Case Study 5: Mixed Membership Modeling

Variational Methods for LDA

Stochastic Variational Inference

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

March 13th, 2014

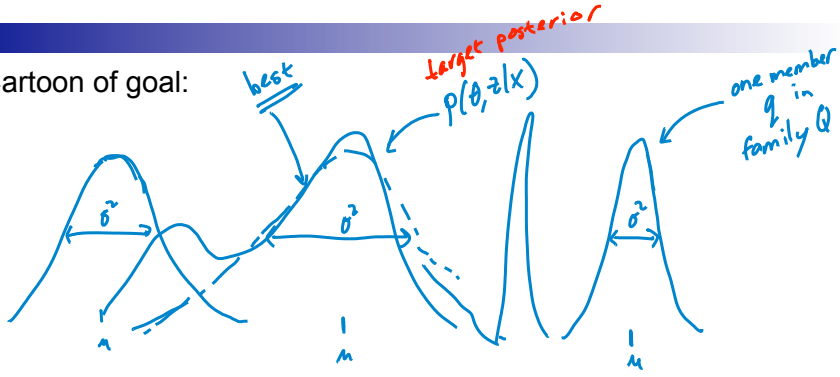
Variational Methods Goal

- Recall task: Characterize the posterior $p(\theta, z \mid x)$ (obs.; params; latent vars)
- Turn posterior inference into an optimization task
- Introduce a "tractable" family of distributions over parameters and latent variables
  - Family is indexed by a set of "free parameters"
  - Find member of the family closest to: $p(\theta, z \mid x)$

Call the family Q and want $q \in Q$ that is closest to $p(\theta, z \mid x)$

Variational Methods Cartoon

- Cartoon of goal:



eg; Q: all Gaussians

- Questions:
  1. How do we measure "closeness"?
  2. If the posterior is intractable, how can we approximate something we do not have to begin with?

Interpretations of Minimizing Reverse KL

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

- Evidence lower bound (ELBO)

log marginal likelihood or "evidence" — ELBO

$$\log p(x) = D(q(z,\theta)||p(z,\theta|x)) + \mathcal{L}(q) \geq \mathcal{L}(q)$$

const. — add to a const. — "ELBO"

- Therefore,
  - ELBO provides a lower bound on marginal likelihood
  - Maximizing ELBO is equivalent to minimizing KL

$\max \mathcal{L}(q) = \min D(q||p) =$ max lower bound of $\log p(x)$

what we can control — depends on what we don't know

Mean Field

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

joint var. q

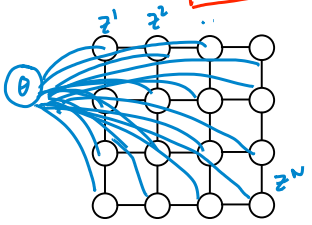
- How do we choose a Q such that the following is tractable?

$$\hat{q} = \arg\max_{q \in Q} \mathcal{L}(q)$$ ← new objective

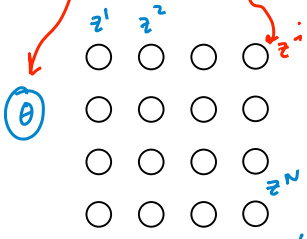
- Simplest case = mean field approximation $\theta, z = \{z^1, \ldots, z^N\}$
  - Assume each parameter and latent variable is conditionally independent given the set of free parameters

$$q(z, \theta) = q(\theta \mid \gamma) \prod_{i=1}^{N} q(z^i \mid \phi^i)$$

$\gamma, \{\phi^i\}$ are "free params" = control knobs in getting q close to p



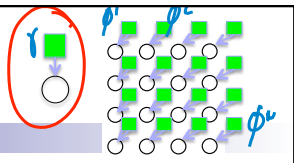
Original graph



Naïve mean field

← can also look at "structured mean field approx (break only some dep.'s)

Mean Field – Optimize γ



- Examine one free parameter, *e.g.*, γ

$$\mathcal{L}(q) = E_q[\log p(\theta \mid z, x)] + E_q[\log p(z, x)] - E_q[\log q(\theta \mid \gamma)] - \sum_i E_q[\log q(z^i \mid \phi^i)]$$

consider θ-full-cond. form

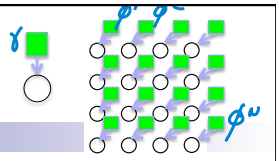
don't depend on γ because under q, $z^i \perp \theta$!

