Latent Dirichlet Allocation

Nikhilkumar Jadhav

Outline

- Parameter Estimation
- Conjugate Distributions
- Modeling Text
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- Gibbs Sampling
- LDA Gibbs Samper

Outline

- **Parameter Estimation**
- Conjugate Distributions
- Modeling Text
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- Gibbs Sampling
- LDA Gibbs Samper

Outline

- **Parameter Estimation**
  - Maximum likelihood estimation (ML)
  - Maximum a posteriori estimation (MAP)
  - Bayesian estimation

Parameter Estimation

- Problem
  - To estimate values for a set of model parameters that can best explain a set of observations
  - Let ϑ be the set of parameters to be estimated and X be set of observations
  - $p(\vartheta \mid X)$
- Bayes' rule
  - $p(\vartheta \mid X) = \frac{p(X|\vartheta)p(\vartheta)}{p(X)}$

Parameter Estimation

- Different view of Bayes' rule
  - $posterior = \frac{likelihood.prior}{evidence}$

Maximum Likelihood estimation (ML)

- Tries to find parameters that maximize the likelihood

$$L(\vartheta|X) = p(X|\vartheta) = \bigcap_{x\in X} p(X = x|\vartheta) = \prod_{x\in X} p(x|\vartheta)$$

- Taking log,

$$\hat{\vartheta}_{ML} = \operatorname*{argmax}_{\vartheta} LL(\vartheta|X) = \operatorname*{argmax}_{\vartheta} \sum_{x\in X} logp(x|\vartheta)$$

Maximum a posteriori estimation

- Includes some prior belief,

$$\hat{\vartheta}_{MAP} = \underset{\vartheta}{\operatorname{argmax}} \frac{p(X|\vartheta)p(\vartheta)}{p(X)} \quad | \quad p(X) \neq f(\vartheta)$$

$$= \underset{\vartheta}{\operatorname{argmax}}\ p(X|\vartheta)p(\vartheta)$$

$$= \underset{\vartheta}{\operatorname{argmax}} \left\{ \sum_{x \in X} \log p(x|\vartheta) + logp(\vartheta) \right\}$$

Bayesian Estimation

- Allows a distribution over the parameter set ϑ instead of making a direct estimate
- Calculation of posterior according to Bayes' rule

Outline

- Parameter Estimation
- **Conjugate Distributions**
- Modeling Text
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- Gibbs Sampling
- LDA Gibbs Samper

Outline

- **Conjugate Distributions**
  - Conjugacy
  - Coin Tossing
    - Bernoulli likelihood
    - Beta Distribution
  - Multivariate case
    - Multinomial likelihood
    - Dirichlet distribution
  - Modeling text

Conjugate Distributions

- Calculation of Bayesian models often becomes quite difficult
  - Summation or integrals of marginal likelihood (evidence) are intractable or there are unknown variables
- Advantage of Bayesian estimation
  - Freedom while encoding prior belief
- Conjugate prior distributions are used to facilitate model inference

Conjugacy

- A conjugate prior, $p(\vartheta)$ of a likelihood, $p(X|\vartheta)$ is a distribution that results in a posterior, $p(\vartheta|X)$ of the same form

Coin Tossing

- Consider a set C of N Bernoulli experiments with unknown parameter p.
- **Bernoulli density function** (likelihood) for the r.v. C for one experiment is,
- $p(C = c|p) = \; p^c(1-p)^{1-c} \triangleq Bern(c|p)$
- for N Bernoulli experiments,

$$= \; p(C = 1|p)^{n^{(1)}} p(C = 0|p)^{n^{(0)}}$$

$$= \; p^{n^{(1)}}(1-p)^{n^{(0)}}$$

where, c=1 means heads and c=0 means tails

- We want to encode a prior belief about p

Beta Distribution

- $p(p|\alpha,\beta) = \frac{1}{B(\alpha,\beta)} p^{\alpha-1}(1-p)^{\beta-1} \triangleq Beta(p|\alpha,\beta)$

  with the beta function,

  – $B(\alpha,\beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)} = \int_0^1 x^{\alpha-1} x^{\beta-1}$

- The beta distribution supports the interval [0,1] and is therefore used to generate normalized probability values
- It is used as a prior for the parameter p in the coin tossing experiment

