CMPT 884
Machine Learning in the Life Sciences

Martin Ester
Spring 2018

Probabilistic Graphical Models

Introduction

[Koller and Friedman 2009]

- A probabilistic graphical model (PGM) is a collection of random variables, a graph describing their conditional dependencies, and the parameters governing the corresponding probability distributions.
- The probabilistic approach explicitly models the real world uncertainty.
- A PGM models the joint probability distribution of the set of random variables.
- Two types of random variables: observed, latent.
- Values of the unobserved (latent) variables can be inferred, given the values of the observed variables.

Introduction

- Two main types of PGMs: directed and undirected (graphs).
- Examples of directed PGMs
  - Bayesian networks
  - Hidden Markov models
  - Probabilistic matrix factorization
- Examples of undirected PGMs
  - Markov random fields (MRF)
  - Conditional random fields (CRF)
- Parameters of the PGM are learnt from the training data so that the data likelihood or the a-posteriori probability of the parameters is maximized.

Bayesian Networks

Introduction

• Directed graph with node = *random variable* (attribute) and edge = *potential dependency.*

→ Graph is acyclic.

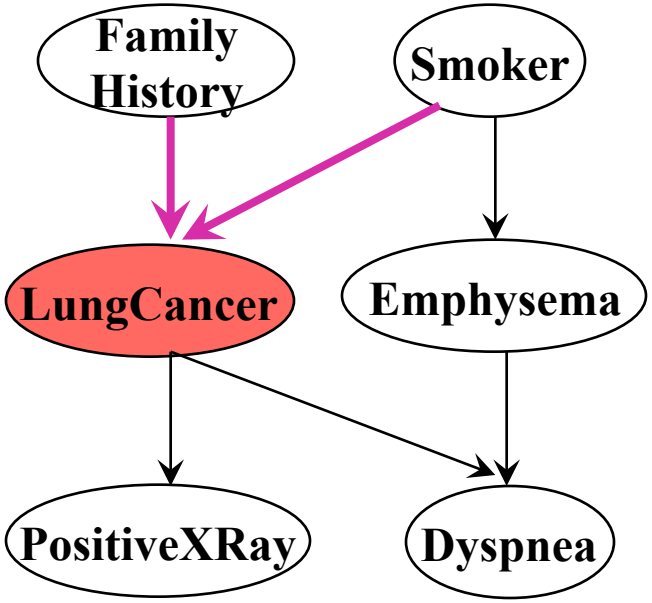
• Each random variable is, given its parent variables, conditionally independent from all non descendant variables.

• For each node (random variable): conditional probability distribution given values of the parent variables.

Bayesian network can represent *causal knowledge.*

Bayesian Networks

Example



	(FH,S)	(FH,~S)	(~FH,S)	(~FH,~S)
LC	0.8	0.5	0.7	0.1
~LC	0.2	0.5	0.3	0.9

Conditional probabilities for LungCancer

For given values of FamilyHistory and Smoker, the value of Emhysema does not provide any additional information about LungCancer.

Bayesian Networks

Training Bayesian Networks

- With given network structure and fully observable random variables
  - all attribute values of the training examples known,
  - estimate parameters of the conditional probability distributions by the relative frequencies.
- With given network structure and partially known random variables
  - some attribute values of the training examples unknown,
  - expectation maximization (EM) algorithm to infer unknown values.

Bayesian Networks

Training Bayesian Networks

- Expectation maximization (EM) algorithm

```
Initialize the latent variables and the parameters;
  Iterate until convergence
    E-step: compute expectation of latent vars
            given the current parameter values
    M-step: determine maximum likelihood parameters
            given the values of latent variables
```

Bayesian Networks

Training Bayesian Networks

- With apriori unknown network structure

  - assume fully observable random variables,
  - heuristic scoring functions for alternative network structures,
  - too complex network structures (too many edges) lead to overfitting and are penalized.

