IMP: A Message-Passing Algorithm for Matrix Completion

Henry D. Pfister
(joint with Byung-Hak Kim and Arvind Yedla)

Electrical and Computer Engineering
Texas A&M University

ISTC 2010
Brest, France
September 9, 2010

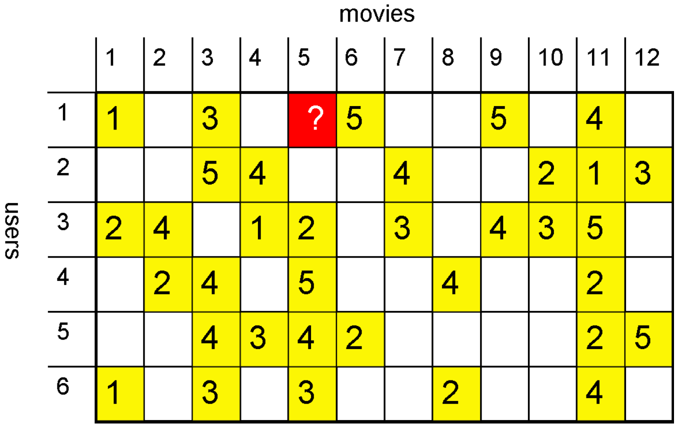
Outline

1. Matrix Completion via Iterative Information Processing
2. Factor Graph Model for Matrix Completion
3. IMP Algorithm and Performance Results
4. Conclusions and Open Problems

Matrix Completion Problem

> *"In its simplest form, the problem is to recover a matrix from a small sample of its entries,...*
>
> *Imagine now that we only observe a few entries of a data matrix. Then is it possible to accurately—or even exactly—guess the entries that we have not seen?..."*
>
> *- Candes and Plan '09*

Motivation: The Netflix Challenge ('06-'09)



	1	2	3	4	5	6	7	8	9	10	11	12
1	1		3		?	5			5		4	
2			5	4			4			2	1	3
3	2	4		1	2		3		4	3	5	
4		2	4		5			4			2	
5			4	3	4	2					2	5
6	1		3		3			2			4	

An overwhelming portion of the user-movie matrix (e.g., 99%) is unknown and the few observed movie ratings are noisy!

Recent Work on Matrix Completion

- Efficient Models and Practical Algorithms
  - Low rank matrix models and clustering models
  - Convex relaxation (SDP) and Bayesian approaches
- Exploration of the Fundamental Limits
  - Relationship between sparse observation and the recovery of missing entries
  - Cold-start problem in recommender systems

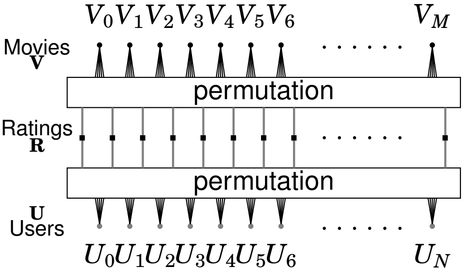
Main Differences

1. Focus on the matrix where the entries (drawn from a finite alphabet) are modeled by a factor graph
2. MP based algorithm to learn the missing information and eventually estimate missing entries

Our Goal: Matrix Completion via Iterative Processing

- Basic Approach
    - Establish a factor-graph model
    - Estimate parameters of model
    - Use for inference via message-passing (IMP)
- Benefits of our factor-graph model
    - Establishes a generative model for data matrices
    - Sparse observations reduce complexity
- Drawbacks of this approach
    - Mixes clustering and message-passing
    - Difficult to analyze behavior