Beta-Bernoulli case

- $p(C|\alpha,\beta) = \int_0^1 p(C|p)p(p|\alpha,\beta)dp$

$$= \int_0^1 p^{n^{(1)}}(1-p)^{n^{(0)}}\frac{1}{B(\alpha,\beta)}p^{\alpha-1}(1-p)^{\beta-1}$$

$$= \frac{1}{B(\alpha,\beta)}\int_0^1 p^{n^{(1)+\alpha-1}}(1-p)^{n^{(0)}+\beta-1}$$

$$= \frac{B(n^{(1)}+\alpha,n^{(0)}+\beta)}{B(\alpha,\beta)}$$

Multivariate case

- Generalizing the no. of possible events from 2 to a finite integer K, we obtain K-dimensional Bernoulli or multinomial experiment.

$$p(\vec{n}|\vec{p},N) = \binom{N}{\vec{n}} \prod_{k=1}^{K} {p_k}^{n^{(k)}} \triangleq Mult(\vec{n}|\vec{p},N)$$

where, $\vec{n}$ is the multinomial count vector

with $\binom{N}{\vec{n}} = \frac{N!}{\prod_k n^{(k)}!}, \sum_k p_k = 1 \ , \sum_k n^{(k)} = N$

- Multinomial coefficient counts the no. of configurations of individual trials that lead to the total, N

Multivariate case

- Single multinomial trial generalizes the Bernoulli distribution to a discrete categorical distribution

$$p(\vec{n}|\vec{p}) = \prod_{k=1}^{K} {p_k}^{n^{(k)}} = Mult(\vec{n}|\vec{p}, 1)$$

- Where, the count vector, $\vec{n}$ is zero except for a single element $n^{(z)} = 1$

$$p(z|\vec{p}) = p_z \triangleq Mult(z|\vec{p})$$

Multinomial Likelihood

- Introducing the multinomial r.v. C, the likelihood of N repetitions of a multinomial experiment becomes,

$$p(C|\vec{p}) = Mult(C = z_i|\vec{p}) = \prod_{i=1}^{N} p_{z_i} = \prod_{k=1}^{K} {p_k}^{n^{(k)}}$$

- This is the same as Multinomial distribution without the multinomial coefficient
- Here, we consider sequence of outcomes of N experiments instead of getting probability of particular multinomial count vector $\vec{n}$, which could be generated by $\binom{N}{\vec{n}}$ sequences.

Dirichlet distribution

- For parameters $\vec{p}$, the conjugate prior is the Dirichlet distribution
- Dirichlet distribution generalizes the beta distribution form 2 to K dimensions
- $p(\vec{p}|\vec{\alpha}) = Dir(\vec{p}|\vec{\alpha}) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_{k=1}^{K} p_k{}^{\alpha_k - 1}$

  $= \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{K} p_k{}^{\alpha_k - 1}$

- where, $\Delta(\vec{\alpha}) = \frac{\Gamma(\sum_{k=1}^{\dim(\vec{\alpha})} \alpha_k)}{\prod_{k=1}^{\dim(\vec{\alpha})} \Gamma(\alpha_k)} = \int_{\sum x_i = 1} \prod_i^N x_i{}^{\alpha_i - 1} d^N x$

Outline

- Parameter Estimation
- Conjugate Distributions
- **Modeling Text**
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- Gibbs Sampling
- LDA Gibbs Samper

Modelling Text

- Consider a set $\mathcal{W}$ of $\boldsymbol{N}$ i.i.d draws from a multinomial random variable $\boldsymbol{W}$. This can be imagined as drawing $\boldsymbol{N}$ words $\boldsymbol{w}$ from a vocabulary $\boldsymbol{V}$ of size $|\boldsymbol{V}|$

- The likelihood of the sample is given by,

$$L(\vec{p}|\vec{w}) = p(\mathcal{W}|\vec{p}) = \prod_{t=1}^{V} {p_t}^{n^{(t)}},$$

$$\sum_{t=1}^{V} n^{(t)} = 1, \qquad \sum_{t=1}^{V} p_t = 1$$

where, $n^{(t)}$ is the number of times term t is observed as a word in the document.