  - Look at terms of ELBO just depending on γ

$$\mathcal{L}^\gamma = E_q[\log p(\theta \mid z, x)] - E_q[\log q(\theta \mid \gamma)] + \text{const.}$$ (const. w.r.t. γ)

really just $q_\theta = q(\theta \mid \gamma)$ needed here

Mean Field – Optimize ϕ^i



- Examine another free parameter, e.g., ϕ^i

$$\mathcal{L}(q) = E_q[\log p(z^i \mid z_{\setminus i}, \theta, x)] + E_q[\log p(z_{\setminus i}, \theta, x)] - E_q[\log q(\theta \mid \gamma)] - \sum_i E_q[\log q(z^i \mid \phi^i)]$$

consider the z^i-full-cond. form

const. wrt ϕ^i

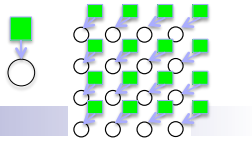
  - Look at terms of ELBO just depending on ϕ^i

$$\mathcal{L}^{\phi^i} = E_q[\log p(z^i \mid z_{\setminus i}, \theta, x)] - E_q[\log q(z^i \mid \phi^i)]$$

really just $q_{z^i} = q(z^i \mid \phi^i)$ here

- This motivates using a coordinate ascent algorithm for optimization
  - Iteratively optimize each free parameter holding all others fixed

Algorithm Outline



- **Initialization:** Randomly select starting distribution $q_\theta^{(0)}$
- **E-Step:** Given parameters, find posterior of hidden data

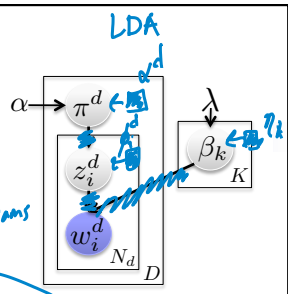
optimize ϕ → $q_z^{(t)} = \arg\max_{q_z} \mathcal{L}(q_z, q_\theta^{(t-1)})$ latent vars z

- **M-Step:** Given posterior distributions, find likely parameters θ

optimize γ → $q_\theta^{(t)} = \arg\max_{q_\theta} \mathcal{L}(q_z^{(t)}, q_\theta)$

- **Iteration:** Alternate E-step & M-step until convergence

Mean Field for LDA



- In LDA, our parameters are $\theta = \{\pi^d\}, \{\beta_k\}$
  $z = \{z_i^d\}$

free params

- The variational distribution factorizes as

$$q(\pi, \beta, z) = \prod_{k=1}^{K} q(\beta_k \mid \eta_k) \prod_{d=1}^{D} \left[q(\pi^d \mid \alpha^d) \prod_{i=1}^{N_d} q(z_i^d \mid \phi_i^d) \right]$$

$\mathrm{Dir}(\eta_{k1}, \ldots, \eta_{kV})$ $\quad \mathrm{Dir}(\alpha_1^d, \ldots, \alpha_K^d)$ $\quad \mathrm{Mult}(\phi_i^d)$

$\sum_{k=1}^{K} \phi_{ik}^d = 1$ ← need to enforce this!

