Statistical Learning for Text Data Analytics
Bayesian Modeling

Yangqiu Song

Hong Kong University of Science and Technology

yqsong@cse.ust.hk

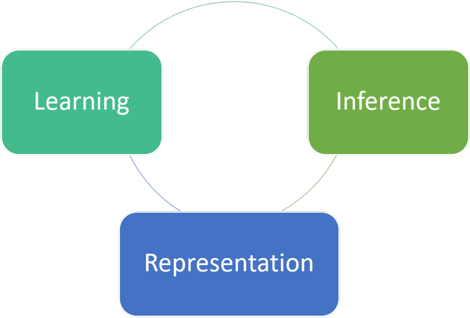
Spring 2018

∗Contents are based on materials created by Noah Smith, Xiaojin (Jerry) Zhu, Chengxiang Zhai, Mark Johnson

Reference Content

- Noah Smith. CSE 517: Natural Language Processing https://courses.cs.washington.edu/courses/cse517/16wi/
- Xiaojin (Jerry) Zhu. CS 769: Advanced Natural Language Processing. http://pages.cs.wisc.edu/~jerryzhu/cs769.html
- Chengxiang Zhai. CS598CXZ Advanced Topics in Information Retrieval. http://times.cs.uiuc.edu/course/598f16/
- Mark Johnson. MLSS "Summer School" Bayesian Inference for Dirichlet-Multinomials and Dirichlet Processes. http://web.science.mq.edu.au/~mjohnson/papers/Johnson11MLSS-talk-extras.pdf

Course Topics



- Representation: language models, word embeddings, topic models
- Learning: supervised learning, semi-supervised learning, sequence models, deep learning, optimization techniques
- Inference: constraint modeling, joint inference, search algorithms

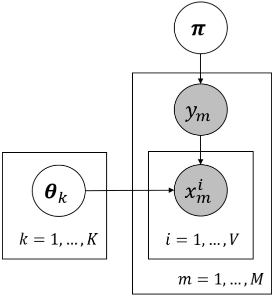
NLP applications: tasks introduced in Lecture 1

Overview

Overview

1. Language Models: Recap
2. Topic Models
3. Probabilistic Latent Semantic Analysis (PLSA)
4. Latent Dirichlet Allocation (LDA)
   - Motivation: Bayesian Modeling

Naive Bayes Classifier: A Generative View



Naive Bayes from Class Conditional Unigram Model

- For $m = 1, \ldots, M$
  - Choose $y_m \sim Multinomial(y_m|1, \boldsymbol{\pi})$
  - Choose $N_m = \sum_j^d x_m^j \sim Poisson(\xi)$
  - For $n = 1, \ldots, N_m$
    - Choose $v \sim Multinomial(v|1, \boldsymbol{\theta}_{*|y_m}) = \prod_{j=1}^d (\theta^j_{*|y_m})^{v=j}$

Both y_m and $\mathbf{x}_m = (x_m^1, \ldots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

Compare Naive Bayes and Mixture Model

In naive Bayes, both y_m and $\mathbf{x}_m = (x_m^1, \dots, x_m^d)^T$ are observed variables; $\boldsymbol{\pi}$ and $\boldsymbol{\theta}_k$ are parameters

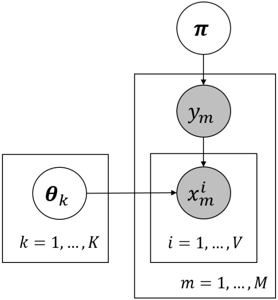


Figure: Native Bayes

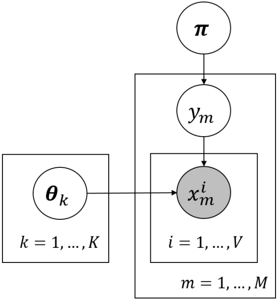
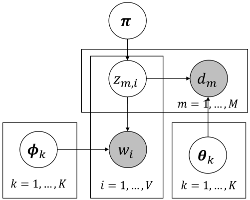


Figure: Mixture Model

However, in clustering problems, y_m is not observed (labeled before feeding into machine learning algorithm)

Probabilistic Latent Semantic Analysis (PLSA)

- PLSA assumes that each document d (with word vector w) is generated from all topics, with documentspecific topic weights.



