

Day 2: Lecture 3

John Paisley

Columbia University

Duke-Tsinghua Machine Learning Summer School

July 26, 2017

Part I

LDA as Matrix Factorization

Q: For a particular document, what is $P(x_{dn} = v | \boldsymbol{\beta}, \theta_d)$?

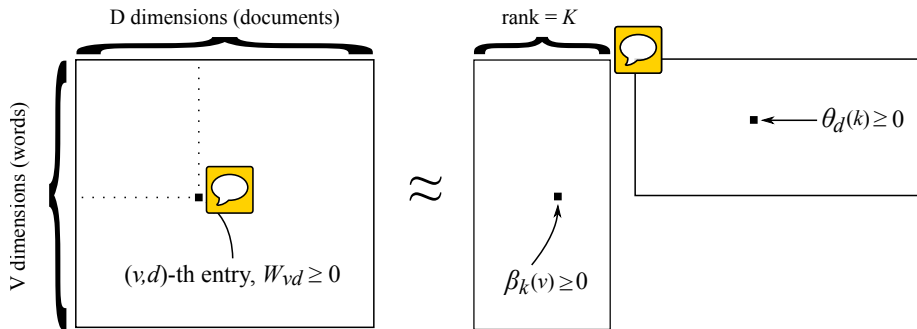
A: Find this by marginalizing out the cluster assignment,

$$\begin{aligned} P(x_{dn} = v | \boldsymbol{\beta}, \theta_d) &= \sum_{k=1}^K P(x_{dn} = v, c_{dn} = k | \boldsymbol{\beta}, \theta_d) \\ &= \sum_{k=1}^K \underbrace{P(x_{dn} = v | \boldsymbol{\beta}, c_{dn} = k)}_{= \beta_k(v)} \underbrace{P(c_{dn} = k | \theta_d)}_{= \theta_d(k)} \end{aligned}$$

Let $B = [\beta_1, \dots, \beta_K]$, $B_{vk} \Leftrightarrow \beta_k(v)$ and $\Theta = [\theta_1, \dots, \theta_D]$, $\Theta_{kd} \Leftrightarrow \theta_d(k)$.
Then

$$P(x_{dn} = v | \boldsymbol{\beta}, \boldsymbol{\theta}) = (B\Theta)_{vd}$$

In other words, we can read the probabilities from a matrix formed by taking the product of topic matrix B and topic proportion matrix Θ .



This representation gives better insight as to why LDA works than before.

LDA is a **low-rank matrix factorization** model.

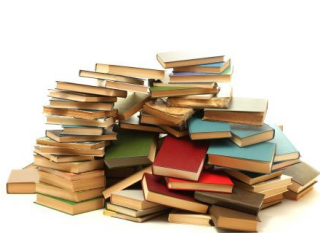
- Low rank because $K \ll V$ and $K \ll D$. Typically $20 \leq K \leq 200$.
- Low rank matrix factorization is well-studied. It tries to represent the columns of W as well as possible using a sum of K columns of B .
- The columns of B (i.e. the topics) are called latent factors in factor analysis parlance. For LDA, W are latent probabilities.

What are some other types of data that can constitute W ?

- Text data (as we've been discussing):
 - Word term-frequency matrices
 - W_{vd} contains the number of times word v appears in document d .
 - In this case, the factorization is of a multinomial parameter matrix.
- Image data:
 - Face identification problems
 - Put each *vectorized* $N \times M$ image of a face on a *column* of W
- Other discrete grouped data:
 - Quantize *continuous* sets of features using K-means
 - W_{vd} counts how many times group d uses cluster v
 - For example: group = song, features = $d \times n$ spectral information

Part II

Scalable variational inference



We often have data that naturally splits into sub-groups:

- Documents — groups of words
- Audio/music — groups of time-evolving frequency content,
- Images — groups of spatially-evolving texture content,
- ...

