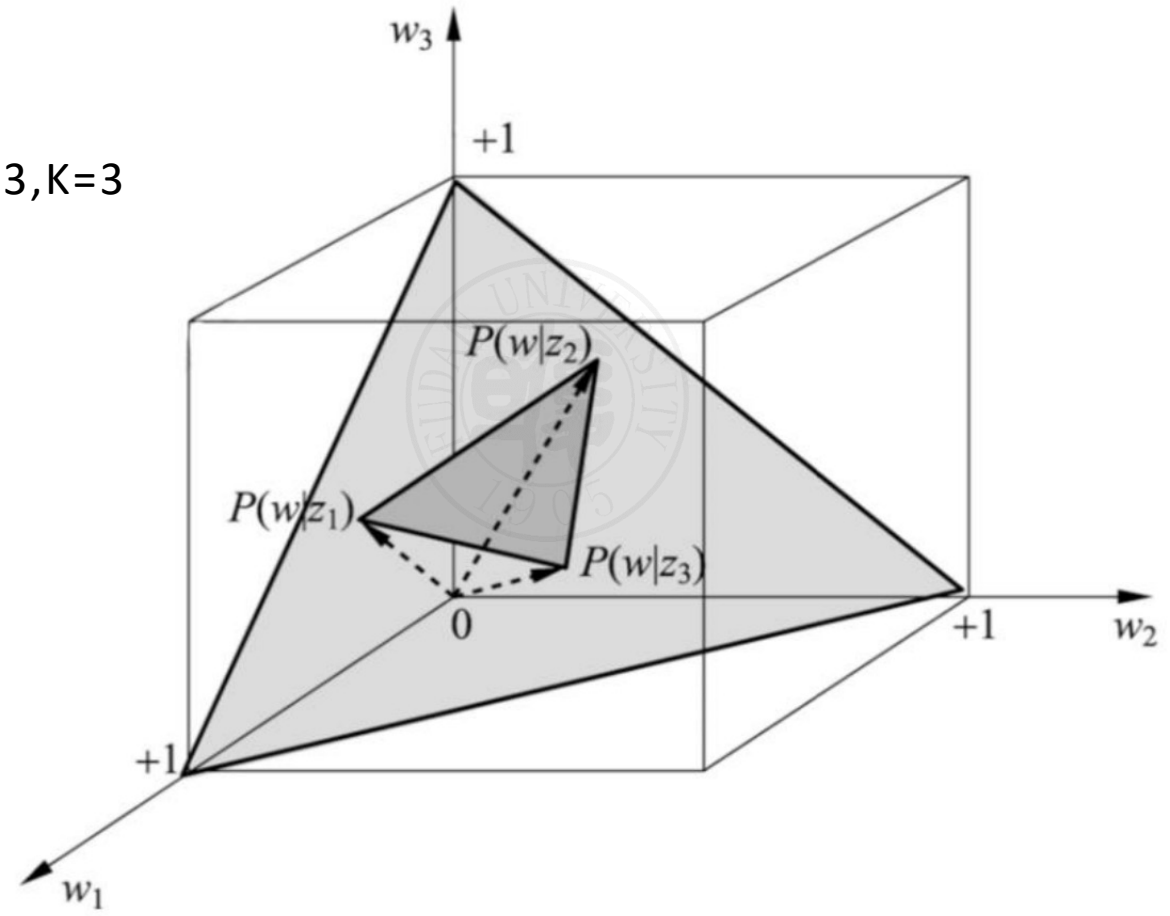
$$P(w|d) = \sum_z P(z|d)P(w|z)$$

$$\sum_z P(z|d) = 1$$

  - 称这K个点构成的单纯形为话题单纯形

概率潜在语义分析（pLSA）

M=3,K=3



概率潜在语义分析（pLSA）

- 模型求解：极大似然估计
  - 最大化单词-文本共现数据T生成概率

$$\max P(T) = \max \prod_{(w,d)} P(w,d)^{n(w,d)}$$

  - 对数似然函数

$$L = \log \prod_{i=1}^{M} \prod_{j=1}^{N} P(w_i, d_j)^{n(w_i,d_j)}$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \log P(w_i, d_j)$$

$$= \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \left[\log P(d_j) + \log \sum_{k=1}^{K} P(w_i \mid z_k) P(z_k \mid d_j) \right]$$

概率潜在语义分析（pLSA）

- 模型求解：EM算法
  - E步：计算Q函数
    - 完全数据的对数似然函数对不完全数据的条件分布的期望
      - 完全数据：$P(w_i, d_j, z_k) = P(d_j)P(w_i|z_k)P(z_k|d_j)$
      - 不完全数据：话题的后验概率 $P(z_k|w_i, d_j)$

$$Q = \sum_{k=1}^{K} \left\{ \sum_{i=1}^{M} \sum_{j=1}^{N} n(w_i, d_j) \log \left(P(d_j)P(w_i|z_k)P(z_k|d_j) \right) \right\} P(z_k|w_i, d_j)$$

概率潜在语义分析（pLSA）

- 模型求解：EM算法
  - E步：Q转化为Q′
    - 可以从数据中直接统计得出$P(d_j)$

$$Q' = \sum_{i=1}^{M}\sum_{j=1}^{N} n(w_i, d_j) \sum_{k=1}^{K} P(z_k|w_i, d_j) \log[P(w_i|z_k)P(z_k|d_j)]$$

    - 条件分布概率$P(z_k|w_i, d_j)$：Bayes rule

$$P(z_k|w_i, d_j) = \frac{P(w_i|z_k)P(z_k|d_j)}{\sum_{k=1}^{K} P(w_i|z_k)P(z_k|d_j)}$$

这里的 $P(w|z)$ 和 $P(z|d)$是前一迭代步的值，即为已知的

概率潜在语义分析（pLSA）

- 模型求解：EM算法
  - M步：极大化Q'函数
    - 条件概率分布$P(w_i|z_k)$和$P(z_k|d_j)$未知

$$\sum_{i=1}^{M} P(w_i|z_k) = 1, \quad k = 1, 2, \cdots, K$$

$$\sum_{k=1}^{K} P(z_k|d_j) = 1, \quad j = 1, 2, \cdots, N$$

概率潜在语义分析（pLSA）

- 模型求解：EM算法
  - M步：极大化Q'函数——拉格朗日乘子法
    - 定义拉格朗日函数Λ

$$\Lambda = Q' + \sum_{k=1}^{K} \tau_k \left(1 - \sum_{i=1}^{M} P(w_i \mid z_k)\right) + \sum_{j=1}^{N} \rho_j \left(1 - \sum_{k=1}^{K} P(z_k \mid d_j)\right)$$

