## Case Study 4: Collaborative Filtering

# Review: Probabilistic Matrix Factorization

Machine Learning for Big Data
CSE547/STAT548, University of Washington
Emily Fox
February 20th, 2014



# Probabilistic Matrix Factorization (PMF)

- A generative process:
  - Pick user u factors $L_{u_1}, \ldots, L_{u_k}$ $\quad L_{u_i} \overset{iid}{\sim} N(0, \sigma_u^2) \leftarrow P(L)$ prior on user factors
  - Pick movie v factors $R_{v_1}, \ldots, R_{v_k}$ $\quad R_{v_i} \overset{iid}{\sim} N(0, \sigma_v^2) \leftarrow P(R)$
  - For each (user,movie) pair observed:
    - Pick rating as $L_u \cdot R_v$ + noise $\rightarrow r_{uv} \mid L_u, R_v \sim N(L_u \cdot R_v, \sigma_r^2)$
    - noise: $N(0, \sigma_r^2)$
    - $P(X \mid L, R)$ likelihood
- Joint probability:

$$P(L,R,X) = P(L)\,P(R)\,P(X \mid L, R)$$

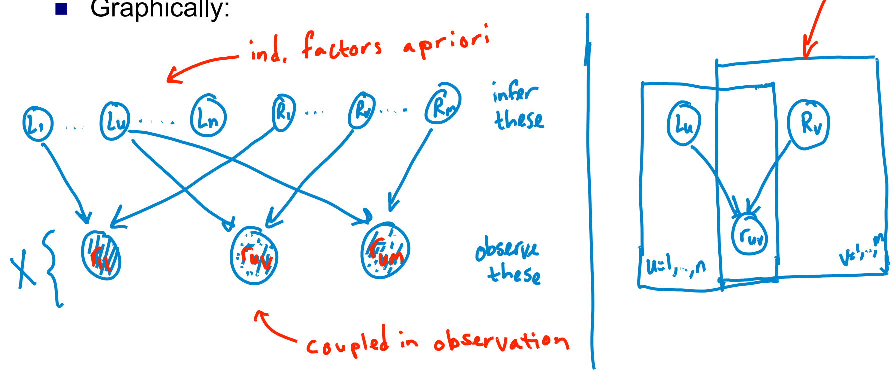
## PMF Graphical Model

$$\overbrace{P(L, R \mid X) \propto P(L)P(R)P(X \mid L, R)}^{\propto P(L,R,X)}$$

posterior $\propto$ joint prob.

- Graphically:

"plates"








## MAP versus Regularized Least-Squares for Matrix Completion

- MAP under Gaussian Model:

$$\max_{L,R} \log P(L, R \mid X) =$$

$$-\frac{1}{2\sigma_u^2}\sum_u \sum_i L_{u_i}^2 - \frac{1}{2\sigma_v^2}\sum_v \sum_i R_{v_i}^2 - \frac{1}{2\sigma_r^2}\sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2 + \text{const}$$

- Least-squares matrix completion with $L_2$ regularization:

$$\min_{L,R} \frac{1}{2}\sum_{r_{uv}} (L_u \cdot R_v - r_{uv})^2 + \frac{\lambda_u}{2}||L||_F^2 + \frac{\lambda_v}{2}||R||_F^2$$

objectives are equivalent!

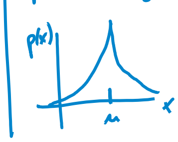
- Understanding as a probabilistic model is very useful! E.g.,
  - Change priors

$L_{u_i} \overset{iid}{\sim} N(0, \sigma_u^2)$, $R_{v_i} \overset{iid}{\sim} N(0, \sigma_v^2)$ } $L_2$ reg

$L_{u_i} \overset{iid}{\sim} \text{Laplace}(0, \sigma_u)$, $R_{v_i} \overset{iid}{\sim} \text{Laplace}(0, \sigma_v)$ } $L_1$ reg

$\text{Laplace}(\mu, b)$

$p(x) \propto e^{-\frac{|x-\mu|}{b}}$




  - Incorporate other sources of information or dependencies

very key! many extensions of PMF

# Posterior Computations

maintain this

- MAP estimation focuses on point estimation:

$$\hat{\theta}^{MAP} = \arg\max_{\theta} p(\theta \mid x)$$

data

parameters


- What if we want a full characterization of the posterior?
  - Maintain a measure of uncertainty
  - Estimators other than posterior mode (different loss functions)
  - Predictive distributions for future observations

$$p(x^{N+1} \mid x^1,\dots,x^N) = \int p(x^{N+1} \mid \theta)\, p(\theta \mid x^1,\dots,x^N)\, d\theta$$

← assuming $x^i$ iid given $\theta$ (exch.)

belief about $\theta$ having seen obs. $x^1,\dots,x^N$

integrate over uncertainty in model params

Contrast with:

$$p(x^{N+1} \mid \hat{\theta}^{MAP}(x^1,\dots,x^N))$$

← make pred w/ $\hat{\theta}^{MAP}$ after N obs.

