A Group-Specific Recommender System

Xuan Bi

Department of Biostatistics, Yale University

Joint work with Annie Qu, Junhui Wang and Xiaotong Shen

Department of Statistics, University of Florida

December 8, 2017

What is a recommender system?

- A system that recommends items to users
- Track users' preferences and make personalized predictions

					
 Tom	4	?	5	?	?
 Jerry	?	?	?	3	2
 JohnSnow	5	5	5	5	?
 Daenerys	?	?	?	1	?
 Xuan Bi	?	?	?	?	4

Research Goal and Impact

- **Recommendation process:**
  - Complete the user-item matrix
  - Recommend items with high ratings to users
- **Research goal:**
  - **Improve prediction accuracy**
    - Much room for improvement (current accuracy $\approx$ 10%-20%)
    - Identify and address problems (Our model: dependency)
- **Impact:**
  - Even 1% improvement brings huge market value (Netflix Challenge)

Diverse Applications

- **Direct applications:**
  - movies, music, restaurants recommendations
- **Broad applications:**
  - personalized medicine (patient & treatment)
    - Ongoing work with LYG Lab, Northwestern University School of Medicine
    - 6,899 breast cancer patients with electronic medical records
  - election prediction (county & candidate)
  - product sales forecasting (store & product)
    - Bi, Adomavicius, Li and Qu (2017), under revision

Challenges

- Data are extremely high-volume (1M to 1B ratings)
- Extremely sparse
- MovieLens 10M data:
  - 71,567 users over 10,681 movies
  - 10,000,054 observed ratings out of 764,407,127 possible ratings (1.3% observation rate)
- Algorithms need to be scalable

How the data look like?

- A real user-item matrix (blue represents missing ratings):

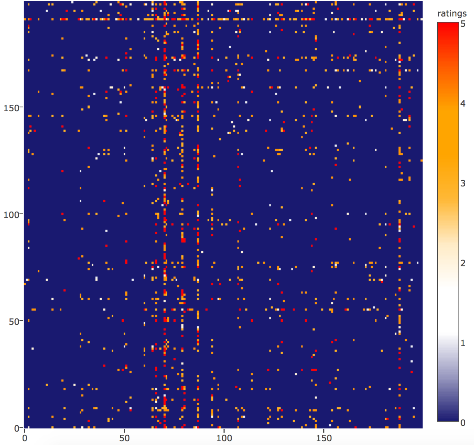


Figure: A random 200×200 sub-matrix of MovieLens 1M data

Existing Approaches

- **Collaborative filtering**
  - Predict users' behaviors based on other users' information (e.g., Funk, 2006; Bell and Koren, 2007)
  - Require more than one user
  - Adopted by Netflix Inc.
- **Content-based filtering**
  - Build item profiles based on domain knowledge, recommend items to users with similar profiles
    (e.g., Lang, 1995; Mooney and Roy, 2000)
  - Require domain knowledge
  - Adopted by Pandora Media, Inc.

- Other types of systems:
  - **Hybrid systems:** use covariates or additional information (Agarwal and Chen, 2009; Zhu et al., 2016; Mao et al., 2017)
  - **Neural networks:** restricted Boltzmann machines (Salakhutdinov et al., 2007)
  - **Ensemble methods:** apply multiple algorithms to enhance prediction accuracy (Paterek, 2007)

Unique Challenges

- A dynamic system (the “cold-start” problem):
  - New ratings are from new users to new items
  - Historical data (the training set) are not representative of future activities (the testing set)
  - MovieLens 10M data: 96% of latest movie ratings are given by new users or on new items
- Strong dependency/Data missing not at random:
  - Popular items attract more users
  - Users are influenced by each other

The Proposed Method: A Group-Specific Method

- A matrix factorization method
- Incorporate between-subject dependency via random effects
- The formulation of dependency also solves the "cold-start" problem
- Propose a new algorithm for scalable computing

Notation and Basic Assumption

- Let $\mathbf{R} = (r_{ui})_{n \times m}$ be a user-item matrix
- r_{ui} is user u's rating on item i, usually non-negative and finite
  - Pre-adjusted by covariates or subject main effects
- n is the number of users; m is the number of items
- Low-rank approximation:
  - $\text{rank}(\mathbf{R}) = K < \min(n, m)$

Existing Matrix Factorization

- Traditional factorization: $\mathbf{R}_{n\times m} \approx \mathbf{P}_{n\times K}(\mathbf{Q}_{m\times K})'$
- A product of a user-preference matrix and an item-preference matrix
- $\mathbf{p}_u$ and $\mathbf{q}_i$ are K-dimensional latent factors, representing user and item preference
- Estimate each rating: $\hat{r}_{ui} = \mathbf{p}_u'\mathbf{q}_i$

The Proposed Method

- Prespecify N user groups and M item groups
- We formulate:

$$r_{ui} \approx (\mathbf{p}_u + \mathbf{s}_{v_u})'(\mathbf{q}_i + \mathbf{t}_{j_i})$$

- $v_u \in \{1, \ldots, N\}$ and $j_i \in \{1, \ldots, M\}$ are group labels
- E.g., $v_{u_1} = v_{u_2}$ if u_1 and u_2 are in the same user group
- $\mathbf{s}_{v_u}$ and $\mathbf{t}_{j_i}$ are vectors of group effects for users and items