Matrix Factorization

Introduction [Koren et al. 2009]

- Given N users and M items, and numerical rating values.
- $R_{i,j}$ rating of user i for item j.
- Goal is to learn latent feature (factor) matrices for users $U \in R^{D \times N}$ and for items $V \in R^{D \times M}$ such that $R \sim U^T V$.
- $D << N, D << M$.

→ Dimensionality reduction

Low-rank approximation

Matrix Factorization

Introduction

- Matrix *R* is typically very sparse, i.e. most ratings are not observed.
- Task 1: prediction of missing/unobserved ratings.
- Task 2: cluster users (and items) into groups such that cluster elements have similar ratings.
  → users/items for which *i* is the largest factor belong to cluster *i*

Matrix Factorization

Methods

- Many different methods.
- Non-negative matrix factorization (NMF)

  factors cannot take on negative values.
- Probabilistic Latent Semantic Analysis (PLSA)

  assumes probabilistic generative model,
  factor vector is probability distribution.
- Probabilistic matrix factorization (PMF)

  assumes probabilistic generative model,
  ratings are generated probabilistically from corresponding user and item factors.

Probabilistic Matrix Factorization

Introduction [Salakhutdinov and Mnih 2007]

- A probabilistic graphical model.
- Solid statistical foundation.
- Employ wealth of existing parameter learning and inference methods.
- Algorithms scale linearly with the number of observations.
- Accurate prediction for sparse and imbalanced datasets.

Probabilistic Matrix Factorization

Graphical Model

- Assume that ratings are generated from a linear probabilistic model with Gaussian observation noise:

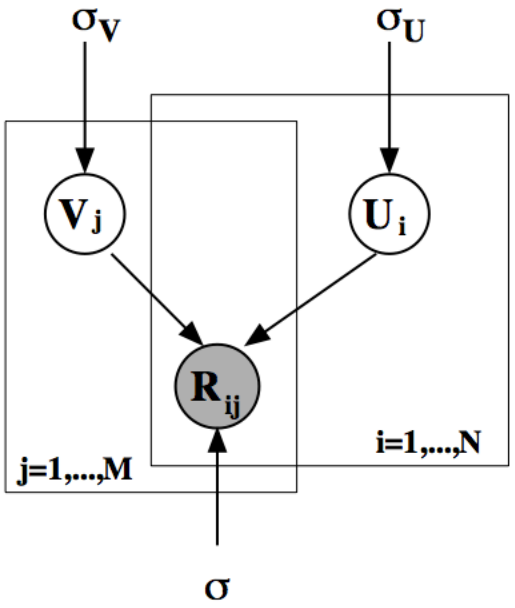
$$p(R \mid U, V, \sigma^2) = \prod_{i=1}^{N} \prod_{j=1}^{M} [N(R_{ij} \mid U_i^T V_j, \sigma^2)]^{I_{ij}}$$

- $N(x \mid \mu, \sigma^2)$: probability density function of Gaussian distribution with mean μ and variance σ^2
- I_{ij} indicator function: = 1 if user *i* has rated item *j*, otherwise = 0.
- Zero-mean spherical Gaussian priors on feature vectors

$$p(U \mid \sigma_U^2) = \prod_{i=1}^{N} N(U_i \mid 0, \sigma_U^2 I), \quad p(V \mid \sigma_V^2) = \prod_{i=1}^{N} N(U_i \mid 0, \sigma_U^2 I),$$

Probabilistic Matrix Factorization

Graphical Model



Hyperparameters: $\sigma^2, \sigma_U^2, \sigma_V^2$
typically provided as user input

