Transfer Learning for Collaborative Filtering via a Rating-Matrix Generative Model

Bin Li[1]
Joint work with Qiang Yang[2] and Xiangyang Xue[1]

[1]Fudan University, Shanghai 200433, China

[2]Hong Kong University of Science & Technology, Hong Kong, China

Outline

Outline

Sparsity Problem in Collaborative Filtering

- Most CF methods are based on the idea of discovering latent user/item clusters and sharing rating knowledge within clusters
- However, in real-world recommender systems, the rating-matrix is usually too sparse to well cluster

	a	b	c	d	e	f
1	?	3	?	3	2	3
2	3	1	2	2	?	1
3	3	?	2	?	3	1
4	3	?	1	1	?	2
5	2	3	3	?	2	?
6	3	2	?	1	3	2

Sparse rating-matrix

→

Clustering

	a	e	b	f	c	d
2	3	3	1	1	2	2
3	3	3	1	1	2	2
1	2	2	3	3	3	3
5	2	2	3	3	3	3
4	3	3	2	2	1	1
6	3	3	2	2	1	1

Expected clustering result

Outline

1. Motivation
   - Sparsity Problem
   - Knowledge Sharing
2. Rating-Matrix Generative Model
   - Model Construction
   - Learning & Prediction
3. Experiments

Main Idea

Pool the rating data from multiple related CF domains to discover a better latent structure

User Group \ Item Group	A	A	B	B
I	3	3	1	1
I	3	3	1	1
II	2	2	3	3
II	2	2	3	3

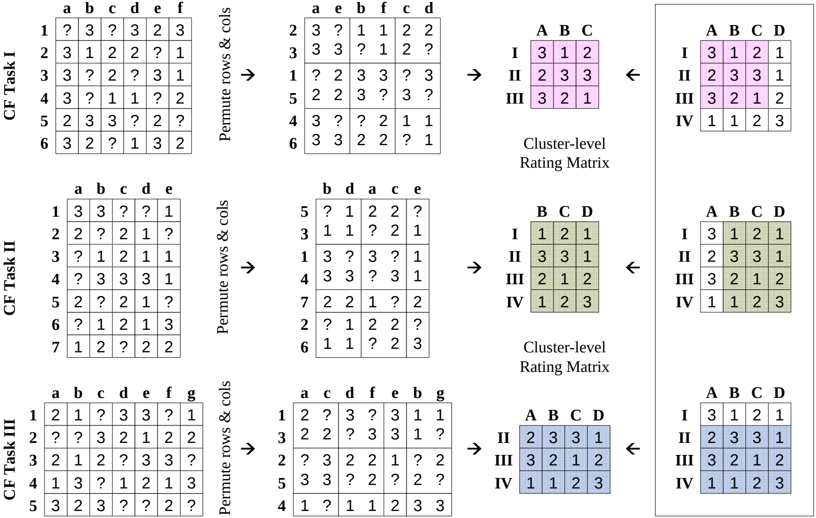
This can be from either **MOVIE** or **BOOK** website

A	Romance movies		Romance books
B	Sci-Fi movies		Sci-Fi books
I	Girls in IMDB	**OR**	Girls in Amazon
II	Boys in IMDB		Boys in Amazon

What "relatedness"

Users are related in Interest; Items are related in Genre

Cluster-Level Rating Matrix as Knowledge Sharing



Outline

Problem Setting

Given Z rating matrices in related domains, in the z-th domain

- User set $U_z = \{u_1^{(z)}, \ldots, u_{n_z}^{(z)}\} \subset \mathcal{U}$
- Item set $V_z = \{v_1^{(z)}, \ldots, v_{m_z}^{(z)}\} \subset \mathcal{V}$
- Rating data $D_z = \{(u_1^{(z)}, v_1^{(z)}, r_1^{(z)}), \ldots, (u_{s_z}^{(z)}, v_{s_z}^{(z)}, r_{s_z}^{(z)})\}$

Assume $\bigcap_z U_z = \emptyset$ and $\bigcap_z V_z = \emptyset$ and ratings in $\{D_1, \ldots, D_Z\}$ should be in the same rating scales R (e.g., $1 - 5$)