"plug-in estimator"

- Often ~~no closed-form characterization~~ (e.g., mixture models, PMF, etc.)



# Bayesian PMF Example

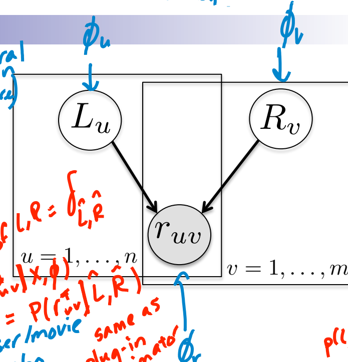
full Bayesian approach place priors on $\phi$ as well!

- Latent user and movie factors:

$L_u \sim N(\mu_u, \Sigma_u)$ $u=1,\dots,n$

$R_v \sim N(\mu_v, \Sigma_v)$ $v=1,\dots,m$

(more general than before)




- Observations $r_{uv} \sim N(L_u' R_v, \sigma_r^2)$
- Hyperparameters:

$$\rightarrow \phi = \{\underbrace{\mu_u, \Sigma_u}_{\phi_u}, \underbrace{\mu_v, \Sigma_v}_{\phi_v}, \underbrace{\sigma_r^2}_{\phi_r}\}$$

if post. of $L,R = \delta_{\hat{L},\hat{R}}$

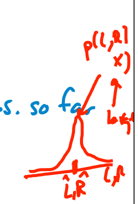
$\Rightarrow p(r^*_{uv} \mid X, \phi) = p(r^*_{uv} \mid \hat{L},\hat{R})$

same as plug-in estimator

new user/movie combo

- Want to predict new movie rating:

posterior given obs. so far




$$p(r^*_{uv} \mid X, \phi) = \int p(r^*_{uv} \mid L_u, R_v)\, p(L,R \mid X, \phi)\, dL\, dR$$

new rating

obs. ratings

integrate over posterior uncertainty in user/movie factors

# Bayesian PMF Example

$$p(r^*_{uv} \mid X, \phi) = \int p(r^*_{uv} \mid L_u, R_v)\, p(L, R \mid X, \phi)\, dL\, dR$$

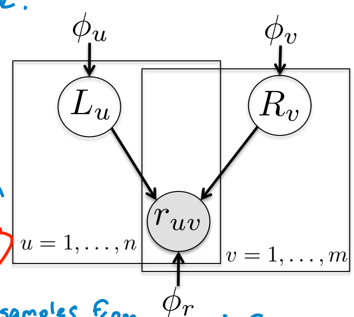
ANALYTICALLY INTRACTABLE!

- Monte Carlo methods:

Approx. as

$$p(r^*_{uv} \mid X, \phi) \approx \frac{1}{N} \sum_{k=1}^{N} p(r^*_{uv} \mid L_u^{(k)}, R_v^{(k)})$$

samples from posterior ... how?




- Ideally: $(L^{(k)}, R^{(k)}) \overset{iid}{\sim} p(L, R \mid X, \phi)$ ← ind samples from posterior

$$p(L, R \mid X) = \frac{p(X \mid L, R)\, p(L)\, p(R)}{p(X)}$$

$$p(X) = \int p(X \mid L, R)\, p(L)\, p(R)\, dL\, dR$$ ← issue!

