

PAPER PRESENTATION:

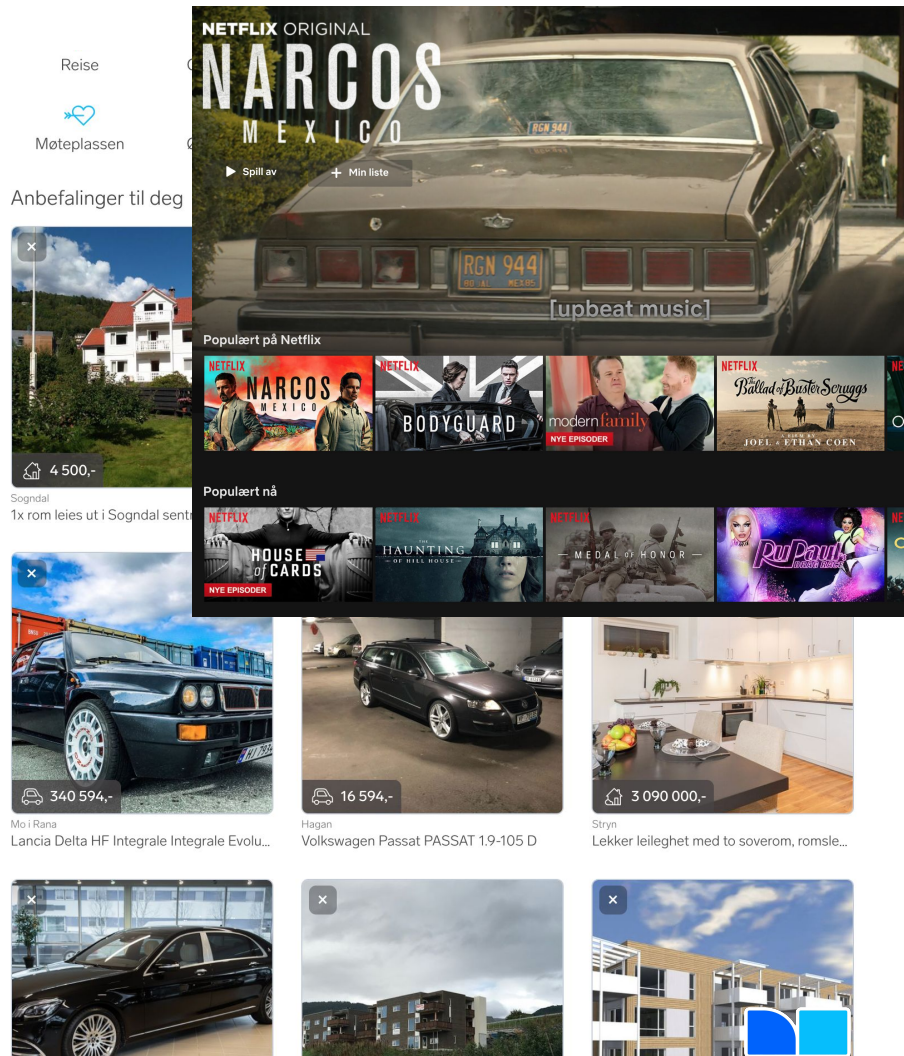
BAYESIAN PROBABILISTIC MATRIX FACTORIZATION USING MARKOV CHAIN MONTE CARLO

**Paper by: Salakhutdinov, Mnih
Conference on Machine Learning, 2008.**

Simen Eide

MOTIVATION

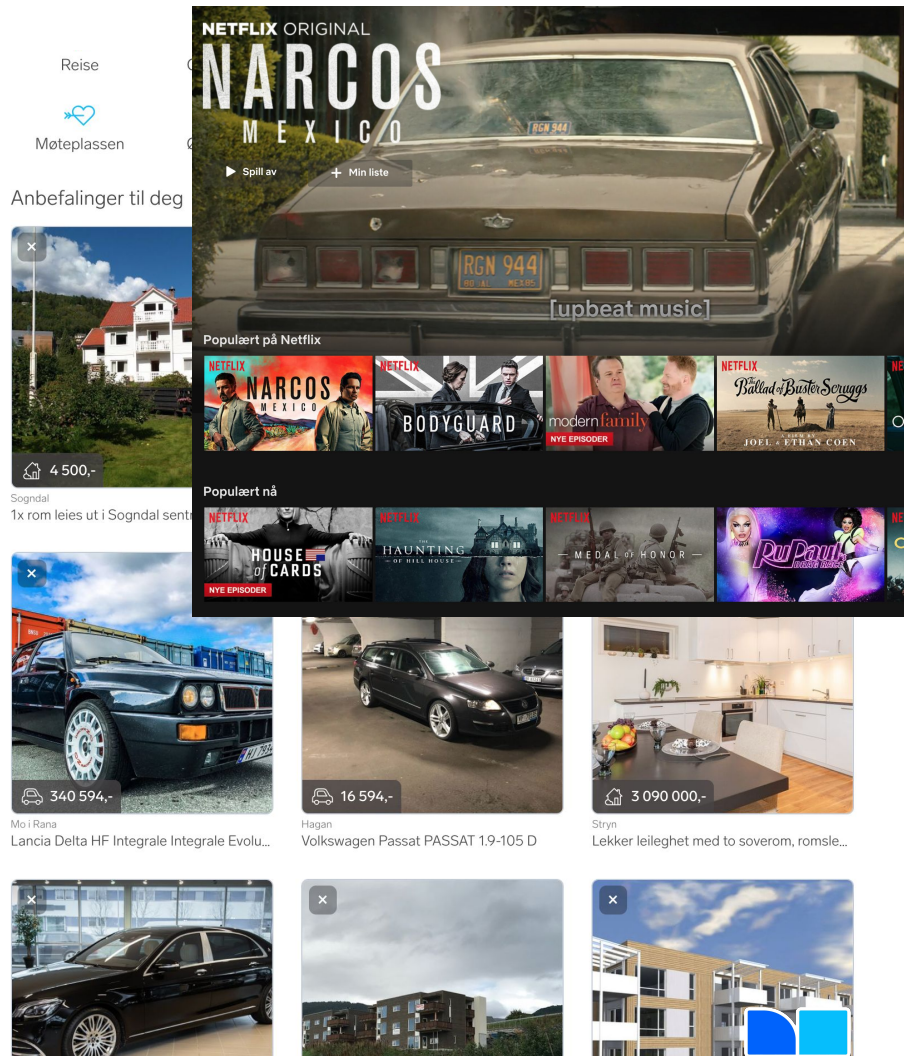
- There is too many choices for people to make
- Help users
- Help advertisers
- 30% amazon sales
- Facebook feed
- Hard problem:
 - Millions of classes
 - Little data per user
- Creepy and stupid?



MOTIVATION

- Content vs collaborative
 - Image, text, tags, gender, age
- Collaborative is main driver
 - Explicit
 - Users rate items
 - Implicit
 - Users click / listen / watch an item




“Users that looked at things similar to you also looked at...”




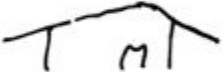

Optimization problem

$$P(r_{ui})$$
$$r_{ui} = \begin{cases} 1 & \text{if RELEVANT} \\ 0 & \text{ELSE} \end{cases}$$

During Recommendation...

			
ERIK	0.4	0.1	0.9
MARIA	0.05	0.93	0.1

During Recommendation...

			
ERIK	0.4	0.1	0.9
MARIA	0.05	0.93	0.1

Back to paper

- Motivation
- Baseline method (PMF)
- Method (Bayes PMF)
- Experimental Results

(sorry for all the formulas)

Bayesian Probabilistic Matrix Factorization using Markov Chain Monte Carlo

Ruslan Salakhutdinov
Andriy Mnih

Department of Computer Science, University of Toronto, Toronto, Ontario M5S 3G4, Canada

RSALAKHU@CS.TORONTO.EDU
AMNIH@CS.TORONTO.EDU

Abstract

Low-rank matrix approximation methods provide one of the simplest and most effective approaches to collaborative filtering. Such models are usually fitted to data by finding a MAP estimate of the model parameters, a procedure that can be performed efficiently even on very large datasets. However, unless the regularization parameters are tuned carefully, this approach is prone to overfitting because it finds a single point estimate of the parameters. In this paper we present a fully Bayesian treatment of the Probabilistic Matrix Factorization (PMF) model in which model capacity is controlled automatically by integrating over all model parameters and hyperparameters. We show that Bayesian PMF models can be efficiently trained using Markov chain Monte Carlo methods by applying them to the Netflix dataset, which consists of over 100 million movie ratings. The resulting models achieve significantly higher prediction accuracy than PMF models trained using MAP estimation.

