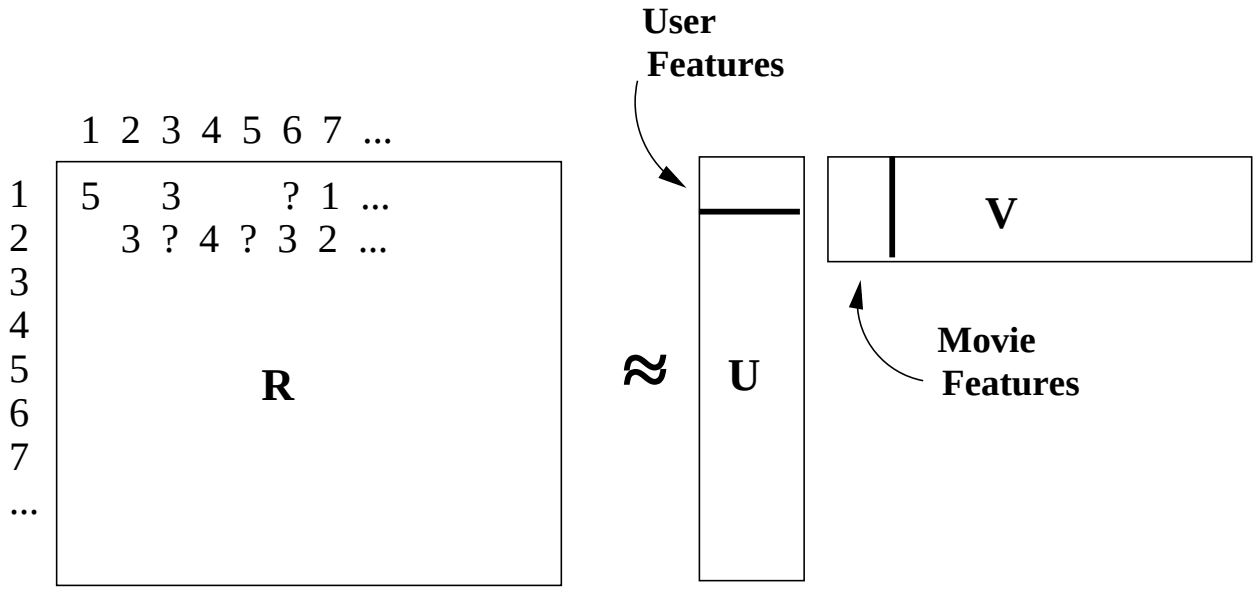
Bayesian Probabilistic Matrix Factorization using MCMC

Ruslan Salakhutdinov

joint work Andriy Mnih

Machine Learning, University of Toronto

Preliminaries



- Suppose we have M movies, N users, and integer rating values from 1 to K.

- Let R_{ij} be the rating of user i for movie j, and $U \in R^{D \times N}$, $V \in R^{D \times M}$ be latent user and movie feature matrices.

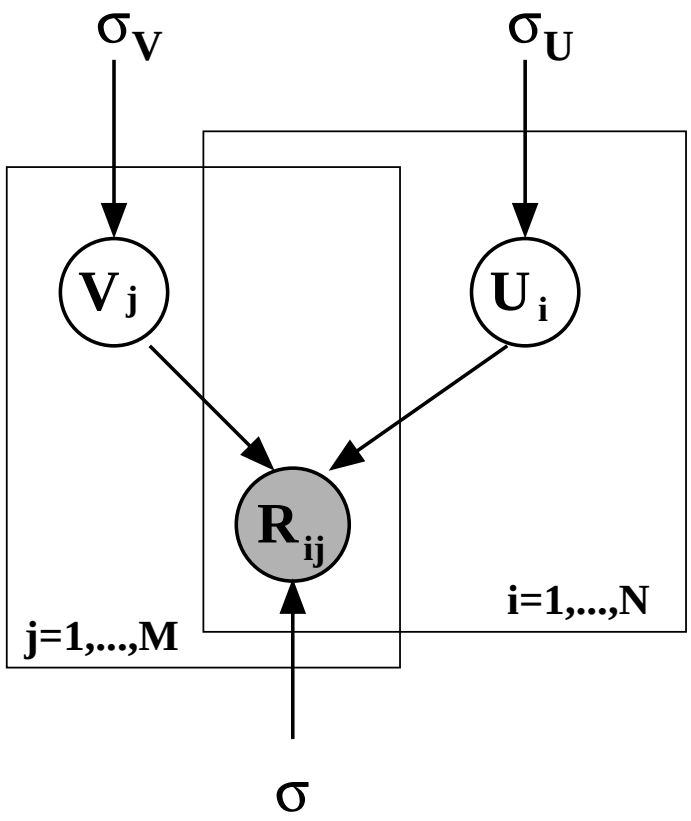
- We will use U_i and V_j to denote the latent feature vectors for user i and movie j respectively.