    - 将拉格朗日函数Λ分别对$P(w_i|z_k)$和$P(z_k|d_j)$求偏导数，并令其等于0

$$\sum_{j=1}^{N} n(w_i, d_j) P(z_k|w_i, d_j) - \tau_k P(w_i|z_k) = 0, \quad i = 1, 2, \cdots, M; \quad k = 1, 2, \cdots, K$$

$$\sum_{i=1}^{M} n(w_i, d_j) P(z_k|w_i, d_j) - \rho_j P(z_k|d_j) = 0, \quad j = 1, 2, \cdots, N; \quad k = 1, 2, \cdots, K$$

概率潜在语义分析（pLSA）

- 模型求解：EM算法
  - M步：极大化Q'函数

$$P(w_i|z_k) = \frac{\sum_{j=1}^{N} n(w_i, d_j) P(z_k|w_i, d_j)}{\sum_{m=1}^{M} \sum_{j=1}^{N} n(w_m, d_j) P(z_k|w_m, d_j)}$$

$$P(z_k|d_j) = \frac{\sum_{i=1}^{M} n(w_i, d_j) P(z_k|w_i, d_j)}{n(d_j)}$$

概率潜在语义分析（pLSA）

输入：设单词集合为 $W=\{w_1,w_2,\cdots,w_M\}$，文本集合为 $D=\{d_1,d_2,\cdots,d_N\}$，话题集合为 $Z=\{z_1,z_2,\cdots,z_K\}$，共现数据 $\{n(w_i,d_j)\},i=1,2,\cdots,M,j=1,2,\cdots,N$；

输出：$P(w_i|z_k)$ 和 $P(z_k|d_j)$。

（1）设置参数 $P(w_i|z_k)$ 和 $P(z_k|d_j)$ 的初始值。

（2）迭代执行以下 E 步，M 步，直到收敛为止。

E 步：

$$P(z_k|w_i,d_j)=\frac{P(w_i|z_k)P(z_k|d_j)}{\sum_{k=1}^{K}P(w_i|z_k)P(z_k|d_j)}$$

M 步：

$$P(w_i|z_k)=\frac{\sum_{j=1}^{N}n(w_i,d_j)P(z_k|w_i,d_j)}{\sum_{m=1}^{M}\sum_{j=1}^{N}n(w_m,d_j)P(z_k|w_m,d_j)}$$

$$P(z_k|d_j)=\frac{\sum_{i=1}^{M}n(w_i,d_j)P(z_k|w_i,d_j)}{n(d_j)}$$

概率潜在语义分析（pLSA）

- 实例：9个文档，11个单词

1. The Neatest Little **Guide** to **Stock Market Investing**
2. **Investing** For **Dummies**, 4th Edition
3. The Little **Book** of Common Sense **investing**: The Only Way to Guarantee Your Fair Share of **Stock Market** Returns
4. The Little **Book** of **Value Investing**
5. **Value Investing**: From Graham to Buffett and Beyond
6. **Rich Dad's Guide** to **Investing**: What the **Rich** Invest in, That the Poor and the Middle Class Do Not!
7. **Investing** in **Real Estate**, 5th Edition
8. **Stock Investing** For **Dummies**
9. **Rich Dad**'s Advisors: The ABC's of **Real Estate Investing**: The Secrets of Finding Hidden Profits Most Investors Miss

Code of this demo is available at: https://github.com/fengdu78/lihang-code

概率潜在语义分析（pLSA）

- 实例：9个文档，11个单词
  - 输入：
    - 单词-文本矩阵
    - 话题个数K=2
  - 初始化条件概率分布$P(w_i|z_k)$和$P(z_k|d_j)$

Index Words	Titles								
	T1	T2	T3	T4	T5	T6	T7	T8	T9
book			1	1					
dads						1			1
dummies		1						1	
estate							1		1
guide	1					1			
investing	1	1	1	1	1	1	1	1	1
market	1		1						
real							1		1
rich						2			1
stock	1		1					1	
value				1	1				

$$\sum_{i=1}^{M} P(w_i|z_k) = 1, \quad k = 1, 2, \cdots, K$$

$$\sum_{k=1}^{K} P(z_k|d_j) = 1, \quad j = 1, 2, \cdots, N$$

Code of this demo is available at: https://github.com/fengdu78/lihang-code

概率潜在语义分析（pLSA）

- 实例：9个文档，11个单词
  - 输出
    - $P(w_i|z_k)$

```
array([[0.64238757, 0.05486094, 0.18905573, 0.24047994, 0.41230822,
        0.38136674, 0.81525232, 0.74314243, 0.32465342, 0.19798429,
        0.72010476],
       [0.6337431 , 0.79442181, 0.96755364, 0.22924392, 0.99367301,
        0.20277986, 0.40513752, 0.51164374, 0.73750246, 0.22300907,
        0.17339099]])
```

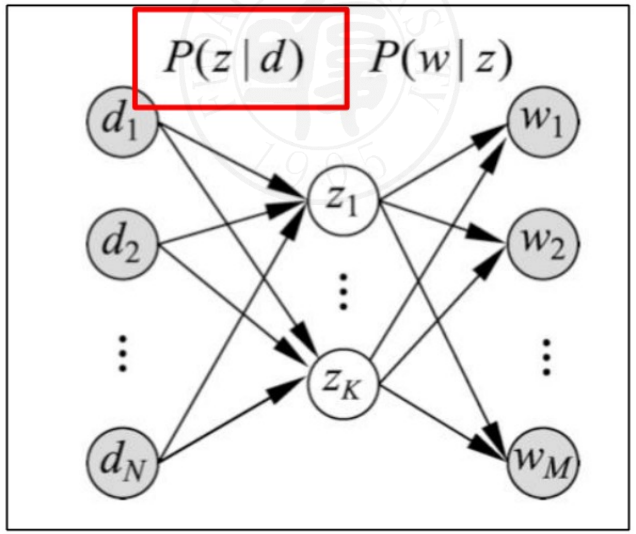
    - $P(z_k|d_j)$

```
array([[7.14884177e-01, 2.85115823e-01],
       [5.38307075e-02, 9.46169293e-01],
       [1.00000000e+00, 3.40624611e-11],
       [1.00000000e+00, 1.12459358e-24],
       [1.00000000e+00, 5.00831891e-42],
       [1.66511004e-19, 1.00000000e+00],
       [1.00000000e+00, 8.02144289e-15],
       [1.04149223e-02, 9.89585078e-01],
       [5.96793031e-03, 9.94032070e-01]])
```