- The joint distribution factorizes as

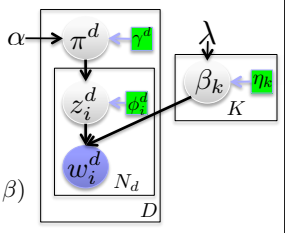
$$p(\pi, \beta, z, w) = \prod_{k=1}^{K} p(\beta_k \mid \lambda) \prod_{d=1}^{D} p(\pi^d \mid \alpha) \prod_{i=1}^{N_d} p(z_i^d \mid \pi^d) p(w_i^d \mid z_i^d, \beta)$$

$\mathrm{Dir}(\lambda_1, \ldots, \lambda_V)$ $\quad \mathrm{Dir}(\alpha_1, \ldots, \alpha_K)$ $\quad \mathrm{Mult}(\pi^d)$

Mean Field for LDA

$$q(\pi, \beta, z) = \prod_{k=1}^{K} q(\beta_k \mid \eta_k) \prod_{d=1}^{D} q(\pi^d \mid \gamma^d) \prod_{i=1}^{N_d} q(z_i^d \mid \phi_i^d)$$

$$p(\pi, \beta, z, w) = \prod_{k=1}^{K} p(\beta_k \mid \lambda) \prod_{d=1}^{D} p(\pi^d \mid \alpha) \prod_{i=1}^{N_d} p(z_i^d \mid \pi^d) p(w_i^d \mid z_i^d, \beta)$$



- Examine the ELBO

$$\mathcal{L}(q) = \sum_{k=1}^{K} E_q[\log p(\beta_k \mid \lambda)] + \sum_{d=1}^{D} E_q[\log p(\pi^d \mid \alpha)]$$

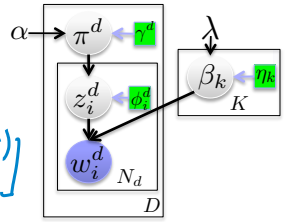
$$+ \sum_{d=1}^{d} \sum_{i=1}^{N_d} E_q[\log p(z_i^d \mid \pi^d)] + E_q[\log p(w_i^d \mid z_i^d, \beta)]$$

from joint dist.

$$- \sum_{k=1}^{K} E_q[\log q(\beta_k \mid \eta_k)] - \sum_{d=1}^{D} E_q[\log q(\pi^d \mid \gamma^d)] - \sum_{d=1}^{d} \sum_{i=1}^{N_d} E_q[\log q(z_i^d \mid \phi_i^d)]$$

all terms from q

Mean Field for LDA



- Let's look at some of these terms

$E_q[\log p(z_i^d \mid \pi^d)] = E_q[\log \pi^d_{z_i^d}] = E_q[\sum_{k=1}^{K} (\log \pi_k^d)^{I(z_i^d=k)}]$

$= \sum_{k=1}^{K} E_q[I(z_i^d=k) \log \pi_k^d] = \sum_{k=1}^{K} E_q[I(z_i^d=k)] E_q[\log \pi_k^d]$

$z_i^d \perp\!\!\!\perp \pi_k^d$ under q ⇒ why mean field is important

$\Pr(z_i^d=k) = \phi_{ik}^d$

$\psi(\gamma_k^d) - \psi(\sum_k \gamma_k^d)$

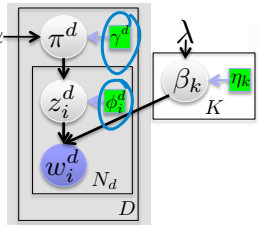
$E_q[\log q(z_i^d \mid \phi_i^d)]$

$= \sum_k E_q[I(z_i^d=k) \log \phi_{ik}^d] = \sum_k \phi_{ik}^d \log \phi_{ik}^d$

given

- Other terms follow similarly

Optimize via Coordinate Ascent



- Algorithm:

for $d = 1, \ldots, D$

$\frac{\partial \mathcal{L}}{\partial \gamma^d} = 0 \rightarrow \gamma^{d(t+1)} = \alpha + \sum_{i=1}^{N_d} \phi_i^{d(t)}$

for $i = 1, \ldots, N_d$

$\frac{\partial \mathcal{L}}{\partial \phi_i^d} = 0 \rightarrow \phi_i^d \propto \exp\{\psi(\gamma_{1:K}^{d(t+1)}) + \psi(\eta^{(t)}_{1:K, w_i^d}) - \psi(\sum_v \eta^{(t)}_{1:K,v})\}$