Again, intractable "



# Bayesian PMF Gibbs Sampler

← return dependent samples from the posterior eventually

- Outline of Bayesian PMF sampler

1. Init $L^{(1)}, R^{(1)}$
2. For $k = 1, \dots, N_{iter}$

   (i) Sample hyperparams $\phi^{(k)} = \{\phi_u^{(k)}, \phi_v^{(k)}, \phi_r^{(k)}\}$ ← from cond. posterior

   (ii) For each user $u = 1, \dots, n$ sample in parallel

   $$L_u^{(k+1)} \sim p(L_u \mid X, R^{(k)}, \phi^{(k)})$$

   (iii) For each movie $v = 1, \dots, m$ sample in parallel

   $$R_v^{(k+1)} \sim p(R_v \mid X, L^{(k+1)}, \phi^{(k)})$$

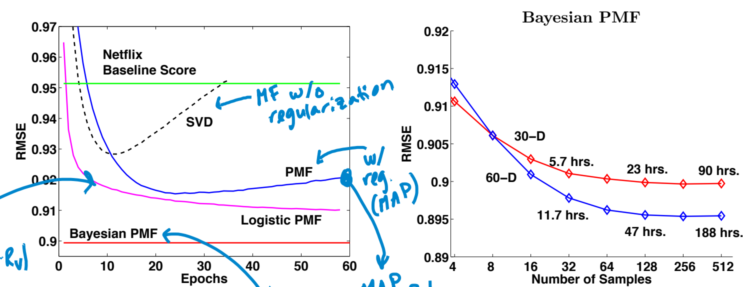
just Gaussian dist.

Very similar to ideas of ALS (systematically)

# Bayesian PMF Results

From Salakhutdinov and Mnih, ICML 2008

- Netflix data with:
  - Training set = 100,480,507 ratings from 480,189 users on 17,770 movie titles
  - Validation set = 1,408,395 ratings.
  - Test set = 2,817,131 user/movie pairs with the ratings withheld.




MF w/o regularization

w/ reg. (MAP)

uses $r_{uv} = \sigma(L_u \cdot R_v)$

logistic fcn

init. to MAP soln

*Figure 2.* Left panel: Performance of SVD, PMF, logistic PMF, and Bayesian PMF using 30D feature vectors, on the Netflix validation data. The y-axis displays RMSE (root mean squared error), and the x-axis shows the number of epochs, or passes, through the entire training set. Right panel: RMSE for the Bayesian PMF models on the validation set as a function of the number of samples generated. The two curves are for the models with 30D and 60D feature vectors.



# Bayesian PMF Results

From Salakhutdinov and Mnih, ICML 2008

- Bayesian model better controls for overfitting by averaging over possible parameters (instead of committing to one)

dim of user/movie factors

| D | Valid. RMSE PMF | Valid. RMSE BPMF | % Inc. | Test RMSE PMF | Test RMSE BPMF | % Inc. |
|---|---|---|---|---|---|---|
| 30 | 0.9154 | 0.8994 | 1.74 | 0.9188 | 0.9029 | 1.73 |
| 40 | 0.9135 | 0.8968 | 1.83 | 0.9170 | 0.9002 | 1.83 |
| 60 | 0.9150 | 0.8954 | 2.14 | 0.9185 | 0.8989 | 2.13 |
| 150 | 0.9178 | 0.8931 | 2.69 | 0.9211 | 0.8965 | 2.67 |
| 300 | 0.9231 | 0.8920 | 3.37 | 0.9265 | 0.8954 | 3.36 |

Bayes. model improves

*Table 1.* Performance of Bayesian PMF (BPMF) and linear PMF on Netflix validation and test sets.

Note: Each sampling stage of BPMF requires an $O(D^3)$ operation, so not for free

# What you need to know…

- Idea of full posterior inference vs. MAP estimation
- Gibbs sampling as an MCMC approach
- Example of inference in Bayesian probabilistic matrix factorization model