1. Introduction

Factor-based models have been used extensively in the domain of collaborative filtering for modelling user preferences. The idea behind such models is that preferences of a user are determined by a small number of unobserved factors. In a linear factor model, a user's rating of an item is modelled by the inner product of an item factor vector and a user factor vector. This means that the $N \times M$ preference matrix of ratings that N users assign to M movies is modeled by the product of an $D \times N$ user coefficient matrix U and a $D \times M$ factor matrix V (Rennie & Srebro, 2005; Srebro

& Jaakkola, 2003). Training such a model amounts to finding the best rank- D approximation to the observed $N \times M$ target matrix R under the given loss function.

A variety of probabilistic factor-based models have been proposed (Hofmann, 1999; Marlin, 2004; Marlin & Zemel, 2004; Salakhutdinov & Mnih, 2008). In these models factor variables are assumed to be marginally independent while rating variables are assumed to be conditionally independent given the factor variables. The main drawback of such models is that inferring the posterior distribution over the factors given the ratings is intractable. Many of the existing methods resort to performing MAP estimation of the model parameters. Training such models amounts to maximizing the log-posterior over model parameters and can be done very efficiently even on very large datasets.

In practice, we are usually interested in predicting ratings for new user/movie pairs rather than in estimating model parameters. This view suggests taking a Bayesian approach to the problem which involves integrating out the model parameters. In this paper, we describe a fully Bayesian treatment of the Probabilistic Matrix Factorization (PMF) model which has been recently applied to collaborative filtering (Salakhutdinov & Mnih, 2008). The distinguishing feature of our work is the use of Markov chain Monte Carlo (MCMC) methods for approximate inference in this model. In practice, MCMC methods are rarely used on large-scale problems because they are perceived to be very slow by practitioners. In this paper we show that MCMC can be successfully applied to the large, sparse, and very imbalanced Netflix dataset, containing over 100 million user/movie ratings. We also show that it significantly increases the model's predictive accuracy, especially for the infrequent users, compared to the standard PMF models trained using MAP with regularization parameters that have been carefully tuned on the validation set.

Previous applications of Bayesian matrix factorization methods to collaborative filtering (Lim & Teh, 2007; Raiko et al., 2007) have used variational approxima-

Appearing in *Proceedings of the 25th International Conference on Machine Learning*, Helsinki, Finland, 2008. Copyright 2008 by the author(s)/owner(s).

Probabilistic Matrix Factorization

- N users: User represented by vector $U : N \times D$
- M items: Item represented by vector $V : M \times D$
- Rating = dot(uservec, itemvec) + gaussian noise

$$p(R|U, V, \alpha) = \prod_{i=1}^N \prod_{j=1}^M \left[\mathcal{N}(R_{ij} | U_i^T V_j, \alpha^{-1}) \right]^{I_{ij}} \quad (1)$$

$$p(U|\alpha_U) = \prod_{i=1}^N \mathcal{N}(U_i | 0, \alpha_U^{-1} I) \quad (2)$$

$$p(V|\alpha_V) = \prod_{j=1}^M \mathcal{N}(V_j | 0, \alpha_V^{-1} I), \quad (3)$$

