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

Tony Song

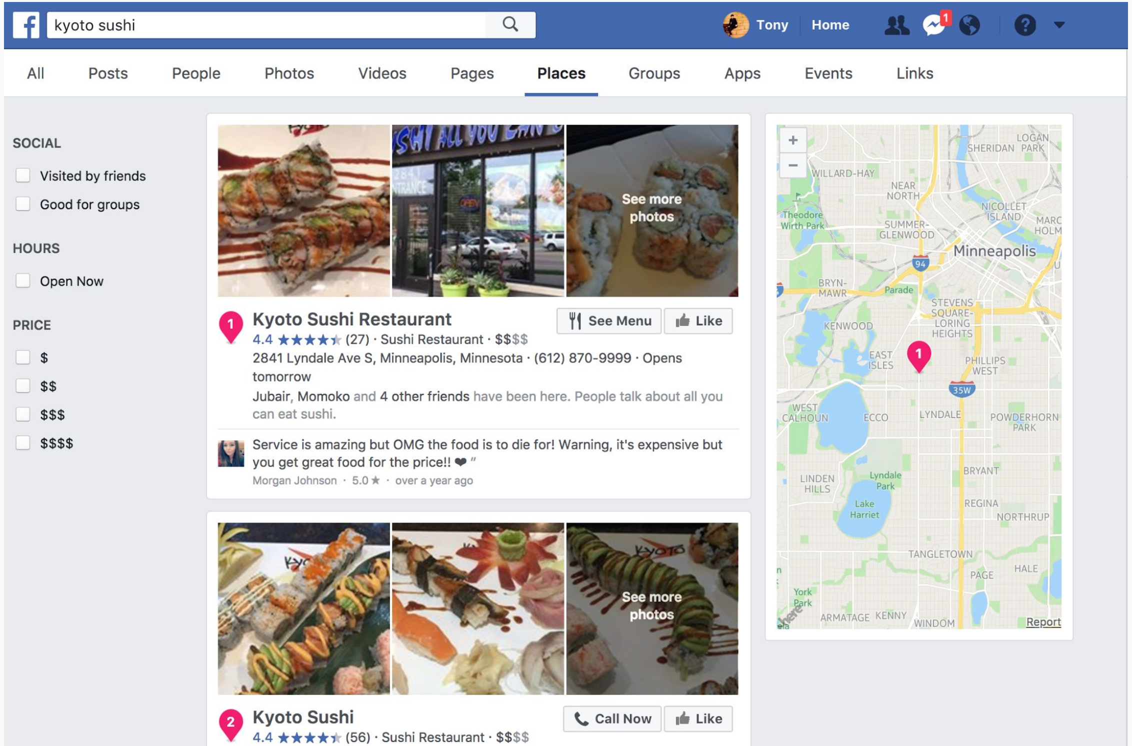
LBSN? POI?

