

# An Information-Theoretic Approach to Collaborative Filtering



Byung-Hak Kim

Department of Electrical and Computer Engineering, Texas A&M University



## Introduction

- Emerging applications, such as recommender systems, motivate us to consider of novel information-theoretic approaches to learning and inference. **Collaborative Filtering (CF)** is chosen to test our approaches to these types of problems.
- In particular, **an information-theoretic framework** is proposed to analyze a model of collaborative filtering problems. The motivating example, used throughout, is the movie-rating prediction problem popularized by the Netflix Prize.
- Consider a collection of  $N$  users and  $M$  movies with  $L + T$  noisy ratings, each generated by one user watching one movie. The main theoretical question is, “**How large should the number of observed ratings,  $L$ , be to estimate the  $T$  hidden ratings within some distortion  $\delta$ ?**”

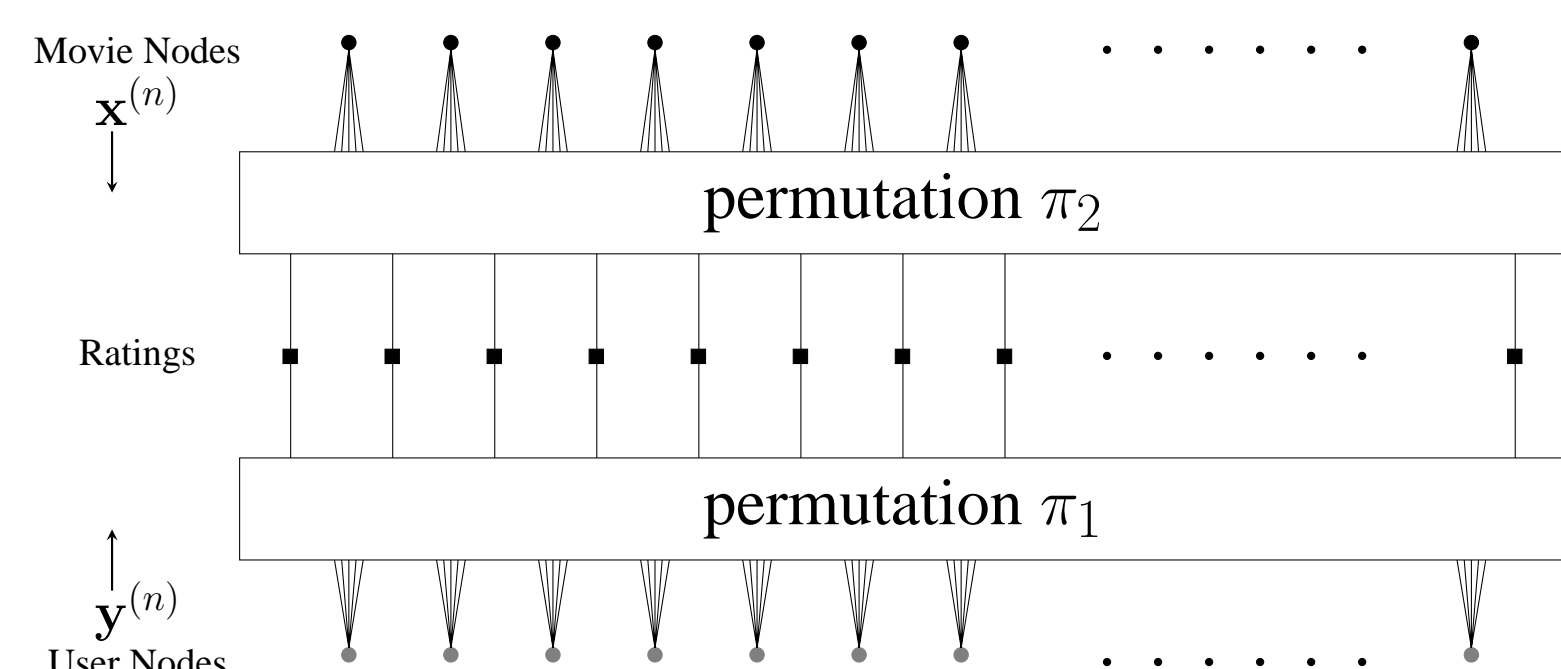
## Model Description

Introducing a *generative and probabilistic model* for the movie ratings:

- The basic idea is that a *hidden (or latent)* variable is introduced for each user and each movie, and the movie ratings are conditionally independent given these hidden variables. It is convenient to **think of the hidden variable for any user (or movie) as the user group (or movie group) of that user (or movie)**. In this context, the rating associated with a user-movie pair depends only on the user group and the movie group. This conditional independence assumption in the model implies that

$$\Pr(\mathbf{R}_O | \mathbf{U}, \mathbf{V}) \triangleq \prod_{i=1}^L w\left(R_{\pi(i), \sigma(i)} | U_{\pi(i)}, V_{\sigma(i)}\right).$$

- Since graphs with local factorization properties (e.g., HMMs) have many advantages, we adopt a probabilistic model based on a graphical model composed of 3 layers (see Figure below). **These layers separate the influence of user groups, movie groups, and observed ratings.** Moreover, for synthetic data, we could randomly pick user (or movie) groups via random permutations.