For such data, *mixed membership* and *latent factor* models are useful.

music band songs rock album jazz pop song singer night	book life novel story books man stories love children family	art museum show exhibition artist artists paintings painting century works	game knicks nets points team season play games night coach	show film television movie series says life man character know
---	---	---	---	---

We will discuss some general inference ideas in the context of LDA.


Latent Dirichlet Allocation

- 1 Generate K topics: $\beta_k \sim \text{Dirichlet}(\gamma)$
- 2 Generate document distributions: $\theta_d \sim \text{Dirichlet}(\alpha)$
- 3 For the n th word in the d th document,
 - a) Assign to topic, $c_{dn} \sim \text{Discrete}(\theta_d)$
 - b) Generate observation, $x_{dn} \sim \text{Discrete}(\beta_{c_{dn}})$


Many topic models build on this general structure.

LDA is part of a class of models called “mixed membership models.” From a matrix factorization perspective it is also a “latent factor model.”

These classes of models break down into the following:

- 1 Groups of data: $\mathbf{x}_1, \dots, \mathbf{x}_n$
- 2 Global variables: β
- 3 Local variables: z_1, \dots, z_n 
- 4 Other fixed parameters (we will ignore these)

Topic models:

- 1 global variables are topics (LDA: $\beta = \{\beta_1, \dots, \beta_K\}$)
- 2 local variables are document variables (LDA: z_d  $\{\theta_d, \mathbf{c}_d\}$)

Our goal is to find the posterior distribution of the hidden variables given the observations, $p(\beta, z|x)$. This requires approximations.

A property these models share is their factorization of the joint likelihood,

$$p(x, \beta, z) = p(\beta) \prod_{d=1}^D p(x_d, z_d | \beta)$$

- Given global β , the local (x_d, z_d) are conditionally independent.
- We can process the local variables separately before updating β .
- This is naturally parallelizable, but also amenable to stochastic optimization when using certain inference techniques.

As we've discussed, mean-field variational Bayes is an optimization-based approach to approximate posterior inference.

There are three fundamental steps:

- 1 Define a factorized distribution to approximate the posterior,

$$p(\boldsymbol{\beta}, \mathbf{z} | \mathbf{x}) \approx q(\boldsymbol{\beta}) \prod_d q(z_d).$$

- 2 Define the variational objective function,

$$\mathcal{L} = \mathbb{E}_q[\ln p(\mathbf{x}, \boldsymbol{\beta}, \mathbf{z})] - \mathbb{E}_q[\ln q(\boldsymbol{\beta}, \mathbf{z})]$$

- 3 Maximize \mathcal{L} with respect to parameters of each q .



By maximizing \mathcal{L} , we are equivalently minimizing the KL divergence between $p(\boldsymbol{\beta}, \mathbf{z} | \mathbf{x})$ and $q(\boldsymbol{\beta}) \prod_d q(z_d)$.

Batch inference

For this model structure, the **variational objective** is

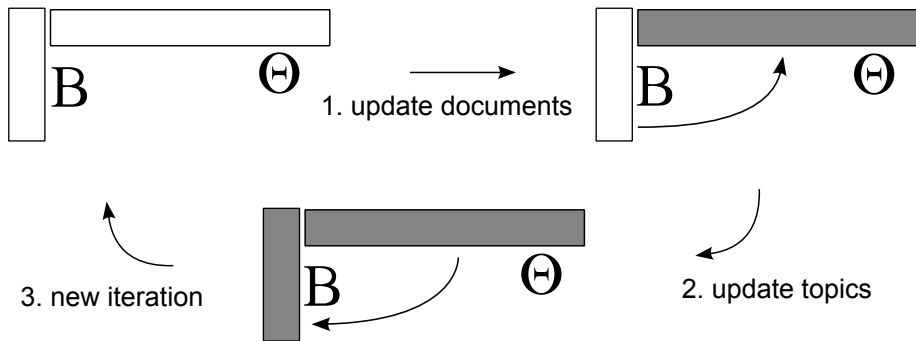
$$\mathcal{L} = \sum_{d=1}^D \mathbb{E}_q \left[\ln \frac{p(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{\beta})}{q(\mathbf{z}_d)} \right] - \mathbb{E}_q \left[\ln \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right]$$

A typical *batch* inference algorithm is:

- 1 For each d ,  optimize $q(\mathbf{z}_d)$
- 2 Optimize $q(\boldsymbol{\beta})$ 
- 3 Repeat

“Batch” because in each iteration we process all of the data “in one batch.”