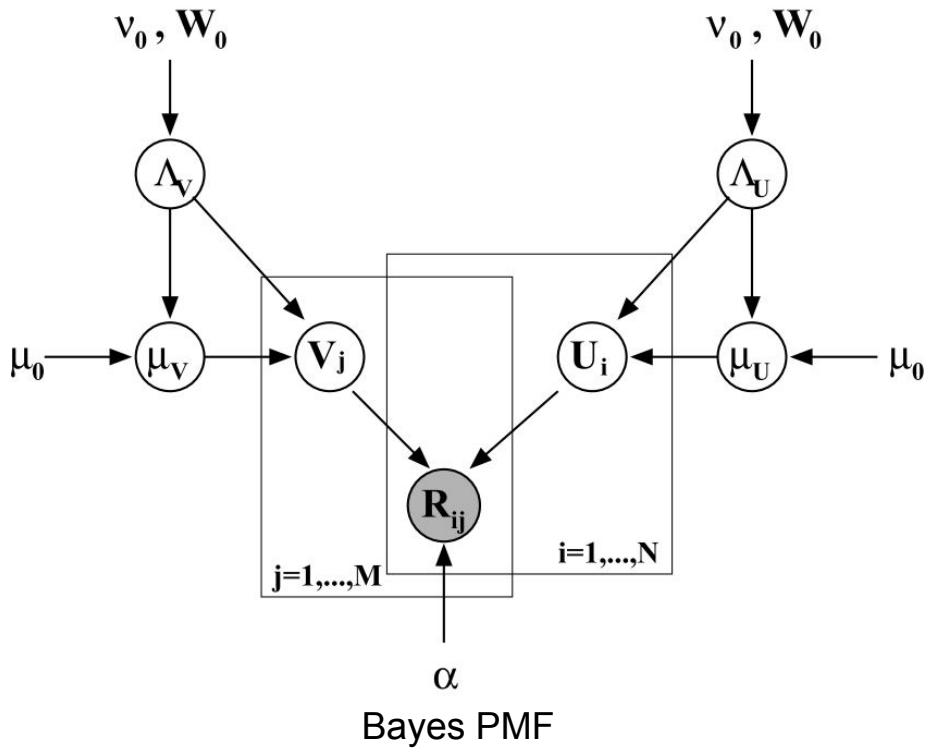
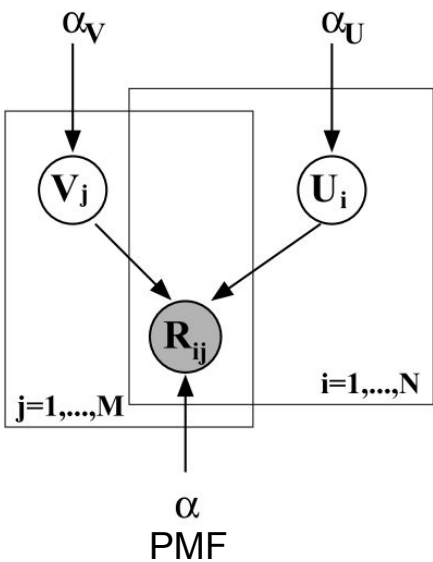
Probabilistic Matrix Factorization

- Maximum a posteriori
- Pick your favourite gradient based optimizer
- Minimize a regularized sum of squares
- How to set lambda?
 - Grid search? Argue expensive
 - Hyperparameters
- Another benchmark: PMF with sigmoid

$$E = \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^M I_{ij} (R_{ij} - U_i^T V_j)^2 + \frac{\lambda_U}{2} \sum_{i=1}^N \|U_i\|_{\text{Fro}}^2 + \frac{\lambda_V}{2} \sum_{j=1}^M \|V_j\|_{\text{Fro}}^2, \quad (4)$$

$$p(R|U, V, \alpha) = \prod_{i=1}^N \prod_{j=1}^M \left[\mathcal{N}(R_{ij} | \sigma(U_i^T V_j), \alpha^{-1}) \right]^{I_{ij}}. \quad (15)$$

Bayesian Probabilistic Matrix Factorization



BPME model

- User and item vectors are gaussian
- Hyperparameters are

Normal-Wishart

- Multi-dimensional version of gamma-normal
 - Conjugate family
-
- Want to find predictive posterior on a new rating:

$$p(R_{ij}^*|R, \Theta_0) = \iint p(R_{ij}^*|U_i, V_j)p(U, V|R, \Theta_U, \Theta_V)p(\Theta_U, \Theta_V|\Theta_0)d\{U, V\}d\{\Theta_U, \Theta_V\}. \quad (9)$$

Likelihood:

$$p(R|U, V, \alpha) = \prod_{i=1}^N \prod_{j=1}^M \left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1}) \right]^{I_{ij}} \quad (1)$$