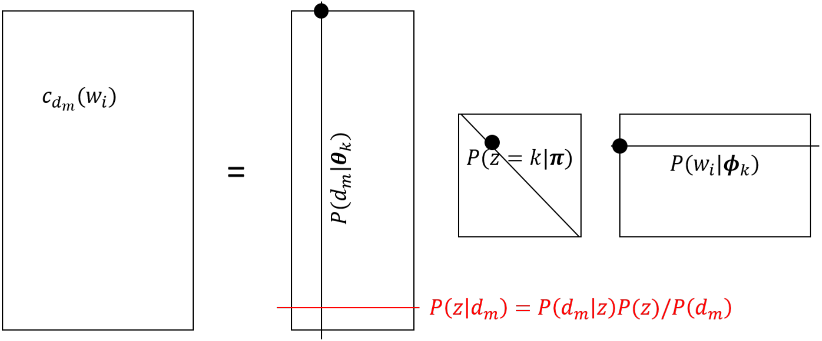
- Choose a $z_{m,i} = k$ from topic distribution $\boldsymbol{\pi}$
- Choose a document from $d_m \sim Multinomial(d_m|1, \boldsymbol{\theta}_k)$
- Choose a word w_i from $w_i \sim Multinomial(w_i|1, \phi_k)$
- Add one count of word w_i to document d_m
- Repeat until we generate the document-word matrix

Under this process, the probability of picking the corpus is:

$$
\begin{aligned}
P(\mathcal{D}, \mathcal{W}) &= \prod_{m=1}^{M} \prod_{i=1}^{N_m} \sum_{k=1}^{K} P(z_{m,i} = k) P(d_m|\boldsymbol{\theta}_k) P(w_i|\phi_k) \\
&= \prod_{m=1}^{M} \prod_{i=1}^{V} \left(\sum_{k=1}^{K} P(z_{m,i} = k) P(d_m|\boldsymbol{\theta}_k) P(w_i|\phi_k) \right)^{c_{d_m}(w_i)}
\end{aligned}
$$

A Matrix Factorization View

$$
\begin{aligned}
P(\mathcal{D}, \mathcal{W}) &= \prod_{m=1}^{M} \prod_{i=1}^{N_m} \sum_{k=1}^{K} P(z_{m,i} = k) P(d_m|\boldsymbol{\theta}_k) P(w_i|\phi_k) \\
&= \prod_{m=1}^{M} \prod_{i=1}^{V} \left(\sum_{k=1}^{K} P(z_{m,i} = k) P(d_m|\boldsymbol{\theta}_k) P(w_i|\phi_k) \right)^{c_{d_m}(w_i)}
\end{aligned}
$$



Maximize Log Likelihood

- Log likelihood:

$$P(\mathcal{D},\mathcal{W}) = \prod_{m=1}^{M} \prod_{i=1}^{V} \left(\sum_{k=1}^{K} P(z_{m,i} = k) P(d_m|\boldsymbol{\theta}_k) P(w_i|\phi_k) \right)^{c_{d_m}(w_i)}$$

- To reduce the notation complexity, we denote:

$$\log P(\mathcal{D},\mathcal{W}) = \sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \log \left(\sum_{k=1}^{K} P(z)P(d|z)P(w|z) \right)$$

- We denote the parameters as $\Theta = \{\boldsymbol{\pi}, \phi_k, \boldsymbol{\theta}_k, k = 1, \ldots, K\} = \{P(z), P(d|z), P(w|z)\}$
- Note here z is a hidden variable, and note that the sum is inside the log
- We can apply EM algorithm to maximize the likelihood

Lower Bound and E-Step

- Remember Jensens inequality

$$\log \sum_i P_i f_i(x) \geq \sum_i P_i \log f_i(x)$$

- We first compute the lower bound of the log likelihood:

$$\begin{aligned}
& \log P(\mathcal{D}, \mathcal{W}) \\
= & \textstyle\sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \log \left(\sum_{k=1}^{K} P(z)P(d|z)P(w|z) \right) \\
= & \textstyle\sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \log \left(\sum_{k=1}^{K} q_{z,d,w}(\Theta) \frac{P(z)P(d|z)P(w|z)}{q_{z,d,w}(\Theta)} \right) \\
\geq & \textstyle\sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \sum_{k=1}^{K} q_{z,d,w}(\Theta) \left(\log \frac{P(z)P(d|z)P(w|z)}{q_{z,d,w}(\Theta)} \right)
\end{aligned}$$