$\frac{1}{1+\cdots+1}$

use Lagrange multipliers to enforce ϕ_i^d is a pmf

DATA PARALLEL

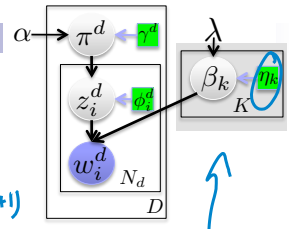
Optimize via Coordinate Ascent

- Algorithm:

for $k = 1, \ldots, K$

$$\frac{\partial \mathcal{L}}{\partial \eta_k} = 0 \rightarrow \eta_k^{(t+1)} = \lambda + \sum_{d=1}^{D} \sum_{i=1}^{N_d} w_i^d \phi_i^{d(t+1)}$$

aggregate



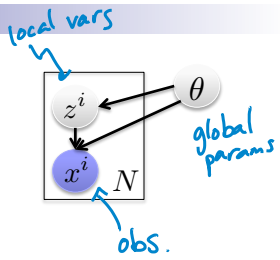
hard "global param"

Map Reduce

Generalizing

- Many Bayesian model have this form:

$$p(\theta, z^{1:N}, x^{1:N}) = p(\theta) \prod_{i=1}^{N} p(z^i \mid \theta) p(x^i \mid z^i, \theta)$$

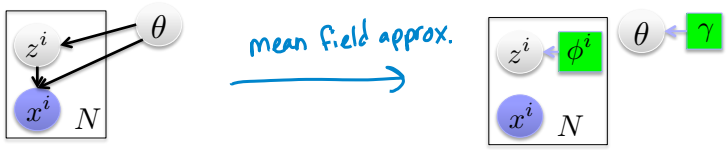


- Goal is to compute $p(\theta, z \mid x)$
- Assume each complete conditional is in the exponential family

$$p(z^i \mid \theta, x^i) = h(z^i) \exp\{\eta_\ell(\theta, x^i)^T z^i - a(\eta_\ell(\theta, x^i))\}$$ "local"

$$p(\theta \mid z, x) = h(\theta) \exp\{\eta_g(z, x)^T \theta - a(\eta_g(z, x))\}$$ "global"

Generalizing



- Mean field variational approximation

$$q(z, \theta) = q(\theta \mid \gamma) \prod_{i=1}^{N} q(z^i \mid \phi^i)$$

- Match each component to have same family as model conditional

$$p(\theta \mid z, x) = h(\theta) \exp\{\eta_g(z, x)^T \theta - a(\eta_g(z, x))\}$$

$$q(\theta \mid \gamma) = h(\theta) \exp\{\gamma^T \theta - a(\gamma)\}$$

- Same for local variational terms, too

Generalizing

- Under these exponential family assumptions, the gradient is:

$$\nabla_\gamma \mathcal{L} = a''(\gamma)(E_\phi[\eta_g(z, x)] - \gamma)$$

- This leads to a simple coordinate update (Ghahramani and Beal, 2001)

$$\nabla_\gamma \mathcal{L} = 0 \rightarrow \gamma^* = E_\phi[\eta_g(z, x)]$$

similarly for local vars

General Coord. Ascent Algorithm

Initialize γ randomly.

Repeat until the ELBO converges

1. For each data point, update the local variational parameters:

$$\phi_i^{(t)} = \mathrm{E}_{\gamma^{(t-1)}}[\eta_\ell(\theta, x_i)] \text{ for } i \in \{1, \ldots N\}.$$

2. Update the global variational parameters:

$$\gamma^{(t)} = \mathrm{E}_{\phi_i^{(t)}}[\eta_g(Z_{1:N}, x_{1:N})]$$

Note: cycle through each obs. before updating global param.