Parameters: U, V, learnt from data

Data: R_{ij}, observed

Probabilistic Matrix Factorization

Parameter Learning

- Learn U,V with maximum posterior probability (MAP) conditioned on the observed data and given hyperparameters

$$\underset{U,V}{\text{argmax}}\ p(U,V \mid R,\sigma^2,\sigma_U^2,\sigma_V^2)$$

$$= \underset{U,V}{\text{argmax}}\ p(U,V,R,\sigma^2,\sigma_U^2,\sigma_V^2)$$

$$= \underset{U,V}{\text{argmax}}\ p(R \mid U,V,\sigma^2)p(U \mid \sigma_U^2)p(V \mid \sigma_V^2)$$

$$= \underset{U,V}{\text{argmax}} \prod_{i=1}^{N}\prod_{j=1}^{M}[N(R_{ij} \mid U_i^T V_j,\sigma^2)]^{I_{ij}}$$

$$\prod_{i=1}^{N} N(U_i \mid 0,\sigma_U^2 I)\prod_{j=1}^{M} N(V_j \mid 0,\sigma_V^2 I)$$

Probabilistic Matrix Factorization

Parameter Learning

- Substituting the Gaussian probability density functions, and taking the log of the posterior, we obtain the objective function

$$\underset{U,V}{\text{argmax}} \; -\frac{1}{2\sigma^2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2$$

$$-\frac{1}{2\sigma_U^2}\sum_{i=1}^{N} U_i^T U_i - \frac{1}{2\sigma_V^2}\sum_{j=1}^{M} V_j^T V_i$$

Probabilistic Matrix Factorization

Parameter Learning

- Maximizing the log posterior is equivalent to minimizing the squared error with quadratic regularization term:

$$\underset{U,V}{\operatorname{argmin}} \; E =$$

$$\underset{U,V}{\operatorname{argmin}} \; \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}(R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N}\|U_i\|^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M}\|V_j\|^2$$

$$\lambda_U = \frac{\sigma^2}{\sigma_U^2},\ \lambda_V = \frac{\sigma^2}{\sigma_V^2},\ \|\ \|^2 \ \textit{Frobenius norm}$$

- Minimization of E through gradient descent in U and V.

Probabilistic Matrix Factorization

Complexity Control

- The complexity of the PMF model is controlled through the hyper-perameters $\sigma^2, \sigma_U^2, \sigma_V^2$
- How to set these hyper-parameters?
- Manual approach

  - Determine a set of reasonable values of hyper-parameters,

  - train a model for each setting of hyper-parameters, and

  - choose the model that performs best on a validation set.
- Drawback: computationally very expensive.
- Automatic approach
  Introduce priors for hyper-parameters, and learn hyper-parameters simultaneously with parameters.

Probabilistic Matrix Factorization

Automatic Complexity Control

- Objective function

$$\underset{U,V,\sigma^2,\sigma_U^2,\sigma_V^2}{\text{argmax}} \; p(U,V,\sigma^2,\sigma_U^2,\sigma_V^2 \mid R)$$

$$= \underset{U,V,\sigma^2,\sigma_U^2,\sigma_V^2}{\text{argmax}} \; p(U,V,R,\sigma^2,\sigma_U^2,\sigma_V^2)$$

$$= \underset{U,V,\sigma^2,\sigma_U^2,\sigma_V^2}{\text{argmax}} \; p(R \mid U,V,\sigma^2)p(U \mid \sigma_U^2)p(V \mid \sigma_V^2)p(\sigma^2)p(\sigma_U^2)(\sigma_V^2)$$

$$= \underset{U,V,\sigma^2,\sigma_U^2,\sigma_V^2}{\text{argmax}} \; \ln p(R \mid U,V,\sigma^2) + \ln p(U \mid \sigma_U^2) + \ln p(V \mid \sigma_V^2)$$

$$+ \ln p(\sigma^2) + \ln p(\sigma_U^2) + \ln p(\sigma_V^2)$$

Probabilistic Matrix Factorization

Automatic Complexity Control

- Optimize objective function by alternating the following steps:
  - optimize hyper-parameters given the current parameters,
    If the prior is Gaussian, there is a closed form solution.
  - optimize parameters given the current hyper-parameters.
    Apply gradient ascent.
- Priors for hyper-parameters
  Simplest choice: Gaussian priors.
  Zero-mean, spherical prior.
- But can also choose diagonal matrices or even full covariance matrices.