Priors:

$$p(U|\mu_U, \Lambda_U) = \prod_{i=1}^N \mathcal{N}(U_i|\mu_U, \Lambda_U^{-1}), \quad (5)$$

$$p(V|\mu_V, \Lambda_V) = \prod_{i=1}^M \mathcal{N}(V_i|\mu_V, \Lambda_V^{-1}). \quad (6)$$

$$p(\Theta_U|\Theta_0) = p(\mu_U|\Lambda_U)p(\Lambda_U) = \mathcal{N}(\mu_U|\mu_0, (\beta_0\Lambda_U)^{-1})\mathcal{W}(\Lambda_U|W_0, \nu_0), \quad (7)$$

$$p(\Theta_V|\Theta_0) = p(\mu_V|\Lambda_V)p(\Lambda_V) = \mathcal{N}(\mu_V|\mu_0, (\beta_0\Lambda_V)^{-1})\mathcal{W}(\Lambda_V|W_0, \nu_0). \quad (8)$$

BPMPF Inference

- Gibbs Sampling
- Each user vector is indep of others

$$p(U_i|R, V, \Theta_U, \alpha) = \mathcal{N}(U_i|\mu_i^*, [\Lambda_i^*]^{-1}) \quad (11)$$
$$\sim \prod_{j=1}^M \left[\mathcal{N}(R_{ij}|U_i^T V_j, \alpha^{-1}) \right]^{I_{ij}} p(U_i|\mu_U, \Lambda_U),$$

where

$$\Lambda_i^* = \Lambda_U + \alpha \sum_{j=1}^M [V_j V_j^T]^{I_{ij}} \quad (12)$$

$$\mu_i^* = [\Lambda_i^*]^{-1} \left(\alpha \sum_{j=1}^M [V_j R_{ij}]^{I_{ij}} + \Lambda_U \mu_U \right). \quad (13)$$

- Hyperparameters are normal-wishart (conjugacy)

Gibbs sampling for Bayesian PMF

1. Initialize model parameters $\{U^1, V^1\}$

2. For $t=1, \dots, T$

- Sample the hyperparameters (Eq. 14):

$$\Theta_U^t \sim p(\Theta_U|U^t, \Theta_0)$$

$$\Theta_V^t \sim p(\Theta_V|V^t, \Theta_0)$$

- For each $i = 1, \dots, N$ sample user features in parallel (Eq. 11):

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

- For each $i = 1, \dots, M$ sample movie features in parallel:

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

Experimental Results

- Netflix dataset
 - 100m ratings
 - 500k users
 - 18k movies
 - Also valuation data (1.4m) and test data (2.8m)
- Evaluation metric:
 - Root Mean Square Error