A General Criterion Function

- In matrix form, let $\mathbf{\Theta} = \hat{\mathbf{R}} = (\mathbf{P} + \mathbf{S})_{n \times K}(\mathbf{Q} + \mathbf{T})'_{m \times K}$
- $\mathbf{\Theta}$ is the parameter of interest
- Let R^o be the set of observed ratings, and $\Omega = \{(u, i) : r_{ui} \in R^o\}$
- We estimate $(\mathbf{P}, \mathbf{Q}, \mathbf{S}, \mathbf{T})$ to minimize

$$L(\mathbf{\Theta}|R^o) = \sum_{(u,i)\in\Omega} (r_{ui} - \theta_{ui})^2 + \lambda(\|\mathbf{P}\|_F^2 + \|\mathbf{Q}\|_F^2 + \|\mathbf{S}\|_F^2 + \|\mathbf{T}\|_F^2),$$

where λ is a tuning parameter, and $L(\cdot|R^o)$ is non-convex

A New Algorithm

- The alternating-least-squares algorithm involves large-matrix operation and storage ($> 100{,}000 \times 10{,}000$)
- A new algorithm embeds back-fitting (BF) into alternating least squares (ALS)
  - A two-step algorithm
  - Solve $(\mathbf{P} + \mathbf{S})$ and $(\mathbf{Q} + \mathbf{T})$ iteratively through ALS
    - Solve $\mathbf{P}$ and $\mathbf{S}$ iteratively through BF
    - Solve $\mathbf{Q}$ and $\mathbf{T}$ iteratively through BF
- Scalable through parallel computing

Model Training

- ALS step: $\hat{\theta}_{ui} = (\hat{\mathbf{p}}_u + \hat{\mathbf{s}}_{v_u})'(\hat{\mathbf{q}}_i + \hat{\mathbf{t}}_{j_i})$
- BF step: fix $\mathbf{P}$ and $\mathbf{S}$ and estimate $\hat{\mathbf{q}}_i$ and $\hat{\mathbf{t}}_j$ iteratively:

$$\hat{\mathbf{q}}_i = \underset{q_i}{\operatorname{argmin}} \sum_{u \in U_i} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{q}_i\|_2^2, i \in J_j,$$

$$\hat{\mathbf{t}}_j = \underset{t_j}{\operatorname{argmin}} \sum_{i \in J_j} \sum_{u \in U_i} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{t}_j\|_2^2.$$

- BF step: fix $\mathbf{Q}$ and $\mathbf{T}$ and estimate $\hat{\mathbf{p}}_u$ and $\hat{\mathbf{s}}_v$ iteratively:

$$\hat{\mathbf{p}}_u = \underset{p_u}{\operatorname{argmin}} \sum_{i \in I_u} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{p}_u\|_2^2, u \in V_v,$$

$$\hat{\mathbf{s}}_v = \underset{s_v}{\operatorname{argmin}} \sum_{u \in V_v} \sum_{i \in I_u} (r_{ui} - \theta_{ui})^2 + \lambda \|\mathbf{s}_v\|_2^2.$$

Identifiability

- $\mathbf{\Theta}$ is identifiable
- Indeterminacies among $(\mathbf{P}, \mathbf{Q}, \mathbf{S}, \mathbf{T})$:
  - Scaling: For a constant $c \neq 0$,
    $$\tilde{\mathbf{P}} = c\mathbf{P}, \text{ and } \tilde{\mathbf{Q}} = \mathbf{Q}/c$$
  - Addition: For a matrix $\mathbf{\Delta}$,
    $$\tilde{\mathbf{P}} = \mathbf{P} + \mathbf{\Delta}, \text{ and } \tilde{\mathbf{S}} = \mathbf{S} - \mathbf{\Delta}$$
  - Rotation: For a unitary matrix $\mathbf{\Omega}$,
    $$\tilde{\mathbf{P}} = \mathbf{P}\mathbf{\Omega}, \text{ and } \tilde{\mathbf{Q}} = \mathbf{Q}\mathbf{\Omega}$$
- Scaling and addition are addressed by the L_2 penalty
- Rotation can be addressed by a singular value decomposition of $\mathbf{P}$ (doable, but not necessary)

The Grouping Method

- Rule of thumb: the mean difference between groups is significant
- Grouping by covariates
  - Use existing categories
  - Evenly split individuals by quantiles
  - Clustering/Bi-clustering methods
- No covariate information
  - Grouping by missing patterns
    (The number of ratings is associated with the mean of ratings)

A Special Case: One Group for All

- Our contribution is not on how to group
- Gain additional information as long as the grouping structure is imposed
- A special case:
  - All subjects are mistakenly assigned to the same group
  - $\mathbf{s}_{v_u} = \mathbf{s}$ and $\mathbf{t}_{j_i} = \mathbf{t}$, then:

$$\hat{r}_{ui} = \mathbf{s}'\mathbf{t} + \mathbf{s}'\mathbf{q}_i + \mathbf{p}_u'\mathbf{t} + \mathbf{p}_u'\mathbf{q}_i$$

  - Compared with $\hat{r}_{ui} = \mu + \alpha_i + \beta_u + \mathbf{p}_u'\mathbf{q}_i$
  - Still better than traditional matrix factorization

Advantages on Prediction

- For **new users/items**:
  - Assign group labels (e.g., number of ratings, release date)
  - Group effects $\mathbf{s}_{v_u}$ and $\mathbf{t}_{j_i}$ become available
    (Personal effect $\mathbf{p}_u$ or $\mathbf{q}_i$ is not)
- Group effects are estimated via existing subjects' information
- An effective solution for the "cold-start" problem

Implementation

- Coded in MATLAB; run through parallel computing on cluster computers
- **Running time** (fixed λ): 0.8 minutes for 1 million ratings;
  8.3 minutes for 10 million ratings
- **Storage:** one group each time
- Tuning parameters are selected by minimizing RMSE on a validation set

Algorithmic Convergence

Proposition 1 (Convergence)

Suppose the parameter space for $\boldsymbol{\Theta}$ is compact, then any cluster point of the algorithm is a stationary point.

