Case Study 4: Collaborative Filtering

Review: Cold Start Problem

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

Emily Fox

February 18th, 2014

Cold-Start Problem

- **Challenge:** Cold-start problem (new movie or user)
- **Methods:** use features of movie/user

Cold-Start Problem More Formally

- Consider a new user *u'* and predicting that user's ratings
  - No previous observations
  - Objective considered so far:

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

  - Optimal user factor:
  - Predicted user ratings:

An Alternative Formulation

- A simpler model for collaborative filtering
  - We would not have this issue if we assumed all users were identical
  - What about for new movies? What if we had side information?
  - What dimension should *w* be?
  - Fit linear model:
  - Minimize:

Personalization

- If we don't have any observations about a user, use wisdom of the crowd
  - **Address cold-start problem**

For user u', predict $r_{u'v} \approx w \cdot \phi(v)$

- Clearly, not all users are the same ... shared w is strong assumption
- Just as in personalized click prediction, consider model with global and user-specific parameters

Consider user-specific deviations w_u from the crowd w

$r_{uv} \approx (w + w_u) \cdot \phi(v)$ (w_u init. to 0)

- As we gain more information about the user, forget the crowd

w_u more informed

User Features…

- In addition to movie features, may have information about the user:

$\phi(u) = (25, F, MSc, A^+, \ldots)$
age, gender, education, grade in Big Data

- Combine with features of movie:

$\phi(u,v) = (\ldots, \phi(u), \ldots, \ldots, \phi(v), \ldots, \ldots$ cross features...$)$

- Unified linear model:

$r_{uv} \approx (w + w_u) \cdot \phi(u,v)$

Feature-based Approach versus Matrix Factorization

- Feature-based approach:
  - Feature representation of user and movies fixed
  - Can address cold-start problem
- Matrix factorization approach:
  - Suffers from cold-start problem
  - User & movie features are learned from data
- A unified model:

Unified Collaborative Filtering via SGD

$$\min_{L,R,w,\{w_u\}_u} \frac{1}{2}\sum_{r_{uv}}(L_u \cdot R_v + (w + w_u)\cdot\phi(u,v) - r_{uv})^2 + \frac{\lambda_u}{2}||L||_F^2 + \frac{\lambda_v}{2}||R||_F^2 + \frac{\lambda_w}{2}||w||_2^2 + \frac{\lambda_{wu}}{2}\sum_u ||w_u||_2^2$$

- Gradient step observing r_{uv}
  - For L,R $\begin{bmatrix} L_u^{(t+1)} \\ R_v^{(t+1)} \end{bmatrix} \leftarrow \begin{bmatrix} (1-\eta_t\lambda_u)L_u^{(t)} - \eta_t\epsilon_t R_v^{(t)} \\ (1-\eta_t\lambda_v)R_v^{(t)} - \eta_t\epsilon_t L_u^{(t)} \end{bmatrix}$
  - For w and w_u:

What you need to know…

- Cold-start problem
- Feature-based methods for collaborative filtering
  - Help address cold-start problem
- Unified approach

Case Study 4: Collaborative Filtering

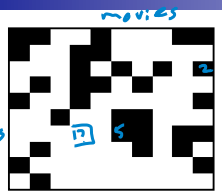
Connections with Probabilistic Matrix Factorization

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

Emily Fox

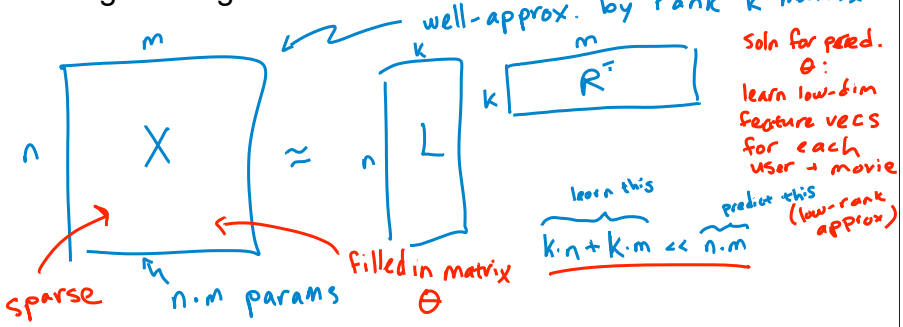
February 18th, 2014

Matrix Completion Problem



X_{ij} known for black cells
X_{ij} unknown for white cells
Rows index users
Columns index movies

- Filling missing data?



Coordinate Descent for Matrix Factorization: Alternating Least-Squares

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

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

$$\min_{L_u} \sum_{v \in V_u} (L_u \cdot R_v - r_{uv})^2 + \lambda_u \|L\|$$

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

$$\min_{R_v} \sum_{u \in U_v} (L_u \cdot R_v - r_{uv})^2 + \lambda_v \|R\|$$

- System may be underdetermined: use regularization
- Converges to local optima

Probabilistic Matrix Factorization (PMF)

- A generative process:
  - Pick user factors
  - Pick movie factors
  - For each (user,movie) pair observed:
    - Pick rating as $L_u R_v$ + noise
- Joint probability:

PMF Graphical Model

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

- Graphically:

Maximum A Posteriori for Matrix Completion

$$P(L,R|X) \propto P(L,R,X) = p(L)p(R)p(X \mid L,R)$$
$$\propto e^{\frac{-1}{2\sigma_u^2}\sum_{u=1}^{n}\sum_{i=1}^{k}L_{ui}^2} e^{\frac{-1}{2\sigma_v^2}\sum_{v=1}^{m}\sum_{i=1}^{k}R_{vi}^2} e^{\frac{-1}{2\sigma_r^2}\sum_{r_{uv}}(L_u \cdot R_v - r_{uv})^2}$$