Modelling Text

- Assuming Cojugacy, the parameter vector $\vec{p}$ can be modelled with a Dirichlet distribution, $\vec{p} \sim Dir(\vec{p}|\vec{\alpha})$
- $$p(\vec{p}|\mathcal{W},\vec{\alpha}) = \frac{\prod_{n=1}^{N} p(w_n|\vec{p})p(\vec{p}|\vec{\alpha})}{\int_P \prod_{n=1}^{N} p(w_n|\vec{p})p(\vec{p}|\vec{\alpha})}$$
$$= \frac{\prod_{t=1}^{V} p(w{=}t|\vec{p})p(\vec{p}|\vec{\alpha})}{\int_P \prod_{t=1}^{V} p(w{=}t|\vec{p})p(\vec{p}|\vec{\alpha})}$$
$$= \frac{1}{Z}\prod_{t=1}^{V} p^{n^{(t)}} \frac{1}{\Delta(\vec{\alpha})} p^{\alpha_t - 1}$$
$$= \frac{\Delta(\vec{\alpha})}{\Delta(\vec{\alpha}+\vec{n})}\prod_{t=1}^{V} \frac{1}{\Delta(\vec{\alpha})} p^{\alpha_t + n^{(t)} - 1}$$
$$= \frac{1}{\Delta(\vec{\alpha}+\vec{n})}\prod_{t=1}^{V} p^{\alpha_t + n^{(t)} - 1} = Dir(\vec{p}|\vec{\alpha}+\vec{n})$$

Outline

- Parameter Estimation
- Conjugate Distributions
- Modeling Text
- **Bayesian networks and generative models**
- Latent Dirichlet Allocation
- Gibbs Sampling
- LDA Gibbs Samper

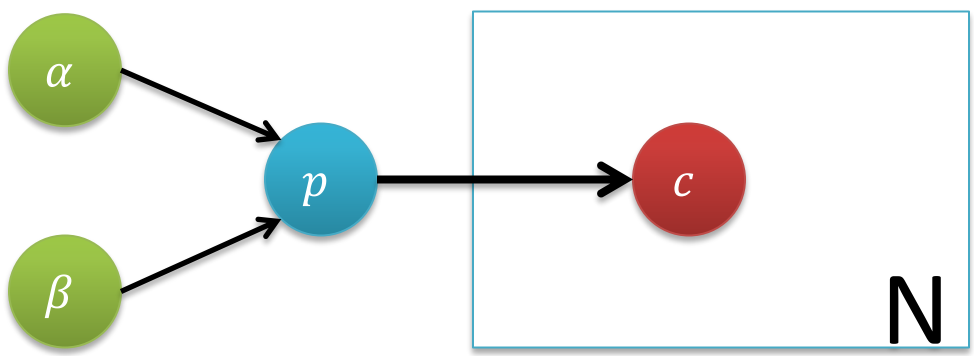
Bayesian networks

- Graphical network used to express
  - joint distribution of random variables
  - their conditional dependencies

  as a directed graph
- It forms Directed Acyclic Graph (DAG)
  - Nodes are random variables
  - Edges are conditional probability distributions
- Evidence nodes are those which are observed
- Hidden nodes correspond to latent variables
- Replication of nodes can be denoted by plates with a replication count in the lower right corner

Coin Experiment example

- Bayesian network for the coin tossing experiment with beta-distribution as a prior

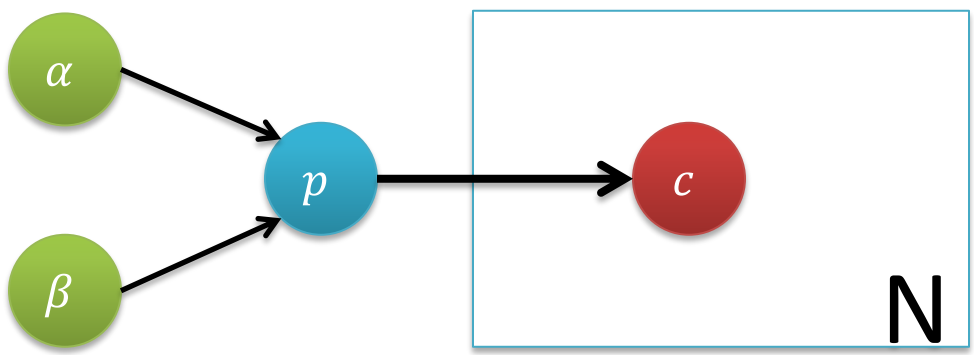


Generative Models

- State how observations could have been generated
- Bayesian networks provide an intuitive description of an observed phenomenon as a generative model
- Observations are generated by realizations of r.v.s and their propagation along directed edges of the network
- Bayesian inference
  - Inverts the generative models
  - Estimate the parameter values
  - Coping with hidden variables