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

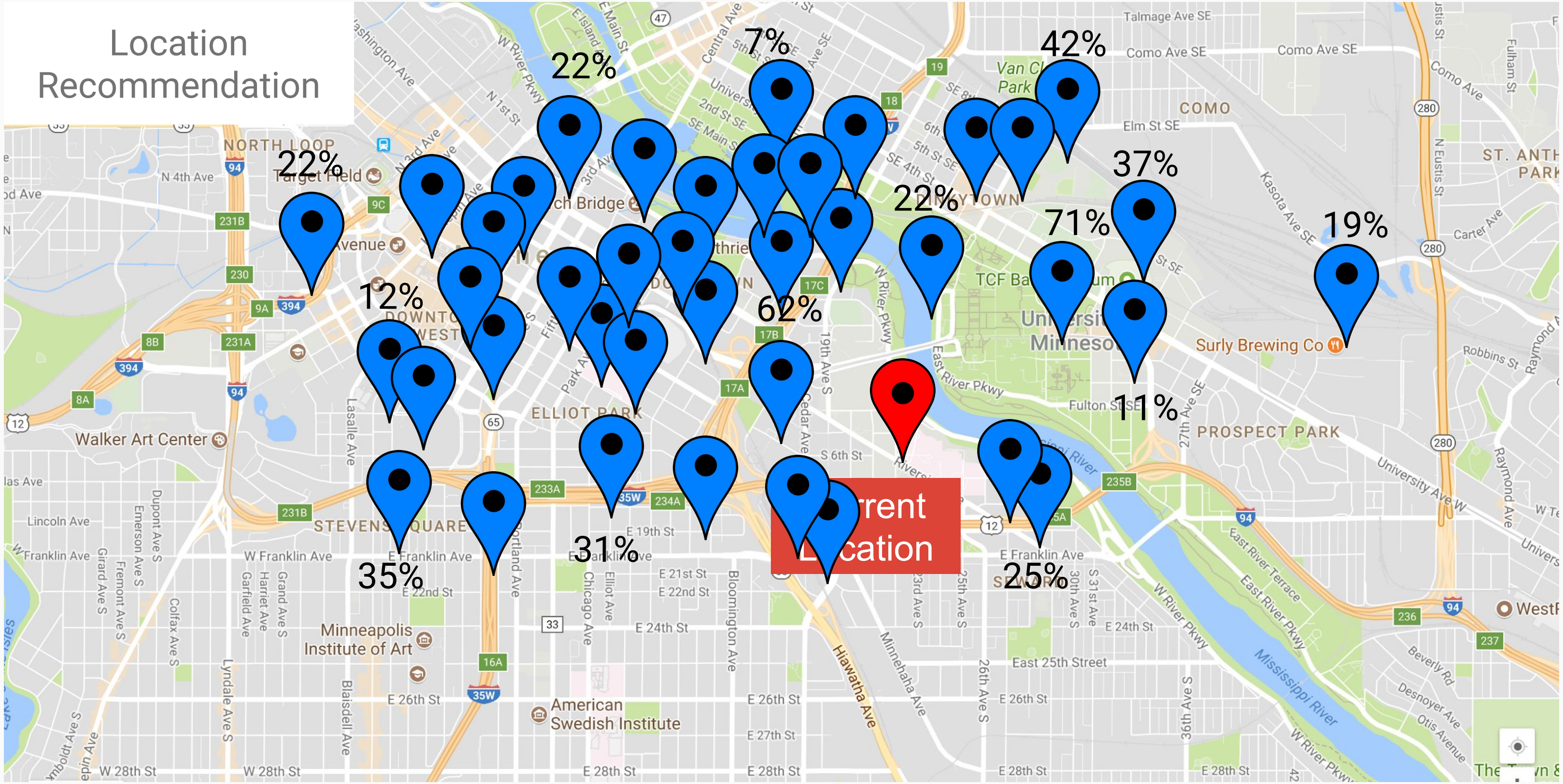
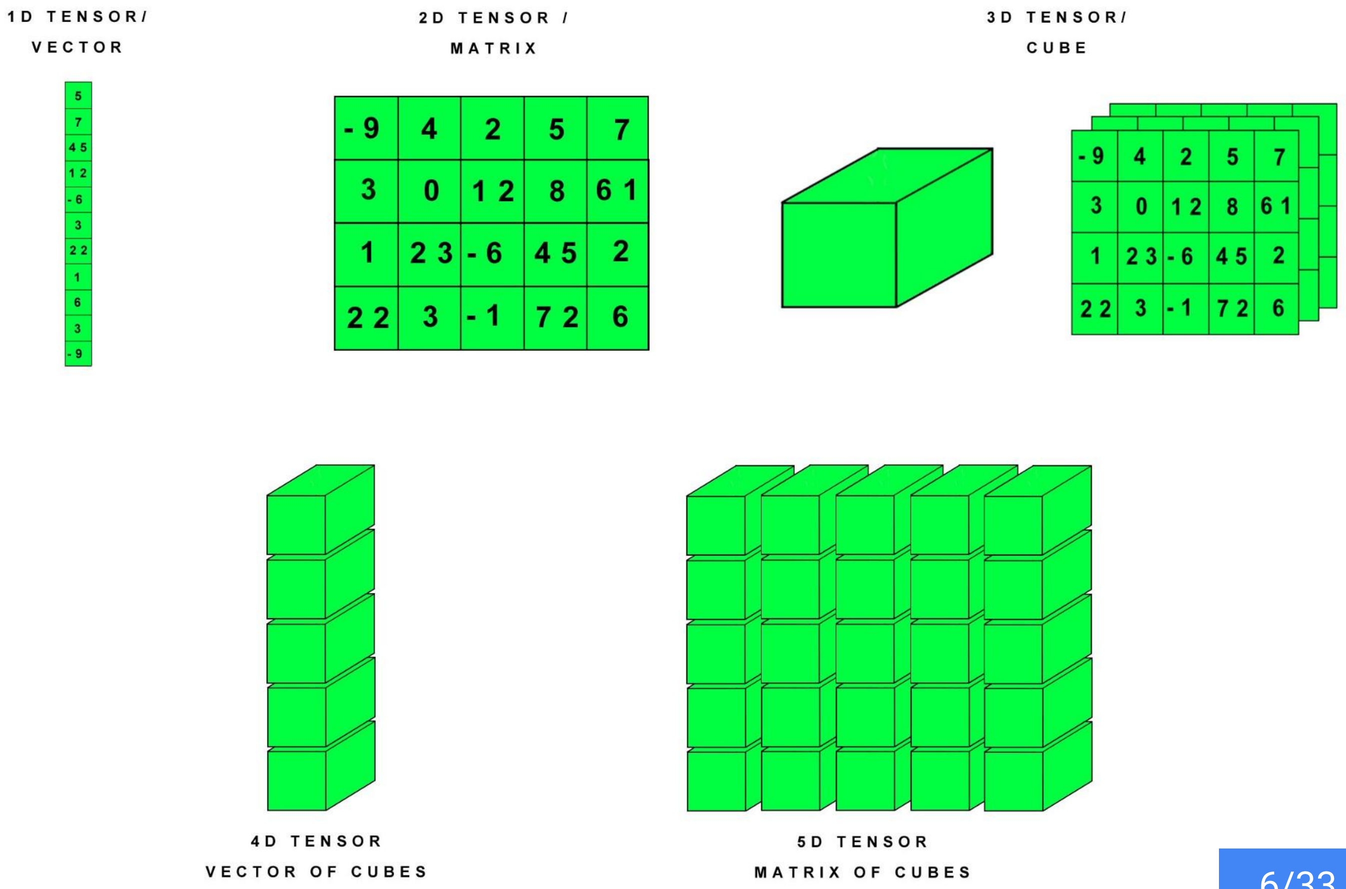