Step 1 can take a very long time when D is large and a non-trivial amount of work is needed to optimize each $q(\mathbf{z}_d)$.



For example, we want to factorize a term-frequency matrix $W \approx B\Theta$. If there are millions of documents, updating Θ can be very slow.

A common solution is to throw away data.

The reasoning is:

- The data I observe is generated i.i.d. by some natural process
- If I select a random subset of data, I have a smaller sample from nature
- For practical purposes, this smaller data set has all the statistical information contained in the larger data set.

Conclusion: “Learning a model more quickly with a smaller data set shouldn’t make much difference compared with using all the data.”

However, in reality it does make a difference, particularly when we have complex models with many parameters.



Stochastic inference

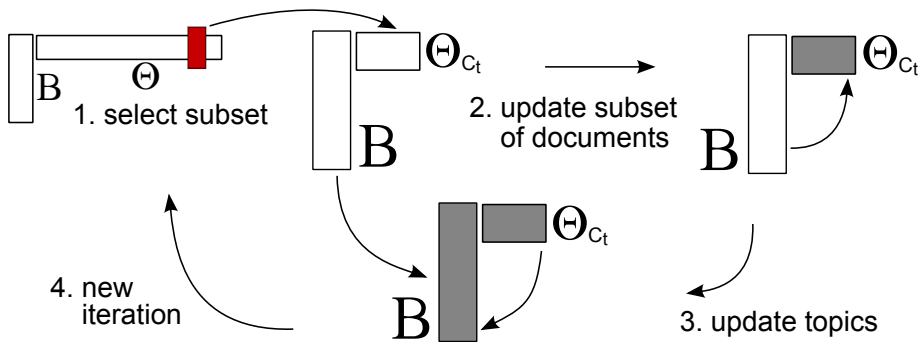
Observation: Because the likelihood factorizes, the objective function splits into a sum over local variable terms,

$$\mathcal{L} = \sum_{d=1}^D \underbrace{\mathbb{E}_q \left[\ln \frac{p(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{\beta})}{q(\mathbf{z}_d)} \right]}_{\text{local variables for } q(\mathbf{z}_d)} - \mathbb{E}_q \left[\ln \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right].$$

This suggests that we can use stochastic optimization to maximize \mathcal{L} .

Stochastic algorithm:

- 1 Select a small subset of $q(\mathbf{z}_d)$ at random and optimize 
- 2 Make a  slight *modification* of the parameters in $q(\boldsymbol{\beta})$
- 3 Repeat



The amount of work we do for Θ in each iteration is a tiny fraction.

Even though it takes more iterations to learn B this way, the reduced running time per iteration can more than compensate for this.

The sub-sample is a red block because the columns have been randomized.

Stochastic variational inference (SVI)

At iteration t , we uniformly sample a subset of indexes, $C_t \subset \{1, \dots, D\}$. We then construct the objective function

$$\mathcal{L}_t = \frac{L}{|C_t|} \sum_{d \in C_t} \mathbb{E}_q \left[\ln \frac{p(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{\beta})}{q(\mathbf{z}_d)} \right] - \mathbb{E}_q \left[\ln \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right].$$

Let $q(\boldsymbol{\beta})$ have a variational parameter vector $\boldsymbol{\psi}$. In iteration t ,

- 1 First, optimize variational parameters of all $q(\mathbf{z}_d)$ for which $d \in C_t$
- 2 Then update $\boldsymbol{\psi}$ as follows:

$$\boldsymbol{\psi} \leftarrow \boldsymbol{\psi} + \rho_t M \nabla_{\boldsymbol{\psi}} \mathcal{L}_t$$

The step size $\rho_t > 0$, $\sum_t \rho_t = \infty$ and $\sum_t \rho_t^2 < \infty$. M is a matrix and there are many choices. (SVI uses the inverse Fisher information of q .)