Leaderboard

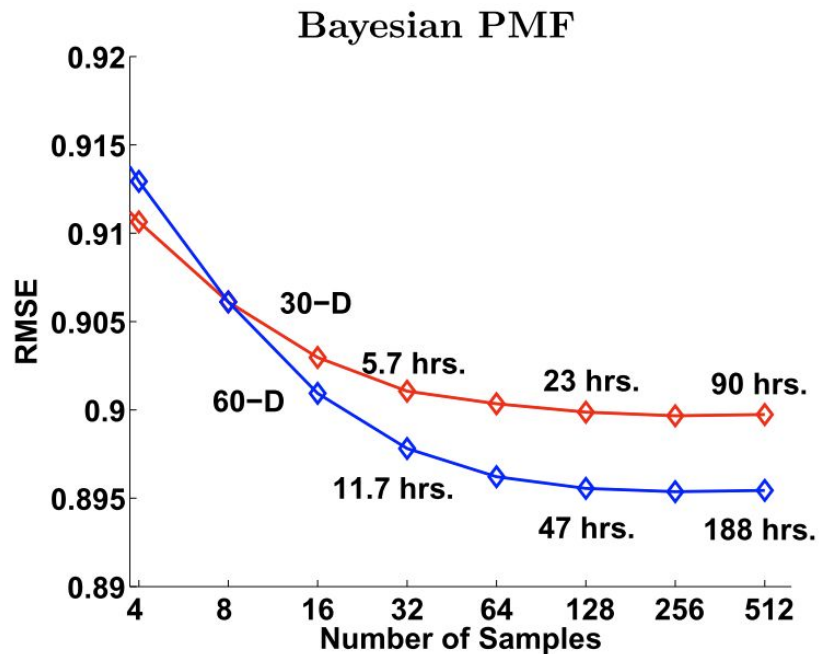
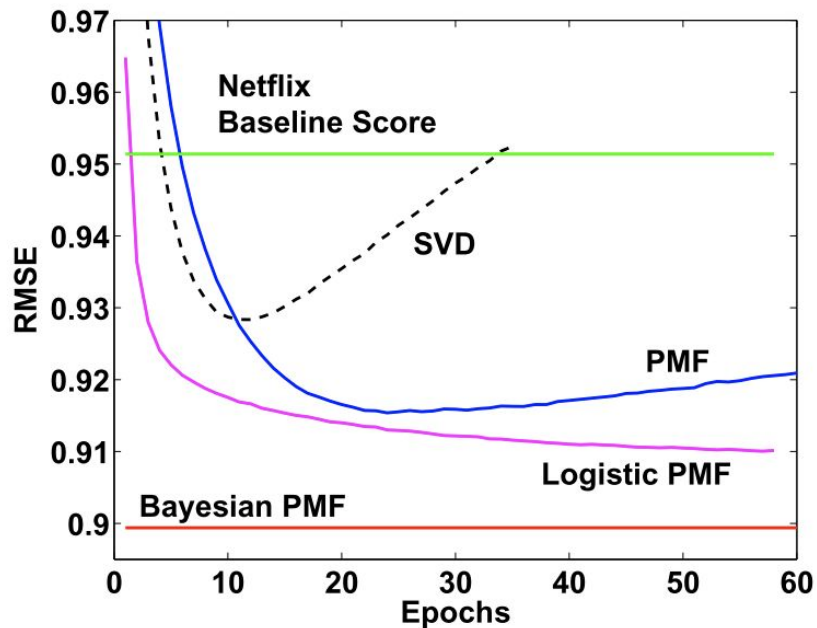
Showing Test Score. [Click here to show quiz score](#)

Display top leaders.

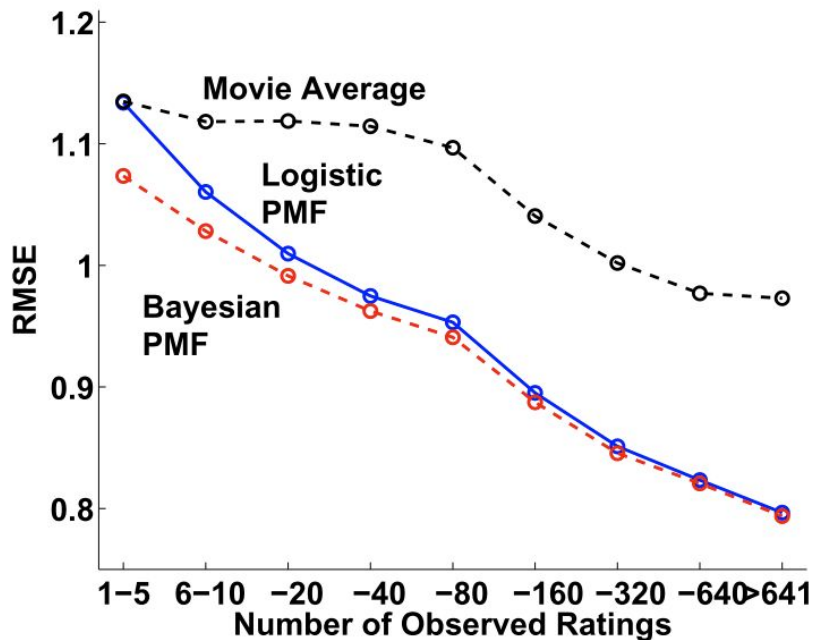
Rank	Team Name	Best Test Score	% Improvement	Best Submit Time
Grand Prize - RMSE = 0.8567 - Winning Team: BellKor's Pragmatic Chaos				
1	BellKor's Pragmatic Chaos	0.8567	10.06	2009-07-26 18:18:28
2	The Ensemble	0.8567	10.06	2009-07-26 18:38:22
3	Grand Prize Team	0.8582	9.90	2009-07-10 21:24:40
4	Opera Solutions and Vandelay United	0.8588	9.84	2009-07-10 01:12:31
5	Vandelay Industries !	0.8591	9.81	2009-07-10 00:32:20
6	PragmaticTheory	0.859		
7	BellKor in BigChaos	0.860		
8	Dace	0.861		
9	Feeds2	0.862		
10	BigChaos	0.862		
11	Opera Solutions	0.862		
12	BellKor	0.862		