Table of Contents

Table of Contents

Tensor

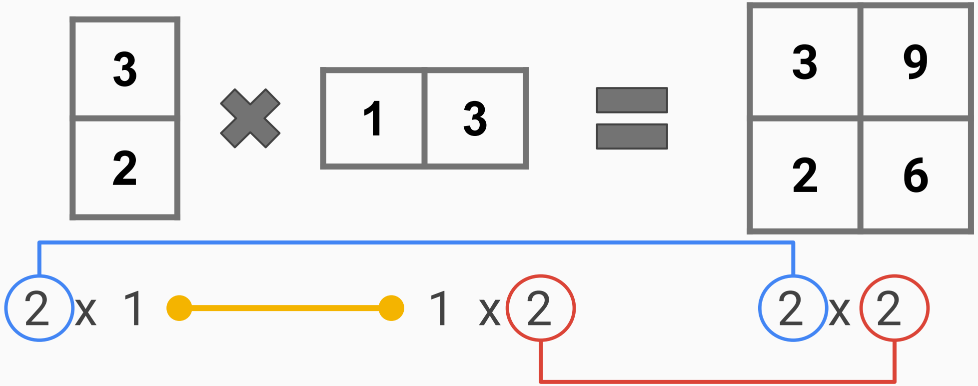
- Generalization of vectors
- Multi-dimensional arrays in Java



Matrix Factorization (MF)

Review:

Matrix multiplication



Matrix Factorization (MF)

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

$$\begin{bmatrix} 3 & 9 & 3 \\ 2 & 6 & 2 \\ 4 & 12 & 4 \end{bmatrix} = \begin{bmatrix} 3 \\ 2 \\ 4 \end{bmatrix} \times \begin{bmatrix} 1 & 3 & 1 \end{bmatrix}$$

3 x 3   3 x ?   ? x 3

Matrix Factorization (MF)

Question:

Given a matrix of user ratings for different locations, **predict empty ratings** *0 (n/a)*

Location\User	User 1	User 2	User 3
Tony's Sushi	5	4	0 *(n/a)*
Theatre	3	3	1
Dollar Store	0 *(n/a)*	0 *(n/a)*	2

Matrix Factorization (MF)

Matrix *L*

Location\Feature	*shopping*	*food*
Tony's Sushi	?	?
Theatre	?	?
Dollar Store	?	?

×

Matrix *U*

Feature\User	User 1	User 2	User 3
shopping	?	?	?
food	?	?	?

=

Location\User	User 1	User 2	User 3
Tony's Sushi	5	4	0 *(n/a)*
Theatre	3	3	1
Dollar Store	0 *(n/a)*	0 *(n/a)*	2

Matrix Factorization (MF)

Matrix *L*

Location\Feature	*shopping*	*food*
Tony's Sushi	1	5
Theatre	2	1
Dollar Store	5	2

×

Matrix *U*

Feature\User	User 1	User 2	User 3
shopping	1	1	0.1
food	0.8	0.6	0.75

=

Location\User	User 1	User 2	User 3
Tony's Sushi	5	4	0 *(n/a)*
Theatre	3	3	1
Dollar Store	0 *(n/a)*	0 *(n/a)*	2

Matrix Factorization (MF)

Matrix *L*

Location\Feature	*shopping*	*food*
Tony's Sushi	1	5
Theatre	2	1
Dollar Store	5	2

×

Matrix *U*

Feature\User	User 1	User 2	User 3
shopping	1	1	0.1
food	0.8	0.6	0.75

=

Location\User	User 1	User 2	User 3
Tony's Sushi	5	4	1 * 0.1 + 5 * 0.75 = 4
Theatre	3	3	1
Dollar Store	0 *(n/a)*	0 *(n/a)*	2

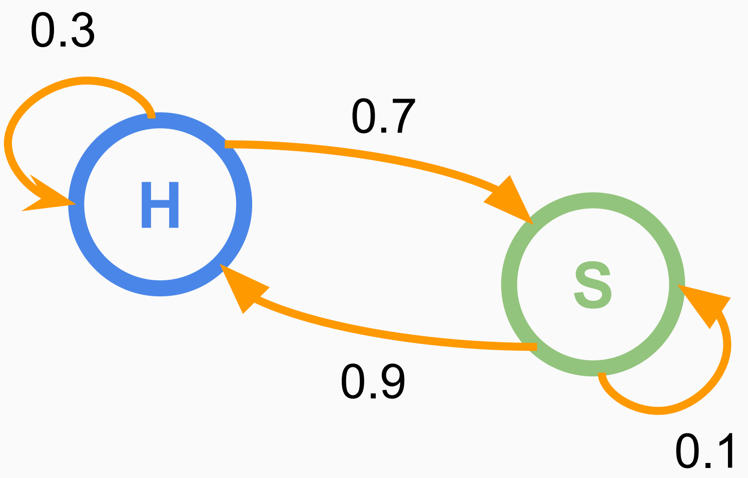
Matrix Factorization (MF)

What's the point?