## EM Learning Algorithm

**Step I: Initialization**

$$f_n^{(0)}(u) = p_U(u), h_m^{(0)}(v) = p_V(v), w^{(0)}(r|u, v)$$

**Step II: Recursive update**

$$f_n^{(i+1)}(u) \propto \sum_{m \in \mathcal{V}_n} f_n^{(i)}(u) \sum_{v \in [g_m]} w^{(i)}(r_{n,m}|u, v) h_m^{(i)}(v)$$

$$h_m^{(i+1)}(v) \propto \sum_{n \in \mathcal{U}_m} h_m^{(i)}(v) \sum_{u \in [g_u]} w^{(i)}(r_{n,m}|u, v) f_n^{(i)}(u)$$

$$w^{(i+1)}(r|u, v) \propto \sum_{(n,m): r_{n,m}=r} w^{(i)}(r_{n,m}|u, v) f_n^{(i+1)}(u) h_m^{(i+1)}(v)$$

**Step III: Output**

$$\hat{p}_{R_{nm}|\mathbf{R}_O}^{(i+1)}(r) \propto \sum_{u,v} f_n^{(i+1)}(u) h_m^{(i+1)}(v) w^{(i+1)}(r|u, v)$$

$$\hat{p}_{U_n|\mathbf{R}_O}^{(i+1)}(u) = f_n^{(i+1)}(u), \hat{p}_{V_m|\mathbf{R}_O}^{(i+1)}(v) = h_m^{(i+1)}(v)$$

## MP Learning Algorithm

**Step I: Initialization**

$$\mathbf{x}_{m \rightarrow n}^{(0)}(v) = \mathbf{x}_m^{(0)}(v) = p_V(v), \mathbf{y}_{n \rightarrow m}^{(0)}(u) = \mathbf{y}_n^{(0)}(u) = p_U(u)$$

**Step II: Recursive update**

$$\mathbf{y}_{n \rightarrow m}^{(i+1)}(u) \propto \mathbf{y}_n^{(i)}(u) \prod_{k \in \mathcal{V}_n \setminus m} \sum_v w(r_{n,m}|u, v) \mathbf{x}_{k \rightarrow n}^{(i)}(v)$$

$$\mathbf{x}_{m \rightarrow n}^{(i+1)}(v) \propto \mathbf{x}_m^{(i)}(v) \prod_{k \in \mathcal{U}_m \setminus n} \sum_u w(r_{n,m}|u, v) \mathbf{y}_{k \rightarrow m}^{(i)}(u)$$

**Step III: Output**

$$\hat{p}_{R_{nm}|\mathbf{R}_O}^{(i+1)}(r) \propto \sum_{u,v} \mathbf{y}_{n \rightarrow m}^{(i+1)}(u) \mathbf{x}_{m \rightarrow n}^{(i+1)}(v) w(r|u, v)$$

$$\hat{p}_{U_n|\mathbf{R}_O}^{(i+1)}(u) \propto \mathbf{y}_n^{(i)}(u) \prod_{k \in \mathcal{V}_n} \sum_v w(r_{n,m}|u, v) \mathbf{x}_{k \rightarrow n}^{(i)}(v)$$

$$\hat{p}_{V_m|\mathbf{R}_O}^{(i+1)}(v) \propto \mathbf{x}_m^{(i)}(v) \prod_{k \in \mathcal{U}_m} \sum_u w(r_{n,m}|u, v) \mathbf{y}_{k \rightarrow m}^{(i)}(u)$$

## Theoretical Performance Analysis

- Density evolution (DE)** is well-known technique for analyzing probabilistic message-passing inference algorithms that was originally developed to analyze belief-propagation decoding of error-correcting codes and was later extended to more general inference problems. It works under the assumption that **the local neighborhood of each node is a tree**. The DE update equations for degree  $d$  user and movie nodes are shown in below equations.

$$\mu_d^{(i+1)}(u, B) = \int \sum_{r_1, \dots, r_d} I(G((b_1, r_1), \dots, (b_d, r_d); a) \in B) \mu^{(0)}(u, da) \prod_{j=1}^d \sum_v \nu^{(i)}(v, db_j) w(r_j|u, v)$$

$$\nu_d^{(i+1)}(v, A) = \int \sum_{r_1, \dots, r_d} I(F((a_1, r_1), \dots, (a_d, r_d); b) \in A) \nu^{(0)}(v, db) \prod_{j=1}^d \sum_u \mu^{(i)}(u, da_j) w(r_j|u, v)$$

where  $F_d(a_1, r_1, \dots, a_d, r_d; b) \triangleq \frac{b(v) \prod_{j=1}^d \sum_u a_j(u) w(r_j|u, v)}{\sum_v b(v) \prod_j \sum_u a_j(u) w(r_j|u, v)}$ ,  $G_d(b_1, r_1, \dots, b_d, r_d; a) \triangleq \frac{a(u) \prod_{j=1}^d \sum_v b_j(v) w(r_j|u, v)}{\sum_u a(u) \prod_j \sum_v b_j(v) w(r_j|u, v)}$ .