Bayesian PMF

Introduction [Salakhutdinov and Mnih 2008]

- For rating prediction, we are not interested in the values of model parameters and hyper-parameters.
- Bayesian approach: do not learn one "best" value (point estimate) of the parameters and hyper-parameters, but consider all possible values weighted by their probability.

→ integrate out model parameters and hyper-parameters

Bayesian PMF

Graphical Model

- Assume that the factors are generated from a Gaussian distribution with mean and variance given by the hyper-parameters:

$$p(U \mid \mu_U, \Lambda_U) = \prod_{i=1}^{N} N(U_i \mid \mu_U, \Lambda_U^{-1})$$

- The hyper-parameters are generated from a Gaussian and from a Wishart distribution with priors μ_0, υ_0, W_0 :

$$p(\mu_U \mid \mu_0, \Lambda_U) = N(\mu_U \mid \mu_0, (\beta_0 \Lambda_U)^{-1}),$$

$$p(\Lambda_U \mid \upsilon_0, W_0) = W(\Lambda_U \mid W_0, \upsilon_0)$$

where W is a Wishart distribution with v_0 degrees of freedom and a DxD scale matrix W_0.

Bayesian PMF

Graphical Model

- The Wishart distribution is the conjugate prior to the precision matrix Λ, the inverse of the covariance matrix of a multi-variate Normal distribution.

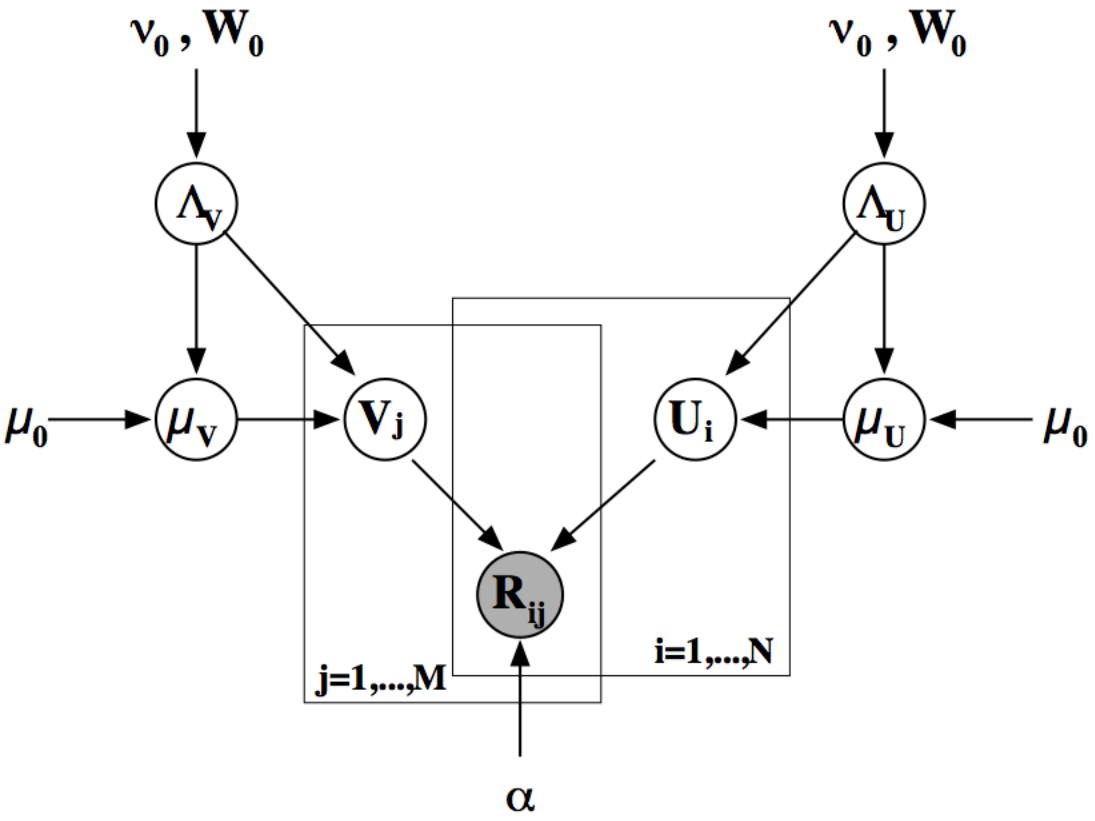
$$W(\Lambda \mid W_0, \upsilon_0) = \frac{1}{C}|\Lambda|^{(\upsilon_0 - D - 1)/2} \exp(-\frac{1}{2} Tr(W_0^{-1}\Lambda))$$