- Note Jensen's inequality involves computing $q_{z,d,w}(\Theta) = P(z|d, w, \Theta^t)$, which computes the probability of topics separately for each cell, under the current parameters Θ^t
- This is exactly the E-step:

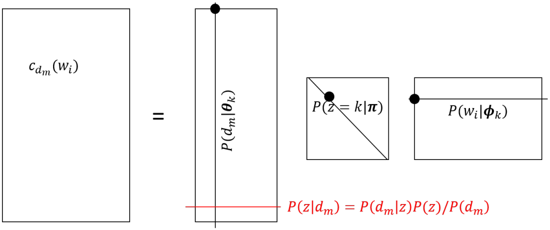
$$P(z|d, w, \Theta^t) \propto P(z|\Theta^t)P(d|z, \Theta^t)P(w|z, \Theta^t)$$

M-Step

$$
\begin{aligned}
& \log P(\mathcal{D}, \mathcal{W}) \\
= \; & \sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \log \left(\sum_{k=1}^{K} P(z) P(d|z) P(w|z) \right) \\
= \; & \sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \log \left(\sum_{k=1}^{K} P(z|d, w, \Theta^t) \frac{P(z)P(d|z)P(w|z)}{P(z|d,w,\Theta^t)} \right) \\
= \; & \sum_{d=1}^{M} \sum_{w=1}^{V} c_d(w) \sum_{k=1}^{K} P(z|d, w, \Theta^t) \left(\log \frac{P(z)P(d|z)P(w|z)}{P(z|d,w,\Theta^t)} \right)
\end{aligned}
$$

- Maximizing the right of the above inequality by setting the gradient to zero amounts to the M-step, which gives
  - $P(z) \propto \sum_d \sum_w c_d(w) P(z|d, w, \Theta^t)$
  - $P(d|z) \propto \sum_w c_d(w) P(z|d, w, \Theta^t)$
  - $P(w|z) \propto \sum_d c_d(w) P(z|d, w, \Theta^t)$

Use of PLSA



- Once the model is trained, we can look at it in the following way
  - $P(w|z)$ are the topics. Each topic is defined by a word multinomial. Often people find that the topics seem to have distinct semantic meanings.
  - From $P(d|z)$ and $P(z)$, we can compute $P(z|d) \propto p(d|z)p(z)$. $P(z|d)$ is the topic wights for document d.
- One drawback of PLSA is that it is transductive in nature. That is, there is no easy way to handle a new document that is not already in the collection
- This motivates us to introduce a Bayesian modeling of topic models

Overview

Recall Unigram Language Modeling

- Data corpus: a collection of words, $\mathcal{W} = \{w_1, w_2, \ldots, w_N\}$
- Model: multinomial distribution $P(\mathcal{W}|\boldsymbol{\theta})$ with parameters $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_V)$, where
  - $\theta_i = P(v_i)$
  - $v_i \in \mathcal{V}$
  - $\mathcal{V}$ is the vocabulary
  - $|\mathcal{V}| = V$
- Count of words in corpus $\mathbf{u} = (u_1, \ldots, u_V)$ where $u_i = c(v_i)$ is the count of v_i shown in $\mathcal{W}$, $\sum_i u_i = N$

Unigram Modeling

- "Bag of words" assumes the words are sampled from a multinomial distribution $\mathbf{u} \sim \mathrm{Multi}(\boldsymbol{\theta})$

$$P(\mathbf{u}|\boldsymbol{\theta}) = \left(\begin{array}{c} N \\ \mathbf{u} \end{array} \right) \prod_{i=1}^{V} \theta_i^{u_i} \triangleq \mathrm{Mult}(\mathbf{u}|\boldsymbol{\theta}, N), \textit{where} \left(\begin{array}{c} N \\ \mathbf{u} \end{array} \right) = \frac{N!}{\prod_i u_i!}$$