Code of this demo is available at: https://github.com/fengdu78/lihang-code

概率潜在语义分析（pLSA）

- 缺点
  - pLSA可以生成其所在数据集的文档的模型，但却不能生成新文档的模型



Outline

- 单词向量空间和话题向量空间
- 期望最大化算法 (EM)
- 概率潜在语义分析(pLSA)
- 潜在狄利克雷分布(LDA)

潜在狄利克雷分布（LDA）

- LDA模型是文本集合的生成概率模型:文本生成话题，话题生成单词
- 假设每个文本由话题的一个多项分布表示，每个话题由单词的一个多项分布表示
  - 话题分布的先验分布是狄利克雷分布
  - 单词分布的先验分布是狄利克雷分布

潜在狄利克雷分布（LDA）

- LDA vs. pLSA
  - 可类比VAE vs. AE
  - **pLSA 与 AE**：
    - 学习从输入到隐空间的确定性映射
    - 隐变量只解释输入数据，而不是分布的一部分
  - **LDA 与 VAE**：
    - 显式引入了一个隐变量的先验分布
    - 从先验分布中采样=创建新样本，使得模型具备采样生成能力

潜在狄利克雷分布（LDA）

- 为什么先验分布都是狄利克雷分布？
  - 话题分布和单词分布都是多项分布
  - 共轭先验：狄利克雷分布

定义 20.1（多项分布） 若多元离散随机变量 $X = (X_1, X_2, \cdots, X_k)$ 的概率质量函数为

$$P(X_1 = n_1, X_2 = n_2, \cdots, X_k = n_k) = \frac{n!}{n_1! n_2! \cdots n_k!} p_1^{n_1} p_2^{n_2} \cdots p_k^{n_k}$$

$$= \frac{n!}{\prod_{i=1}^{k} n_i!} \prod_{i=1}^{k} p_i^{n_i} \tag{20.1}$$

其中 $p = (p_1, p_2, \cdots, p_k)$，$p_i \geqslant 0, i = 1, 2, \cdots, k$，$\sum_{i=1}^{k} p_i = 1$，$\sum_{i=1}^{k} n_i = n$，则称随机变量 X 服从参数为 (n, p) 的多项分布，记作 $X \sim \mathrm{Mult}(n, p)$。

潜在狄利克雷分布

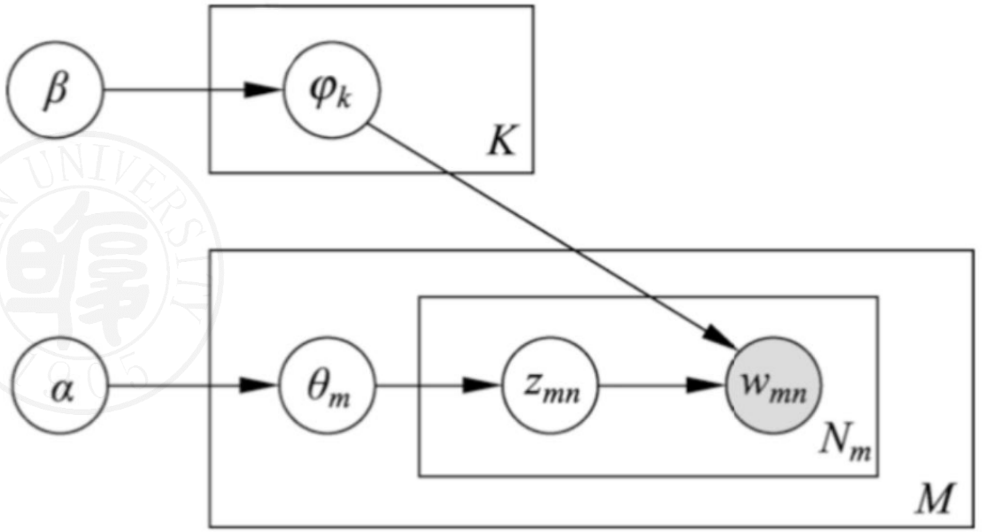
- LDA的记号M表示文本数，V表示单词集合的单词数，与前面记号不同
  - LDA使用了三个集合
    - 单词集合$W = \{w_1, \ldots, w_V\}$
    - 话题集合$\mathrm{Z} = \{z_1, \ldots, z_K\}$
    - 文本集合$\mathcal{D} = \{d_1, d_2, \ldots, d_M\}$
      - d_m文档的单词序列记$\boldsymbol{w}_m = \{w_{m1}, \ldots, w_{mn} \ldots, w_{mN_m}\}$

潜在狄利克雷分布

- 文本集合的自动生成过程：
  - 首先，基于单词分布的狄利克雷先验（超参数 β）生成 K 个单词分布 ϕ_k，即决定 K 个话题各自的内容；
  - 之后，对每篇文本m，基于话题分布的狄利克雷先验（超参数 α）生成话题分布 θ_m，即决定每篇文本各自的话题构成；
  - 然后，对该文本的位置n，按话题分布 θ_m 生成一个话题 z_mn，接着根据单词分布ϕ_{z_mn}生成单词w_mn，整体构成单词序列，即生成一篇文本
  - 重复这个过程生成所有文本