- Typical settings of the priors:

$$\mu_0 = 0, \ \upsilon_0 = D, W_0 = \mathrm{I}.$$

Bayesian PMF

Graphical Model



Priors

Hyper-parameters

Parameters

Data

Bayesian PMF

Predictions

- Predictive distribution of the unobserved rating of user *i* for item *j*:

$$p(R_{ij}^* \mid R, \Theta_0) = \iint p(R_{ij}^* \mid U_i, V_j)\, p(U, V \mid R, \Theta_U, \Theta_V)$$
$$p(\Theta_U, \Theta_V \mid \Theta_0)\, d\{U, V\}\, d\{\Theta_U, \Theta_V\}$$

where $\Theta_0 = \{\mu_0, \upsilon_0, W_0\}$, $\Theta_U = \{\mu_U, \Lambda_U\}$, $\Theta_V = \{\mu_V, \Lambda_V\}$.

- Exact evaluation is analytically intractable.
- Need approximate inference.
- MCMC-based approach

$$p(R_{ij}^* \mid R, \Theta_0) \approx \frac{1}{K} \sum_{k=1}^{K} p(R_{ij}^* \mid U_i^{(k)}, V_j^{(k)})$$

Bayesian PMF

Inference

- Samples $\{U_i^k, V_j^k\}$ are generated from a Markov chain with stationary distribution

$$p(U, V, \Theta_U, \Theta_V \mid R, \Theta_0).$$

- Asymptotically exact results.
- Gibbs Sampling
  Simple MCMC method that iterates over the random variables, sampling one of them given the current values of all others.
- Due to use of conjugate priors, conditional distributions are easy to sample from.
- Sample hyper-parameters, given parameters.
- Sample parameters, given hyper-parameters.

Bayesian PMF

Inference

- Sample U_i $\quad p(U_i \mid V, \Theta_U, \Theta_V, R, \Theta_0)$

$$\sim \prod_{j=1}^{M} [N(R_{ij} \mid U_i^T V_j, \alpha^{-1})]^{I_{ij}} \; p(U_i \mid \mu_U, \Lambda_U)$$

$$= N(U_i \mid \mu_i^*, [\Lambda_i^*]^{-1})$$

where $\quad \Lambda_i^* = \Lambda_U + \alpha \sum_{j=1}^{M} [V_j V_j^T]^{I_{ij}}$

$$\mu_i^* = [\Lambda_i^*]^{-1} \left(\alpha \sum_{j=1}^{M} [V_j R_{ij}]^{I_{ij}} + \Lambda_U \mu_U \right)$$