- Also a local minimizer along each block direction
- Generalize to an arbitrary initial point if the Hessian matrix is bounded

Algorithmic Convergence

Proposition 2 (Local Linear Convergence Rate)

Let $\mathbf{\Theta}_0$ be a strict local minimizer of $L(\cdot|R^o)$. For a neighborhood $\mathcal{V}$ of $\mathbf{\Theta}_0$ and the t_0-th iteration, suppose $\mathbf{\Theta}_{(t_0)} \in \mathcal{V}$. Then $\{\mathbf{\Theta}_{(t)}\}_{t \geq t_0} \subset \mathcal{V}$ exists, and there exists a $\mu \in [0, 1)$, such that

$$\|\mathbf{\Theta}_{(t+1)} - \mathbf{\Theta}_0\| \leq \mu \|\mathbf{\Theta}_{(t)} - \mathbf{\Theta}_0\|.$$

- The number of iterations is upper bounded by:

$$n_{iter} \sim O\left(\left\{\log \epsilon - \log \|\mathbf{\Theta}_{(t_0)} - \mathbf{\Theta}_0\|\right\} / \log \mu\right),$$

  where ϵ is the tolerance error
- May use branch-and-bound and random-start-point techniques to search for a good initial point

Theoretical Settings

- A general framework: the exponential family
- $\theta_{ui} = (\mathbf{p}_u + \mathbf{s}_{v_u})'(\mathbf{q}_i + \mathbf{t}_{j_i})$
- Formulate the inverse link function:

$$E(r_{ui}) = \mu(\theta_{ui})$$

- Formulate the criterion function:

$$\mathcal{L}(\boldsymbol{\Theta}|R^o) = -\sum_{(u,i)\in\Omega} \log f_{ui} + \lambda_{|\Omega|} D(\gamma),$$

where $f_{ui} = f(\cdot|\theta_{ui})$ is the density function of r_{ui}, $\lambda_{|\Omega|}$ is a penalization coefficient, $D(\cdot)$ is a penalty function, and $\gamma = \text{vec}(\mathbf{P}, \mathbf{Q}, \mathbf{S}, \mathbf{T})$

Major Theoretical Result

Theorem 1 (Consistency)

Given uniform continuity of f and $\lambda_{|\Omega|} < \frac{1}{2k}\epsilon^2_{|\Omega|}$, we have:

1. *The best possible convergence rate of $\hat{\boldsymbol{\Theta}}$ is*

$$\epsilon_{|\Omega|} \sim \frac{\sqrt{(n+m)K}}{|\Omega|^{1/2}} \left\{ \log\left(\frac{|\Omega|}{\sqrt{nmK}} \right) \right\}^{1/2}.$$

2. *There exists a constant $c > 0$, such that*

$$P\left(h(\hat{\boldsymbol{\Theta}}, \boldsymbol{\Theta}) \geq \epsilon_{|\Omega|} \right) \leq 7\exp(-c|\Omega|\epsilon^2_{|\Omega|}),$$

where $h(\cdot,\cdot)$ is the Hellinger metric, and $|\Omega| \to \infty$ is the total number of ratings.

- Only $(\log|\Omega|)^{1/2}$ slower than MLE; Same rate as MLE if f is smooth
- Applies to L_2 and Kullback-Leibler pseudo distance

Simulation study 1

- Compare with four competitive matrix factorization / latent factor models:
- Regularized singular-value decomposition (RSVD; Funk, 2006; Koren et al., 2009)
- A regression-based latent factor model (AC; Agarwal and Chen, 2009)
- Nuclear-norm matrix completion (MHT; Mazumder et al., 2010)
- A latent factor model with sparsity pursuit (ZSY; Zhu et al., 2016)

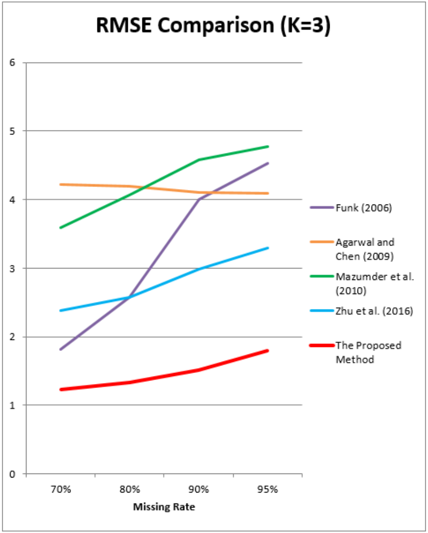
Simulation study 1

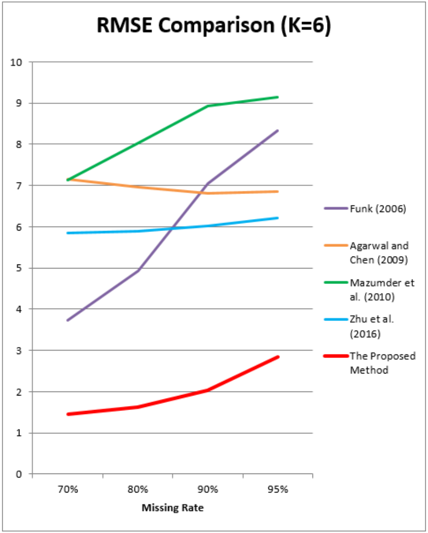
- $n = 650$, $m = 660$, $N = 13$, $M = 11$, and $K = 3$ or 6
- Generate $\mathbf{p}_u, \mathbf{q}_i \overset{iid}{\sim} \mathrm{N}(\mathbf{0}, \mathbf{I}_K)$
- Equally distanced group effects:
  $\mathbf{s}_v = (-3.5 + 0.5v)\mathbf{1}_K$, $\mathbf{t}_j = (-3.6 + 0.6j)\mathbf{1}_K$;
  each group has the same size
- $r_{ui} = (\mathbf{p}_u + \mathbf{s}_{v_u})'(\mathbf{q}_i + \mathbf{t}_{j_i})/3 + \varepsilon$, $\varepsilon \sim \mathrm{N}(0, 1)$