Case Study 5: Mixed Membership Modeling

Stochastic Variational Inference

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

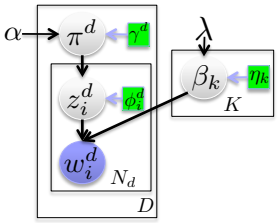
March 13th, 2014

Limitations of Batch Variational Methods

interested in huge datasets (e.g. millions of docs)

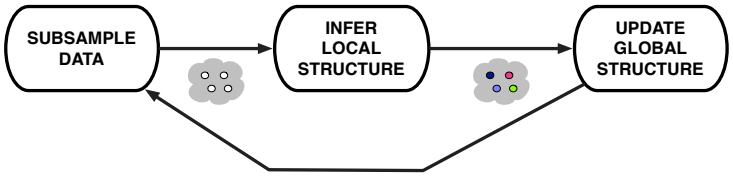
Limitations of Batch Variational Methods

- *Example = LDA*
  - Start from randomly initialized η_k (topics)
  - Analyze whole corpus before updating η_k again
- Streaming data: can't compute one iteration!



- *More generally…*
  - Do some local computation for each data point.
  - Aggregate these computations to re-estimate global structure.
  - Repeat.
- Inefficient, and cannot handle massive data sets.

Stochastic Variational Inference



- Stochastic variational inference harnesses:
  - Idea #1: **Stochastic optimization** (Robbins and Monro, 1951)
  - Idea #2: **Natural gradients** (Amari, 1998)

just like in SGD

Alternative Optimization Schemes

- Didn't have to do coord. ascent. Could have used gradient ascent.

$\hookrightarrow \nabla_\gamma \mathcal{L} = 0$

$\gamma^{(t+1)} = \gamma^{(t)} + \epsilon_t \nabla_\gamma \mathcal{L}$

- Here,

$$\mathcal{L}(q) = E_q[\log p(\theta)] - E_q[\log q(\theta)] - \sum_{i=1}^{N} E_q[\log p(z^i, x^i \mid \theta)] - E_q[\log q(z^i)]$$

touches all of the data! (docs)

- Use **stochastic gradient** step instead
  - Consider one data point x^t sampled uniformly at random and define:

$$\mathcal{L}_t(q) = E_q[\log p(\theta)] - E_q[\log q(\theta)] - N(E_q[\log p(z^t, x^t \mid \theta)] - E_q[\log q(z^t)])$$

"t-ELBO"

$E[\nabla \mathcal{L}_t] = \nabla \mathcal{L}$ ← using all data

over all subsets ↑ using a subset

Alternative Optimization Schemes

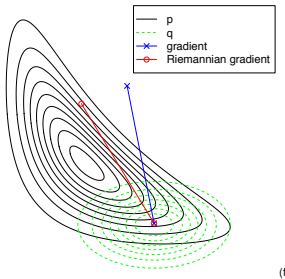
- Recall the gradient of the ELBO for the global parameter:

$$\nabla_\gamma \mathcal{L} = a''(\gamma)(E_\phi[\eta_g(z,x)] - \gamma)$$

- Even using just one data point, issue for scalability:

must compute $a''(\gamma)$...
computationally intensive

Natural Gradient of the ELBO



(from Honkela et al., 2010)

doubly beneficial:
① takes geometry into account
② removes $a''(\gamma)$ term

- The **natural gradient** accounts for the geometry of parameter space
- Natural gradient of the ELBO:

$$\hat{\nabla}_\gamma \mathcal{L} = E_\phi[\eta_g(z,x)] - \gamma$$

↑ natural grad.

Noisy Natural Gradients

- Let $\eta_t(z^t, x^t)$ be the conditional distribution of the global variable for the model where the observations are *N* replicates of x^t
- With this, the noisy natural gradient of the ELBO is

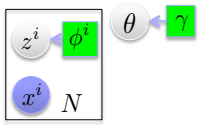
$$\hat{\nabla}_\gamma \mathcal{L}_t = \mathrm{E}_{\phi^t}[\eta_t(z^t, x^t)] - \gamma$$

($\mathcal{L}_t$: t-ELBO)

- Notes:
  - It only requires the local variational parameters of one data point.
  - In contrast, the full natural gradient requires all local parameters.
  - Thanks to conjugacy it has a simple form.