Goal

To learn a rating-matrix generative model (RMGM) for the given related tasks on the pooled rating data $\bigcup_z D_z$ and predict missing values

User-Item Joint Mixture Model

Users/Items can simultaneously belong to multiple clusters

- Users may have multiple Personalities
- Items may have multiple Attributes

Suppose there are K user clusters $\{c_{\mathcal{U}}^{(1)}, \ldots, c_{\mathcal{U}}^{(K)}\}$ and L item clusters $\{c_{\mathcal{V}}^{(1)}, \ldots, c_{\mathcal{V}}^{(L)}\}$, the marginal distributions for users and items are

$$P_{\mathcal{U}}(u) = \sum_k P(c_{\mathcal{U}}^{(k)})P(u|c_{\mathcal{U}}^{(k)}), \quad P_{\mathcal{V}}(v) = \sum_l P(c_{\mathcal{V}}^{(l)})P(v|c_{\mathcal{V}}^{(l)})$$

User-Item Joint Mixture Model

$$(u_i^{(z)}, v_i^{(z)}) \sim \sum_{k,l} P(c_{\mathcal{U}}^{(k)})P(c_{\mathcal{V}}^{(l)})P(u|c_{\mathcal{U}}^{(k)})P(v|c_{\mathcal{V}}^{(l)}) \tag{1}$$

Cluster-Level Rating Model

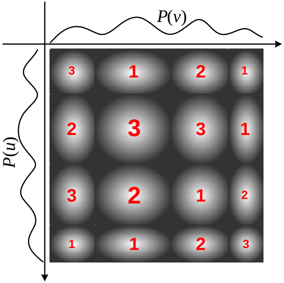
Cluster-Level Rating Model

$$r_i^{(z)} \sim P(r|c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)}) \quad (2)$$

4×4 Cluster-level Rating Matrix

	A	B	C	D
I	3	1	2	1
II	2	3	3	1
III	3	2	1	2
IV	1	1	2	3

Extended to a cluster-level rating model →



Combing (1) and (2) gives rating-matrix generative model (RMGM)

Rating-Matrix Generating Process

CF Task I

	a	e	b	f	c	d
2	3	?	1	1	2	2
3	3	3	?	1	2	?
1	?	2	3	3	?	3
5	2	2	3	?	3	?
4	3	?	?	2	1	1
6	3	3	2	2	?	1

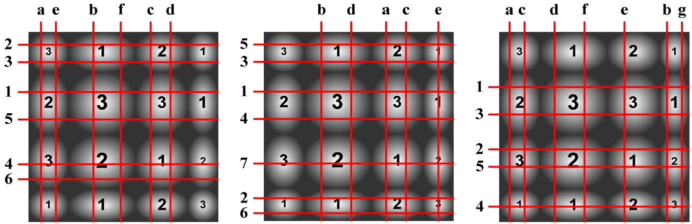
CF Task II

	b	d	a	c	e
5	?	1	2	2	?
3	1	1	?	2	1
1	3	?	3	?	1
4	3	3	?	3	1
7	2	2	1	?	2
2	?	1	2	2	?
6	1	1	?	2	3

CF Task III

	a	c	d	f	e	b	g
1	2	?	3	?	3	1	1
3	2	2	?	3	3	1	?
2	?	3	2	2	1	?	2
5	3	3	?	2	?	2	?
4	1	?	1	1	2	3	3

Draw users and items from the same user-item joint mixture model for related tasks



Outline

1. Motivation
   - Sparsity Problem
   - Knowledge Sharing
2. Rating-Matrix Generative Model
   - Model Construction
   - Learning & Prediction
3. Experiments

Learning the RMGM

Five sets of parameters in RMGM need to learn:
$P(c_{\mathcal{U}}^{(k)})$, $P(c_{\mathcal{V}}^{(l)})$, $P(u|c_{\mathcal{U}}^{(k)})$, $P(v|c_{\mathcal{V}}^{(l)})$, and $P(r|c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)})$
for $k = 1, \ldots, K$; $l = 1, \ldots, L$; $u \in \bigcup_z U_z$; $v \in \bigcup_z V_z$; and $r \in R$