Simulation study 1

- Simulated dependency:
  - r_{ui} is assigned a value with probability 0.85 if $\bar{r}_{\cdot i} > 0.5$
  - Otherwise probability=0.2
- Simulated "cold-start":
  - Later-generated ratings are most likely from new IDs
- Missing rates are 0.7, 0.8, 0.9 and 0.95
- Results are based on 500 replications

Comparison under the "Cold-Start" Problem





Comparison under the "Cold-Start" Problem

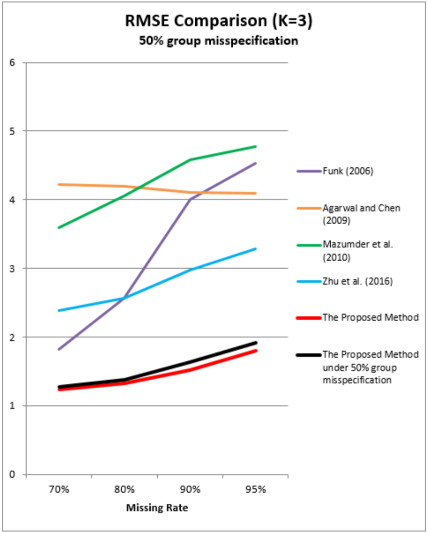
Table: RMSE (standard error) of the proposed method compared with four existing methods under different missing rates

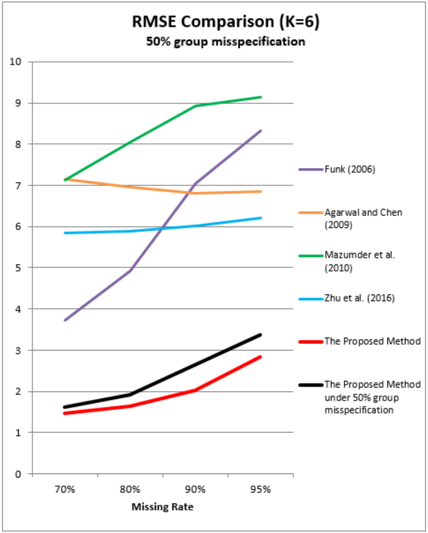
K	$\bar{\pi}$	Our Method	RSVD	AC	MHT	ZSY
$K = 3$	70%	**1.23 (0.03)**	1.82 (0.32)	4.22 (0.09)	3.59 (0.18)	2.38 (0.08)
	80%	**1.33 (0.04)**	2.57 (0.51)	4.19 (0.09)	4.06 (0.14)	2.57 (0.09)
	90%	**1.52 (0.07)**	4.00 (0.69)	4.11 (0.10)	4.58 (0.12)	2.98 (0.10)
	95%	**1.80 (0.10)**	4.53 (0.17)	4.09 (0.10)	4.77 (0.12)	3.29 (0.10)
$K = 6$	70%	**1.46 (0.04)**	3.73 (0.19)	7.16 (0.13)	7.13 (0.29)	5.84 (0.66)
	80%	**1.63 (0.06)**	4.93 (0.27)	6.96 (0.13)	8.04 (0.27)	5.89 (0.15)
	90%	**2.03 (0.14)**	7.05 (0.27)	6.81 (0.14)	8.93 (0.17)	6.02 (0.42)
	95%	**2.84 (0.39)**	8.32 (0.27)	6.85 (0.15)	9.14 (0.18)	6.21 (0.15)

Simulation Study 2

- Robustness against group misspecification
- Misassign users and items to adjacent groups with 50% probability
- Adjacent groups are the groups with the closest group effects

Comparision under Group Misspecification





Comparision under Group Misspecification

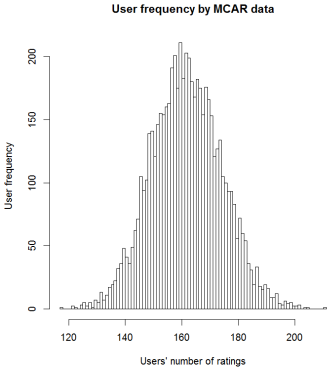
Table: RMSE (standard error) of the proposed method under group misassignment

		Probability of Group Misspecification			
K	$\bar{\pi}$	0%	10%	30%	50%
$K = 3$	70%	1.23 (0.03)	1.24 (0.03)	1.25 (0.03)	1.27 (0.04)
	80%	1.39 (0.04)	1.34 (0.05)	1.36 (0.05)	1.38 (0.05)
	90%	1.52 (0.07)	1.54 (0.18)	1.59 (0.16)	1.63 (0.26)
	95%	1.80 (0.10)	1.81 (0.12)	1.87 (0.10)	1.92 (0.09)
$K = 6$	70%	1.46 (0.04)	1.50 (0.05)	1.56 (0.05)	1.62 (0.06)
	80%	1.63 (0.06)	1.70 (0.07)	1.82 (0.07)	1.91 (0.09)
	90%	2.03 (0.14)	2.23 (0.20)	2.43 (0.15)	2.65 (0.15)
	95%	2.84 (0.39)	3.04 (0.30)	3.25 (0.24)	3.37 (0.18)

