Point-of-Interest Recommendation Systems in Location-Based Social Networks

Tony Song
LBSN? POI?

Location-based social networks (LBSNs):
Social networks tied with geographical information

Point-of-interest (POI):
A specific location where users show interest (restaurant, bar, cinema, etc.)
Motivation - Big question

Location Recommendation

- Current Location
- Location Prediction

<table>
<thead>
<tr>
<th>Location</th>
<th>Probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>22%</td>
<td>22%</td>
</tr>
<tr>
<td>35%</td>
<td>31%</td>
</tr>
<tr>
<td>12%</td>
<td>62%</td>
</tr>
<tr>
<td>7%</td>
<td>22%</td>
</tr>
<tr>
<td>37%</td>
<td>22%</td>
</tr>
<tr>
<td>19%</td>
<td>71%</td>
</tr>
<tr>
<td>11%</td>
<td>11%</td>
</tr>
</tbody>
</table>
Table of Contents

● Backgrounds
  ○ Tensor
  ○ Matrix Factorization
  ○ Markov Chain

● Algorithms
  ○ FPMC
  ○ FPMC-LR
  ○ TAD-FPMC

● Performance Analysis
Table of Contents

- Backgrounds
  - Tensor
  - Matrix Factorization
  - Markov Chain
- Algorithms
  - FPMC
  - FPMC-LR
  - TAD-FPMC
- Performance Analysis
Tensor

- Generalization of vectors
- Multi-dimensional arrays in Java
Matrix Factorization (MF)

Review:

Matrix multiplication

\[
\begin{pmatrix}
3 \\
2
\end{pmatrix}
\times
\begin{pmatrix}
1 & 3
\end{pmatrix}
= 
\begin{pmatrix}
3 & 9 \\
2 & 6
\end{pmatrix}
\]
Matrix Factorization (MF)

A technique to factor a matrix as a product of multiple matrices.

\[
\begin{bmatrix}
3 & 9 & 3 \\
2 & 6 & 2 \\
4 & 12 & 4
\end{bmatrix} =
\begin{bmatrix}
3 \\
2 \\
4
\end{bmatrix} 
\times 
\begin{bmatrix}
1 & 3 & 1
\end{bmatrix}
\]
Question:
Given a matrix of user ratings for different locations, **predict empty ratings** 0 (n/a)

<table>
<thead>
<tr>
<th>Location</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>5</td>
<td>4</td>
<td>0 (n/a)</td>
</tr>
<tr>
<td>Theatre</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>0 (n/a)</td>
<td>0 (n/a)</td>
<td>2</td>
</tr>
</tbody>
</table>
Matrix Factorization (MF)

Matrix $L$

<table>
<thead>
<tr>
<th>Location\Feature</th>
<th>shopping</th>
<th>food</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Theatre</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Matrix $U$

<table>
<thead>
<tr>
<th>Feature\User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>shopping</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>food</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
</tbody>
</table>

Location\User

<table>
<thead>
<tr>
<th>Location\User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>5</td>
<td>4</td>
<td>0 ($n/a$)</td>
</tr>
<tr>
<td>Theatre</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>0 ($n/a$)</td>
<td>0 ($n/a$)</td>
<td>2</td>
</tr>
</tbody>
</table>
Matrix Factorization (MF)

Matrix $L$

<table>
<thead>
<tr>
<th>Location \ Feature</th>
<th>shopping</th>
<th>food</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Theatre</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Matrix $U$

<table>
<thead>
<tr>
<th>Feature \ User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>shopping</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>food</td>
<td>0.8</td>
<td>0.6</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Location \ User

<table>
<thead>
<tr>
<th>Location \ User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>5</td>
<td>4</td>
<td>0 (n/a)</td>
</tr>
<tr>
<td>Theatre</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>0 (n/a)</td>
<td>0 (n/a)</td>
<td>2</td>
</tr>
</tbody>
</table>
Matrix Factorization (MF)

Matrix $L$

<table>
<thead>
<tr>
<th>Location \ Feature</th>
<th>shopping</th>
<th>food</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>1</td>
<td>5</td>
</tr>
<tr>
<td>Theatre</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>

Matrix $U$

<table>
<thead>
<tr>
<th>Feature \ User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>shopping</td>
<td>1</td>
<td>1</td>
<td>0.1</td>
</tr>
<tr>
<td>food</td>
<td>0.8</td>
<td>0.6</td>
<td>0.75</td>
</tr>
</tbody>
</table>

Tony's Sushi: $1 \cdot 0.1 + 5 \cdot 0.75 = 4$

Dollar Store: $0 \ (n/a)$

Theatre: $2$
Matrix Factorization (MF)

What’s the point?
Based on the calculated predictions, we can make **recommendations**.

<table>
<thead>
<tr>
<th>Location\User</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tony's Sushi</td>
<td>5</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Theatre</td>
<td>3</td>
<td>3</td>
<td>1</td>
</tr>
<tr>
<td>Dollar Store</td>
<td>5</td>
<td>5</td>
<td>2</td>
</tr>
</tbody>
</table>
Markov Chain (MC)

Stochastic model to represent different states (locations) and their transitional possibilities.