Expectation Maximization (EM) Algorithm

- E-Step: the joint posterior probability $P(c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)}|u_i^{(z)}, v_i^{(z)}, r_i^{(z)})$ is computed using the five sets of parameters
- M-Step: the five sets of parameters are updated based on $P(c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)}|u_i^{(z)}, v_i^{(z)}, r_i^{(z)})$

Note: all the parameters are computed on the pooled rating data $\bigcup_z D_z$

RMGM-Based Prediction

Predicting Missing Values for An Existing User

$$f_R(u_i^{(z)}, v_i^{(z)}) = \sum_r r \sum_{k,l} P(r|c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)}) P(c_{\mathcal{U}}^{(k)}|u_i^{(z)}) P(c_{\mathcal{V}}^{(l)}|v_i^{(z)})$$

Predicting Missing Values for A New User

Solve a quadratic optimization problem to estimate the user-cluster membership $\mathbf{p}_{u^{(z)}} \in \mathbb{R}^K$ for $u^{(z)}$ based on the given ratings $\mathbf{r}_{u^{(z)}}$

$$\min_{\mathbf{p}_{u^{(z)}}} \left\| [\mathbf{B}\mathbf{P}_{V_z}]^\top \mathbf{p}_{u^{(z)}} - \mathbf{r}_{u^{(z)}} \right\|^2_{\mathbf{W}_{u^{(z)}}}, \quad \text{s.t. } \mathbf{p}_{u^{(z)}}^\top \mathbf{1} = 1$$

where $\mathbf{B}_{kl} = \sum_r r P(r|c_{\mathcal{U}}^{(k)}, c_{\mathcal{V}}^{(l)})$ and $[\mathbf{P}_{V_z}]_{li} = P(c_{\mathcal{V}}^{(l)}|v_i^{(z)})$

Outline

Experimental Setup

Compared Methods

- Pearson Correlation Coefficients (PCC) - Baseline
- Flexible Mixture Model (FMM) - Single-task
- Rating-Matrix Generative Model (RMGM) - Multi-task

Data Sets

Randomly select 500 users and 1000 items from each of the following real-world data sets: 1) MovieLens (Movie); 2) EachMovie (Movie); 3) Book-Crossing (Book) - Three 500 $\times$ 1000 rating matrices

Evaluation Protocol

- First 100/200/300 users for training; last 200 users for testing
- Given 5/10/15 observable ratings for each test user
- Evaluation metric: Mean Absolute Error (MAE)

Experimental Results

Table: MAE Comparison on MovieLens (ML)

Train	Method	Given5	Given10	Given15
ML100	PCC	0.930	0.908	0.895
	FMM	0.908	0.868	0.846
	RMGM	0.868	0.822	0.808
ML200	PCC	0.934	0.899	0.888
	FMM	0.890	0.863	0.847
	RMGM	0.859	0.821	0.806
ML300	PCC	0.935	0.896	0.888
	FMM	0.885	0.868	0.846
	RMGM	0.857	0.820	0.804

Experimental Results

Table: MAE Comparison on EachMovie (EM)

Train	Method	Given5	Given10	Given15
EM100	PCC	0.996	0.952	0.936
	FMM	0.969	0.937	0.924
	RMGM	0.942	0.908	0.895
EM200	PCC	0.983	0.943	0.930
	FMM	0.955	0.933	0.923
	RMGM	0.934	0.905	0.890
EM300	PCC	0.976	0.937	0.933
	FMM	0.952	0.930	0.924
	RMGM	0.934	0.906	0.890

Experimental Results

Table: MAE Comparison on Book-Crossing (BX)

Train	Method	Given5	Given10	Given15
BX100	PCC	0.617	0.599	0.600
	FMM	0.619	0.592	0.583
	RMGM	0.612	0.583	0.573
BX200	PCC	0.621	0.612	0.620
	FMM	0.617	0.602	0.596
	RMGM	0.615	0.591	0.583
BX300	PCC	0.621	0.619	0.630
	FMM	0.615	0.604	0.596
	RMGM	0.612	0.590	0.581

Summary

Summary

- Relate in cluster-level rating patterns
- Bridge via a cluster-level rating model
- Transfer rating knowledge
- Benefit from one another
- Alleviate sparsity problem

Future Work

- Quantify the “relatedness”
- Asymmetric setting: Dense $\longrightarrow$ Sparse