MovieLens Data

- Viewers' ratings on movies, collected by GroupLens Research from 2000 to 2009
- **MovieLens 1M data:**
  - 1,000,209 ratings collected from 6,040 users over 3,883 items
  - Users' age, gender, occupation and zipcode, and items' genres and release dates
- **MovieLens 10M data:**
  - 10,000,054 ratings collected from 71,567 users over 10,681 items
  - User information is not included

MovieLens Data: Dependency



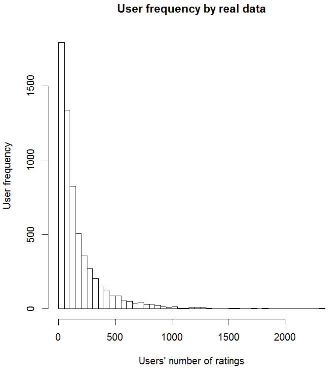


Figure: Missing-completely-at-random data vs. MovieLens 1M data

- In real data, the number of users' ratings has a large range (0-2500), and is highly-skewed

MovieLens Data: Dependency

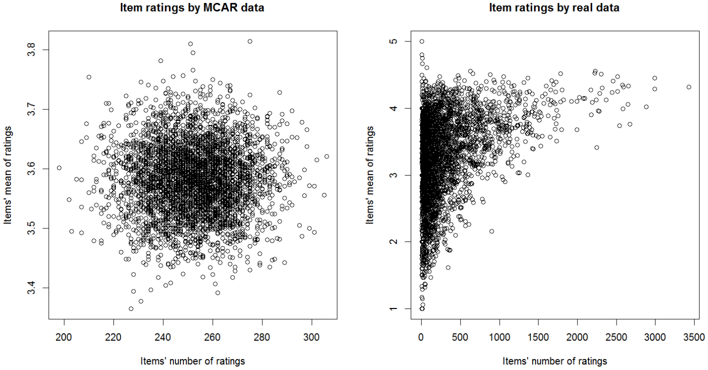


Figure: Missing-completely-at-random data vs. MovieLens 1M data

- In real data, the value of ratings is highly associated with the number of ratings (p-value $\approx 10^{-112}$)
- Popular items have high ratings

MovieLens Data

- For the proposed method, we set the number of groups as $N = 12$ and $M = 10$
- Group users by the number of their ratings, and group items by their release dates
- The results are robust if grouping by demographic information or k-means

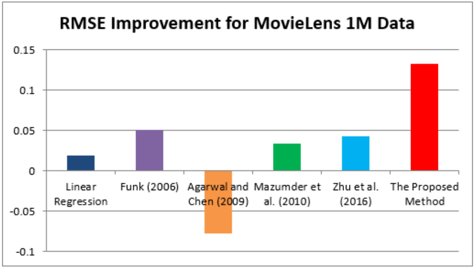
MovieLens Data

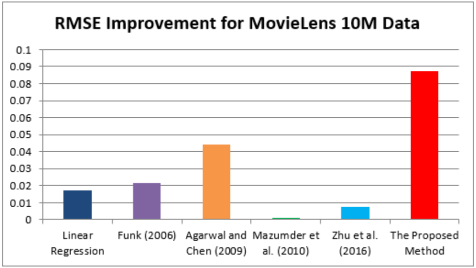
Table: RMSE of the proposed method compared with six existing methods for MovieLens 1M and 10M data.

	MovieLens 1M	MovieLens 10M
Grand Mean Imputation	1.1112	1.0185
Linear Regression	1.0905	1.0007
Funk (2006)	1.0552	0.9966
Agarwal and Chen (2009)	1.1974	0.9737
Mazumder et al. (2010)	1.0737	1.0177
Zhu et al. (2016)	1.0635	1.0108
The Proposed Method	**0.9644**	**0.9295**

- Improve prediction accuracy by **4.5%** – **19.5%**

MovieLens Data





The improvement is calculated by $1 - \dfrac{\text{RMSE of method}}{\text{RMSE of grand mean imputation}}$

The "Cold-Start" Problem

Table: Investigation of the "cold-start" problem on MovieLens 10M data.

Root Mean Square Error	Old Ratings	New Ratings	The Entire Testing Set
Funk (2006)	0.8062	1.0039	0.9966
Agarwal and Chen (2009)	1.3324	0.9553	0.9737
Mazumder et al. (2010)	0.8160	1.0252	1.0177
Zhu et al. (2016)	0.8018	1.0189	1.0108
The Proposed Method	**0.7971**	**0.9348**	**0.9295**

- "Old ratings" (4%): from existing users to existing items
- "New ratings" (96%): either from new users or to new items
- The proposed method has better prediction accuracy on the "new ratings" and the entire testing set

Extension to Tensor

- Additional contextual information:
  - Time
  - Locations, companions, promotion strategies
- Context-aware recommender systems
  - Incorporate contextual information to improve predictions
- Diverse applications:
  - Computer-aided diagnosis: Disease recurrence prediction
  - Product forecasting: New product introduction
  - Personalized marketing: Returning customers prediction

- From user-item matrix to user-item-context tensor

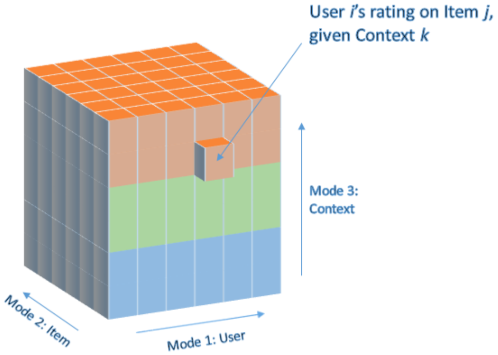


Figure: An illustration of a third-order tensor

Challenges

- **Tensor is not as well-defined as matrix:**
  - One cannot acquire tensor rank and orthonormal bases simultaneously
  - Identifiability issues
  - Unlike matrix, best low-rank approximation does not exist in general
- **Practical challenges:**
  - Higher volume (e.g., 130GB IRI marketing data)
  - Higher sparsity ($< 0.1\%$ observation rate)
  - Forecasting future events (latent factors not available)