SVI Algorithm Overview

Initialize global parameters γ randomly.
Set the step-size schedule ϵ_t appropriately.
Repeat forever



1. Sample a data point uniformly,

$$x_t \sim \mathrm{Uniform}(x_1, \ldots, x_N).$$

2. Compute its local variational parameter,

$$\phi_t = \mathrm{E}_{\gamma^{(t-1)}}[\eta_l(\theta, x_t)]$$

(← just as in coord. ascent, but just for local var only)

3. Pretend its the only data point in the data set,

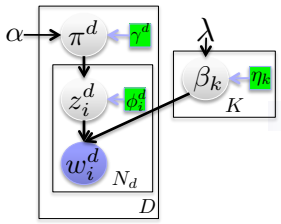
$$\hat{\gamma} = \mathrm{E}_{\phi_t}[\eta_t(Z_t, x_t)]$$

4. Update the current global variational parameter, (← noisy natural gradient step)

$$\gamma^{(t)} = (1 - \epsilon_t)\gamma^{(t-1)} + \epsilon_t \hat{\gamma}.$$

$$\gamma^{(t)} = \gamma^{(t-1)} + \epsilon_t \underbrace{\left[\mathrm{E}_{\phi^t}[\eta(z^t, x^t)] - \gamma^{(t-1)}\right]}_{\hat{\nabla}_{\gamma^{(t-1)}} \mathcal{L}_t}$$

SVI for LDA



- In LDA, the full ELBO is given by

$$\mathcal{L} = E_q[\log p(\beta)] - E_q[\log q(\beta)] + \sum_{d=1}^{D} E_q[\log p(\pi^d)] - E_q[\log q(\pi^d)]$$
$$+ \sum_{d=1}^{D} E_q[\log p(z^d, x^d \mid \pi^d, \beta)] - E_q[\log q(z^d)]$$

ELBO, as before

- Assuming *D* documents, consider one sampled at random

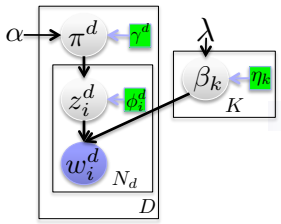
as if we viewed it D times

just one doc t

$$\mathcal{L}_t = E_q[\log p(\beta)] - E_q[\log q(\beta)] + D\left(E_q[\log p(\pi^t)] - E[\log q(\pi^t)]\right)$$
$$+ D\left(E_q[\log p(z^t, x^t \mid \pi^t, \beta)] - E_q[\log q(z^t)]\right)$$

t-ELBO

SVI for LDA



- Initialize $\eta^{(0)}$ randomly.
- Repeat (indefinitely):
  - Sample a document **d** uniformly from the data set.
  - For all *k*, initialize $\gamma_k^d = 1$
  - Repeat until converged
    - For i=1,…,N_d

local free params for doc d

$$\phi_{ik}^d \propto \exp\{E[\log \pi_k^d] + E[\log \beta_{k,w_i^d}]\}$$

Set $\gamma^d = \alpha + \sum_{i=1}^{N_d} \phi_i^d$

just like in coord. asc. for this doc.

  - Take a stochastic gradient step $\eta^{(t)} = \eta^{(t-1)} + \epsilon_t \nabla_\eta \mathcal{L}_d$

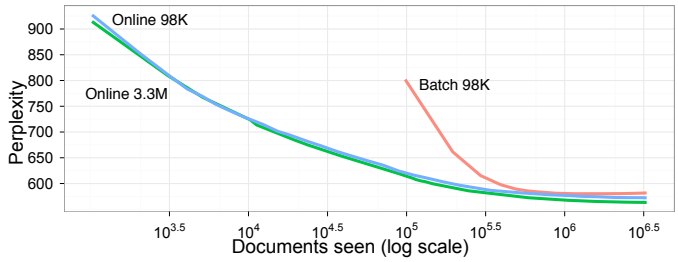
global vars

$$\lambda + D\sum_{i=1}^{N_d} \phi_i^d w_i^d - \eta^{(t-1)}$$

$$\eta^{(t)} = (1-\epsilon_t)\eta^{(t-1)} + \epsilon_t(\cdot)$$

looks exactly like what we had in coord. asc. update for doc. d (here D terms)

