Self-Introduction

• Yong Zheng, Ph.D. Candidate, DePaul University
• Research: Context-aware Collaborative Filtering
• Supervisor: Dr. Bamshad Mobasher
• Currently 5th Year at DePaul
• Expected Graduation: Summer, 2015
Outline

• Context-aware Recommender Systems (CARS)
• Context-aware Collaborative Filtering
• Contextual SLIM (CSLIM) Algorithms
• Current Work and Ongoing Work
• Challenges and Work in the Future
• Context-aware Recommender Systems (CARS)
• Context-aware Collaborative Filtering
• Contextual SLIM (CSLIM) Algorithms
• Current Work and Ongoing Work
• Challenges and Work in the Future
Context-aware Recommender Systems (CARS)

• What is Context?
  “Any information that can be used to characterize the situation of an entity”, Abowd and Dey, 1999

Sample of popular contextual variables in Recommender Systems (RS):
Time (weekday/weekend/holiday), Location, Companion (kids/family), Mood (happy/sad/excited/upset), Weather, etc
Context-aware Recommender Systems (CARS)
Context-aware Recommender Systems (CARS)

• Motivation Behind
Users’ preferences change from contexts to contexts

• Basic Assumptions
RS should learn users’ preferences in contexts c from others’ preferences in the same c, e.g. context-aware collaborative filtering.

• Challenges
How to incorporate contexts into RS? E.g. CF, MF, etc
How to develop effective CARS?
How to Interpret contextual effects from the model?
Sparsity problems in CARS.
Outline

• Context-aware Recommender Systems (CARS)
• Context-aware Collaborative Filtering
• Contextual SLIM (CSLIM) Algorithms
• Current Work and Ongoing Work
• Challenges and Work in the Future
Context-aware Collaborative Filtering (CACF)

• Collaborative Filtering (CF)
  CF is one of the most popular recommendation algorithms in the traditional RS.
  1). Memory-based CF, e.g., ItemKNN CF
  2). Model-based CF, e.g., matrix factorization
  3). Hybrid CF

• Context-aware Collaborative Filtering (CACF)
Intro. Collaborative Filtering (CF)

There are three series of most popular CF algorithms

• Neighborhood-based CF
  ItemKNN-based CF, Sarwar, et al, 2001

• Matrix factorization (MF)-based CF
  Matrix Factorization, Sarwar, et al., 2000; Koren, et al., 2009
  Tensor Factorization, Symeonidis, et al., 2008

• Sparse Linear Method (SLIM), Ning, et al., 2011
Intro. Collaborative Filtering (CF)

- Neighborhood-based CF
- ItemKNN-based CF (ItemKNN), Sarwar, et al, 2001
- UserKNN-based CF (UserKNN), Resnick, et al, 1994

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
</tr>
</thead>
<tbody>
<tr>
<td>U1</td>
<td>2</td>
<td>?</td>
<td>5</td>
<td>4</td>
</tr>
<tr>
<td>U2</td>
<td>4</td>
<td>3</td>
<td>5</td>
<td>3</td>
</tr>
<tr>
<td>U3</td>
<td>4</td>
<td>3</td>
<td>1</td>
<td>4</td>
</tr>
</tbody>
</table>
Intro. Collaborative Filtering (CF)

• ItemKNN, Sarwar, et al, 2001

Rating Prediction in ItemKNN: \( P_{u,i} = \frac{\sum_{j \in N_i} R_{u,j} \times \text{sim}(i,j)}{\sum_{j \in N_i} \text{sim}(i,j)} \)

Pros:
1). Straightforward
2). The explainability of the results

Cons:
1). Sparsity problem
2). Predictions rely on similarity (from co-ratings)
Intro. Collaborative Filtering (CF)


Both users and items are represented by a vector of weights on each Latent factors.
- User vector ➞ How users like some features
- Item vector ➞ How items obtain/capture those features
Intro. Collaborative Filtering (CF)


![Matrix Factorization Diagram]

**Pros:**
1. Work effectively and efficiently (e.g., MapReduce)
2. Being easy/flexible to incorporate side information

**Cons:**
1. Cold-start problem
2. Difficult to interpret the Latent factors
Intro. Collaborative Filtering (CF)

- Sparse Linear Model (SLIM), Ning, et al., 2011

\[ S_{i,j} = R_{i,:} \cdot W_{:,j} = \sum_{h=1, h \neq j}^{N} R_{i,h}W_{h,j} \]

Matrix \( R \) = User-Item Rating matrix; \( W \) = Item-Item Coefficient matrix
Intro. Collaborative Filtering (CF)

- **Sparse Linear Model (SLIM),** Ning, et al., 2011

Rating Prediction in ItemKNN:

\[
P_{u,i} = \frac{\sum_{j \in N_i} R_{u,j} \times \text{sim}(i, j)}{\sum_{j \in N_i} \text{sim}(i, j)}
\]

Ranking Prediction in SLIM-I:

\[
\hat{S}_{i,j} = R_{i,:} \cdot W_{:,j} = \sum_{h=1, h \neq j}^{N} R_{i,h} W_{h,j}
\]