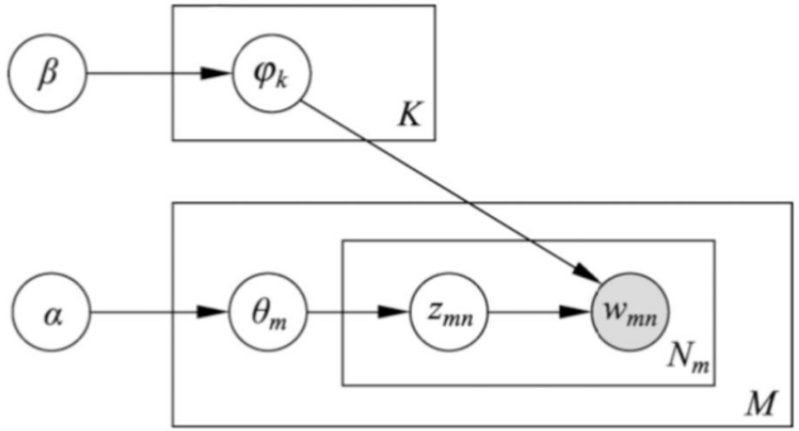
潜在狄利克雷分布

- 概率图模型
  - 结点表示随机变量
    - 实心结点是观测变量
    - 空心结点是隐变量
  - 有向边表示概率依存关系
  - 矩形（板块）表示重复，板块内数字表示重复的次数



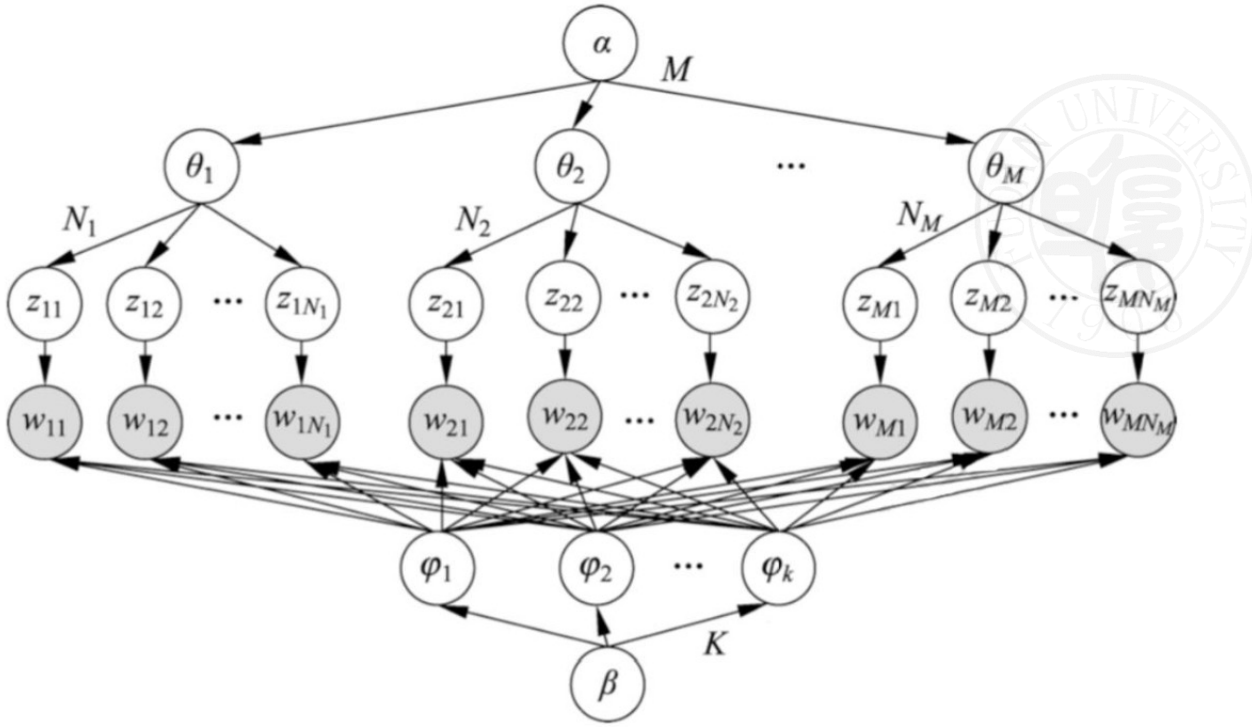
潜在狄利克雷分布

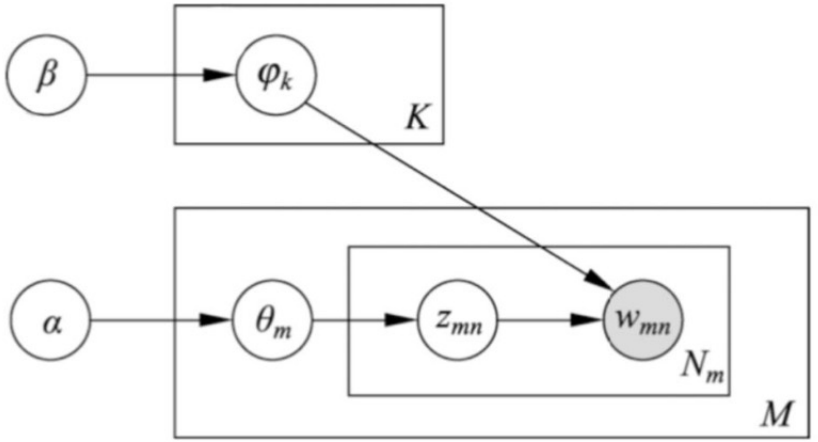
- 概率图模型
  - α, β：狄利克雷先验分布的超参数
  - φ_k：第k个话题的单词分布
  - θ_m：第m个文本的话题分布
  - z_{mn}：第m个文本中第n个单词对应的话题
  - w_{mn}：生成单词



潜在狄利克雷分布

- 概率图模型：有向图





潜在狄利克雷分布

- 文本生成算法

算法 20.1（LDA 的文本生成算法）

（1）对于话题 z_k（$k = 1, 2, \cdots, K$）：

生成多项分布参数 $\varphi_k \sim \mathrm{Dir}(\beta)$，作为话题的单词分布 $p(w|z_k)$；

（2）对于文本 d_m（$m = 1, 2, \cdots, M$）：

生成多项分布参数 $\theta_m \sim \mathrm{Dir}(\alpha)$，作为文本的话题分布 $p(z|d_m)$；

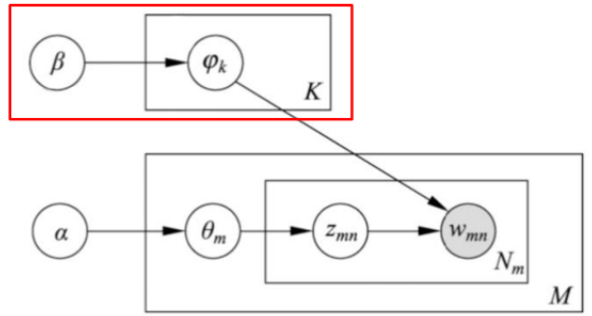
（3）对于文本 d_m 的单词 w_{mn}（$m = 1, 2, \cdots, M$，$n = 1, 2, \cdots, N_m$）：