Contributions

- Generalize the group-specific method from matrix to tensor
- A scalable algorithm based on maximum block improvement
- Asymptotic consistency without algorithmic global optimum

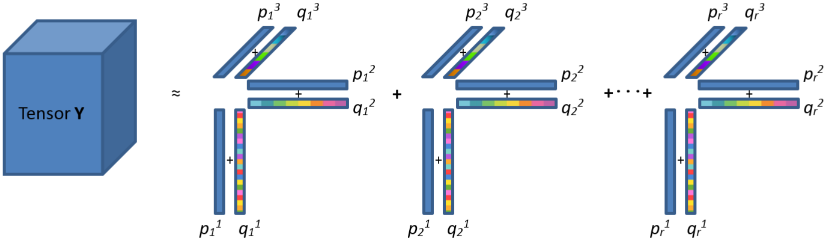


Figure: Illustration of the proposed method

An Example: IRI Marketing Data

- 116.3 million observations of average sales volumes
- 2447 grocery stores
- 161,114 products (consumer packaged goods)
- 30 promotion strategies

Table: Comparison with context-aware recommender systems. Criteria include root mean square error (RMSE), mean absolute error (MAE), and computational time in hours (hrs).

	RMSE	MAE	hrs	Languages
The Proposed Method	0.637	0.209	3.9	Matlab
MF	0.969	0.371	0.7	Matlab
GCPD	0.640	0.229	5.4	Matlab
BPTF	0.782	0.209	8.4	Matlab & C++
libFM	0.705	0.236	0.5	C++

Summary

- A group-specific method
    - **Key idea:** Incorporate dependency
    - **Key advantage:** Address the “cold-start” problem
    - A new algorithm for scalable computing
    - Algorithmic convergence and asymptotic consistency
    - Excellent numerical performance
    - Applicable to both matrix and tensor

References

Bi, X., Qu, A, Wang, J. and Shen, X. (2017). A group-specific recommender system. *Journal of the American Statistical Association,* 112(519):1344-1353.
Bi, X., Qu, A and Shen, X. (2017). Multilayer tensor factorization with applications to recommender systems. *Annals of Statistics,* accepted.

Agarwal, D. and Chen, B.-C. (2009). Regression-based latent factor models. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 19–28. ACM.

Bell, R. M. and Koren, Y. (2007). Scalable collaborative filtering with jointly derived neighborhood interpolation weights. In *Proceedings of the 2007 7th IEEE International Conference on Data Mining*, pages 43–52. IEEE.

Chen, B., He, S., Li, Z., and Zhang, S. (2012). Maximum block improvement and polynomial optimization. *SIAM Journal on Optimization*, 22(1):87–107.

Funk, S. (2006). Netflix update: Try this at home. *URL http://sifter.org/~simon/journal/20061211.html*.

Koren, Y., Bell, R., and Volinsky, C. (2009). Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Lang, K. (1995). Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Conference on Machine Learning*, pages 331–339.

Mao, X., Chen, S. X., and Wong, R. K. W. (2017). Matrix completion with covariate information. *Journal of the American Statistical Association*, to appear.

Mazumder, R., Hastie, T., and Tibshirani, R. (2010). Spectral regularization algorithms for learning large incomplete matrices. *The Journal of Machine Learning Research*, 11:2287–2322.

Mooney, R. J. and Roy, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 195–204. ACM.

Paterek, A. (2007). Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pages 5–8.

Salakhutdinov, R., Mnih, A., and Hinton, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th International Conference on Machine Learning*, pages 791–798. ACM.

Zhu, Y., Shen, X., and Ye, C. (2016). Personalized prediction and sparsity pursuit in latent factor models. *Journal of the American Statistical Association*, 111(513):241–252.

Thank You!

IRI Marketing Data: Original

IRI_KEY	WEEK	SY	GE	VEND	ITEM	UNITS	DOLLARS	F	D	PR
681530	1373	0	1	28400	4874	2	1.98	NONE	0	0
681530	1373	0	1	28400	4853	7	6.93	NONE	0	0
681530	1373	0	1	28400	4361	20	40.00	A	0	1
681530	1373	0	1	28400	4852	1	0.99	NONE	0	0
681530	1373	0	1	28400	4363	5	10.00	A	0	1
681530	1373	0	1	28400	4854	3	2.97	NONE	0	0
681530	1373	0	1	28400	4855	1	0.99	NONE	0	0
681530	1373	0	1	28400	4365	8	16.00	A	0	1

Figure: A snapshot of the original IRI marketing data

- 130 Gigabytes weekly transaction data
  - Collect from 2001 to 2011 and cover 47 U.S. markets

Canonical Polyadic Decomposition

- A Canonical Polyadic (CP) Decomposition represents a tensor $\mathbf{Y}$ as a sum of r rank-1 tensors:

$$\mathbf{Y} \approx \sum_{j=1}^{r} \mathbf{p}_j^1 \circ \mathbf{p}_j^2 \circ \cdots \circ \mathbf{p}_j^d,$$

where $\mathbf{p}_j^k$, $j = 1, \ldots, r$, is the latent factor corresponding to the k-th mode, $k = 1, \ldots, d$.