Q: First, why do we construct \mathcal{L}_t in this way, and why scale it by $\frac{D}{|C_t|}$?

A: Let $C_t \sim p(C_t)$. The answer is found by showing $\mathbb{E}_p[\mathcal{L}_t] = \mathcal{L}$.

We can write \mathcal{L}_t using indicators.

$$\mathcal{L}_t = \frac{D}{|C_t|} \sum_{d=1}^D \mathbb{1}(d \in C_t) \mathbb{E}_q \left[\ln \frac{p(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{\beta})}{q(\mathbf{z}_d)} \right] - \mathbb{E}_q \left[\ln \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right].$$

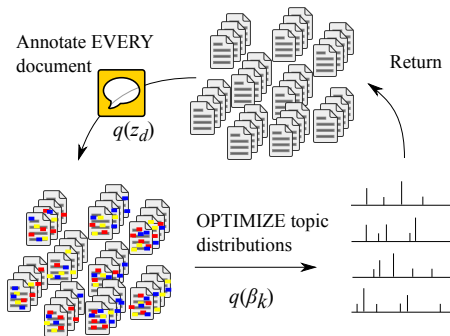
Therefore,

$$\mathbb{E}_p[\mathcal{L}_t] = \frac{D}{|C_t|} \sum_{d=1}^D \mathbb{E}_p[\mathbb{1}(d \in C_t)] \mathbb{E}_q \left[\ln \frac{p(\mathbf{x}_d, \mathbf{z}_d | \boldsymbol{\beta})}{q(\mathbf{z}_d)} \right] - \mathbb{E}_q \left[\ln \frac{p(\boldsymbol{\beta})}{q(\boldsymbol{\beta})} \right].$$

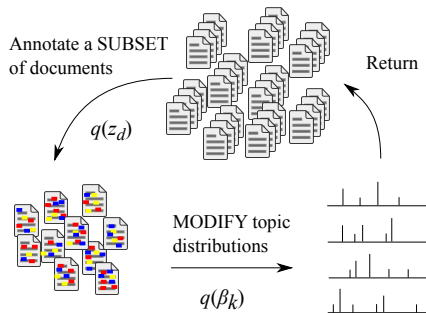
A fundamental result from probability says that $\mathbb{E}_p[\mathbb{1}(d \in C_t)] = P(d \in C_t)$.

There are $\binom{D}{|C_t|}$ equally probable subsets; d appears in $\binom{D-1}{|C_t|-1}$ of them...

For LDA, we end up with this type of variational inference structure.



Batch inference



Stochastic inference

At a high-level, the difference between batch and stochastic inference for models like LDA amounts to parameter averaging.

Batch inference at iteration t

$$q(\beta_k) = \text{Dirichlet}(\gamma_k), \quad \gamma_k = \gamma + \sum_{d=1}^D \lambda_d^{(k)} \leftarrow \text{work done for doc } d$$