Based on the calculated predictions, we can make **recommendations**.

Location\User	User 1	User 2	User 3
Tony's Sushi	5	4	4
Theatre	3	3	1
Dollar Store	5	5	2

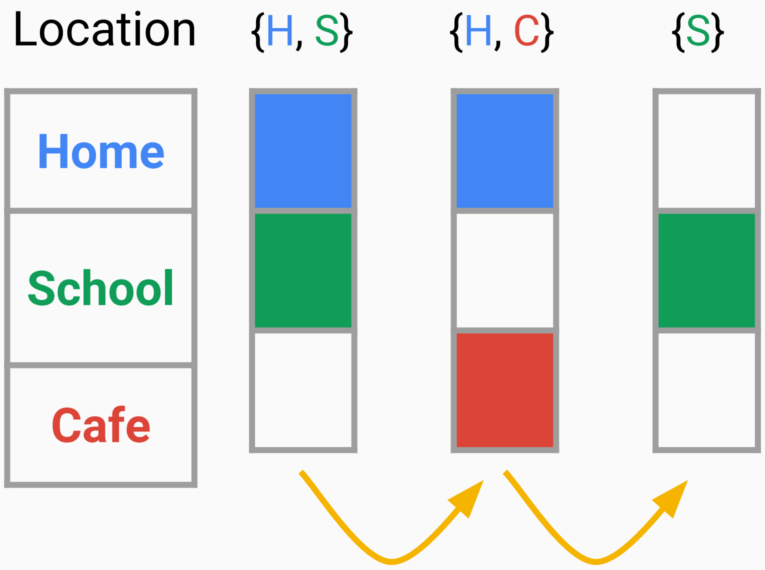
Markov Chain (MC)



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

	Home	School
Home	0.3	0.7
School	0.9	0.1

Markov Chain (MC) for Sets



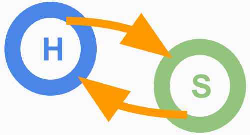
Each state is now a set of locations instead of a single location.

	Home	School	Cafe	#
Home	1/2	1/2	1/2	2
School	1/1	0/1	1/1	1
Cafe	0/1	1/1	0/1	1

Table of Contents

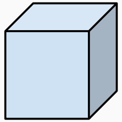
POI Recommendation History over Time

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

From location

To location

User 1

0.2	?	?
0.5	0.1	?
?	?	0.3

User 2

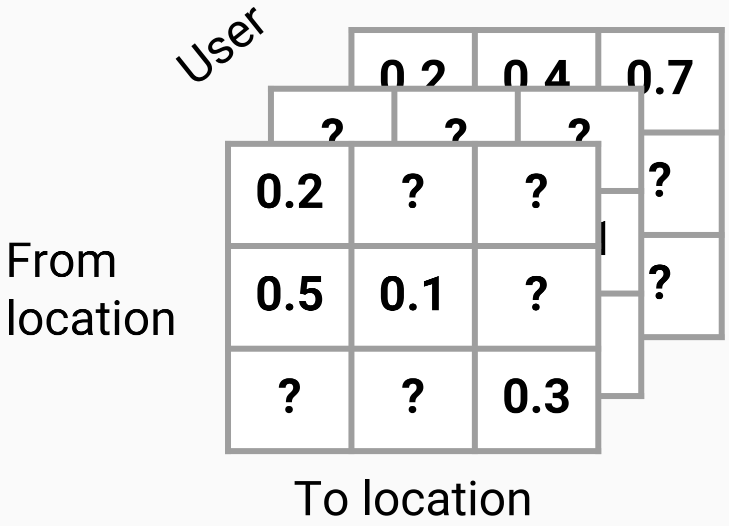
?	?	?
0.3	?	0.1
0.6	?	?

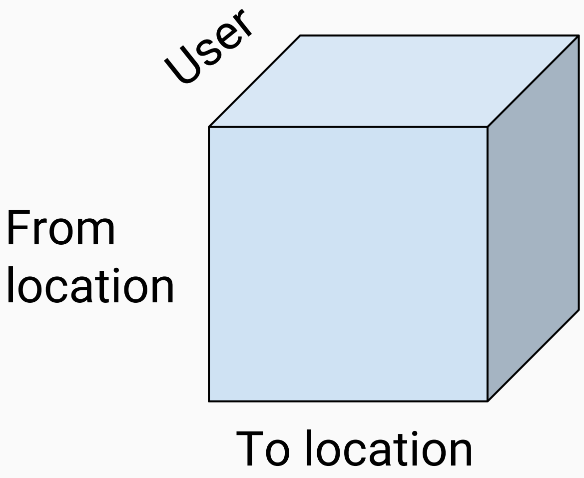
User 3

0.2	0.4	0.7
?	0.1	?
?	0.5	?