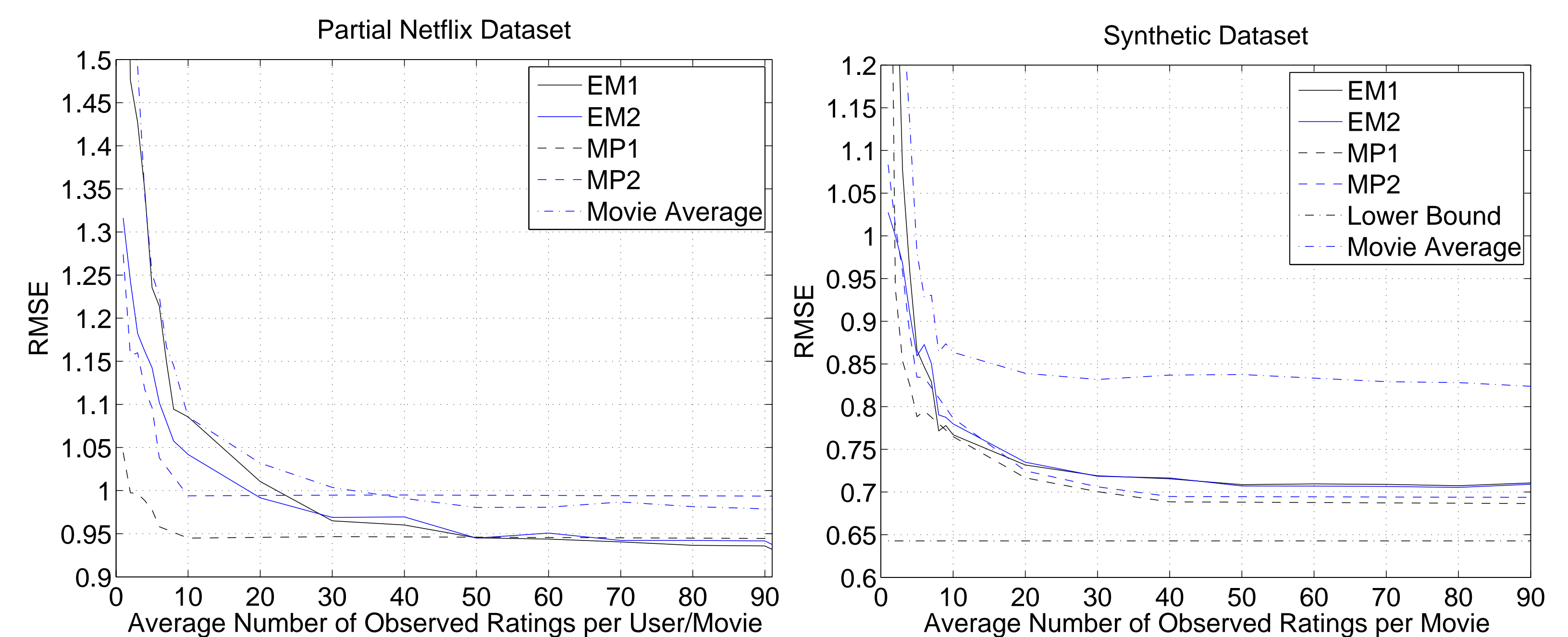
- Since computation of ratings prediction matrix can be viewed as the product of three matrices, we consider the simplified class of tri-factorized matrices to derive preliminary **generalization error bounds of the model in terms of the training error** as

$$\chi_{g_u, g_v} \triangleq \left\{ X | X = U^T G V \text{ where } U \in [0, 1]^{g_u \times N}, V \in [0, 1]^{g_v \times M}, G \in \{\pm 1\}^{g_u \times g_v} \right\}.$$

**Theorem 1 [Generalization Error Bound]** For any matrix  $Y \in \{\pm 1\}^{N \times M}$ ,  $N, M > 2$ ,  $\delta > 0$  and integers  $g_u$  and  $g_v$ , with probability at least  $1 - \delta$  over choosing a subset  $O$  of entries in  $Y$  uniformly among all subsets of  $|O|$  entries

$$\forall X \in \chi_{g_u, g_v}, |D_H(X, Y) - D_O(X, Y)| < \sqrt{\left\{ (N g_u + M g_v + g_u g_v) \log \frac{12eM}{\min(g_u, g_v)} - \log \delta \right\} / (2|O|)}.$$

## Experimental Performance Analysis



## Conclusions

- Our results show that, while both methods perform similarly with large amounts of data, **the MP algorithm is superior for small amounts of data**. Another advantage of the MP algorithm is that it can be analyzed using the technique of DE that was originally developed for MP decoding of error-correcting codes.
- One interesting aspect is the similar asymptotic behaviors between partial Netflix and synthetic dataset. This implies **the proposed model approximates Netflix data generation more closely than other simpler factor (or low rank matrix) models** since noise process is built in the model. Note that this is a *generative model* which allows one to evaluate different learning algorithms on synthetic data and compare the results with theoretical bounds.

Acknowledgments: This poster is based on joint work with Henry D. Pfister and Arvind Yedla.