Case Study 4: Collaborative Filtering

GraphLab Review,
ALS, SGD, Gibbs Sampling for MF in GraphLab

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 27th, 2014

©Emily Fox 2014

The GraphLab Framework

1. Graph Based Data Representation
2. Update Functions User Computation
3. Scheduler
4. Consistency Model

©Emily Fox 2014
Women on the Verge of a Nervous Breakdown

The Celebration

City of God

Wild Strawberries

La Dolce Vita

Interpreting Low-Rank Matrix Completion (aka Matrix Factorization)

\[ X = LR \]

movie topic i

“romance”

how much user u likes topic i

how much movie v is about topic i

©Emily Fox 2014
Matrix Completion as a Graph

\[
X_{ij} \text{ known for black cells}
\]

\[
X_{ij} \text{ unknown for white cells}
\]

Rows index movies
Columns index users

\[
X = \begin{pmatrix}
X_{ij}
\end{pmatrix}
\]

black cells in data matrix = edges in graph (value in cell = weight on edge)

Coordinate Descent for Matrix Factorization: Alternating Least-Squares

\[
\min_{L,R} \sum_{(u,v):r_{uv} \neq ?} (L_u \cdot R_v - r_{uv})^2 + \lambda_u \|L_u\|_1 + \lambda_v \|R_v\|_1
\]

- Fix movie factors, optimize for user factors
- Independent least-squares over users

\[
\min_{L_u} \sum_{u \in U} (L_u \cdot R_v - r_{uv})^2
\]

- Fix user factors, optimize for movie factors
- Independent least-squares over movies

\[
\min_{R_v} \sum_{v \in V} (L_u \cdot R_v - r_{uv})^2
\]

- System may be underdetermined:
- Use regularization
- Converges to local optima
Alternating Least Squares Update Function

\[
\min_{L_u, R_v} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} (L_u \cdot R_v - r_{uv})^2
\]

\[
\min_{L_u, R_v} \sum_{u \in \mathcal{U}} \sum_{v \in \mathcal{V}} (L_u \cdot R_v - r_{uv})^2
\]

update \( \langle i, \text{scope} \rangle \)

read current factors for nhbrs

\[ x = \left[ \begin{array}{c} \overline{r}_u \end{array} \right] \] all movies rated by user \( i \)

read all ratings on edges

\[ y = \left[ \begin{array}{c} \overline{r}_v \end{array} \right] \] \( u \) \( v \) \( u \) \( v \)

Solve local regression problem

\[ X = (X^TX + \lambda I)^{-1}X^TY \]

SGD for Matrix Factorization in GraphLab

\[
\epsilon_t = L_u^{(t)} \cdot R_v^{(t)} - r_{uv}^{(t)}
\]

\[
\begin{bmatrix}
L_u^{(t+1)} \\
R_v^{(t+1)}
\end{bmatrix}
\leftarrow
\begin{bmatrix}
(1 - \eta \lambda_u) L_u^{(t)} - \eta \epsilon_t R_v^{(t)} \\
(1 - \eta \lambda_v) R_v^{(t)} - \eta \epsilon_t L_u^{(t)}
\end{bmatrix}
\]

GraphLab operates on vertices

Update \( \langle i, \text{scope} \rangle \)

Perform SGD update

for each neighbor of \( i \)

(for every edge connected to \( i \))
Bayesian PMF Example

- Latent user and movie factors:
  \[ L_u \sim N(m_u, \Sigma_u) \quad u = 1, \ldots, n \]
  \[ R_v \sim N(m_v, \Sigma_v) \quad v = 1, \ldots, m \]

- Observations:
  \[ r_{uv} \sim N(L_u R_v, \sigma^2) \]

- Hyperparameters:
  \[ \phi = \{ m_u, \Sigma_u, m_v, \Sigma_v, \sigma^2 \} \]

- Want to predict new movie rating:
  \[ p(r_{uv}^{new} | l_u, r_v) = \int p(r_{uv}^{new} | l_u, r_v) \cdot p(l_u, r_v | x, \phi) \, dl_u dr_v \]