（a）生成话题 $z_{mn} \sim \mathrm{Mult}(\theta_m)$，作为单词对应的话题；

（b）生成单词 $w_{mn} \sim \mathrm{Mult}(\varphi_{z_{mn}})$。

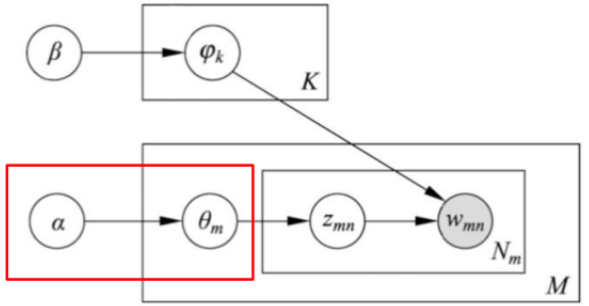
潜在狄利克雷分布

- 基本思想
  - 话题生成单词：单词分布
    - 随机生成K个话题的单词分布$p(w|z_k), k = 1, \dots, K$
    - 分布$p(w|z_k)$服从多项分布，其参数为φ_k
    - 参数φ_k服从超参数为β的狄利克雷先验分布
    - 参数$\varphi_k = (\varphi_{k1}, \dots, \varphi_{kV})$是一个V维向量，其中$\varphi_{kv}$表示话题$z_k$生成单词$w_v$的概率
    - 超参数β也是一个V维向量
    - 所有话题的参数向量构成一个$K \times V$矩阵$\varphi = \{\varphi_k\}_{k=1}^{K}$



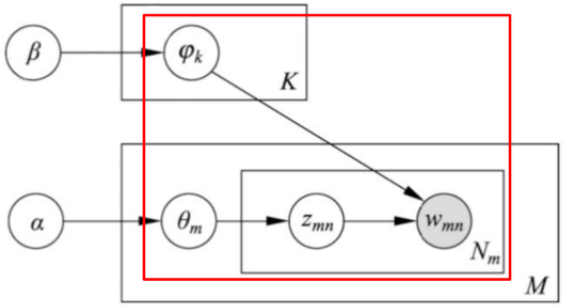
潜在狄利克雷分布

- 基本思想
  - 文本生成话题：话题分布
    - 随机生成M个文本的话题分布$p(z|\ d_m\ ), m = 1, \dots, M$
    - 分布 $p(z|\ d_m\ )$服从多项分布，其参数为θ_m
    - 参数θ_m服从狄利克雷先验分布，其超参数为α
    - 参数$\theta_m = (\theta_{m1}, \dots, \theta_{mK})$是一个K维向量，其中$\theta_{mk}$ 表示文本 d_m 生成话题z_k的概率
    - 超参数α也是一个K维向量
    - 所有文本的参数向量构成一个 M x K 矩阵$\theta = \{\theta_m\}_{m=1}^{M}$



潜在狄利克雷分布

- 基本思想
  - 生成文本的单词序列
    - 文本 d_m 中的每一个单词w_{mn}由该文本的话题分布 $p(z|\,d_m\,)$ 以及所有话题的单词分布 $p(w|z_k)$ 决定
    - 按文本的话题分布 $p(z|\,d_m\,)\sim\mathrm{Mult}(\theta_m)$ 随机生成一个话题序列$\boldsymbol{z}_m = (z_{m1}, z_{m2}, \ldots, z_{mN_m})$
    - 按话题的单词分布 $p(w|z_k)\sim\mathrm{Mult}(\varphi_k)$ 随机生成一个单词序列$\boldsymbol{w}_m = (w_{m1}, w_{m2}, \ldots, w_{mN_m})$



潜在狄利克雷分布

- 生成模型求解
  - 最大后验概率估计（MAP）

$$p(\theta, \varphi, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = \frac{p(\boldsymbol{w}, \boldsymbol{z}, \theta, \varphi|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)}$$

潜在狄利克雷分布

- LDA模型整体是由观测变量和隐变量组成的联合概率分布

$$p(\mathbf{w}, \mathbf{z}, \theta, \varphi|\alpha, \beta) = \prod_{k=1}^{K} p(\varphi_k|\beta) \prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N_m} p(z_{mn}|\theta_m)p(w_{mn}|z_{mn}, \varphi)$$

  - 观测变量**w**表示所有文本中的单词序列
  - 隐变量z表示所有文本中的话题序列
  - 隐变量$\theta = \{\theta_m\}_{m=1}^{M}$表示所有文本的话题分布的参数
  - 隐变量$\varphi = \{\varphi_k\}_{k=1}^{K}$表示所有话题的单词分布的参数
  - α 和 β 是超参数

潜在狄利克雷分布

- 第m个文本的联合概率分布可以表为

$$p(\mathbf{w}_m, \mathbf{z}_m, \theta_m, \varphi | \alpha, \beta) = \prod_{k=1}^{K} p(\varphi_k|\beta) p(\theta_m|\alpha) \prod_{n=1}^{N_m} p(z_{mn}|\theta_m) p(w_{mn}|z_{mn}, \varphi)$$

  - 观测变量$\boldsymbol{w}_m$表示第m个文本中的单词序列
  - 隐变量$\boldsymbol{z}_m$表示第m个文本中的话题序列
  - 隐变量θ_m表示第m个文本的话题分布的参数

潜在狄利克雷分布

- 参数θ_m和φ给定条件下第m个文本的生成概率

$$p(\mathbf{w}_m|\theta_m,\varphi)=\prod_{n=1}^{N_m}\left[\sum_{k=1}^{K}p(z_{mn}=k|\theta_m)p(w_{mn}|\varphi_k)\right]$$