Probabilistic Matrix Factorization (PMF)

- PMF is a simple probabilistic linear model with Gaussian observation noise.

- Given the feature vectors for the user and the movie, the distribution of the corresponding rating is:

$$p(R_{ij}|U_i, V_j, \sigma^2) = \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2).$$

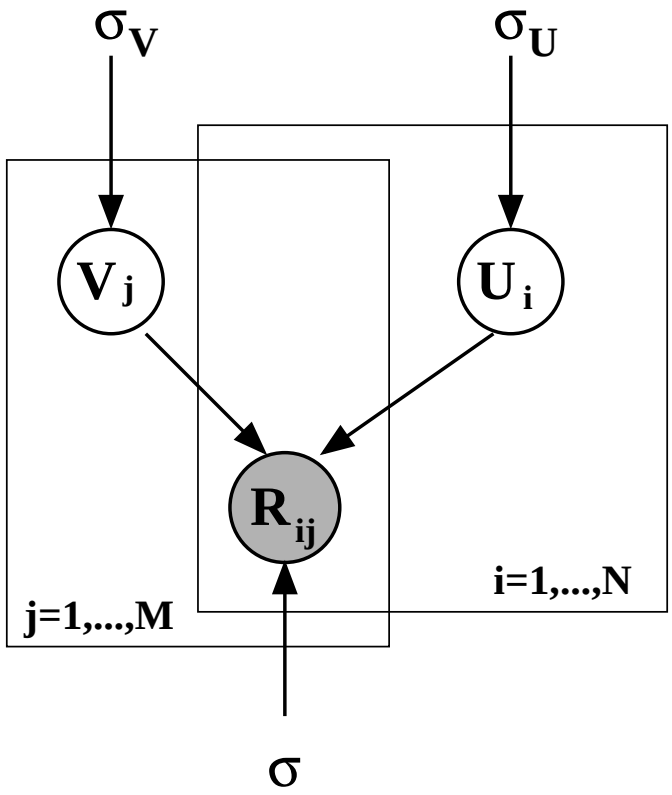


- The user and movie feature vectors are given zero-mean spherical Gaussian priors:

$$p(U|\sigma_U^2) = \prod_{i=1}^{N} \mathcal{N}(U_i|0, \sigma_U^2 \mathbf{I}), \quad p(V|\sigma_V^2) = \prod_{j=1}^{M} \mathcal{N}(V_j|0, \sigma_V^2 \mathbf{I}).$$

Learning (I)

• MAP Learning: Maximize the log-posterior over movie and user features with fixed hyperparameters.



• Equivalent to minimizing the sum-of-squared-errors with quadratic regularization terms:

$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij} - U_i^T V_j\right)^2 + \\ + \frac{\lambda_U}{2}\sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \| V_j \|_{Fro}^2,$$

$\lambda_U = \sigma^2/\sigma_U^2$, $\lambda_V = \sigma^2/\sigma_V^2$, and $I_{ij} = 1$ if user i rated movie j and is 0 otherwise.