Factor Graph Model (1): Conditional Independence



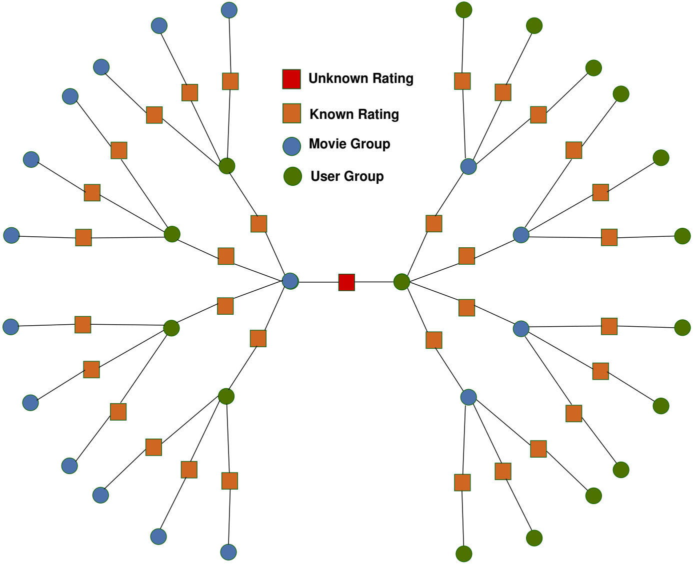
Assumption

The movie ratings R_{nm} (user n, movie m) are *conditionally independent* given the user group U_n and the movie group V_m

$$\Pr\left(\mathbf{R}_O | \mathbf{U}, \mathbf{V}\right) \triangleq \prod_{(n,m)\in O} w\left(R_{nm} | U_n, V_m\right)$$

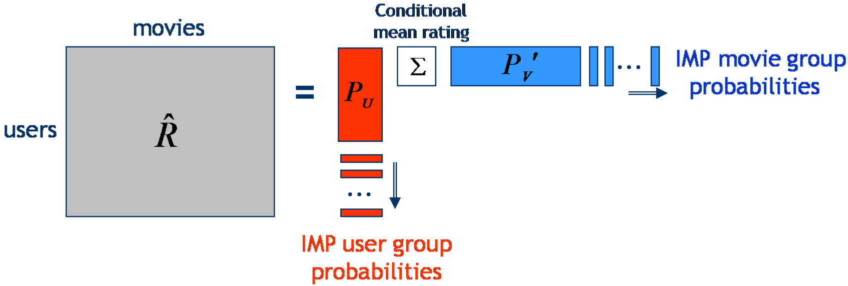
where $w(r|u,v) \triangleq \Pr(R_{nm} = r | U_n = u, V_m = v)$

Factor Graph Model (2): Computational Tree



Factor Graph Model (3): Matrix Decomposition

- MMSE estimates can be written as a matrix factorization



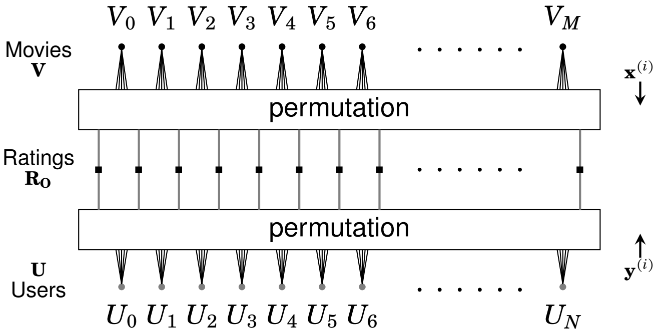
- In contrast to the standard low-rank matrix model, this adds non-negativity and normalization constraints

Inference by Message-Passing (IMP) Algorithm (1)

Step I. Initialize Group Rating Probabilities $w(r|u, v)$

- Cluster users (and movies) using variable-dimension vector quantization (VDVQ), which is a variant of VQ where the codebook vectors are full, but training vectors are sparse.
- Training is based on based on the generalized Lloyd algorithm (GLA) and the distance computed only on overlapping entries.
- For user clustering, the K codebook vectors can be thought of as K critics which have rated every movie.
- After clustering users/movies each into user/movie groups, estimate $w(r|u, v)$ from the frequencies of each user-group / movie-group / rating triple.