Factorized Personalized Markov Chain (FPMC)

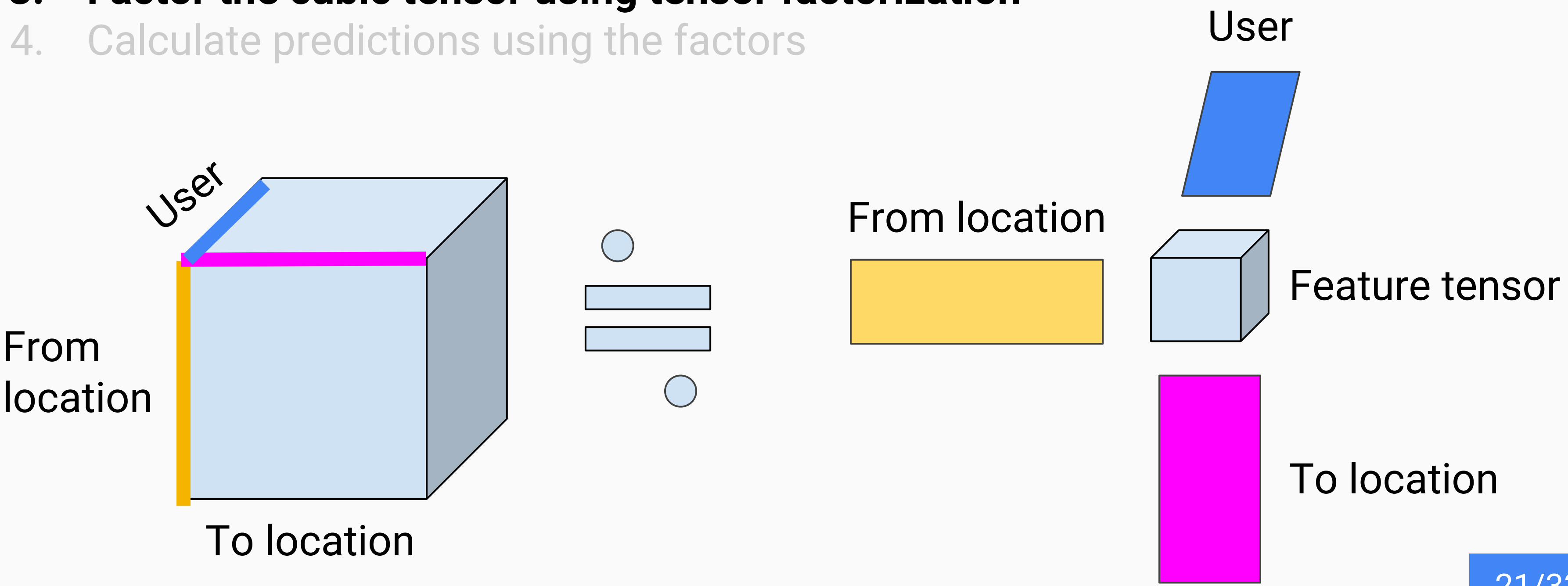
1. Make a Markov chain for each user
2. **Stack them one onto another**
3. Factor the cubic tensor using tensor factorization
4. Calculate predictions using the factors





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

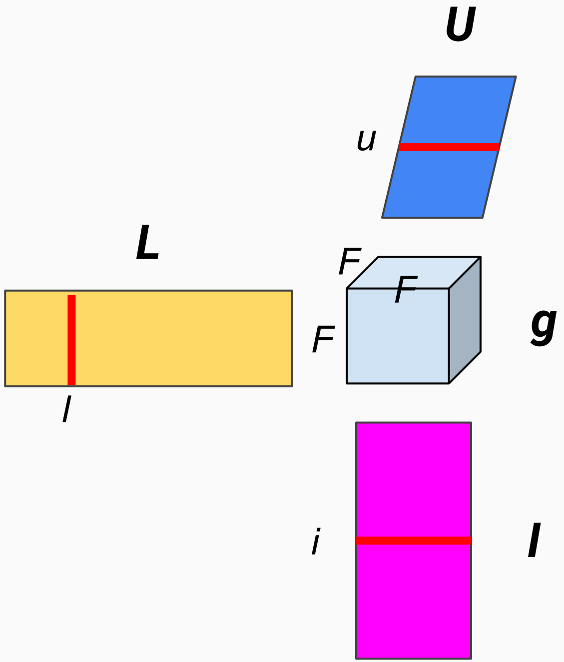


Factorized Personalized Markov Chain (FPMC)

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***

$$x_{l,i,u} \approx \sum_{p=1}^{F} \sum_{q=1}^{F} \sum_{r=1}^{F} g_{pqr} L_{lp} I_{iq} U_{ur}$$