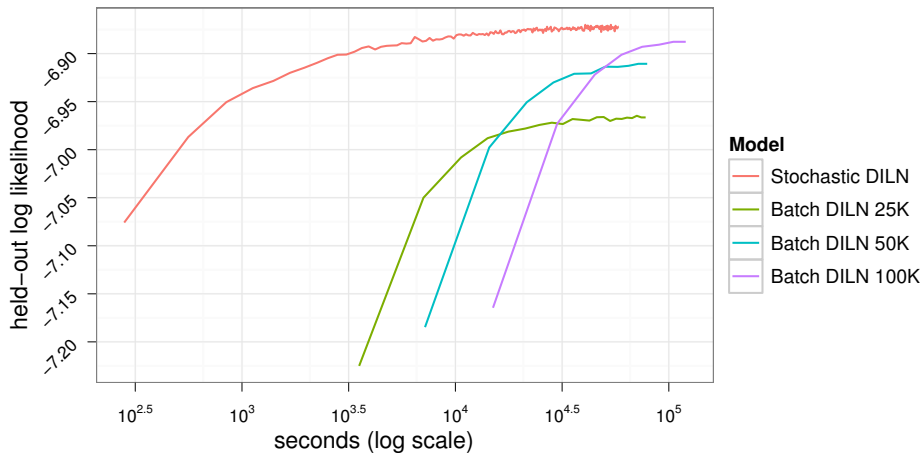
Stochastic inference at iteration t

$$q(\beta_k) = \text{Dirichlet}(\gamma_k), \quad \gamma_k = (1 - \rho_t)\gamma_k + \rho_t\left(\gamma + \frac{D}{|C_t|} \sum_{d \in C_t} \lambda_d^{(k)}\right)$$

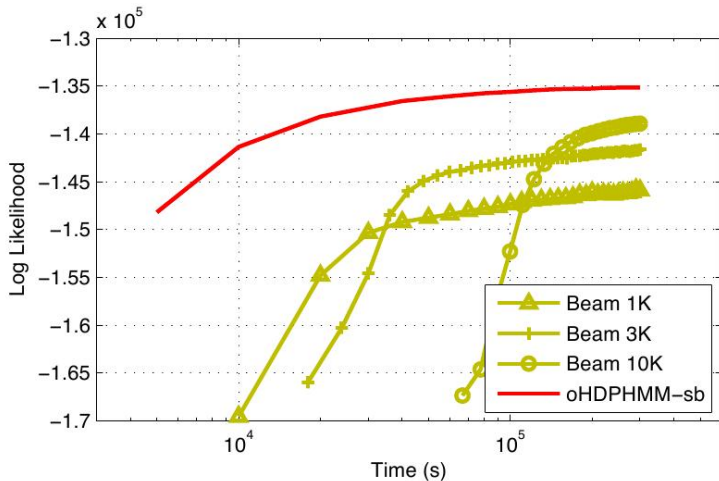
Recall: In Lecture 2 we showed that the V -dimensional λ_d corresponds to

$$\lambda_d^{(k)}(v) = \sum_n \mathbb{E}_q[\mathbb{1}(c_{dn} = k)] \mathbb{1}(x_{dn} = v)$$

Batch versus stochastic inference for a topic model.



Stochastic inference using all data versus MCMC using a subset of data.



Homework

Problem 1:

In Lecture 2, slide #5 gives the joint likelihood of the LDA model. On slide #11 we are told how to update the q distribution for a general variable and the following two slides (#12 and #13) show these q distributions for the three variables of LDA before concluding on slide #15.

Show the connection between slide #5 and the “answer” on slides #12 and #13 by writing out the math necessary to make the transition. In other words, double-check that what I have written on slides #12 and #13 is correct.

Problem 2:

Verify the last claim on slide #18 of Lecture 3 and finish the derivation by showing that $\mathbb{E}_p[\mathcal{L}_t] = \mathcal{L}$.

SOME MORE DETAILS FOR REFERENCE

Summary: Conjugate exponential family models

- 1 Exponential family:

$$p(w_d|\eta) = h(w_d) \exp \left\{ \eta^T t(w_d) - A(\eta) \right\}$$

η is the natural parameter and $A(\eta)$ is the log-partition function.

- 2 Conjugate prior for η :

$$p(\eta|\chi, \nu) = f(\chi, \nu) \exp\{\eta^T \chi - \nu A(\eta)\}$$

- 3 Posterior distribution given D i.i.d. samples (from Bayes rule):

$$p(\eta|\chi', \nu') = f(\chi', \nu') \exp\{\eta^T \chi' - \nu' A(\eta)\},$$

with $\chi' = \chi + \sum_{j=1}^D t(w_d)$ and $\nu' = \nu + D$.