If we focus on a single trial, we have:

$$P(w|\boldsymbol{\theta}) = P(w = v_i) = \prod_{i=1}^{V} \theta_i^{\delta_{w=v_i}} \triangleq \mathrm{Mult}(w|\boldsymbol{\theta})$$

- Maximum likelihood estimator: $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{W}|\boldsymbol{\theta})$

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_{j=1}^{N} P(w_j|\boldsymbol{\theta}) = \prod_{i=1}^{V} P(v_i)^{u_i} = \prod_{i=1}^{V} \theta^{u_i}$$

Maximum Likelihood Estimation: $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{W}|\boldsymbol{\theta})$

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_i^V \theta_i^{u_i}$$

(log likelihood)

$$\Rightarrow \log P(\mathcal{W}|\boldsymbol{\theta}) = \sum_i^V u_i \log \theta_i$$

(Lagrange multiplier to make $\boldsymbol{\theta}$ be a distribution)

$$\Rightarrow L(\mathcal{W}, \boldsymbol{\theta}) = \log P(\mathcal{W}|\boldsymbol{\theta}) = \sum_i^V u_i \log \theta_i + \lambda(\sum_i \theta_i - 1)$$

(Set partial derivatives to zero)

$$\Rightarrow \frac{\partial L}{\partial \theta_i} = \frac{u_i}{\theta_i} + \lambda$$

Since $\sum_i^V \theta_i = 1$, we have $\lambda = -\sum_i^V u_i$

$$\Rightarrow \theta_i = \frac{u_i}{\sum_i^V u_i} = \frac{u_i}{N} \; (Maximum\ Likelihood\ Estimation\ , MLE)$$

Generalization: Add-K smoothing

Problem: Add-one moves too much probability mass from seen to unseen events!

- Variant of Add-One smoothing
  - Add a constant k to the counts of each word
  - For any $k > 0$ (typically, $k < 1$), a unigram model is

$$\Rightarrow \theta_i = \frac{u_i + k}{\sum_i^V u_i + kV} = \frac{u_i + k}{N + kV}$$

- If $k = 1$
  - "Add one" Laplace smoothing
- This is still too simplistic to work well.

Any explanation?

Bayesian Interpretation

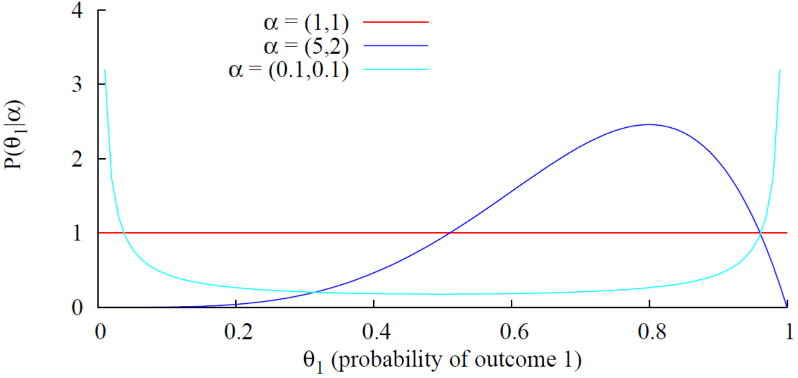
- Conjugate distribution
  - Adding a conjugate prior to a likelihood will result in a posterior in the same distribution family as the prior, then the prior and the likelihood are called conjugate distributions
  - Conjugate distribution makes us easier to formulate Bayesian belief and inference the model