Learning (II)

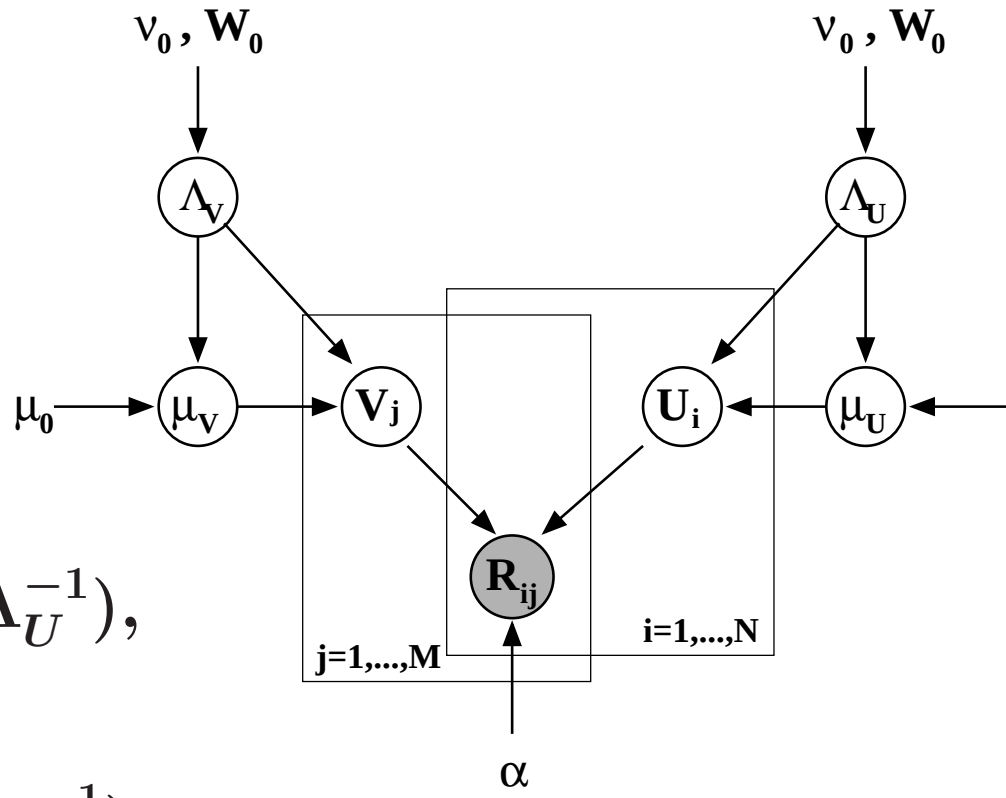
$$E = \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{M} I_{ij}\left(R_{ij} - U_i^T V_j\right)^2 + \frac{\lambda_U}{2}\sum_{i=1}^{N} \| U_i \|_{Fro}^2 + \frac{\lambda_V}{2}\sum_{j=1}^{M} \| V_j \|_{Fro}^2 .$$

- Find local minimum by gradient descent in U and V. Efficient and easy to implement.
- Main drawback is complexity control, which is essential to making the model generalize well.
- Usually interested in predicting ratings for new user/movie pairs, not in estimating model parameters.

Bayesian Probabilistic Matrix Factorization

- Likelihood:

$$p(R_{ij}|U_i, V_j, \sigma^2) = \mathcal{N}(R_{ij}|U_i^T V_j, \sigma^2).$$



- Gaussian Priors:

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

$$p(V|\mu_V, \Lambda_V) = \prod_{i=1}^{M} \mathcal{N}(V_i|\mu_V, \Lambda_V^{-1}).$$

- Gaussian-Wishart priors on the user and movie hyperparameters $\Theta_U = \{\mu_U, \Lambda_U\}$ and $\Theta_V = \{\mu_V, \Lambda_V\}$.