Summary: The variational take on conjugacy

Imagine that w_d is a latent variable as well, with data $x_d \sim p(w_d)$. For variational inference, we define q distributions on each w_d and η .

Let the q distribution of η have the form of the prior on η :

$$q(\eta|\chi', \nu') = f(\chi', \nu') \exp\{\eta^T \chi' - \nu' A(\eta)\}.$$

Calculating $\nabla \mathcal{L} = \nabla \mathbb{E}_q \left[\ln \frac{p(\eta) \prod_d p(x_d, w_d|\eta)}{q(\eta) \prod_d q(w_d)} \right]$ with respect to $[\chi', \nu']^T$,

$$\nabla \mathcal{L} = - \begin{bmatrix} \frac{\partial^2 \ln f(\chi', \nu')}{\partial \chi' \partial \chi'^T} & \frac{\partial^2 \ln f(\chi', \nu')}{\partial \chi' \partial \nu'} \\ \frac{\partial^2 \ln f(\chi', \nu')}{\partial \nu' \partial \chi'^T} & \frac{\partial^2 \ln f(\chi', \nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \chi + \sum_d \mathbb{E}_q[t(w_d)] - \chi' \\ \nu + D - \nu' \end{bmatrix}.$$

Setting this to zero we can read off the update to $q(\eta|\chi', \nu')$. In this case q is just the posterior using the expected statistics $t(w_d)$ under $q(w_d)$.

Summary: The stochastic variational take on conjugacy

For the case where we subsample w_d to update $q(\eta)$:

$$[\chi', \nu']^T = [\chi'_{\text{old}}, \nu'_{\text{old}}]^T + \rho_t M \nabla \mathcal{L}_t$$

In this case, the gradient is:

$$\nabla \mathcal{L}_t = - \begin{bmatrix} \frac{\partial^2 \ln f(\chi', \nu')}{\partial \chi' \partial \chi'^T} & \frac{\partial^2 \ln f(\chi', \nu')}{\partial \chi' \partial \nu'} \\ \frac{\partial^2 \ln f(\chi', \nu')}{\partial \nu' \partial \chi'^T} & \frac{\partial^2 \ln f(\chi', \nu')}{\partial \nu'^2} \end{bmatrix} \begin{bmatrix} \chi + \frac{D}{|C_t|} \sum_{j \in C_t} \mathbb{E}_q t(w_d) - \chi' \\ \nu + D - \nu' \end{bmatrix}$$

Setting M to be the inverse of the matrix, we get

$$\begin{aligned} \chi' &= (1 - \rho_t) \chi'_{\text{old}} + \rho_t \left(\chi + \frac{D}{|C_t|} \sum_{d \in C_t} \mathbb{E}_q t(w_d) \right), \\ \nu' &= (1 - \rho_t) \nu'_{\text{old}} + \rho_t (\nu + D) \end{aligned}$$

A Dirichlet-multinomial example

Consider the following *portion* of a model:

$$\theta \sim \text{Dirichlet}(\chi), \quad z_d \stackrel{iid}{\sim} \text{Multinomial}(\theta), \quad q(\theta) = \text{Dirichlet}(\chi')$$

z_d itself is unknown, e.g., a latent indicator variable.

$$\text{Batch update: } \chi' = \chi + \sum_{d=1}^D \mathbb{E}_q[z_d]$$

- Requires fresh update of each $q(z_d)$.

$$\text{Stochastic update: } \chi'_t = (1 - \rho_t)\chi'_{\text{old}} + \rho_t\left(\chi + \frac{D}{|C_t|} \sum_{d \in C_t} \mathbb{E}_q[z_d]\right)$$

- Requires fresh update only of $q(z_d)$ for which $d \in C_t$.