Coin Experiment example

- Generative model for the coin tossing experiment with beta-distribution as a prior



- $p(c) = p(C = c|p)p(p|\alpha, \beta)$
- $p(C) = p^{n^{(1)}}(1-p)^{n^{(0)}}p(p|\alpha, \beta)$

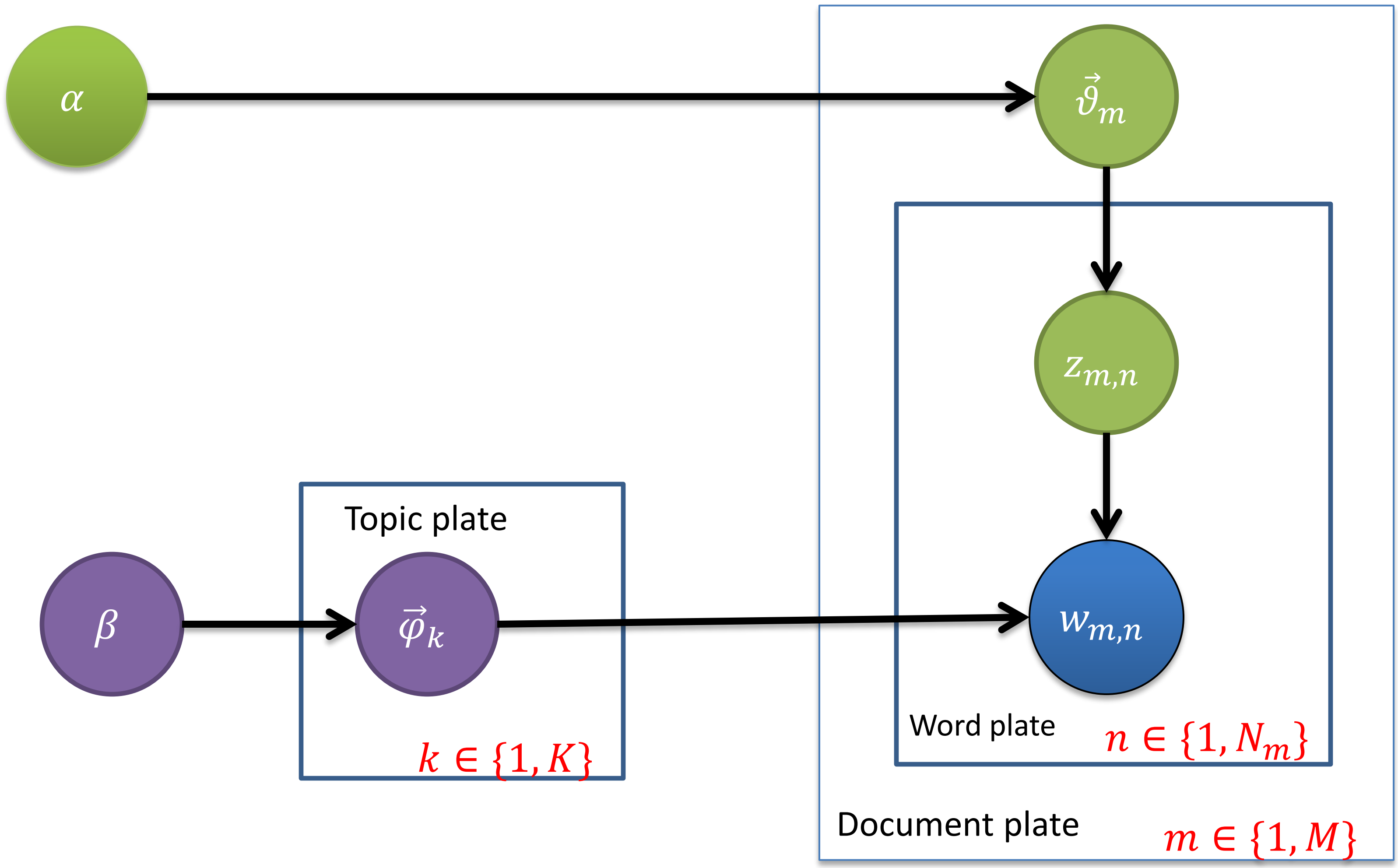
Outline

- Parameter Estimation
- Conjugate Distributions
- Modeling Text
- Bayesian networks and generative models
- **Latent Dirichlet Allocation**
- Gibbs Sampling
- LDA Gibbs Samper