Bayesian PMF

Inference

- Sample Θ_U

$$p(\Theta_U \mid U, V, \Theta_V R, \Theta_0)$$

$$= p(\Theta_U \mid U, \Theta_0)$$

$$= N(\mu_U \mid \mu_0^*, (\beta_0^* \Lambda_U)^{-1}) W(\Lambda_U \mid W_0^*, \upsilon_0^*)$$

where $\mu_0^* = \frac{\beta_0 \mu_0 + N\bar{U}}{\beta_0 + N}, \ \beta_0^* = \beta_0 + N, \ \upsilon_0^* = \upsilon_0 + N$

$$[W_0^*]^{-1} = W_0^{-1} + N\bar{S} + \frac{\beta_0 N}{\beta_0 + N}(\mu_0 - \bar{U})(\mu_0 - \bar{U})^T$$

$$\bar{U} = \frac{1}{N}\sum_{i=1}^{N} U_i, \ \bar{S} = \sum_{i=1}^{N}(U_i - \bar{U})(U_i - \bar{U})^T$$

Bayesian PMF

Gibbs Sampling for Bayesian PMF

1. Initialize latent factors U^1, V^1
2. For $k = 1, \ldots, K$

- Sample the hyper-parameters

$$\Theta_U^k \sim p(\Theta_U \mid U^k, \Theta_0)$$

$$\Theta_V^k \sim p(\Theta_V \mid V^k, \Theta_0)$$

- For $i = 1, \ldots, N$ sample user factors

$$U_i^{k+1} \sim p(U_i \mid V^k, \Theta_U^k, R)$$

- For $j = 1, \ldots, M$ sample item factors

$$V_j^{k+1} \sim p(V_j \mid U^{k+1}, \Theta_V^k, R)$$

Bayesian PMF

Experimental Evaluation

- Netflix dataset
  100M ratings, 480K users, 17700 movies
- Evaluation metric

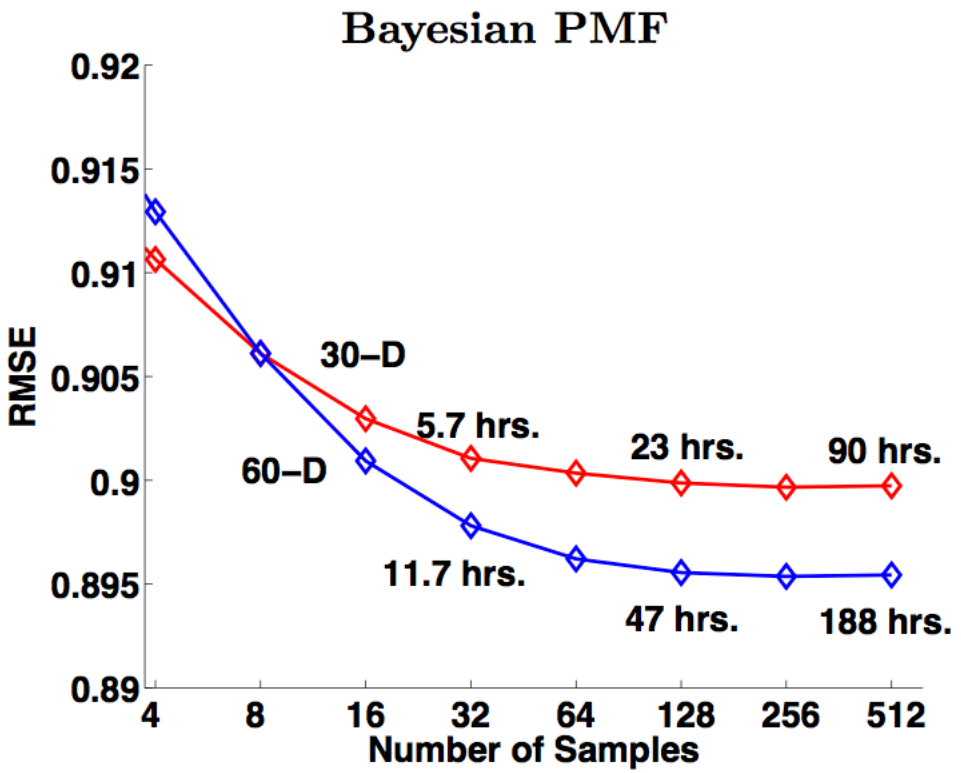
  Root Mean Square Error (RMSE)
- Comparison partners
  SVD
    no regularization