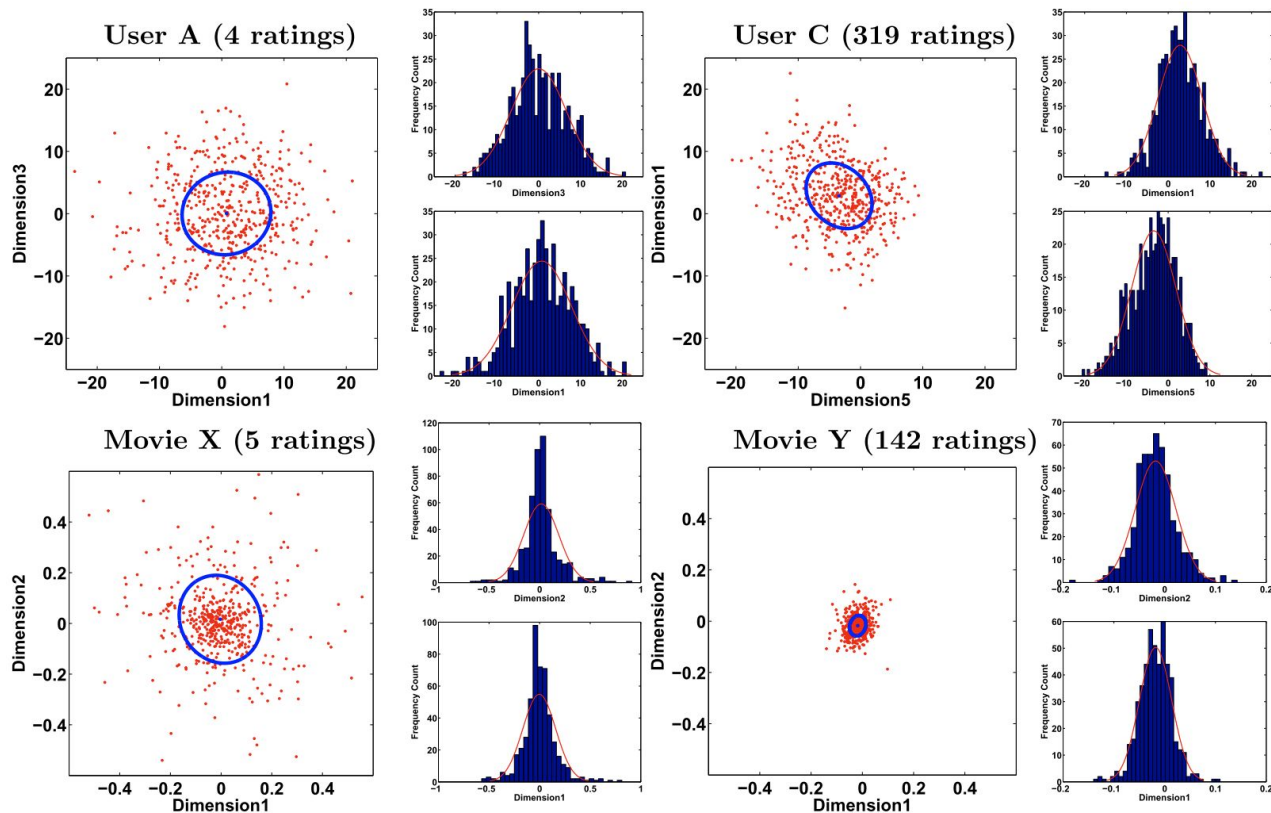
Results



Works better for users with little data



Inspection on some user vector indices



Conclusions

- Full bayesian recommender with MCMC
- Actually works on 100m dataset
- Uncertain with little data (cold start)
 - Potentially more exploration

My conclusions

- RMSE is not a reliable eval metric
- MCMC is heavy (1-2days), try Variational Inference?
- Why not MCMC on PMF model?