Latent Dirichlet Allocation

- Probabilistic generative model
- Unsupervised topic modeling
- Used to perform **Latent Semantic Analysis (LSA)**
  - To find latent structure of **topics** or **concepts**
  - Co-occurrence structure of terms is used to recover this structure

Generative model of LDA



Likelihoods

- Probability that a particular word $w_{m,n}$ instantiates a particular term t given the LDA parameters is,
- $p\left(w_{m,n} = t \middle| \vec{\vartheta}_m, \underline{\phi}\right) = \sum_{k=1}^{K} p\left(w_{m,n} = t \middle| \vec{\varphi}_k\right) p(z_{m,n} = k|\vec{\vartheta}_m)$
- This corresponds to one iteration on the word plate of the Bayesian network
- Joint distribution of all know and hidden variables given the hyperparameters,
- $p\left(\overrightarrow{w}_m, \vec{z}_m, \vec{\vartheta}_m, \underline{\phi} \middle| \overrightarrow{\alpha}, \vec{\beta}\right) = \prod_{n=1}^{N_m} \overbrace{\underbrace{p\left(w_{m,n} \middle| \vec{\varphi}_{z_{m,n}}\right) . p\left(z_{m,n} \middle| \vec{\vartheta}_m\right)}_{\text{Word plate}} . p\left(\vec{\vartheta}_m \middle| \vec{\alpha}\right)}^{\text{Document plate (1 document)}} . \underbrace{p(\underline{\phi}|\vec{\beta})}_{\text{Topic plate}}$

Likelihoods

- Likelihood of the document,

$$p(\vec{w}_m|\vec{\alpha}|\vec{\beta}) = \iint p(\vec{\vartheta}_m|\vec{\alpha}).\, p\left(\underline{\phi}\middle|\vec{\beta}\right).\prod_{n=1}^{N_m}\sum_{z_{m,n}} p\left(w_{m,n}\middle|\vec{\varphi}_{z_{m,n}}\right) p(z_{m,n}|\vec{\vartheta}_m)\, d\underline{\phi}\, d\vec{\vartheta}_m$$

$$= \iint p(\vec{\vartheta}_m|\vec{\alpha}).\, p\left(\underline{\phi}\middle|\vec{\beta}\right).\prod_{n=1}^{N_m} p(w_{m,n}|\vec{\vartheta}_m, \underline{\phi})\, d\underline{\phi}\, d\vec{\vartheta}_m$$

- Likelihood of the corpus, $\mathcal{W} = \{\vec{w}_m\}_{m=1\ldots M}$,

$$p\left(\mathcal{W}\middle|\vec{\alpha},\vec{\beta}\right) = \prod_{m=1}^{M} p(\vec{w}_m|\vec{\alpha},\vec{\beta})$$

Outline

- Parameter Estimation
- Conjugate Distributions
- Modeling Text
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- **Gibbs Sampling**
- LDA Gibbs Samper

Gibbs Sampling

- Makov-chain Monte Carlo (MCMC) can emulate high-dimensional probability distributions $p(\vec{x})$ by the stationary distribution of a Markov chain
- Each sample is generated for each transition in the chain
  - After a stationary state of the chain has been reached
  - This happens after so-called “burn-in period” which eliminates the effect of initialization parameters
- Gibbs sampling is a special case of MCMC where
  - The dimensions x_i of the distribution are sampled alternately one at a time, conditioned on the values of all other dimensions, denoted by $\vec{x}_{\neg i}$

Bivariate Case

- Consider a bivariate random variable (x, y), and suppose we wish to compute the marginals, $p(x)$ and $p(y)$
- The idea behind the sampler
  - Easier to consider a sequence of distributions, $p(x|y)$ and $p(y|x)$
  - Than obtaining marginal by integration, $p(x) = \int p(x, y)dy$
- Steps
  - Start with some initial value y_0 for y
  - Obtain x_0 by generating a random variable from the conditional distribution, $p(x|y = y_0)$
  - Use x_0 to generate a new value of y_1, drawing from the conditional distribution, $p(y|x = x_0)$