Bayesian Interpretation

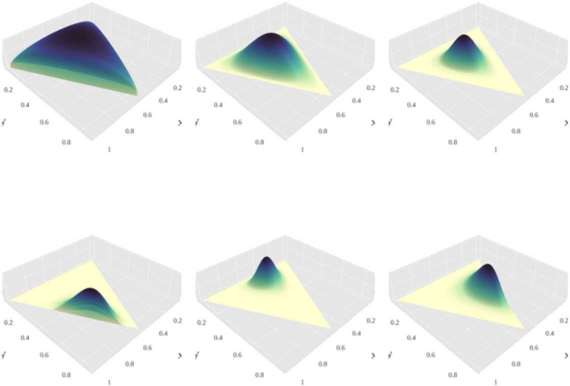
- The conjugate prior of a multinomial is Dirichlet distribution:
  $P(\boldsymbol{\theta}|\boldsymbol{\alpha}) = \mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}) \triangleq \frac{\Gamma(\sum_{i=1}^{V}\alpha_i)}{\prod_{i=1}^{V}\Gamma(\alpha_i)}\prod_{i=1}^{V}\theta_i^{\alpha_i-1} \triangleq \frac{1}{\Delta(\boldsymbol{\alpha})}\prod_{i=1}^{V}\theta_i^{\alpha_i-1}$
  - The "Dirichlet Delta function" $\Delta(\boldsymbol{\alpha})$ is introduced for convenience
  - $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_V)^\top \in \mathbb{R}^V$
  - The Gamma function satisfies $\Gamma(x+1) = x\Gamma(x)$
    - For integer variable, Gamma function is $\Gamma(x) = (x-1)!$
    - For real numbers, it is $\Gamma(x) = \int_0^\infty t^{x-1}e^{-t}\mathrm{d}t$
- The Dirichlet distribution can be seen as the *"distribution of a distribution"*
  - We can sample a multinomial distribution from Dirichlet distribution, satisfied the constraint $\sum_i \theta_i = 1$

Called Beta distribution when there are two choices of variable values

$$P(\boldsymbol{\theta} \mid \boldsymbol{\alpha}) = \frac{\Gamma(\sum_{j=1}^{m} \alpha_j)}{\prod_{j=1}^{m} \Gamma(\alpha_j)} \prod_{k=1}^{m} \theta_k^{\alpha_k - 1}$$



- The Dirichlet distribution can be seen as the *"distribution of a distribution"*
  - We can sample a multinomial distribution from Dirichlet distribution, satisfied the constraint $\sum_i \theta_i = 1$



Bayesian Estimation

- Remember Maximum likelihood estimator: $\hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} P(\mathcal{W}|\boldsymbol{\theta})$

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_{j=1}^{N} P(w_j|\boldsymbol{\theta}) = \prod_{i=1}^{V} P(v_i)^{u_i} = \prod_{i=1}^{V} \theta^{u_i} (\theta_i = \frac{u_i}{\sum_i^V u_i} = \frac{u_i}{N})$$

- The posterior of the parameters $\boldsymbol{\theta}$ based on the prior and the observation of N words:

$$\begin{aligned} P(\boldsymbol{\theta}|\mathcal{W},\boldsymbol{\alpha}) &= \frac{P(\mathcal{W}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\mathcal{W}|\boldsymbol{\alpha})} \\ &= \frac{\prod_{i=1}^{N} P(w_i|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})}{\int_{\boldsymbol{\theta}} \prod_{i=1}^{N} P(w_i|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha}) \mathrm{d}\boldsymbol{\theta}} \\ &= \frac{\prod_{i=1}^{N} P(w_i|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})}{Z} \\ &= \frac{1}{Z} \prod_{i=1}^{V} \theta_i^{u_i} \frac{1}{\Delta(\boldsymbol{\alpha})} \prod_{i=1}^{V} \theta_i^{\alpha_i - 1} \\ &= \frac{1}{\Delta(\boldsymbol{\alpha}+\mathbf{u})} \prod_{i=1}^{V} \theta_i^{\alpha_i + u_i - 1} = \mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha}+\mathbf{u}) \end{aligned}$$

- We have MAP (maximum a posterior estimation) estimate as $\theta_i = \frac{u_i + \alpha_i - 1}{\sum_i^V u_i + V(\alpha_i - 1)}$ ($\alpha_i = 1$ equals to MLE)

Graphical Representation of Two Versions

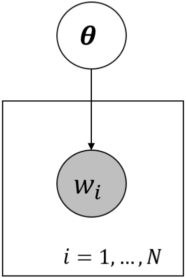


Figure: Unigram Language Model

$$P(\mathcal{W}|\boldsymbol{\theta}) = \prod_{j=1}^{N} P(w_j|\boldsymbol{\theta})$$