- 超参数α和β给定条件下第m个文本的生成概率

$$p(\mathbf{w}_m|\alpha,\beta)=\prod_{k=1}^{K}\int p(\varphi_k|\beta)\left[\int p(\theta_m|\alpha)\prod_{n=1}^{N_m}\left[\sum_{l=1}^{K}p(z_{mn}=l|\theta_m)p(w_{mn}|\varphi_l)\right]\mathrm{d}\theta_m\right]\mathrm{d}\varphi_k$$

- 超参数α和β给定条件下所有文本的生成概率

$$p(\mathbf{w}|\alpha,\beta)=\prod_{k=1}^{K}\int p(\varphi_k|\beta)\left[\prod_{m=1}^{M}\int p(\theta_m|\alpha)\prod_{n=1}^{N_m}\left[\sum_{l=1}^{K}p(z_{mn}=l|\theta_m)p(w_{mn}|\varphi_l)\right]\mathrm{d}\theta_m\right]\mathrm{d}\varphi_k$$

潜在狄利克雷分布

- 模型后验概率：不能直接求解

$$p(\theta, \varphi, \boldsymbol{z}|\boldsymbol{w}, \alpha, \beta) = \frac{p(\boldsymbol{w}, \boldsymbol{z}, \theta, \varphi|\alpha, \beta)}{p(\boldsymbol{w}|\alpha, \beta)}$$

$$= \frac{\prod_{k=1}^{K} p(\varphi_k|\beta) \prod_{m=1}^{M} p(\theta_m|\alpha) \prod_{n=1}^{N_m} p(z_{mn}|\theta_m) p(w_{mn}|z_{mn}, \varphi)}{\prod_{k=1}^{K} \int p(\varphi_k|\beta) \left[\prod_{m=1}^{M} \int p(\theta_m|\alpha) \prod_{n=1}^{N_m} \left[\sum_{l=1}^{K} p(z_{mn}=l|\theta_m) p(w_{mn}|\varphi_l) \right] \mathrm{d}\theta_m \right] \mathrm{d}\varphi_k}$$

潜在狄利克雷分布

- 模型求解
  - 吉布斯抽样算法
    - 马尔科夫链蒙特卡罗法（MCMC）
    - 通过随机抽样的方法近似地计算模型的后验概率
  - 变分EM算法
    - 通过解析的方法计算模型的后验概率的**近似值**

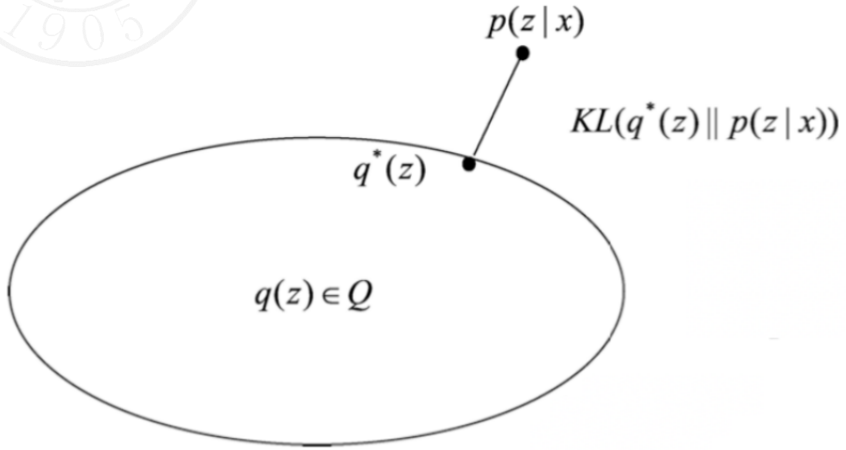
变分推断

- 假设模型是联合概率分布 $p(x, z)$
  - x是观测变量（数据）
  - z是隐变量，包括参数

- 目标是学习模型的后验概率分布$p(z|x)$

变分推断

- 基本思想
  - 用概率分布 $q(z)$ 近似条件概率分布 $p(z|x)$，用KL散度 $KL(q(z)\,||\,p(z|x))$ 计算两者的相似度
  - 如果能找到与 $p(z|x)$ 在KL散度意义下最近的分布$q^*(z)$，则可以用这个分布近似 $p(z|x)$



变分推断

- KL 散度

$$
\begin{aligned}
D(q(z)\|p(z|x)) &= E_q\left[\log q(z)\right] - E_q\left[\log p(z|x)\right] \\
&= E_q\left[\log q(z)\right] - E_q\left[\log p(x,z)\right] + \log p(x) \\
&= \log p(x) - \{E_q\left[\log p(x,z)\right] - E_q\left[\log q(z)\right]\}
\end{aligned}
$$

  - $D(q(z)||p(z|x)) \geq 0$恒成立

$$
\log p(x) \geqslant E_q\left[\log p(x,z)\right] - E_q\left[\log q(z)\right]
$$

  - 证据下界 ELBO

$$
L(q) = E_q\left[\log p(x,z)\right] - E_q\left[\log q(z)\right]
$$

变分推断

- 已知 $\log p(x) = L(q) + D(q(z)\|p(z|x))$
  - $\log p(x)$是常量
- KL散度的最小化：可以通过证据下界的最大化实现
- 因此，变分推断变成求解证据下界最大化的问题

变分推断

- 变分分布$q(z)$
  - 平均场（mean field）
    - 通常假设$q(z)$对z的所有分量都是互相独立的（实际是条件独立于参数），即满足

$$q(z) = q(z_1)q(z_2)\cdots q(z_n)$$

  - KL散度的最小化或证据下界最大化实际是在平均场的集合，即满足独立假设的分布集合

$$Q = \{q(z)|q(z) = \prod_{i=1}^{n} q(z_i)\}$$

进行的

变分推断

- 变分推断步骤：
  - 定义变分分布$q(z)$ $q(z) = q(z_1)q(z_2)\cdots q(z_n)$
  - 推导证据下界 $L(q) = E_q\left[\log p(x,z)\right] - E_q\left[\log q(z)\right]$
  - 解析或迭代优化，最大化证据下界
    得到最优分布$q^*(z)$，作为后验分布$p(z|x)$的近似