Predictions

- Obtain predictive distribution of R^*_{ij}, given observed ratings R.

- This requires marginalization over the model parameters $\{U, V\}$ and hyperparameters $\{\Theta_U, \Theta_V\}$.

- Monte Carlo approximation:

$$p(R^*_{ij}|R) \approx \frac{1}{K} \sum_{k=1}^{K} p(R^*_{ij}|U_i^{(k)}, V_j^{(k)}).$$

- Samples $\{U_i^{(k)}, V_j^{(k)}\}$ are generated by a Markov chain, whose stationary distribution is the posterior distribution over the model parameters.

Inference

- We use Gibbs sampling to generate the samples.
- Due to the use of conjugate priors, the conditional distributions are easy to sample from.
- The posterior distributions of the user and movie latent feature matrices $\boldsymbol{U}$ and $\boldsymbol{V}$ factorize:

$$\boldsymbol{p}(\boldsymbol{U}|\boldsymbol{R}, \boldsymbol{V}, \boldsymbol{\Theta}_U) = \prod_{\boldsymbol{i}=\boldsymbol{1}}^{\boldsymbol{N}} \boldsymbol{p}(\boldsymbol{U}_i|\boldsymbol{R}, \boldsymbol{V}, \boldsymbol{\Theta}_U),$$

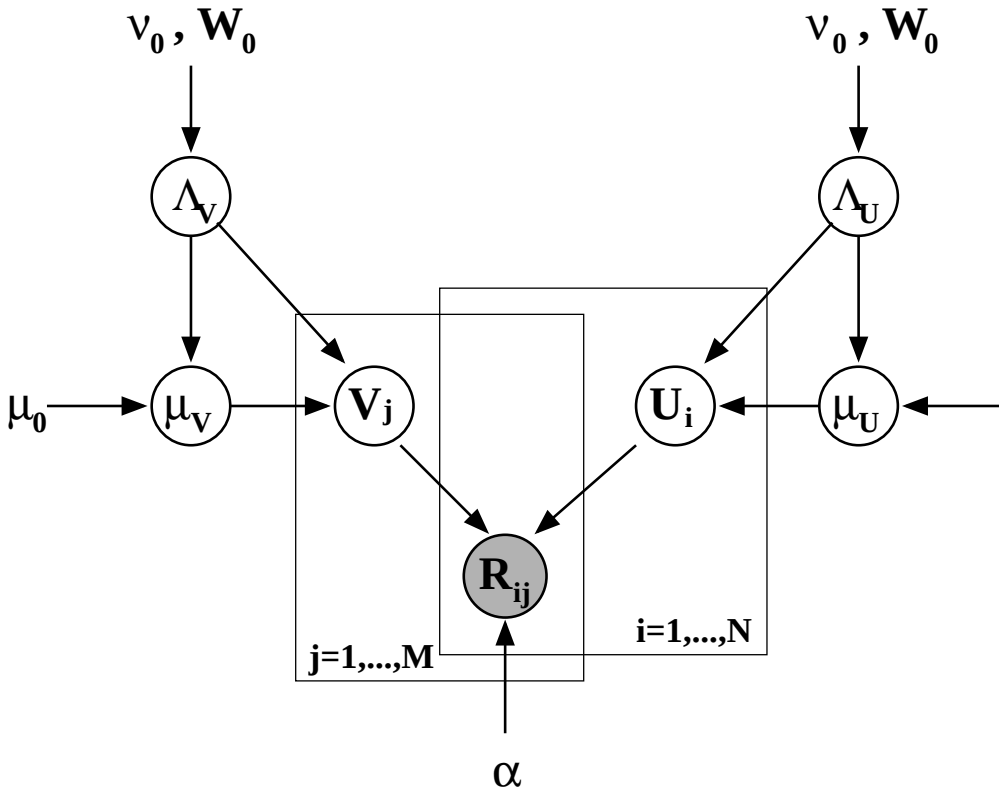
$$\boldsymbol{p}(\boldsymbol{V}|\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{\Theta}_U) = \prod_{\boldsymbol{j}=\boldsymbol{1}}^{\boldsymbol{M}} \boldsymbol{p}(\boldsymbol{V}_i|\boldsymbol{R}, \boldsymbol{U}, \boldsymbol{\Theta}_{\boldsymbol{V}}).$$

- We can speed up the sampler by sampling the feature vectors for different users/movies in parallel.