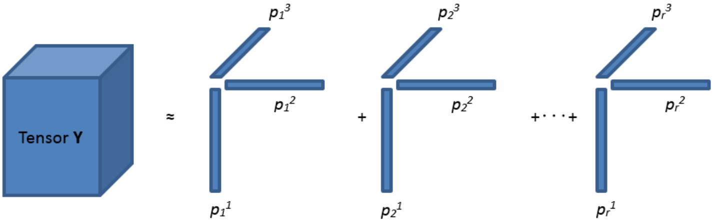


Figure: The CP decomposition of a third-order tensor

The Proposed Model Framework

- For a d-th order tensor, we estimate each element $\hat{y}_{i_1 i_2 \cdots i_d}$ as

$$\hat{y}_{i_1 i_2 \cdots i_d} = \sum_{j=1}^{r} (p^1_{i_1 j} + q^1_{i_1 j})(p^2_{i_2 j} + q^2_{i_2 j}) \cdots (p^d_{i_d j} + q^d_{i_d j})$$

- Here $q^k_{i_k j}$ is the j-th group effect for the i_k-th subject at the k-th mode
- Group members share the same group effects

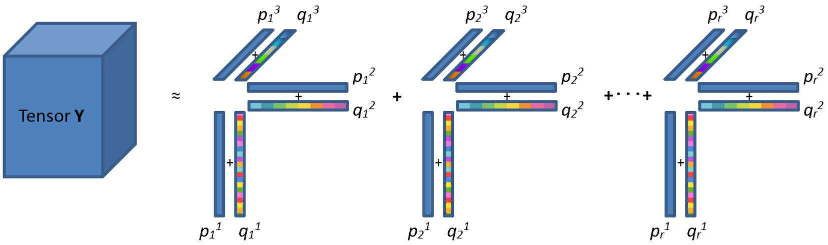


Figure: Illustration of the proposed method

Parameter Estimation

- For each tensor mode $k = 1, \ldots, d$:
  - $\Omega^k_{i_k}$ is the subset of Ω that contains i_k
  - $\mathcal{I}^k_{(u_k)}$ is the subset of subjects in the group u_k
- Parameter estimation through ridge regressions
  - For each subject $i_k = 1, \ldots, n_k$:

$$\hat{\mathbf{p}}^k_{i_k} = \underset{\mathbf{p}^k_{i_k}}{\operatorname{argmin}} \sum_{\Omega^k_{i_k}} (y_{i_1 \cdots i_d} - \hat{y}_{i_1 \cdots i_d})^2 + \lambda \|\mathbf{p}^k_{i_k}\|_2^2,$$

  - For each group $u_k = 1, \ldots, m_k$:

$$\hat{\mathbf{q}}^k_{(u_k)} = \underset{\mathbf{q}^k_{(u_k)}}{\operatorname{argmin}} \sum_{i_k \in \mathcal{I}^k_{(u_k)}} \sum_{\Omega^k_{i_k}} (y_{i_1 \cdots i_d} - \hat{y}_{i_1 \cdots i_d})^2 + \lambda \|\mathbf{q}^k_{(u_k)}\|_2^2$$

- A new block-coordinate-descent algorithm

Algorithm

A revised maximum block improvement algorithm based on Chen et al. (2012)

- *(Latent-factors update)* At the t-th iteration:
  (i) Calculate the improvement of the criterion function, $I^k_{(t)}$, via updating $P^k_{(t-1)}$ to P^k_*.
  (ii) Assign $P^{k_0}_{(t)} \leftarrow P^{k_0}_*$, if $I^{k_0}_{(t)} = \max\{I^1_{(t)}, \ldots, I^d_{(t)}\}$.
- *(Group-effects update)* At the t-th iteration:
  (i) Calculate the improvement of the criterion function, $J^k_{(t)}$, via updating $Q^k_{(t-1)}$ to Q^k_*.
  (ii) Assign $Q^{k_0}_{(t)} \leftarrow Q^{k_0}_*$, if $J^{k_0}_{(t)} = \max\{J^1_{(t)}, \ldots, J^d_{(t)}\}$.
- *(Stopping Criterion)* Stop if

$$\max\{I^1_{(t)}, \ldots, I^d_{(t)}, J^1_{(t)}, \ldots, J^d_{(t)}\} < \varepsilon.$$

Otherwise set $t \leftarrow t + 1$ and repeat.

Statistical Consistency

- $L(\boldsymbol{\Theta}|\mathbf{Y}) \geq 0$, but $\min L(\boldsymbol{\Theta}|\mathbf{Y})$ may not exist
- Instead, suppose the sample estimator $\hat{\boldsymbol{\Theta}}_{|\Omega|}$ satisfies:

$$L(\hat{\boldsymbol{\Theta}}_{|\Omega|}|\mathbf{Y}) \leq \inf_{\Theta} L(\boldsymbol{\Theta}|\mathbf{Y}) + \tau_{|\Omega|},$$

where $\lim_{|\Omega| \to \infty} \tau_{|\Omega|} = 0$

Theorem 2

Let $\rho(\boldsymbol{\Theta}, \boldsymbol{\Theta}_0) = \frac{1}{\sqrt{n_1 \cdots n_d}} \|\boldsymbol{\Theta} - \boldsymbol{\Theta}_0\|_F^2$. *Then we have:*

$$P(\rho(\hat{\boldsymbol{\Theta}}_{|\Omega|}, \boldsymbol{\Theta}_0) \geq \eta_{|\Omega|}) \leq 7 \exp(-c_1 |\Omega| \eta_{|\Omega|}^2),$$

where $c_1 \geq 0$ *is a constant,* $\eta_{|\Omega|} = \max(\varepsilon_{|\Omega|}, \lambda_{|\Omega|}^{1/2})$, *and* $\varepsilon_{|\Omega|} \sim \frac{1}{|\Omega|^{1/2}}$ *is the best possible rate achieved when* $\lambda_{|\Omega|} \sim \varepsilon_{|\Omega|}^2$.