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:

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

- Understanding as a probabilistic model is very useful! E.g.,
  - Change priors
  - Incorporate other sources of information or dependencies

What you need to know…

- Probabilistic model for collaborative filtering
  - Models, choice of priors
  - MAP equivalent to optimization for matrix completion

Case Study 4: Collaborative Filtering

Gibbs Sampling for Bayesian Inference

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

Emily Fox

February 18th, 2014

Posterior Computations

- MAP estimation focuses on point estimation:

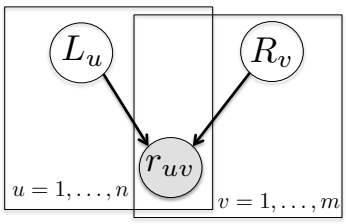
$$\hat{\theta}^{MAP} = \arg\max_{\theta} p(\theta \mid x)$$

- 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

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

Bayesian PMF Example

- Latent user and movie factors:



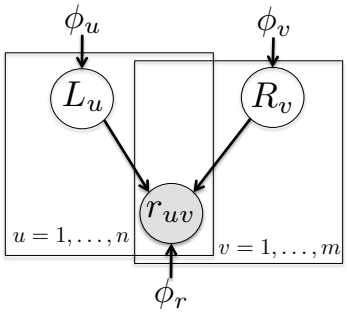
- Observations
- Hyperparameters:

- Want to predict new movie rating:

Bayesian PMF Example

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

- Monte Carlo methods:



- Ideally:

Bayesian PMF Example

- Want posterior samples $(L^{(k)}, R^{(k)}) \sim p(L, R \mid X, \phi)$
- What can we sample from?
  - Hint: Same reasoning as behind ALS, but sampling rather than maximization

Bayesian PMF Example

- For user u:

$$p(L_u \mid X, R, \phi_u) \propto p(L_u \mid \phi_u) \prod_{v \in V_u} p(r_{uv} \mid L_u, R_v, \phi_r)$$

- Symmetrically for R_v conditioned on L (breaks down over movies)
- Luckily, we can use this to get our desired posterior samples

Gibb Sampling

- Want draws:

- Construct Markov chain whose steady state distribution is
- Then, asymptotically correct
- Simplest case:

Bayesian PMF Gibbs Sampler

- Outline of Bayesian PMF sampler

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.

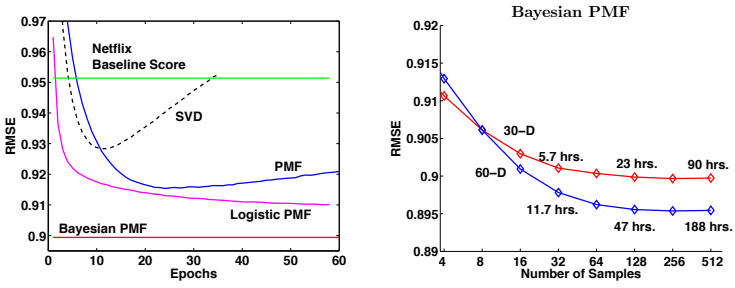


Figure 2. Left panel: Performance of SVD, PMF, logistic PMF, and Bayesian PMF using 30D feature vectors, on the Netflix validation data. The y-axis displays RMSE (root mean squared error), and the x-axis shows the number of epochs, or passes, through the entire training set. Right panel: RMSE for the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

D	Valid. RMSE		%	Test RMSE		%
	PMF	BPMF	Inc.	PMF	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

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

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

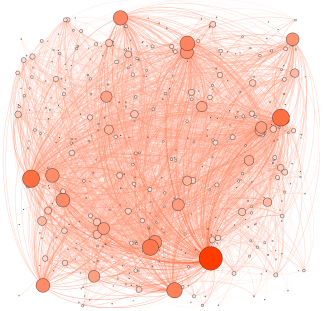
Machine Learning for Big Data
CSE547/STAT548, University of Washington

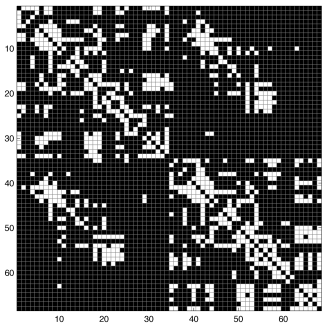
Emily Fox

February 18th, 2014

Network Data

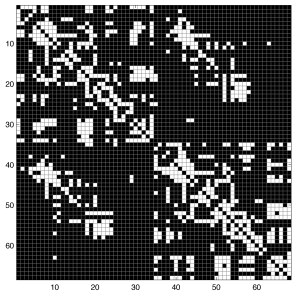
- Structure of network data





Properties of Data Source

- Similarities to Netflix data:
  - Matrix
  - High-dimensional
  - Sparse
- Differences
  - Square
  - Binary



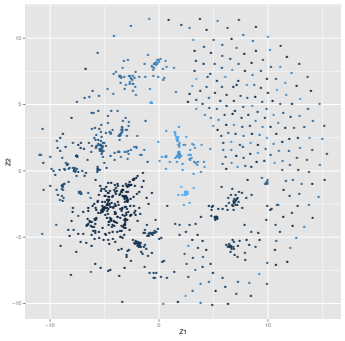
Matrix Factorization for Network Data

- Vanilla matrix factorization approach:
- What to return for link prediction?
- Slightly fancier:

Probabilistic Latent Space Models

- Assume features (covariates) of the user or relationship
- Each user has a "position" in a *k*-dimensional latent space

- Probability of link:



Probabilistic Latent Space Models

- Probability of link:

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

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

- 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