Results of Comparison between ItemKNN and SLIM-I:
1). Coefficients in W are similar to Item-item similarities in ItemKNN;
2). SLIM-I removed the normalization function;
3). The item-item coefficients DO NOT rely on co-ratings;
Intro. Collaborative Filtering (CF)

• Sparse Linear Model (SLIM), Ning, et al., 2011

\[
\hat{S} = \begin{bmatrix}
    u_1 & 0 & 2 & 0 & \ldots & 2 \\
    u_2 & ? & 3 & 2 & \ldots & 0 \\
    \vdots & \vdots & \vdots & \vdots & \ldots & \vdots \\
    u_M & 0 & 4 & 2 & \ldots & 2 \\
\end{bmatrix}
\times
\begin{bmatrix}
    t_1 & 0 & \ldots \\
    t_2 & 0 & \ldots \\
    \vdots & \vdots & \vdots \\
    t_N & \ldots & 0 \\
\end{bmatrix}
\]

SLIM-I: Matrix W is an Item-Item Coefficient Matrix \(\xRightarrow{\text{ItemKNN}}\)
SLIM-U: Matrix W is a User-User Coefficient Matrix \(\xRightarrow{\text{UserKNN}}\)

Pros:
1). Avoid unreliable similarity calculations in ItemKNN/UserKNN
3. Work effectively and efficiently; Obtain explainability.

Cons: Cold-start problems
## CF and Context-aware CF (CACF)

<table>
<thead>
<tr>
<th>CF</th>
<th>Neighborhood-based Collaborative Filtering</th>
<th>Matrix Factorization</th>
<th>SLIM</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pros</strong></td>
<td>Explainability</td>
<td>Effectiveness</td>
<td>Effectiveness</td>
</tr>
<tr>
<td><strong>Cons</strong></td>
<td>Sparsity Problem</td>
<td>Weak Explainability</td>
<td>Cold-start Problem</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Incorporate</th>
<th>Contexts</th>
</tr>
</thead>
</table>

|------------------|--------------------------------------------|----------------------|--------------------------|
CACF: Differential Context Modeling (DCM)

DCM incorporates contexts into neighborhood-based CF (e.g., UserKNN, ItemKNN, Slope One) by applying contextual constraints to different functional components in the algorithms.

For example, the similarity of items can be calculated within the same (or similar) contexts, instead of calculations without considering contexts.

Drawbacks: Overfitting in Top-N Recommendations
CAMF incorporates contexts by adding contextual rating deviations, which is actually a dependent way to model the contextual effects.

\[
\hat{R}_{i,j} = \mu + b_u + b_i + p_u^T q_i
\]

Rating Prediction in CF:

\[
\hat{R}_{i,j, \{c_1, c_2, \ldots, c_N\}} = \mu + b_u + \sum_{j=1}^{N} B_{ijc_j} + p_u^T q_i
\]

Rating Prediction in CAMF:

bi is item’s rating bias, which is replaced by an aggregation of item’s rating biases in different contextual situations. Drawbacks: Difficult to interpret latent factors.
CACF: Tensor Factorization (TF)

TF is an independent way, which directly considers each contextual variable as an individual context in the multi-dimensional space.

Drawbacks:
1). Computational costs increase exponentially with the number of contexts increases.
2). Contexts are usually dependent rather than fully independent.
• Context-aware Recommender Systems (CARS)
• Context-aware Collaborative Filtering
• Contextual SLIM (CSLIM) Algorithms
• Current Work and Ongoing Work
• Challenges and Work in the Future
CACF: Contextual SLIM (CSLIM) Algorithms

CSLIM: Incorporate contexts into SLIM

Ranking Prediction in SLIM-I:
\[ \hat{S}_{i,j} = R_{i,:} \cdot W_{:,j} = \sum_{h=1, h \neq j}^{N} R_{i,h} W_{h,j} \]

CSLIM has a uniform ranking prediction:
\[ \hat{S}_{i,j,c} = \sum_{h=1, h \neq j}^{N} R_{i,h,c} W_{h,j} \]

CSLIM aggregates contextual ratings with item-item coefficients. However, there are two main points:

1). The rating to be aggregated should be placed under same c;
2). Accordingly, W indicates coefficients under same contexts;
Ranking prediction in CSLIM: 

\[ \hat{S}_{i,j,c} = \sum_{h=1, h \neq j}^{N} R_{i,h,c} W_{h,j} \]

The remaining problem is how to calculate \( R_{i,h,c} \), it is because context-aware data set is usually sparse – it is not guaranteed that user also rated other items under the same contexts \( c \).

We use an estimation to obtain \( R_{i,h,c} \): 
1). Deviation-Based CSLIM 
2). Similarity-Based CSLIM
Outline

- Context-aware Recommender Systems (CARS)
- Context-aware Collaborative Filtering
- Contextual SLIM (CSLIM) Algorithms
- Current Work and Ongoing Work
- Challenges and Work in the Future
Deviatıon-Based CSLIM Algorithms

Research Status: Completed.