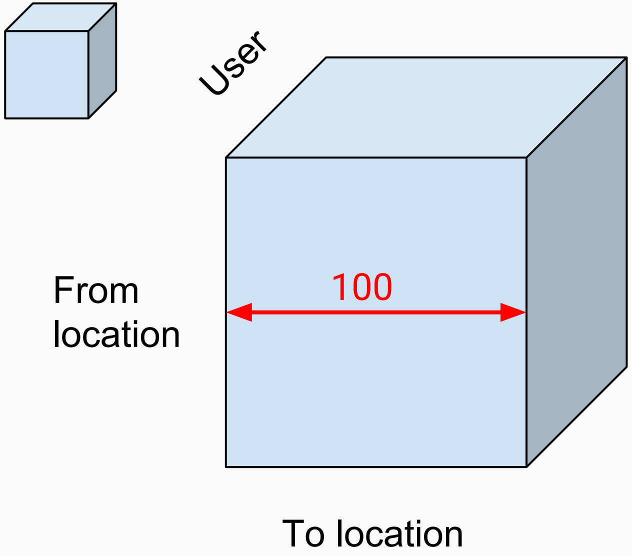


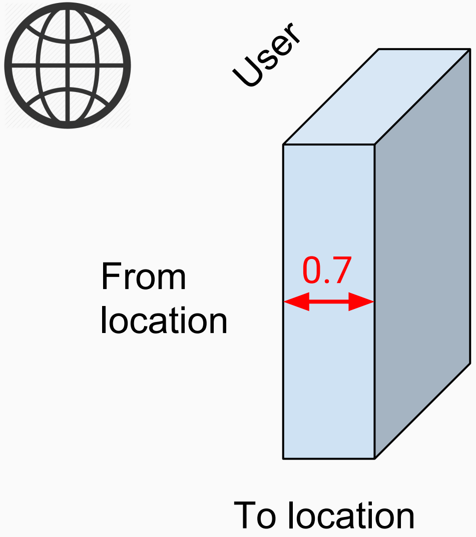
FPMC with Localized Region Constraint (FPMC-LR)

Problems of FPMC		**How FPMC-LR solves the problem**
Generic method for any recommendation systems	→	POI-specific method
Computationally expensive	→	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

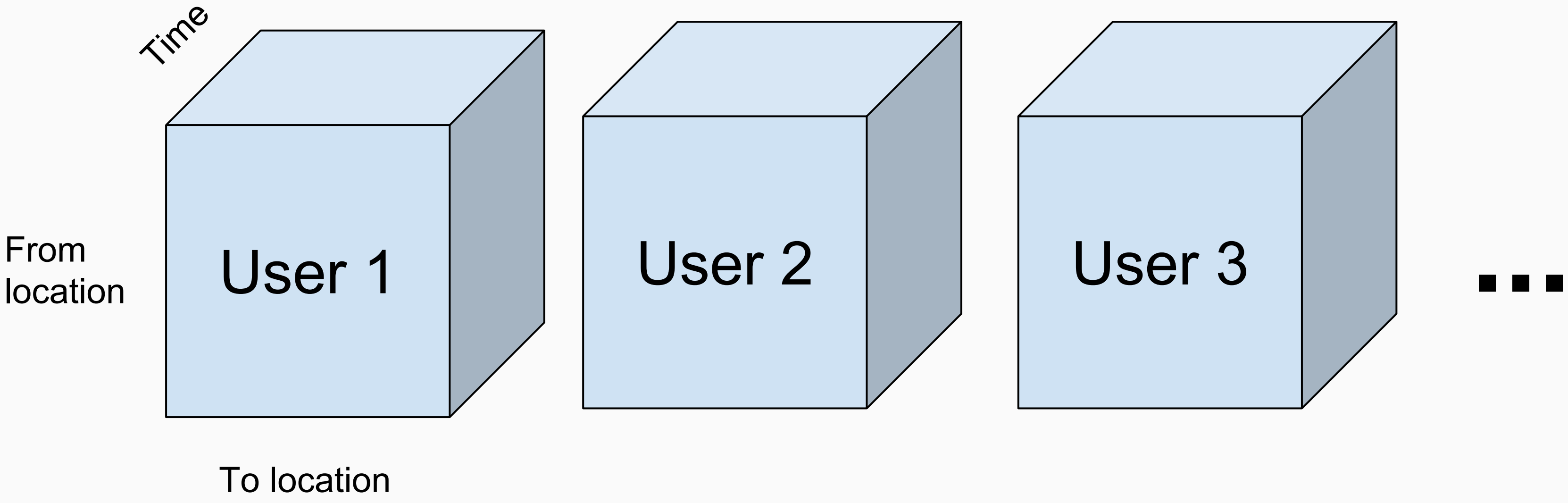
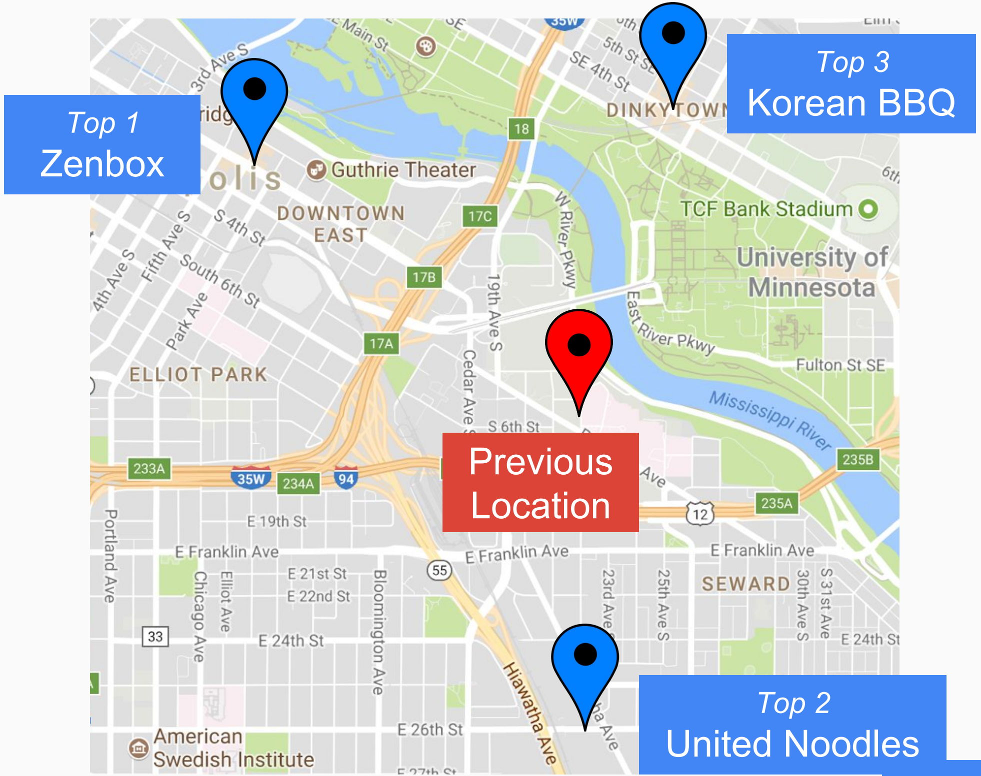