IMP Algorithm (2)



Notation

Let $\mathbf{T}^{(i)}_{m\to n}$ be $m \to n$ computation tree after i iterations, then message from movie m to user n is $\mathbf{x}^{(i)}_{m\to n} \triangleq \Pr(V_m = v|\mathbf{T}^{(i)}_{m\to n})$ and message from user n to movie m is $\mathbf{y}^{(i)}_{n\to m} \triangleq \Pr(U_n = u|\mathbf{T}^{(i)}_{n\to m})$

IMP Algorithm (3)

Step II. Message-Passing Update of Group Vectors

1. Initialization of user/movie to prior group probabilities

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

2. Recursive update for user/movie group probabilities

$$\mathbf{x}_{m\to n}^{(i+1)}(v) \propto \mathbf{x}_m^{(0)}(v) \prod_{k\in\mathcal{U}_m\backslash n} \sum_u w\left(r|u,v\right) \mathbf{y}_{k\to m}^{(i)}(u)$$

$$\mathbf{y}_{n\to m}^{(i+1)}(u) \propto \mathbf{y}_n^{(0)}(u) \prod_{k\in\mathcal{V}_n\backslash m} \sum_v w\left(r|u,v\right) \mathbf{x}_{k\to n}^{(i)}(v)$$

$\mathcal{U}_m$: all users who rated movie m
$\mathcal{V}_n$: all movies whose rating by user n was observed

3. Update $w(r|u,v)$ for group ratings

IMP Algorithm (4)

Step III. Approximate Matrix Completion

After i iterations of message-passing update,

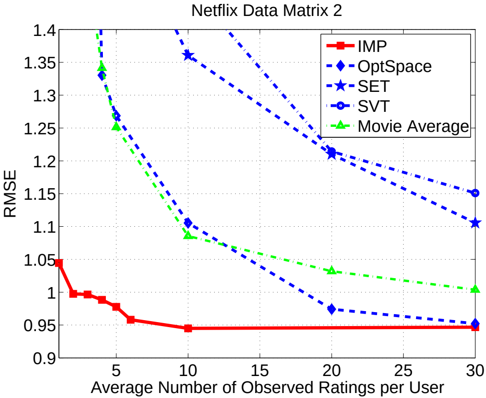
- Output probability of rating R_{nm} given observed ratings

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

- Minimize the mean-squared prediction error with

$$\hat{r}^{(i+1)}_{n,m} = \sum_{r\in\mathcal{R}} r\,\hat{p}^{(i+1)}_{R_{nm}|\mathbf{R}_O}(r)$$

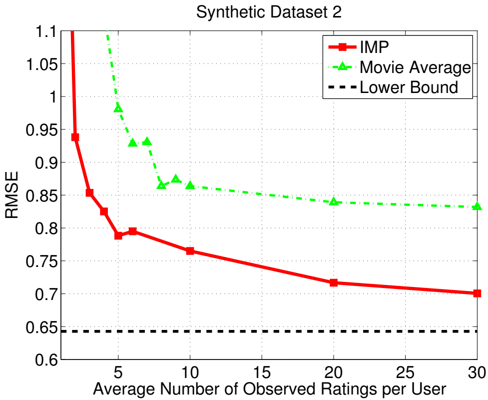
IMP Improves the Cold-Start Problem



Dataset Description

- Netflix submatrix of 5,035 movies and 5,017 users by avoiding movies and users with less than 3 ratings
- 16% of the users and 41% of the movies have less than 10 ratings

Generation of Synthetic Datasets



Dataset Description

- Synthetic Dataset generated after learning Netflix Data Matrix with 16 movie/user groups and randomly subsampled

Conclusion

Main Contributions

1. Factor Graph Model: Probabilistic low-rank matrix model
2. IMP Algorithm: Combines clustering with message-passing

Avenues for Future Research

1. Combine with clustering via message-passing to get a fully iterative approach
2. Obtain performance analysis of IMP algorithm via density evolution

Thank you