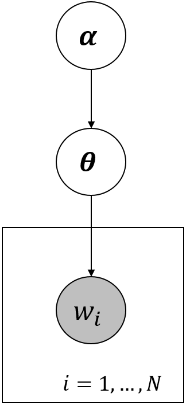
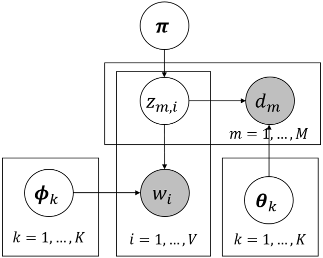
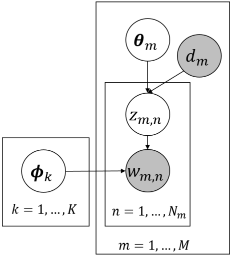


Figure: Bayesian Esitmation

$$P(\boldsymbol{\theta}|\mathcal{W}, \boldsymbol{\alpha}) = \frac{P(\mathcal{W}|\boldsymbol{\theta})P(\boldsymbol{\theta}|\boldsymbol{\alpha})}{P(\mathcal{W}|\boldsymbol{\alpha})}$$

$$
\begin{aligned}
P(\mathcal{D},\mathcal{W}) &= \prod_{m=1}^{M}\prod_{i=1}^{N_m}\sum_{k=1}^{K} P(z_{m,i}=k)P(d_m|\boldsymbol{\theta}_k)P(w_i|\phi_k) \\
&= \prod_{m=1}^{M}\prod_{i=1}^{V}\left(\sum_{k=1}^{K} P(z_{m,i}=k)P(d_m|\boldsymbol{\theta}_k)P(w_i|\phi_k)\right)^{c_{d_m}(w_i)}
\end{aligned}
$$

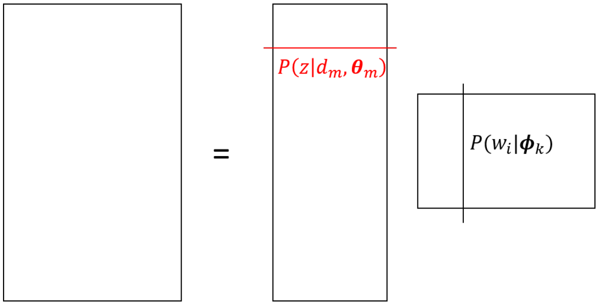




$$
P(\mathcal{D},\mathcal{W}) = \prod_{m=1}^{M}\prod_{i=1}^{V} P(d_m)\left(\sum_{k=1}^{K} P(z_{m,i}=k|\boldsymbol{\theta}_m)P(w_i|\phi_k)\right)^{c_{d_m}(w_i)}
$$

Comparison of Mixture Models and PLSA

$$P(\mathcal{D},\mathcal{W}) = \prod_{m=1}^{M}\prod_{i=1}^{V} P(d_m)\left(\sum_{k=1}^{K} P(z_{m,i}=k|\boldsymbol{\theta}_m)P(w_i|\phi_k)\right)^{c_{d_m}(w_i)}$$



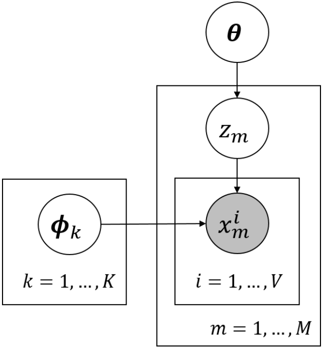


Figure: Mixture Models (with notation change)