SVI for LDA in Practice



Documents analyzed	2048	4096	8192	12288	16384	32768	49152	65536
Top eight words	systems road made service announced national west language	systems health communication service billion language care road	service systems health companies market communication company billion	service systems companies business company billion health industry	service companies systems business company industry market billion	business service companies industry company management systems services	business service companies industry services company management public	business industry service companies services company management public

(Hoffman et al. 2010)

What you need to know…

- Variational methods
  - Mean field for LDA
- Stochastic variational inference
  - General idea of using natural gradients + stochastic optimization
  - Resulting generic algorithm
  - SVI for LDA

Reading

- **Inference in LDA:**
  - Basic LDA and batch variational inference in LDA: Blei, David M., Andrew Y. Ng, and Michael I. Jordan. "Latent dirichlet allocation." the Journal of machine Learning research 3 (2003): 993-1022.
  - Stochastic variational inference: Hoffman, Matt, et al. "Stochastic Variational Inference." arXiv: 1206.7051 (2012).

Acknowledgements

- Thanks to Dave Blei for some material in this lecture relating to SVI

Course Wrapup

Overview of CSE 547 / STAT 548 Topics Covered

Machine Learning for Big Data
CSE547/STAT548, University of Washington

Emily Fox

March 13th, 2014

What you need to know

- Case Study 1: Estimating Click Probabilities
  - Logistic regression
  - Regularization
  - Gradient descent, stochastic gradient decent
  - Hashing and sketching

What you need to know

- Case Study 2: Document Retrieval and Clustering
  - Approach 1: **k-NN**
  - *Algorithm:* Fast k-NN using KD-trees (exact)
  - *Algorithm:* Approximate k-NN using KD-trees and locality sensitive hashing
  - Approach 2: **k-means**
  - Data parallel problems
  - *Algorithm*: MapReduce framework and parallel k-means using MapReduce
  - Approach 3: **Gaussian mixture models (GMM)**
  - *Algorithm*: EM

What you need to know

- Case Study 3: fMRI Prediction
  - Regularized linear models: Ridge regression and LASSO
  - Sparsistency
  - LASSO solvers:
    - LARS
    - Shotgun (stochastic coordinate descent)
    - Hogwild (stochastic gradient descent)
    - Averaging methods
    - ADMM
  - LASSO variants:
    - Fused LASSO
    - Graphical LASSO
  - Coping with large covariances using latent factor models

What you need to know

- Case Study 4: Collaborative Filtering
  - Approach: Matrix factorization
  - *Algorithm*: Alternating least squares (ALS)
  - *Algorithm*: Stochastic gradient descent (SGD)
  - Cold-start problem and feature-based collaborative filtering
  - Model variants:
    - Non-negative matrix factorization
    - Probabilistic matrix factorization
      - *Algorithm*: Gibbs sampling
    - Probabilistic latent space models
  - Graph parallel problems
  - GraphLab framework and application to distributed ALS and Gibbs for matrix factorization

What you need to know

- Case Study 5: Document Mixed Membership Modeling
  - Approach 1: Bayesian document clustering model
  - Conditional independencies in directed graphical models
  - *Algorithm*: Gibbs sampling and collapsed Gibbs sampling
  - Approach 2: Latent Dirichlet allocation
  - *Algorithm*: Collapsed Gibbs sampling
  - *Algorithm*: Variational methods and stochastic variational inference

THANK YOU!!!

- You have been a great, interactive class! …especially for a 9:30am lecture =)
- We're looking forward to the poster session
- Thanks to Alden and Chad, too!