Table of Contents

Performance Analysis - Evaluation Metric

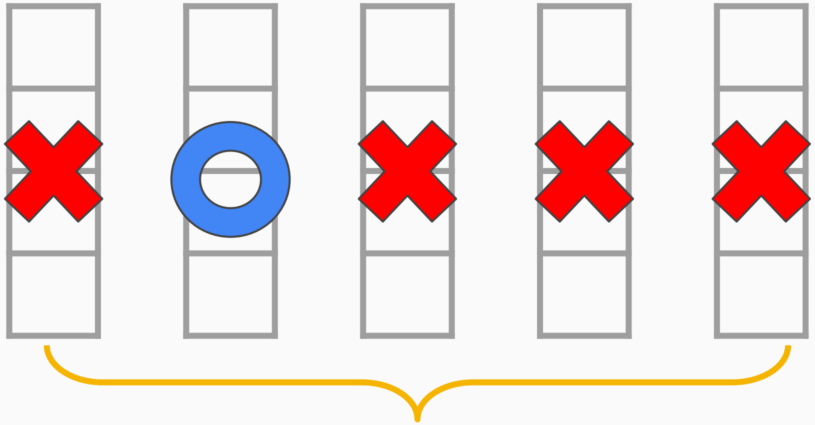
1. Each Algorithm recommends a list of top-N places.
2. Recommendation is correct if the user indeed visited any place in the list at time t.