<table>
<thead>
<tr>
<th></th>
<th>Home</th>
<th>School</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>0.3</td>
<td>0.7</td>
</tr>
<tr>
<td>School</td>
<td>0.9</td>
<td>0.1</td>
</tr>
</tbody>
</table>
Markov Chain (MC) for Sets

Each state is now a set of locations instead of a single location.

<table>
<thead>
<tr>
<th>Location</th>
<th>Home</th>
<th>School</th>
<th>Cafe</th>
<th>#</th>
</tr>
</thead>
<tbody>
<tr>
<td>Home</td>
<td>1/2</td>
<td>1/2</td>
<td>1/2</td>
<td>2</td>
</tr>
<tr>
<td>School</td>
<td>1/1</td>
<td>0/1</td>
<td>1/1</td>
<td>1</td>
</tr>
<tr>
<td>Cafe</td>
<td>0/1</td>
<td>1/1</td>
<td>0/1</td>
<td>1</td>
</tr>
</tbody>
</table>
Table of Contents

- Backgrounds
  - Tensor
  - Matrix Factorization
  - Markov Chain
- Algorithms
  - FPMC
  - FPMC-LR
  - TAD-FPMC
- Performance Analysis
POI Recommendation History over Time

**MF / MC**
Traditionally used for recommendation systems

**FPMC (2010)**
Combining MF + MC

**FPMC-LR (2013)**
FPMC + Only considering nearby locations

**TAD-FPMC (2017)**
FPMC + Capturing complex behavior of users over time
Factorized Personalized Markov Chain (FPMC)

Problem of Markov chain
One Markov chain is used for all users (Recommendation is not personalized)

How FPMC solves the problem
Having a Markov chain for each user
Factorized Personalized Markov Chain (FPMC)

1. **Make a Markov chain for each user**
2. Stack them one onto another
3. Factor the cubic tensor using tensor factorization
4. Calculate predictions using the factors

<table>
<thead>
<tr>
<th>From location</th>
<th>User 1</th>
<th>User 2</th>
<th>User 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.2</td>
<td>?</td>
<td>?</td>
<td>0.2</td>
</tr>
<tr>
<td>0.5</td>
<td>0.1</td>
<td>?</td>
<td>0.4</td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>0.6</td>
<td>0.7</td>
</tr>
<tr>
<td>0.3</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>?</td>
<td>?</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>?</td>
<td>?</td>
<td>0.1</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>To location</th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

19/33
Factorized Personalized Markov Chain (FPMC)

1. Make a Markov chain for each user
2. **Stack them one onto another**
3. Factor the cubic tensor using tensor factorization
4. Calculate predictions using the factors
Factorized Personalized Markov Chain (FPMC)

1. Make a Markov chain for each user
2. Stack them one onto another
3. **Factor the cubic tensor using tensor factorization**
4. Calculate predictions using the factors
Factorized Personalized Markov Chain (FPMC)

1. Make a Markov chain for each user
2. Stack them one onto another
3. Factor the cubic tensor using tensor factorization
4. **Calculate predictions using the factors**

Probability of user $u$ going from location $l$ to location $i$

$$x_{l,i,u} \approx \sum_{p=1}^{F} \sum_{q=1}^{F} \sum_{r=1}^{F} g_{pqr} L_{lp} I_{iq} U_{ur}$$
FPMC with Localized Region Constraint (FPMC-LR)

Problems of FPMC

- Generic method for any recommendation systems
- Computationally expensive

How FPMC-LR solves the problem

- POI-specific method
- Only consider nearby locations when predictions are made
FPMC with Localized Region Constrain (FPMC-LR)

1. Make a Markov chain for each user
2. Stack them one onto another
3. Factor the **cubic tensor** using tensor factorization
4. Calculate predictions using the factors
Time-Aware Decaying FPMC (TAD-FPMC)

Problem of FPMC-LR
Complex user behavior over time is not incorporated

How TAD-FPMC solves the problem
Adding time variable when calculating the recommendations
Time-Aware Decaying FPMC

1. Make a Markov chain for each user
2. Stack them one onto another
3. Factor the **cubic tensor** using tensor factorization
4. Calculate predictions using the factors
Table of Contents

- Backgrounds
  - Tensor
  - Matrix Factorization
  - Markov Chain
- Algorithms
  - FPMC
  - FPMC-LR
  - TAD-FPMC
- Performance Analysis
Performance Analysis - Evaluation Metric

1. Each Algorithm recommends a list of top-N places.

2. Recommendation is correct if the user indeed visited any place in the list at time $t$. 
Performance Analysis - Evaluation Metric

Precision = \frac{\text{# of correct predictions}}{\text{# of recommendation rounds}} = \frac{1}{5}

Top-4 Recommendations

5 Rounds
Performance Analysis

POI specific models (FPMC-LR & TAD-FPMC) show much better performance.
Variations of TAD-FPMC far outperform all existing models
Conclusion

MF / MC
Traditionally used for recommendation systems

FPMC (2010)
Combining MF + MC

FPMC-LR (2013)
FPMC + Only considering nearby locations

TAD-FPMC (2017)
FPMC + Capturing complex behavior of users over time
Questions?
Contact

Tony Song
songx823@morris.umn.edu
https://github.com/frogrammer
References