## Case Study 4: Collaborative Filtering

# Matrix Factorization and Probabilistic LFMs for Network Modeling

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

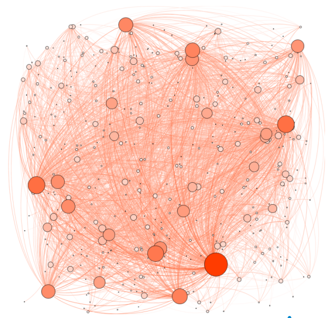
Emily Fox

February 20$^{th}$, 2014

# Network Data

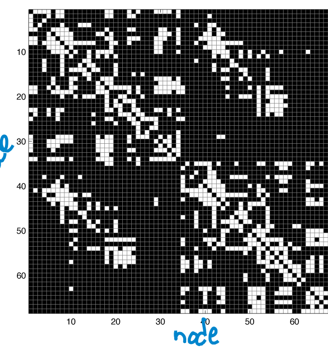
- Structure of network data

white squares = edge between node i and j



nodes in a network w/ undirected edges

node




node

no self edges

Adjacency matrix



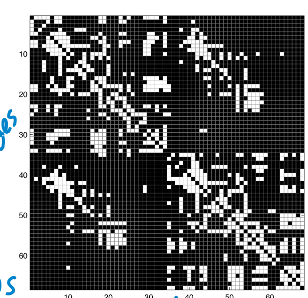
# Properties of Data Source

- Similarities to Netflix data:
  - Matrix -valued data (adj. matrix)
  - High-dimensional many nodes
  - Sparse few links between nodes (eg ppl in social network)
- Differences
  - Square ← same indices for rows + columns
  - Binary

yes/no for link, though you could have multigraph (multiple link instances bt nodes)

If undirected, then matrix is symmetric

nodes




nodes

# Matrix Factorization for Network Data

- Vanilla matrix factorization approach:

In undirected case, just introduce node (eg user) factors $L_u$

$$r_{uv} \approx L_u \cdot L_v \quad \leftarrow \text{edge bt users } u + v$$

In directed (or asymm.) case, introduce sender factors $L_u$ and receiver factors $\tilde{L}_u$ (every node/user has both $L_u$ and $\tilde{L}_u$)

$$r_{uv} \approx L_u \cdot \tilde{L}_v \quad \leftarrow \text{edge from user } u \text{ to user } v$$

- What to return for link prediction?

Is $r_{uv}$ binary? $L_u \in \mathbb{R}^k$ → no.

Many options, but can return top $\tilde{k}$ $r_{uv_1}, \dots, r_{uv_{\tilde{k}}}$ (just use threshold rule)

- Slightly fancier:

More appropriate to have $r_{uv} \in [0,1]$

Use $r_{uv} \approx \sigma(L_u \cdot L_v)$ $\sigma$ = logistic fcn



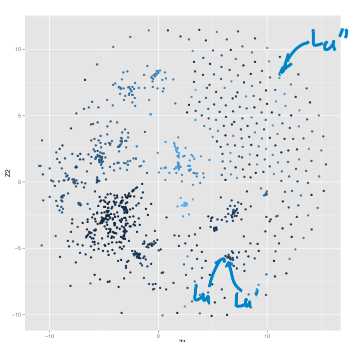

# Probabilistic Latent Space Models

- Assume features (covariates) of the user $x_u$ or relationship $X_{uv}$ ← could be $x_u x_v$
- Each user has a "position" in a *k*-dimensional latent space

unobserved, learn $L_u \in \mathbb{R}^k$, just as in matrix factorization

- Probability of link:

$$\text{log odds } p(r_{uv}=1 \mid L_u, L_v, X_{uv}, \beta)$$
$$= \log \frac{p(r_{uv}=1 \mid L_u, L_v, X_{uv}, \beta)}{p(r_{uv}=0 \mid L_u, L_v, X_{uv}, \beta)}$$
$$= \beta_0 + \beta^T X_{uv} - |L_u - L_v|$$
$$\text{OR}$$
$$= \beta_0 + \beta^T X_{uv} - |L_u' L_v|$$



2D example

## Probabilistic Latent Space Models

- Probability of link:

$$\text{log odds } p(r_{uv} = 1 \mid L_u, L_v, x_{uv}, \beta) = \beta_0 + \beta^T x_{uv} - |L_u - L_v|$$

prob. of link is high for $L_u$ close to $L_v$

$$\text{log odds } p(r_{uv} = 1 \mid L_u, L_v, x_{uv}, \beta) = \beta_0 + \beta^T x_{uv} + |L_u^T L_v|$$

can modify as $\frac{|L_u^T L_v|}{|L_v|}$

prob. of link is high if $\measuredangle$ bt $L_u$ + $L_v$ is small

- Bayesian approach:
  - Place prior on user factors and regression coefficients
  - Place hyperprior on user factor hyperparameters
- Many other options and extensions (e.g., can use GMM for $L_u$ → clustering of users in the latent space)



## What you need to know…

- Representation of network data as a matrix
  - Adjacency matrix
- Similarities and differences between adjacency matrices and general matrix-valued data
- Matrix factorization approaches for network data
  - Just use standard MF and threshold output
  - Introduce link functions to constrain predicted values
- Probabilistic latent space models
  - Model link probabilities using distance between latent factors