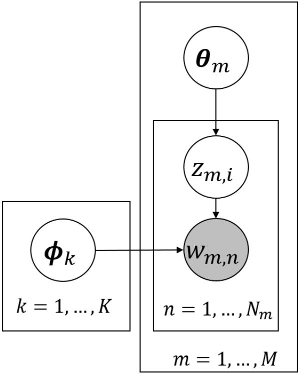


Figure: PLSA

Bayesian Modeling: Language Models

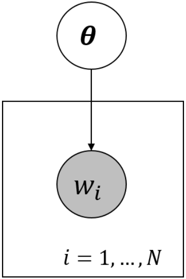


Figure: Unigram Language Model

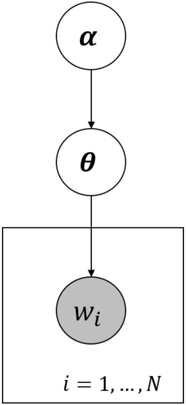


Figure: Bayesian Esitmation

Bayesian Modeling: Mixture Models

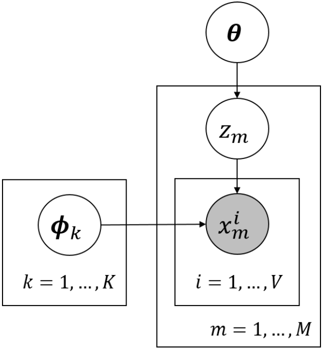


Figure: Unigram Language Model

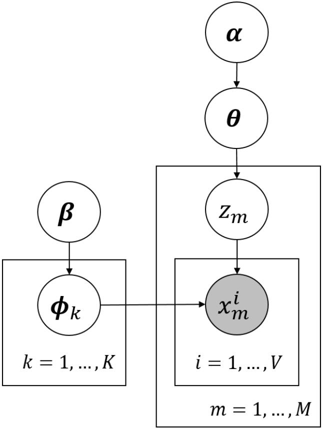


Figure: Bayesian Esitmation

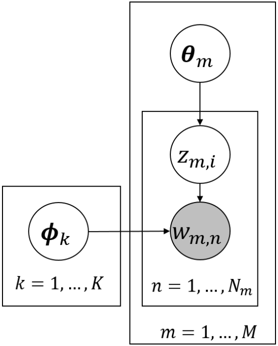


Figure: PLSA

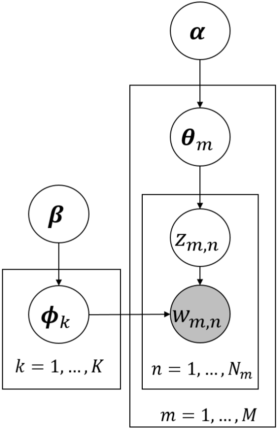


Figure: LDA

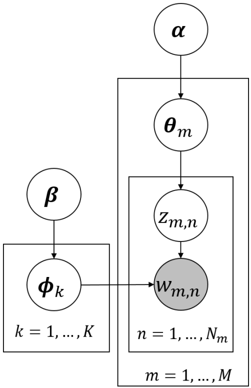


Figure: LDA

- For all clusters/components $k \in [1, K]$:
    - Choose mixture components $\phi_k \sim \mathrm{Dir}(\phi|\beta)$
- For all documents $m \in [1, M]$:
    - Choose $N_m \sim \mathrm{Poisson}(\xi)$
    - Choose mixture probability $\boldsymbol{\theta}_m \sim \mathrm{Dir}(\boldsymbol{\theta}|\boldsymbol{\alpha})$
    - For all words $n \in [1, N_m]$ in document d_m:
        - Choose a component index $z_{m,n} \sim \mathrm{Mult}(z|\boldsymbol{\theta}_m)$
        - Choose a word $w_{m,n} \sim \mathrm{Mult}(w|\phi_{z_{m,n}})$

Bayes' rule

$$P(\text{Hypothesis}|\text{Data}) = \frac{P(\text{Data}|\text{Hypothesis})P(\text{Hypothesis})}{P(\text{Data})}$$

- Bayesian's use Bayes' Rule to update beliefs in hypotheses in response to data
- $P(\text{Hypothesis}|\text{Data})$ is the posterior distribution
- $P(\text{Hypothesis})$ is the prior distribution
- $P(\text{Data}|\text{Hypothesis})$ is the likelihood, and
- $P(\text{Data})$ is a normalizing constant sometimes called the evidence

Computing the Normalizing Constant

$$P(\text{Data}) = \sum_{\text{Hypothesis}' \in \mathcal{H}} P(\text{Data}|\text{Hypothesis})P(\text{Hypothesis})$$

- If set of hypotheses $\mathcal{H}$ is small, can calculate $P(\text{Data})$ by enumeration
- But often these sums are intractable

"Being Bayesian"

- A summary of the Bayesian philosophy in NLP:
  - Because we have finite data, we should be uncertain about every estimated model parameter
  - Bayes' rule gives us a way to manage that uncertainty, if we can define a prior distribution over model parameters
  - Inference is a "simple matter" of estimating posterior distributions
    - But exact inference is almost never tractable, so we need approximations
    - There are many of these, and they tend to be expensive
    - Some of them look like EM, some don't

References I