  (linear) PMF
  logistic PMF
    apply logistic function to the product of the latent factors

  Bayesian PMF

Bayesian PMF

Experimental Evaluation



Bayesian PMF

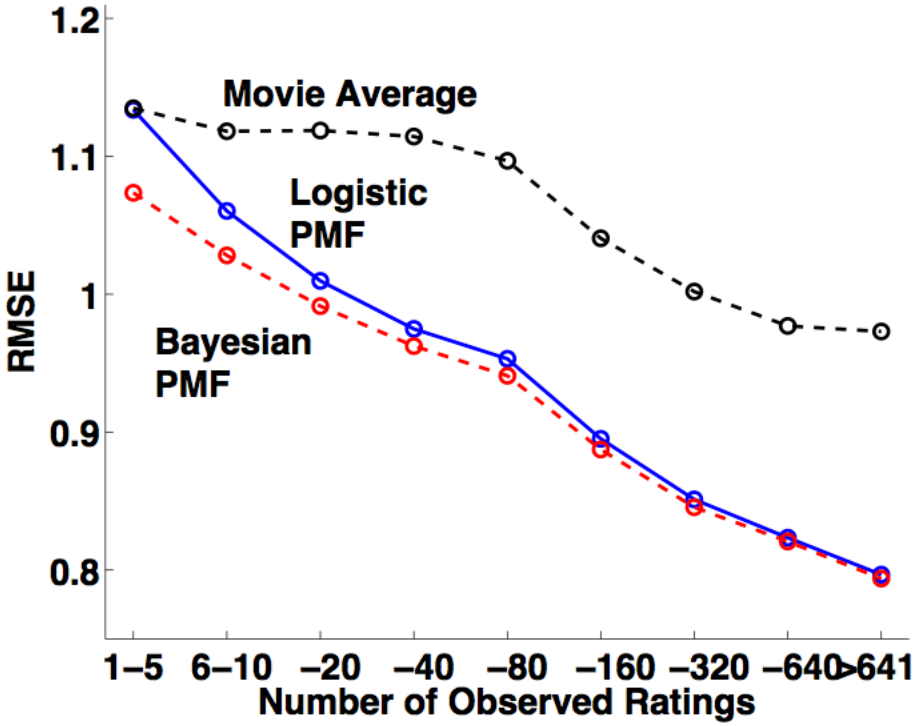
Experimental Evaluation

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

→ Bayesian PMF achieves significant gain in predictive accuracy

Bayesian PMF

Experimental Evaluation



→ Accuracy gains are larger for users with few observed ratings

Probabilistic Matrix Factorization

Discussion

- Probabilistic graphical model with well-established methods for inference and parameter learning.
- MAP approach
  - point estimates of parameters
  - efficient
  - can also learn hyper-parameters
- Bayesian approach
  - infer full posterior distribution of parameters and hyper-parameters
  - computationally (much) more expensive
  - infer predictive distribution instead of only one rating value
- Methods scale to large datasets

References

[Koller and Friedman 2009]
Daphne Koller, Nir Friedman: Probabilistic Graphical Models, MIT Press, 2009.

[Koren et al. 2009]
Yehuda Koren, Robert M. Bell, Chris Volinsky: Matrix Factorization Techniques for Recommender Systems. IEEE Computer 42 (8): 30-37 (2009)

[Salakhutdinov and Mnih 2007]
Ruslan Salakhutdinov, Andriy Mnih: Probabilistic Matrix Factorization.NIPS 2007: 1257-1264

[Salakhutdinov and Mnih 2008]
Ruslan Salakhutdinov, Andriy Mnih: Bayesian probabilistic matrix factorization using Markov chain Monte Carlo. ICML 2008: 880-887