Top-3 recommendations



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

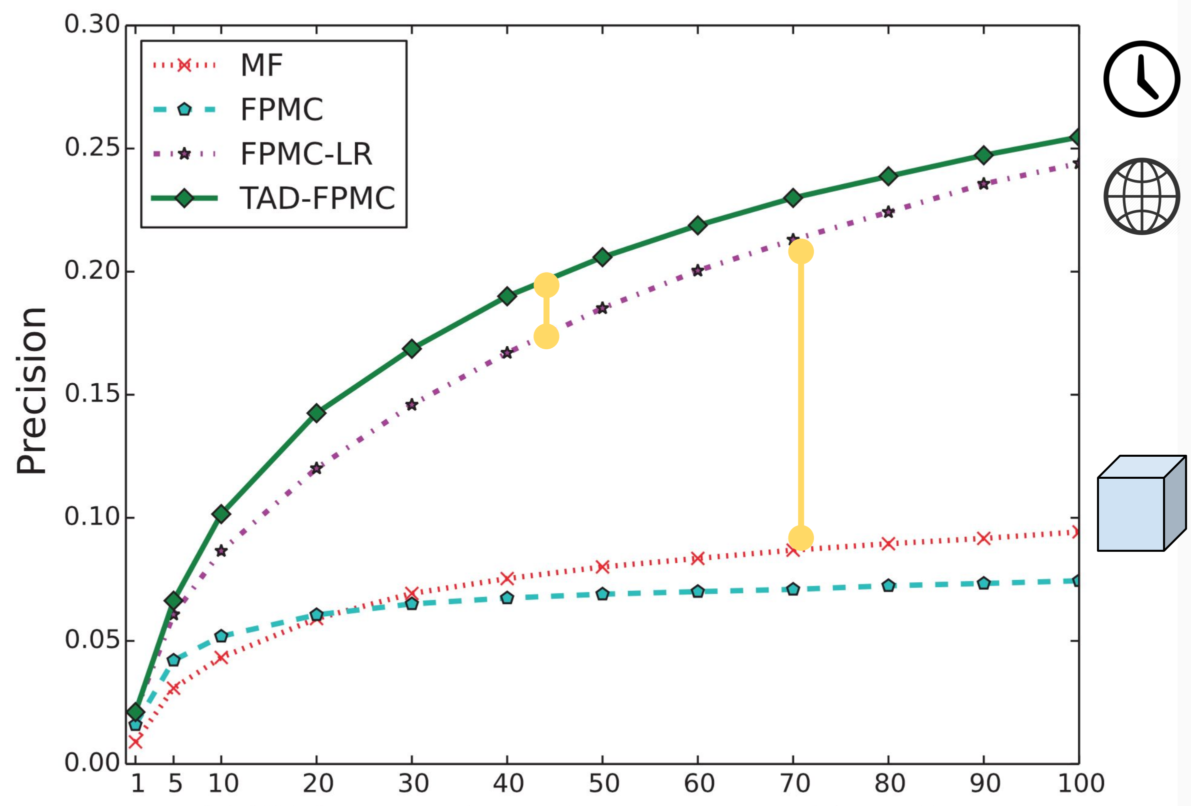
Top-4 Recommendations



5 Rounds

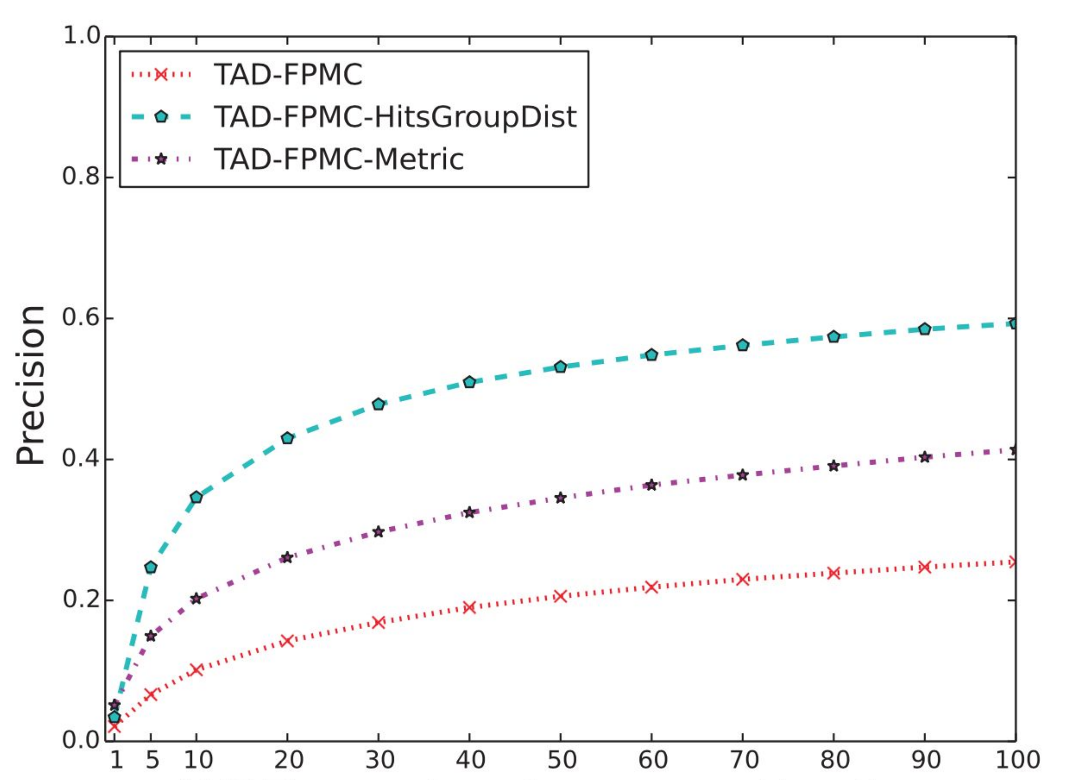
Performance Analysis

POI specific models (FPMC-LR & TAD-FPMC) show much better performance



Size of Top-N list (NYC data)

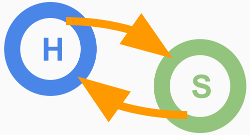
Variations of TAD-FPMC far outperform all existing models



Size of Top-N list (NYC data)

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

[1] C. Cheng, H. Yang, M. R. Lyu, and I. King. Where you like to go next: Successive point-of-interest recommendation. In Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, IJCAI ’13, pages 2605–2611. AAAI Press, 2013.

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

[3] X. Li, M. Jiang, H. Hong, and L. Liao. A time-aware personalized point-of-interest recommendation via high-order tensor factorization. ACM Trans. Inf. Syst., 35(4):31:1–31:23, June 2017.

[4] S. Rendle, C. Freudenthaler, and L. Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In Proceedings of the 19th International Conference on World Wide Web, WWW ’10, pages 811–820, New York, NY, USA, 2010. ACM.

[5] S. Rendle and L. Schmidt-Thieme. Pairwise interaction tensor factorization for personalized tag recommendation. In Proceedings of the Third ACM International Conference on Web Search and Data Mining, WSDM ’10, pages 81–90, New York, NY, USA, 2010. ACM.