Bivariate Case

- The sampler proceeds as follows:
  - $x_i \sim p(x|y = y_{i-1})$
  - $y_i \sim p(y|x = x_i)$
- Repeating this process k times, generates a Gibbs sequence of length k, where
  - a subset of points $\left(x_{j,}y_j\right)\, for\ 1 \leq j \leq m < k$ are taken as the simulated draws from the full joint distribution

Multivariate Case

- The value of the k^{th} variable is drawn from the distribution, $p(\theta^{(k)}|\boldsymbol{\theta}^{(\neg k)})$ where $\boldsymbol{\theta}^{(\neg k)}$ denotes a vector containing all of the variable but k
- We draw from the distribution,

$$\theta_i{}^{(k)} \sim p\left(\theta^{(k)}\middle|\theta^{(1)} = \theta_i{}^{(1)}, \dots, \theta^{(k-1)} = \theta_i{}^{(k-1)}, \theta^{(k+1)} = \theta_{i-1}{}^{(k+1)}\right), \dots, \theta^{(n)} = \theta_{i-1}{}^{(n)}$$

- For example, if there are four variables, (w, x, y, z) the sampler becomes,
  - $w_i \sim p(w \mid x = x_{i-1}, y = y_{i-1}, z = z_{i-1})$
  - $x_i \sim p(w \mid w = w_i, y = y_{i-1}, z = z_{i-1})$
  - $y_i \sim p(w \mid w = w_i, x = x_i, z = z_{i-1})$
  - $z_i \sim p(w \mid w = w_i, x = x_i, y = y_i)$

Gibbs Sampling Algorithm

To get a sample from $p(x)$

1. Choose dimension i (random or by permutation)
2. Sample x_i from $p(x_i \mid \vec{x}_{\neg i})$

- $p(x_i \mid \vec{x}_{\neg i}) = \frac{p(\vec{x})}{p(\vec{x}_{\neg i})}$ with $\vec{x} = \{x_i, \vec{x}_{\neg i}\}$

Gibbs Sampling for models with hidden variables

- For models containing hidden variables $\vec{z}$, their posterior given the evidence, $p(\vec{z}|\vec{x})$ is a distribution commonly wanted

- The general formulation of a Gibbs sampler for such latent-variable models becomes:

$$p(z_i \mid \vec{z}_{\neg i}, \vec{x}) = \frac{p(\vec{z}, \vec{x})}{p(\vec{z}_{\neg i}, \vec{x})}$$

Outline

- Parameter Estimation
- Conjugate Distributions
- Modeling Text
- Bayesian networks and generative models
- Latent Dirichlet Allocation
- Gibbs Sampling
- **LDA Gibbs Sampler**

LDA Gibbs Sampler

- Target of inference is the distribution, $p(\vec{z}|\vec{w})$

$$p(\vec{z}|\vec{w}) = \frac{p(\vec{z},\vec{w})}{p(\vec{w})} = \frac{\prod_{i=1}^{W} p(z_i, w_i)}{\prod_{i=1}^{W} \sum_{k=1}^{K} p(z_i = k, w_i)}$$

- Full conditional, $p(z_i|\vec{z}_{\neg i},\vec{w})$ is used to simulate $p(\vec{z}|\vec{w})$

- This requires the joint distribution,

$$p\left(\vec{w},\vec{z}\middle|\vec{\alpha},\vec{\beta}\right) = p\left(\vec{w}\middle|\vec{z},\vec{\beta}\right)p(\vec{z}|\vec{\alpha})$$

$$p(\overrightarrow{w}|\vec{z},\vec{\beta})$$

- W words of the corpus are observed according to the independent multinomial trials

$$p\left(\overrightarrow{w}\middle|\vec{z},\underline{\phi}\right)=\prod_{t=1}^{W}p(w_i|z_i)=\prod_{t=1}^{W}\varphi_{z_i,w_i}$$