- Same convergence rate as the MLE

Statistical Consistency under General Settings

- Let $l(\mathbf{\Theta}_0, y_{i_1 \cdots i_d})$ be the log-likelihood of $y_{i_1 \cdots i_d}$

Assumption 1

For each $y_{i_1 \cdots i_d}$, suppose $|l(\mathbf{\Theta}_0, y_{i_1 \cdots i_d}) - l(\mathbf{\Theta}, y_{i_1 \cdots i_d})| \leq g(y_{i_1 \cdots i_d}) \|\mathbf{\Theta}_0 - \mathbf{\Theta}\|_F$, where $g(\cdot)$ has a finite moment generating function around 0. In particular, there exists a constant $c_2 > 0$, such that $E\{g^2(y_{i_1, \dots, i_d})\} \leq c_2$ for all $y_{i_1 \cdots i_d}$'s.

- Define the Kullback-Leibler pseudo-distance:

$$\rho^2(\mathbf{\Theta}, \mathbf{\Theta}_0) = \frac{1}{n_1 \cdots n_d} \sum_{i_1=1}^{n_1} \cdots \sum_{i_d=1}^{n_d} \mathrm{E}\left\{ l(\mathbf{\Theta}, y_{i_1 i_2 \cdots i_d}) - l(\mathbf{\Theta}_0, y_{i_1 i_2 \cdots i_d}) \right\}$$

Assumption 2

Suppose there exist $\delta > 0$ and $\beta \in [0, 1)$, such that for a δ-ball centered at $\mathbf{\Theta}_0$, we have $\rho(\mathbf{\Theta}_0, \mathbf{\Theta}) \geq c_3 \|\mathbf{\Theta}_0 - \mathbf{\Theta}\|_F^{\frac{1}{1+\beta}}$, where $c_3 \geq 0$ is a constant.

Statistical Consistency under General Settings

Theorem 3

Suppose Assumptions 1 and 2 hold. Then:

$$P(\rho(\hat{\boldsymbol{\Theta}}_{|\Omega|}, \boldsymbol{\Theta}_0) \geq \eta_{|\Omega|}) \leq 7\exp(-c|\Omega|\eta_{|\Omega|}^2),$$

where $c \geq 0$ is a constant, and $\eta_{|\Omega|} = \max(\varepsilon_{|\Omega|}, \lambda_{|\Omega|}^{1/2})$ with

$$\varepsilon_{|\Omega|} \sim \begin{cases} \left(\frac{1}{|\Omega|^{1/2}}\right)^{\frac{2\omega}{2\omega+1}} & \text{if } \omega > \frac{1}{2} \\ \left(\frac{1}{|\Omega|^{1/2}}\right)^{\omega} & \text{if } \omega \leq \frac{1}{2} \end{cases}$$

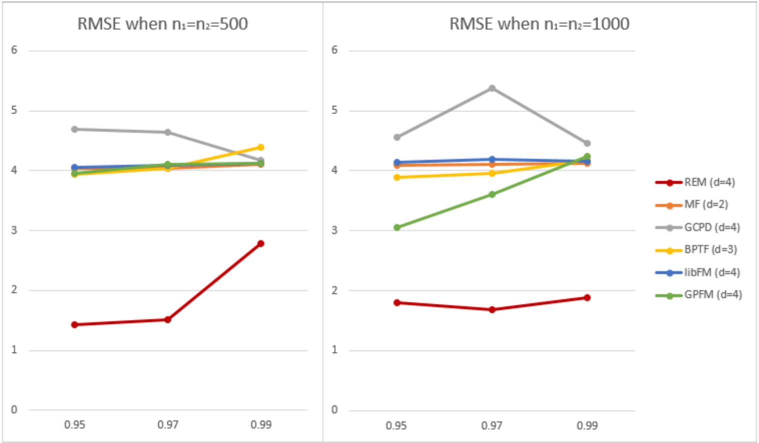
being the best possible rate, achieved when $\lambda_{|\Omega|} \sim \varepsilon_{|\Omega|}^2$. Here $\omega = \alpha/\gamma$, α is the degree of differentiability of $l(\boldsymbol{\Theta}, \cdot)$, and $\gamma = \sum_{k=1}^{d}(n_k + m_k)r$ is the total number of free parameters.

- The best convergence rate $\varepsilon_{|\Omega|} \sim \frac{1}{|\Omega|^{1/2}}$ is achieved when $\omega = \infty$

Simulation Study

- Consider a fourth-order tensor
- Tensor size: $n_1 = n_2 = 500$ or 1000, $n_3 = n_4 = 4$
- The observation rates is 0.01, 0.03, or 0.05
- 10 subgroups for users and items, and 2 subgroups for contextual variables
  - $\mathbf{q}^1_{(u_k)} = \mathbf{q}^2_{(u_k)} = (-5.5 + u_k)\mathbf{1}_r$,
  - $\mathbf{q}^3_{(u_3)} = -0.25 \cdot \mathbf{1}_r$, and $\mathbf{q}^4_{(u_4)} = 0.25 \cdot \mathbf{1}_r$
  - $y_{i_1 i_2 i_3 i_4} = \sum_{j=1}^{r} (p^1_{i_1 j} + q^1_{i_1 j})(p^2_{i_2 j} + q^2_{i_2 j})(p^3_{i_3 j} + q^3_{i_3 j})(p^4_{i_4 j} + q^4_{i_4 j})/4 + \varepsilon$
- Assume that 30% of the items are not available in the training set
- The rest of the settings is the same as Simulation Study 1

Simulation Result Comparison: RMSE



- The x-axis represents the percentage of missing data
- The tensor order of each method is shown in the legend