变分EM算法

- 变分EM算法
  - 变分推断中，通过迭代法最大化证据下界，这时算法是EM算法的推广，即变分EM算法
  - 假设模型是联合概率分布 $p(x, z|\theta)$
    - x是观测变量（数据）
    - z是隐变量
    - θ是参数
  - 最大化目标是观测数据的对数似然 $\log p(x|\theta)$——困难
    - 转而优化变分下界

$$L(q, \theta) = E_q[\log p(x, z|\theta)] - E_q[\log q(z)]$$

变分EM算法

算法 20.3（变分 EM 算法）

循环执行以下 E 步和 M 步，直到收敛。

（1）E 步：固定 θ，求 $L(q,\theta)$ 对 q 的最大化。

（2）M 步：固定 q，求 $L(q,\theta)$ 对 θ 的最大化。

给出模型参数 θ 的估计值。

根据变分推理原理，观测数据的概率和证据下界满足

$$\log p(x|\theta) - L(q,\theta) = D(q(z)\|p(z|x,\theta)) \geqslant 0$$

注： E步最大化q求是尽可能得逼近原始后验概率;$-E_q[\log q]$ 不含 θ，所以M步就退化为最大化 _q[\log $\mathbb{E}_q p(x,z|\theta)]$ —— 和标准 EM 的 M 步完全一样。

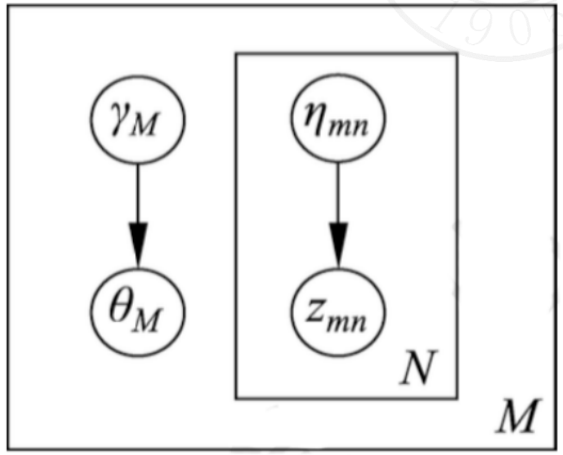
潜在狄利克雷分布

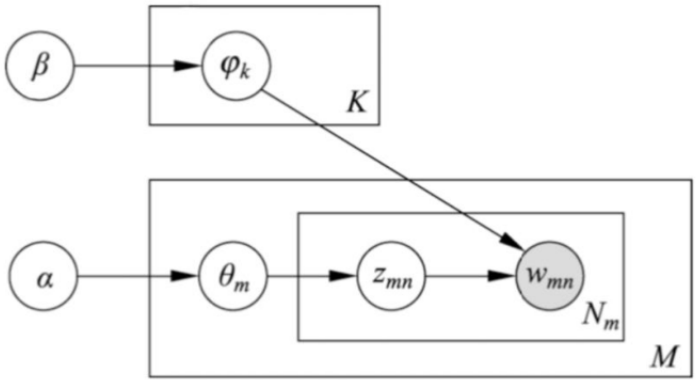
- LDA的变分EM算法
  - 定义变分分布$q(\theta, z|\gamma, \eta)$
    - 平均场假设，θ, z条件独立，从而

$$q(\boldsymbol{\theta}_m, \{z_{mn}\}) = q(\boldsymbol{\theta}_m; \boldsymbol{\gamma}_m) \prod_n q(z_{mn}; \boldsymbol{\eta}_{mn})$$

其中

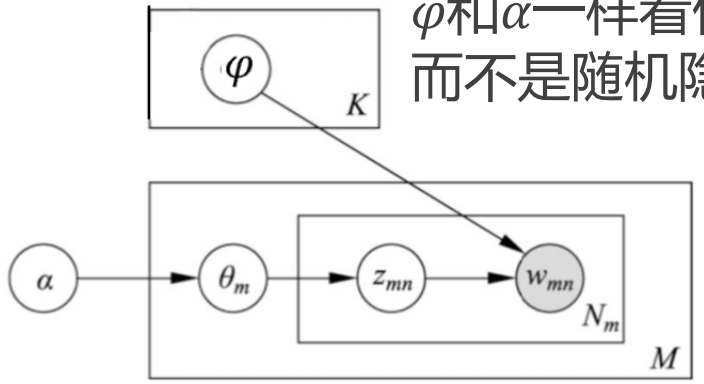
$$q(z_{mn} = k) = \eta_{mnk}, \quad \sum_{k=1}^{K} \eta_{mnk} = 1$$





简化模型

φ和α一样看作是模型参数，而不是随机隐变量



$$p(\boldsymbol{\theta}_m, \{z_{mn}\}, \{w_{mn}\}; \boldsymbol{\alpha}, \{\boldsymbol{\varphi}_k\}) = p(\boldsymbol{\theta}_m \mid \boldsymbol{\alpha}) \prod_n p(z_{mn} \mid \boldsymbol{\theta}_m)\, p(w_{mn} \mid z_{mn}, \boldsymbol{\varphi}_{z_{mn}})$$

潜在狄利克雷分布

- LDA的变分EM算法
  - 一个文本的证据下界

$$\begin{aligned}\mathrm{ELBO}_m =& \mathbb{E}_q\left[\log p(\boldsymbol{\theta}_m \mid \boldsymbol{\alpha})\right] + \sum_n \mathbb{E}_q\left[\log p(z_{mn} \mid \boldsymbol{\theta}_m)\right] + \sum_n \mathbb{E}_q\left[\log p(w_{mn} \mid z_{mn})\right] \\ & - \mathbb{E}_q\left[\log q(\boldsymbol{\theta}_m; \boldsymbol{\gamma}_m)\right] - \sum_n \mathbb{E}_q\left[\log q(z_{mn}; \boldsymbol{\eta}_{mn})\right]\end{aligned}$$

  - 所有文本的证据下界，对数相加

$$\mathrm{ELBO} = \sum_{m=1}^{D} \mathrm{ELBO}_m$$

潜在狄利克雷分布

- LDA的变分EM算法

根据前面的ELBO以及等式约束下的拉格朗日函数求导，便可得到各参数更新的公式

E 步（变分 E-step）

固定当前模型参数 $\boldsymbol{\alpha}, \{\boldsymbol{\varphi}_k\}$；针对每一篇文档单独优化文档变分参数 $\boldsymbol{\gamma}_m$ 和词级变分参数 η_{mnk}，最大化 ELBO_m。在文档内部循环迭代直到单文档收敛。