Bayesian PMF Gibbs Sampler

- Outline of Bayesian PMF sampler
  1. Init: \( L^{(0)} \), \( R^{(0)} \)
  2. For \( k = 1, \ldots, Niter \)
     (i) Sample hyperparams \( \phi^{(k)} \), \( \phi_{ll}^{(k)} \), \( \phi_{ll}^{(k)} \), \( \phi_{lr}^{(k)} \)
     (ii) For each user \( u = 1, \ldots, n \) sample in parallel
        \[ L_u^{(k+1)} \sim p(L_u | x, R^{(k)}, \phi^{(k)}) \]
     (iii) For each movie \( v = 1, \ldots, m \) sample in parallel
        \[ R_v^{(k+1)} \sim p(R_v | x, L^{(k+1)}, \phi^{(k)}) \]

Very similar to ideas of ALS (systematically)
Bayesian PMF Example

- For user $u$: $p(L_u | X, R, \phi_u) \propto p(L_u | \phi_u) \prod_{v \in V_u} p(r_{uv} | L_u, R_v, \phi_R)$

$$p(L_u | X, R, \phi_u) \propto N(L_u | \mu_u, \Sigma_u) \prod_{v \in V_u} N(r_{uv} | L_u R_v, \sigma^2)$$

Symmetrically for $R_v$ conditioned on $L$ (breaks down over movies)

- Luckily, we can use this to get our desired posterior samples

PMF Gibbs Sampling in GraphLab

$$p(L_u | X, R, \phi_u) = N(\mu_u, \Sigma_u)$$

$$\Sigma_u = \Sigma_u^{-1} + \sigma^2 \sum_{v \in V_u} R_u R_v^T$$

$$\mu_u = \Sigma_u^{-1} \sum_{v \in V_u} r_{uv} R_v + \mu_u$$

read current factors for nhbrs $R_j$
read ratings on edges $r_{ij}$
set $\Sigma_i = \Sigma_i^{-1} + \sigma^2 \sum_{j \neq i} r_{ij} R_j^T$

fixed at vertex $i$
set $\mu_i = \Sigma_i^{-1} \sum_{j \neq i} r_{ij} R_j^T$

Sample $L_i \sim N(\hat{\mu}_i, \Sigma_i^{-1})$
Release 2.2 available now
http://graphlab.org
Documentation... Code... Tutorials... (more on the way)

GraphChi 0.1 available now
http://graphchi.org

What you need to know...

- Data-parallel versus graph-parallel computation
- Bulk synchronous processing versus asynchronous processing
- GraphLab system for graph-parallel computation
  - Data representation
  - Update functions
  - Scheduling
  - Consistency model
- ALS, SGD and Gibbs for matrix factorization/PMF in GraphLab
Reading

- Papers under “Case Study IV: Parallel Learning with GraphLab”

- Optional:
  - Parallel Splash BP
  - [http://www.ml.cmu.edu/research/dap-papers/dap-gonzalez.pdf](http://www.ml.cmu.edu/research/dap-papers/dap-gonzalez.pdf)

Acknowledgements

- Slides based on Carlos Guestrin’s GraphLab talk
Case Study 5: Mixed Membership Modeling

Clustering Documents Revisited, Latent Dirichlet Allocation

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 27th, 2014

Document Retrieval

- **Goal:** Retrieve documents of interest
- **Challenges:**
  - Tons of articles out there
  - How should we measure similarity?
Task 1: Find Similar Documents

- First considered:
  - Input: Query article
  - Output: Set of k similar articles

Task 2: Cluster Documents

- Then examined:
  - Cluster documents based on topic
Document Representation

- Bag of words model

\[ d = \text{word} \]
\[ \text{cat, dog, hat, car} \]

\[ x^d = [\cdot \cdot \cdot] \]

- Bag of words model

- Documents:
- Associated topics:
- Parameters: \( \theta = \{ \pi, \beta \} \)

\[ \Pi = [\pi_1, \ldots, \pi_K] \]

\[ \Pr(z^d = k) = \pi_k \]

\[ \beta = \frac{1}{K} \left[ \begin{array}{c} \beta_1 \\ \vdots \\ \beta_K \end{array} \right] \]

A Generative Model
A Generative Model

- Documents: $x^1, \ldots, x^D$
- Associated topics: $z^1, \ldots, z^D$
- Parameters: $\theta = \{\pi, \beta\}$
- Generative model:

  $z^d \sim \pi$  generate topic
  
  $w_i^d | z^d \sim \beta_{z^d}$  \(i = 1, \ldots, N_d\)

Model In Pictures

- Mixture weights (on topics)
- Topic distributions (on words)
- For each document,
  
  $z^d \sim \pi$
  
  $w_i^d | z^d \sim \beta_{z^d}$
Bayesian Document Model

- Model parameters $\pi, \{\beta_k\}$ unknown
- Bayesian approach
  - place priors on parameters
- Need distribution on pmf's

The Simplex in 3D

- The simplex defines the hyperplane of vectors that sum to 1

$\theta = [\theta_1, \ldots, \theta_K]$  
$0 \leq \theta_k \leq 1$  
$\sum_{k=1}^{K} \theta_k = 1$
Dirichlet Distributions

- The Dirichlet distribution is defined on the simplex

\[ \pi \sim \text{Dir}(\alpha_1, \ldots, \alpha_K) \]

\[ \Rightarrow \sum \pi_k = 1 \text{ and } \pi \geq 0 \forall k \]

The probability mass function is given by:

\[ p(\pi | \alpha) = \frac{\Gamma(\sum_k \alpha_k)}{\prod_k \Gamma(\alpha_k)} \prod_k \pi_k^{\alpha_k - 1} \]

- Moments:

\[ E_\alpha[\pi_k] = \frac{\alpha_k}{\alpha_0} \]

\[ \text{Var}_\alpha[\pi_k] = \frac{K - 1}{K^2(\alpha_0 + 1)} \]

Dirichlet Probability Densities

- Dirichlet distributions are used to model probability distributions over discrete outcomes.

- Represented by non-negative vectors that sum to one.

- Picture representation:
  - \((1,0,0)\)
  - \((0,0,1)\)
  - \((1/2,1/2,0)\)
  - \((1/3,1/3,1/3)\)
  - \((1/4,1/4,1/2)\)
  - \((0,1,0)\)

- Come from a Dirichlet distribution.

- Uniform over simplex.

- Mass more concentrated at \(1/3,1/3,1/3\).

- Mass at corners.

- Draws from Dir.
Dirichlet Samples

Samples are \textit{sparse} for small values of $\alpha_i$.

\[ \mathbb{E}_{\alpha}[\pi_k] = \frac{\alpha_k}{\alpha_0} \]

(5D example)

\[ \text{Dir}(\pi | 0.1, 0.1, 0.1, 0.1, 0.1) \]

\[ \text{Dir}(\pi | 1.0, 1.0, 1.0, 1.0, 1.0) \]

Model Summary

- Prior on model parameters
  - E.g., symmetric Dirichlet for $\pi$

- Dirichlet prior for topic parameters

- Sample observations as
  \[ z^d \sim \pi, \quad d = 1, \ldots, D \]
  \[ w_i^d \mid z^d \sim \beta_{z^d}, \quad i = 1, \ldots, N_d \]
Posterior Inference via Sampling

- Iterate between sampling and actual observations.
  \[ \pi \sim p(\pi | \tau, z, \beta, \lambda, w) \]
  For \( k = 1, \ldots, K \)
  \[ \alpha_k \sim p(\alpha_k | \pi, \tau, z, \beta, i, k, j, w) \]
  For \( d = 1, \ldots, D \)
  \[ z^d \sim p(z^d | \pi, \tau, z, \beta, i, k, j, w) \]

- What form do these complete conditionals take?
  - First a look at statements of conditional independence in directed graphical models.

Conditional Independence in Bayes Nets

- Consider 4 different junction configurations:

- Conditional versus unconditional independence:
  \[ p(x, y, z) = p(x)p(y|x, z) \implies p(x, z) = p(x)p(z) \implies x \perp \perp z \]
  \[ p(x, z | y) \propto p(x, y, z) = p(x)p(y|x) \]

- Explaining away:
  - \( x = \) earthquake, \( z = \) burglar, \( y = \) car alarm
  - \( p(x) \) prior, \( p(z) \) prior, \( p(y|x) \) likelihood, \( p(y) \) likelihood
  - If alarm \( (y \text{ falls}) \), an increase in earthquake \( p(x|y) \), means \( p(x|y) \) lower