Gibbs Sampler

- Sample the hyperparameters:

$$\Theta_U^{t+1} \sim p(\Theta_U | U^t),$$
$$\Theta_V^{t+1} \sim p(\Theta_V | V^t).$$



- Sample the feature vectors of users $i = 1, .., N$ in parallel:

$$U_i^{t+1} \sim p(U_i | R, V^t, \Theta_U^{t+1}).$$

- Sample the feature vectors of movies $j = 1, .., M$ in parallel:

$$V_i^{t+1} \sim p(V_i | R, U^{t+1}, \Theta_V^{t+1}).$$

The Netflix Dataset

- The Netflix dataset is large, sparse, and imbalanced.
- The training set: 100,480,507 ratings from 480,189 users on 17,770 movies.
- The validation set: 1,408,395 ratings. The test set: 2,817,131 user/movie pairs with ratings withheld.
- The dataset is very imbalanced. The number of ratings entered by a user ranges from 1 to over 15000.
- Performance is assessed by submitting predictions to Netflix, which prevents accidental cheating since the test answers are known only to Netflix.

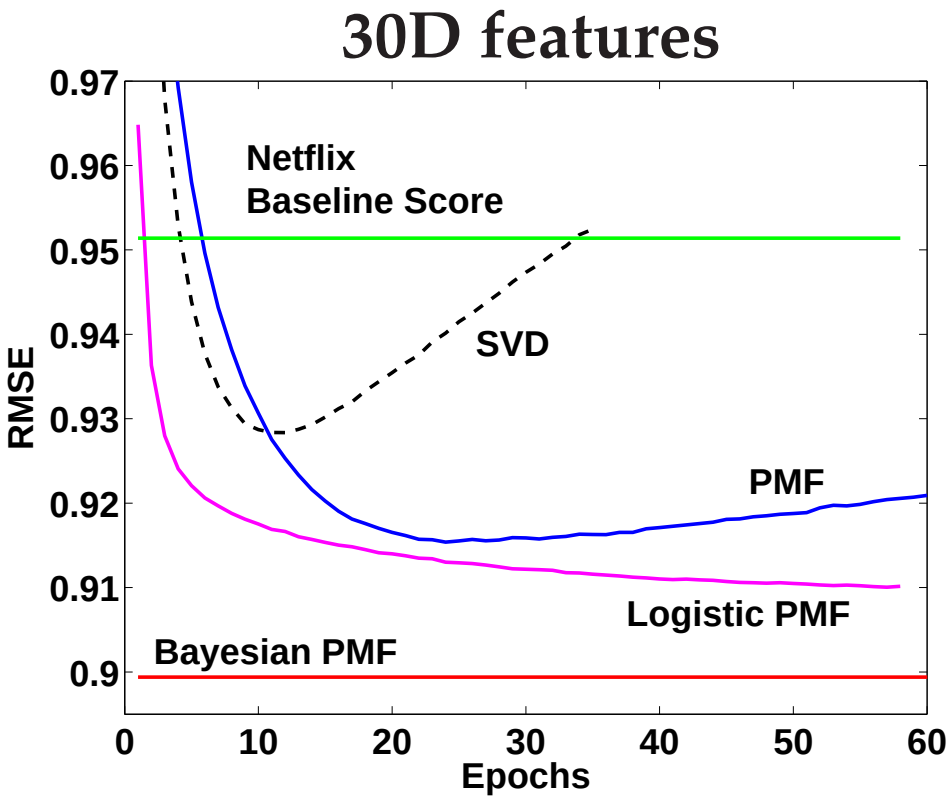
Bayesian PMF

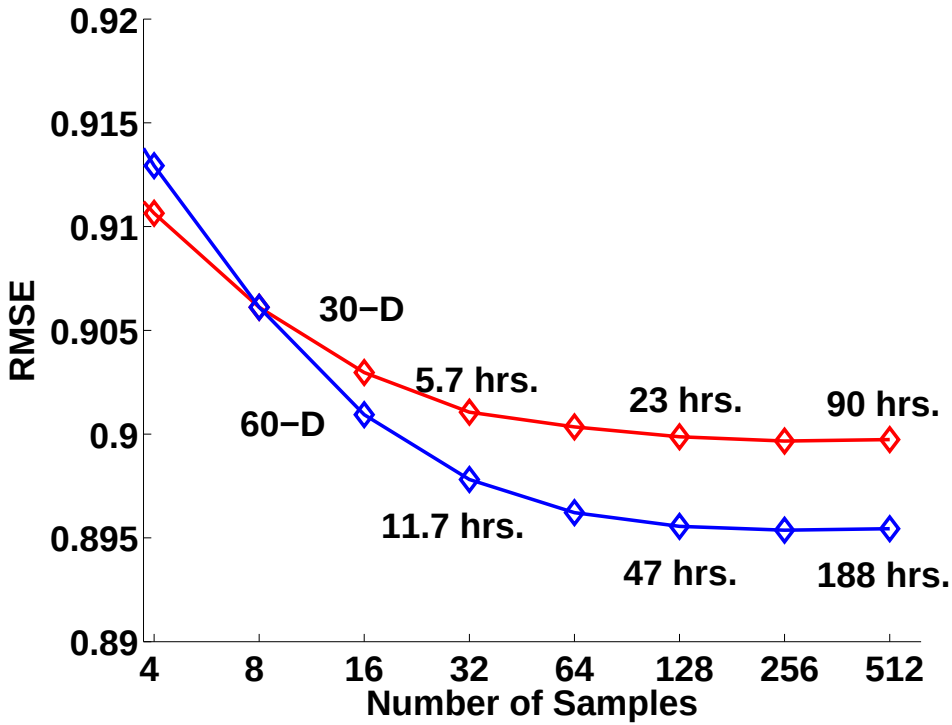
- PMF models are trained efficiently by finding point estimates of model parameters and hyperparameters.

- Does a fully Bayesian approach seem computationally feasible on 100 million ratings, with 0.5 million users, and 18,000 movies?

- MCMC methods are rarely used on large-scale problems. They are perceived to be very slow by practitioners.

Experimental Results





- Performance of SVD, MAP-trained PMF, and Bayesian PMF using 30D feature vectors on the full Netflix validation set.

Experimental Results

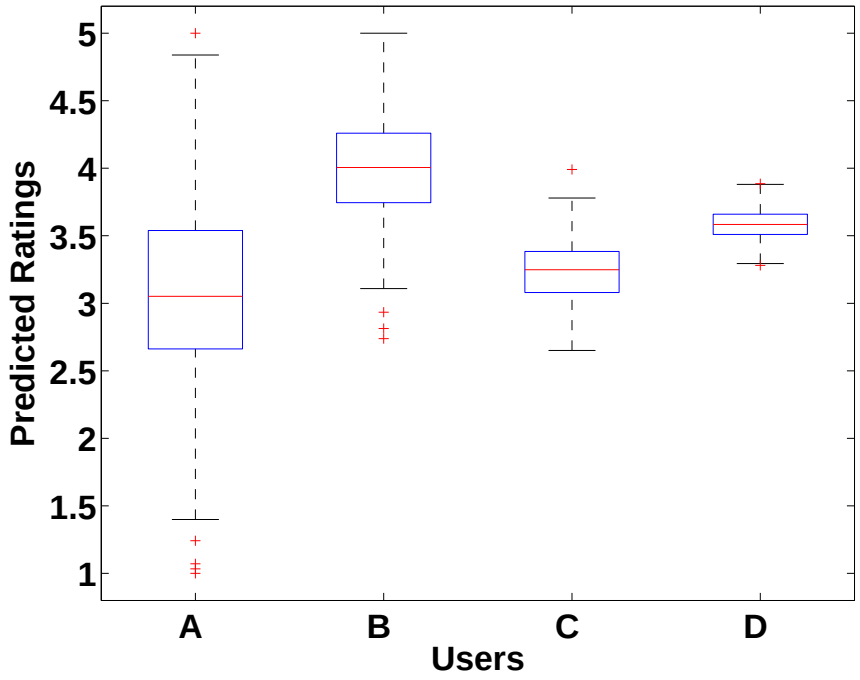
- Performance on the Netflix *test set*.