- Splitting the product over words into product over topics and one over vocabulary,

$$p\left(\overrightarrow{w}\middle|\vec{z},\underline{\phi}\right)=\prod_{k=1}^{K}\prod_{t=1}^{V}p(w_i=t|z_i=k)=\prod_{k=1}^{K}\prod_{t=1}^{V}{\varphi_{k,t}}^{{n_k}^{(t)}}$$

$p(\vec{w}|\vec{z},\vec{\beta})$

- Integrating over $\underline{\phi}$, we get

$$p(\vec{w}|\vec{z},\vec{\beta}) = \int p\left(\vec{w}\middle|\vec{z},\underline{\phi}\right) p(\underline{\phi}|\vec{\beta})\,\mathrm{d}\underline{\phi}$$

$$= \int \prod_{z=1}^{K} \frac{1}{\Delta(\vec{\beta})} \prod_{t=1}^{V} {\varphi_{z,t}}^{{n_z}^{(t)}+\beta_t-1} d\vec{\varphi}_z$$

$$= \prod_{z=1}^{K} \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})}\ , \vec{n}_z = \{{n^{(t)}}_z\}_{t=1\ldots N}$$

$$p(\vec{z}|\vec{\alpha})$$

$$
\begin{aligned}
p\left(\vec{z}\middle|\underline{\theta}\right) &= \prod_{i=1}^{W} p(z_i|d_i) \\
&= \prod_{m=1}^{M}\prod_{k=1}^{K} p(z_i = k|d_i = m) \\
&= \prod_{m=1}^{M}\prod_{k=1}^{K} {\vartheta_{m,k}}^{{n_m}^{(k)}}
\end{aligned}
$$

$$p(\vec{z}|\vec{\alpha})$$

$$p(\vec{z}|\vec{\alpha}) = \int p\big(\vec{z}\big|\underline{\theta}\big)p(\underline{\theta}|\vec{\alpha})\, d\underline{\theta}$$

$$= \int \prod_{m=1}^{M} \frac{1}{\Delta(\vec{\alpha})} \prod_{k=1}^{K} \vartheta_{m,k}{}^{n_m{}^{(k)}+\alpha_k-1} d\vec{\vartheta}_m$$

$$= \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})} \;, \vec{n}_m = \{n^{(k)}{}_m\}_{k=1\ldots K}$$

Joint distribution

$$p\left(\vec{z},\overrightarrow{w}\middle|\overrightarrow{\alpha},\vec{\beta}\right) = \prod_{z=1}^{K} \frac{\Delta(\vec{n}_z + \vec{\beta})}{\Delta(\vec{\beta})} \prod_{m=1}^{M} \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta(\vec{\alpha})}$$

Full conditional

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) = \frac{p(\vec{w}, \vec{z})}{p(\vec{w}, \vec{z}_{\neg i})} = \frac{p(\vec{w}|\vec{z})p(\vec{z})}{p(\vec{w}|\vec{z}_{\neg i})p(\vec{z}_{\neg i})}$$

$$p(z_i = k|\vec{z}_{\neg i}, \vec{w}) \propto \frac{\Delta\left(\vec{n}_z + \vec{\beta}\right)}{\Delta\left(\vec{n}_{z,\neg i} + \vec{\beta}\right)} \cdot \frac{\Delta(\vec{n}_m + \vec{\alpha})}{\Delta\left(\vec{n}_{m,\neg i} + \vec{\alpha}\right)}$$

$$\propto \frac{\Gamma\left({n_k}^{(t)} + \beta_t\right)\Gamma\left(\sum_{t=1}^{V} {n_{k,\neg i}}^{(t)} + \beta_t\right)}{\Gamma\left({n_{k,\neg i}}^{(t)} + \beta_t\right)\Gamma(\sum_{t=1}^{V} {n_k}^{(t)} + \beta_t)} \cdot \frac{\Gamma({n_m}^{(k)} + \alpha_k)\Gamma(\sum_{k=1}^{K} {n_{m,\neg i}}^{(k)} + \alpha_k)}{\Gamma({n_{m,\neg i}}^{(k)} + \alpha_k)\Gamma(\sum_{k=1}^{K} {n_m}^{(k)} + \alpha_k)}$$

$$\propto \frac{{n_{k,\neg i}}^{(t)} + \beta_t}{\sum_{t=1}^{V} {n_{k,\neg i}}^{(t)} + \beta_t} \cdot \frac{{n_{m,\neg i}}^{(k)} + \alpha_k}{\sum_{k=1}^{K} {n_{m,\neg i}}^{(k)} + \alpha_k}$$