Publications:


• Y. Zheng. "Deviation-Based and Similarity-Based Contextual SLIM Recommendation Algorithms", ACM RecSys, Silicon Valley, USA, 2014

• Y. Zheng, B. Mobasher, R. Burke. "Deviation-Based Contextual SLIM Recommenders", ACM CIKM, Shanghai, China, 2014
Deviation-Based CSLIM Algorithms

Ranking prediction in CSLIM:  
\[ \hat{S}_{i,j,c} = \sum_{h=1, h \neq j}^{N} R_{i,h,c} W_{h,j} \]

Example: CSLIM-I-Cl,  
\[ \hat{R}_{i,j,c} = R_{i,j} + \sum_{l=1}^{L} D_{j,l} C_{l} \]

R = non-contextual Rating Matrix
D = Contextual Rating Deviation Matrix
W = Item-item Coefficient Matrix
C = a binary context vector, as below

<table>
<thead>
<tr>
<th>Weekend</th>
<th>Weekday</th>
<th>At Home</th>
<th>At Park</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Center for Web Intelligence, DePaul University, Chicago, USA
Several CSLIM algorithms can be built:

1). Build upon ItemKNN or UserKNN

- ItemKNN → CSLIM-I models, W is item-item matrix
- UserKNN → CSLIM-U models, W is user-user matrix

2). How to define Contextual Rating Deviations?

- Dependent with items → D is a Context-Item (CI) matrix
- Dependent with users → D is a CU matrix
- Individual deviations → D is a vector of deviations

So, there are six models in total:

- CSLIM-I-CI, CSLIM-I-CU, CSLIM-I-C;
- CSLIM-U-CI, CSLIM-U-CU, CSLIM-U-C;
Deviation-Based CSLIM Algorithms (Optional)

Stay tuned:

Y. Zheng, B. Mobasher, R. Burke. "Deviation-Based Contextual SLIM Recommenders", ACM CIKM, Shanghai, China, 2014

New Updates:

1). Discover patterns how to select appropriate CSLIM algorithms in advance based on data characteristics;
2). Develop general CSLIM algorithms, so that deviations can be estimated from any contexts;
Deviation-Based CSLIM Algorithms

Deviation-Based CSLIM algorithms have been demonstrated to outperform the state-of-the-art CACF algorithms in terms of Top-N evaluation metrics, e.g., precision, recall, MAP, NDCG, etc. (over 5 data sets)
Similarity-Based CSLIM Algorithms

Research Status: Ongoing.

Ranking prediction in CSLIM:

\[ \hat{S}_{i,j,c} = \sum_{h=1, h \neq j}^{N} R_{i,h,c} W_{h,j} \]

In this case, \( R_{i,h,c} \), will be estimated by fusing with similarity of two contexts.

<table>
<thead>
<tr>
<th>Contexts</th>
<th>( R_{u,i,c} )</th>
<th>Companion</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Kids</td>
<td>Family</td>
</tr>
<tr>
<td>c-1</td>
<td>?</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>c-2</td>
<td>5</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>c-3</td>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

\[ R_{u,i,c-1} = R_{u,i,c-2} \times \text{Sim}(\text{Kids, Partner}) \times \text{Sim}(\text{Weekday, Weekend}) \]

Of course, there could be many more ways to model sim.
Outline

• Context-aware Recommender Systems (CARS)
• Context-aware Collaborative Filtering
• Contextual SLIM (CSLIM) Algorithms
• Current Work and Ongoing Work
• Challenges and Work in the Future
Challenges

1. Effectiveness

Several reviewers argued about scalability of CSLIM. There are several solutions to reduce costs:
If Large # users/items $\Rightarrow$ KNN, select Top-K neighbors;
If Large # contexts $\Rightarrow$ select relevant ones only;
Besides, it is possible to use coordinate descent for optimization instead of gradient descent.

2. Similarity of contexts

Most related work on the calculations of similarity contexts rely on the existing contextual ratings.
Pros and Cons

1. Pros
Effectiveness and also Explainability

2. Cons
Not all cells can be learned!

Sol: Factorize contexts/items to represent them by vectors
Future Work

1. Explainability
Try to explore and discover the explainability of CSLIM algorithms, e.g., discovering emotional effects compared with other CARS algorithms (Zheng, et al, 2013, RecSys)

2. Scalability
Try to evaluate CSLIM on large data sets.
Notice: there are no real+large data in CARS domain!

3. Similarity-Based CSLIM models
Try to model and learn similarity of contexts in different ways to develop similarity-based CSLIM models.
Acknowledgement

• NSF Student Funding
• ACM RecSys Doctoral Symposium
• Mentors: Dr. Pablo Castells and Dr. Irwin King
• My supervisor: Dr. Bamshad Mobasher
• My parents
Any Questions or Suggestions?

Yong Zheng, DePaul University, Chicago, USA

Oct 10, Doctoral Symposium @ ACM RecSys