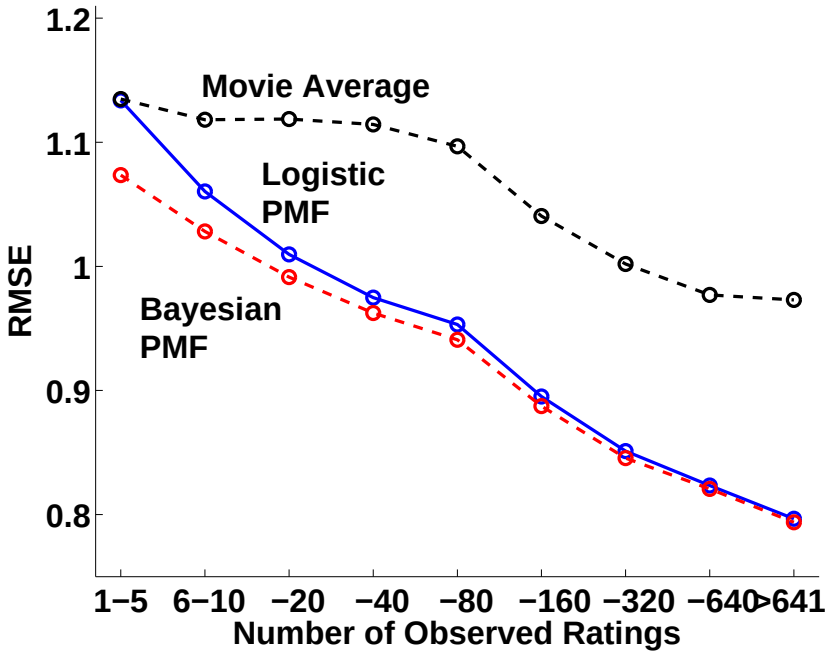
Feature Dim.	PMF	Bayesian PMF	Reduction in %
60	0.9185	0.8989	2.13
150	0.9211	0.8965	2.67
300	0.9265	0.8954	3.36

- Bayesian PMF models significantly outperform their MAP counterparts.
- The predictive accuracy of Bayesian PMF models improves as the model complexity grows.
- The Bayesian approach does not require limiting the complexity of the model based on the training set size.

Experimental Results

- Bayesian PMF models deal with uncertainty more effectively than their non-Bayesian counterparts.





- Right: Predictions on randomly chosen test movies by users who have rated 4, 23, 319, and 660 movies.
- Left: Performance comparison between Bayesian PMF, Logistic PMF, and the movie average algorithm that predicts the average rating of each movie.

Conclusions

- Bayesian PMF models can be successfully applied to a large dataset containing over 100 million movie ratings.
- They achieve significantly higher predictive accuracy than the MAP-trained models.
- They provide a predictive distribution, allowing the prediction confidence to be taken into account when making recommendations.
- One drawback of using MCMC for training Bayesian PMF models is that it is hard to determine when the Markov chain has converged to its equilibrium distribution.

The End