Multinomial parameter sets $\underline{\theta}$ and $\underline{\phi}$

$$p\left(\vec{\vartheta}_m \middle| \mathrm{M}C, \vec{\alpha}\right) = \frac{1}{Z_{\vartheta_m}} \prod_{n=1}^{N_m} p\left(z_{m,n} \middle| \vec{\vartheta}_m\right) p(\vec{\vartheta}_m | \vec{\alpha})$$

$$= \mathrm{Dir}(\vec{\vartheta}_m | \vec{n}_m + \vec{\alpha})$$

$$p\left(\vec{\varphi}_m \middle| \mathrm{M}C, \vec{\beta}\right) = \frac{1}{Z_{\varphi_k}} \prod_{[i:z_i=k]}^{N_m} p(w_i | \vec{\varphi}_k) p(\vec{\varphi}_k | \vec{\beta})$$

$$= \mathrm{Dir}(\vec{\varphi}_k | \vec{n}_k + \vec{\beta})$$

Multinomial parameter sets $\underline{\theta}$ and $\underline{\phi}$

- Using the expectation of the Dirichlet distribution, $Dir(\vec{\alpha}) = a_i / \sum_i a_i$,

$$\varphi_{k,t} = \frac{n^{(t)}{}_{k} + \beta_t}{\sum_{t=1}^{V} n^{(t)}{}_{k} + \beta_t}$$

$$\vartheta_{m,k} = \frac{n^{(k)}{}_{m} + \alpha_k}{\sum_{k=1}^{K} n^{(k)}{}_{m} + \alpha_k}$$

Steps in Gibbs Sampling

- Initialize
  - Assign random topic to each word in the corpus
  - Get the counts
- For a given burn-in period
  - For all documents
    - For all words in the document
      - Decrement counts corresponding to a topic
      - Sample a topic from the multinomial (full conditional)
      - Increment counts corresponding to that topic
- Estimate the parameter sets

Querying

- Operation to
  - Retrieve documents relevant to a query document
- Similarity analysis
  - Find topic distribution of the query document (Query sampling)
  - Rank documents using KL-divergence (Similarity Ranking)

Query Sampling

- Consider a query as a vector of words, $\vec{w}$
- Estimate the posterior distribution of topics, $\vec{z}$ given the word vector, $\vec{w}$ and the LDA Markov state, $MC = \{\vec{z}, \vec{w}\}$: $p(\vec{z}|\vec{w}; MC)$
- Randomly assign topics to words and then perform a number of loops through the Gibbs sampling update,

$$p(\tilde{z}_i = k|\widetilde{w}_i = t, \tilde{z}_{\neg i}, \widetilde{w}_{\neg i}; MC) = \frac{{n_k}^{(t)} + {\tilde{n}_{k,\neg i}}^{(t)} + \beta_t}{\sum_{t=1}^{V} {n_k}^{(t)} + {\tilde{n}_{k,\neg i}}^{(t)} + \beta_t} \cdot \frac{{n_{\widetilde{m},\neg i}}^{(k)} + \alpha_k}{\sum_{k=1}^{K} {n_{\widetilde{m},\neg i}}^{(k)} + \alpha_k}$$

Query Sampling

- Topic distribution of the query document is given by,

$$\vartheta_{m,k} = \frac{n^{(k)}{}_{\tilde{m}} + \alpha_k}{\sum_{k=1}^{K} n^{(k)}{}_{\tilde{m}} + \alpha_k}$$

Similarity Ranking

- Let X be the topic distribution of the query document
- Let *Y* be the topic distribution of the document is the corpus
- Using KL-divergence, *distance(X,Y)* is given by,
  - *distance(X,Y) = ½(KL(X || M) + KL(Y || M))*
  - *where, M = ½(X+Y)*

Future Work

- Query expansion
  - Duplicating the query n times
  - Adding topic words relevant to the query

References

- *Heinrich, Gregor. "Parameter estimation for text analysis." Web: http://www. arbylon. net/publications/text-est. pdf (2005)*
- *Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022*
- Walsh, Brian. "Markov chain monte carlo and gibbs sampling." (2004).