更新公式

1. 更新词级变分参数 η_{mnk}

$$\eta_{mnk} \propto \exp\left(\psi(\gamma_{mk}) + \log \varphi_k[w_{mn}]\right)$$

$\psi(\cdot)$ 代表 digamma 函数；$\varphi_k[w_{mn}]$：主题 k 生成词 w_{mn} 的概率，计算完成之后做归一化，保证 $\sum_k \eta_{mnk} = 1$。

2. 更新文档狄利克雷变分参数 γ_{mk}

$$\gamma_{mk} = \alpha_k + \sum_n \eta_{mnk}$$

潜在狄利克雷分布

- LDA的变分EM算法

M 步（Maximization step）

固定全部文档已经得到的变分分布，最大化整体 ELBO，更新模型参数 $\{\varphi_k\}$；如果需要同时估计超参数，一并优化 $\boldsymbol{\alpha}$。

1. 更新主题词参数 φ_k

统计期望计数：主题 k，词 w 在全部语料里面的期望频次

$n_{k,w} = \sum_m \sum_{\{n|w_{mn}=w\}} \eta_{mnk}$

归一化：

$\varphi_{k,w} \propto n_{k,w}$

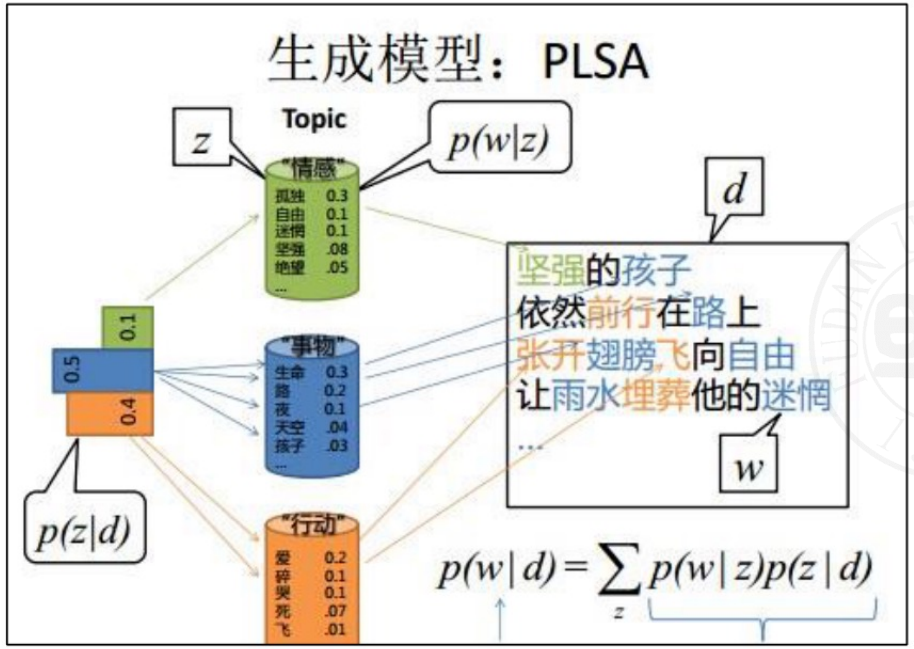
2. 更新 $\boldsymbol{\alpha}$（可选）

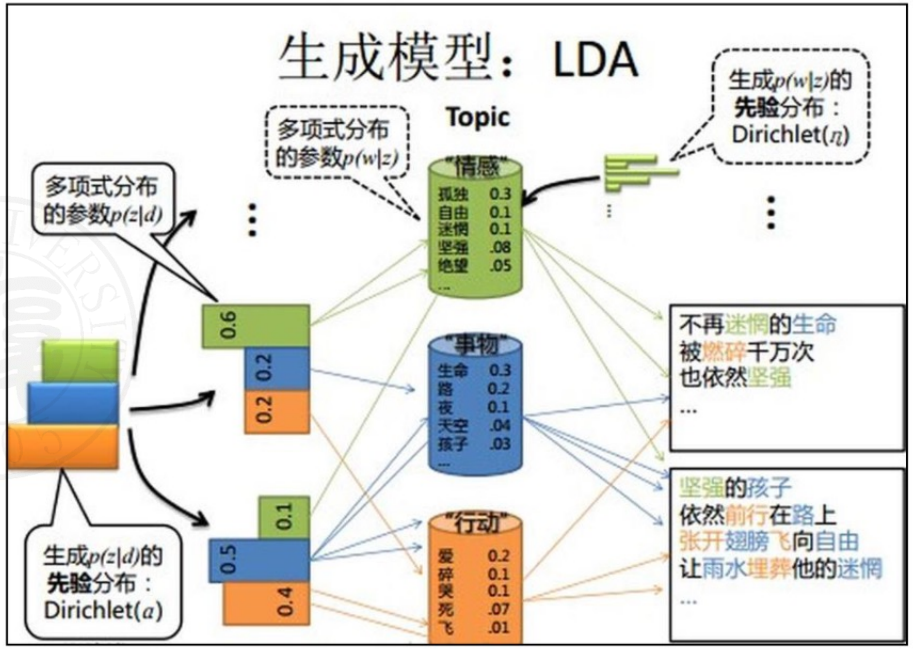
固定当前全套变分分布，以 ELBO 作为目标函数，采用牛顿法做数值优化更新 $\boldsymbol{\alpha}$；如果直接人工预设 $\boldsymbol{\alpha}$，这一步直接跳过。

LDA和pLSA的比较

- LDA和pLSA的比较
  - 可以认为LDA是pLSA的扩展
  - 相同点
    - 假设话题是单词的多项分布，文本是话题的多项分布
  - 不同点
    - LDA使用狄利克雷分布作为先验分布，pLSA不使用先验分布（或者说假设先验分布是均匀分布）
    - 学习过程LDA基于贝叶斯学习，而pLSA基于极大似然估计
  - LDA的优点
    - 具有生成性
    - 先验约束能有效防止过拟合

LDA和pLSA的比较





Thanks!

Questions?