Context-Aware Event Recommendation in Event-Based Social Networks (RecSys 2015)

Augusto Q. Macedo, Leandro B. Marinho and Rodrygo L. T. Santos

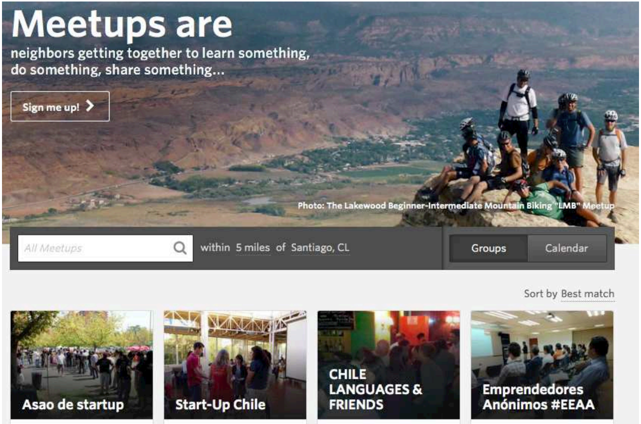
TIAN Jun Feng (田 俊锋)

Department of Computer Science, East China Normal University

November 27, 2015

Event-Based Social Networks (EBSNs)

People can create events of any kind and share it with other users.



Which events best match the user's preferences?

Outline

- Problem Setting and Motivation
- Contextual Models (Social, Content, Location, Time)
- Experimental and Evaluation
- Conclusions

Problem

Event Recommendation is Intrinsically **Cold-Start**

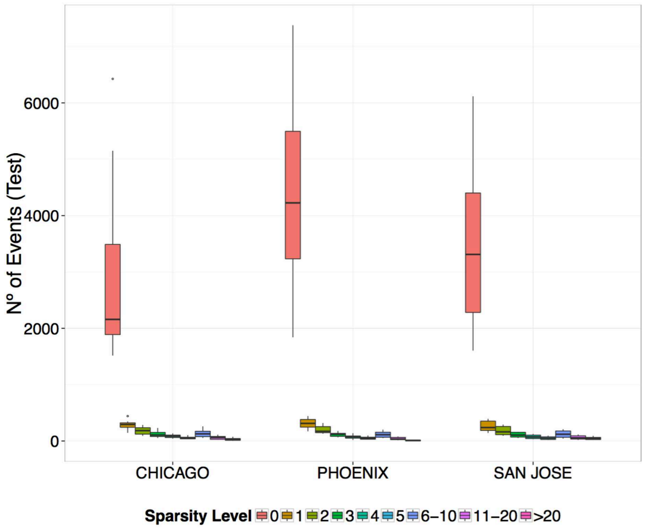
Events are always in the future.

Idea 1: Use RSVP as a Proxy for Event Attendance

RSVP: ”please respond”. After the creation of a public event, any user can RSVP to it with ”yes” or ”no”.

However, RSVP data is very sparse!

Sparsity Level per User



Idea 2: Use Contextual Data

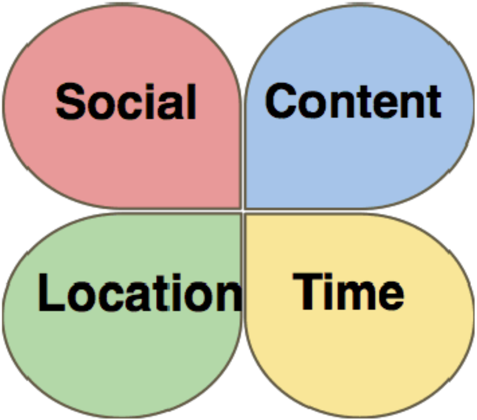
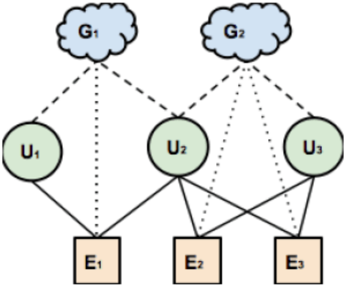


Figure: Multi-Contextual Learning to Rank Events (MCLRE)

Social-Aware: Group Frequency

The more events a user attends in a group, the higher the probability he will attend a new event of this group. Formally:

$$\hat{s}(u, e) := \frac{|E_{u,g_e}|}{E_u}$$



Multi-Relational Matrix Factorization [Drumond, 2012]

$$argmin_{\Theta} \underbrace{\alpha L(R_{UE}, UE^T) + \beta L(R_{UG}, UG^T) + \gamma L(R_{GE}, GE^T)}_{\text{sum of weighted losses}} \underbrace{+\lambda_U \parallel U \parallel + \lambda_E \parallel E \parallel + \lambda_G \parallel G \parallel}_{\text{regularization term}} \quad (1)$$

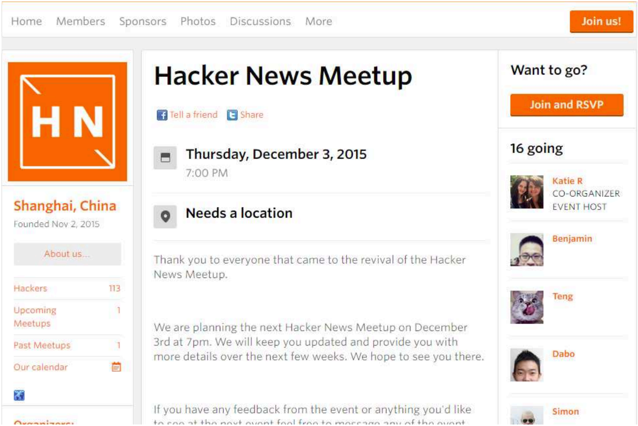
- $R_{XY} \ldots$ Relation between X and Y.
- $\Theta := \{U, E, G\} \ldots$ latent matrices of U, E, G resp.
- $L \ldots$ BPR loss function.

The recommendation score is given by:

$$\hat{s}(u, e) = \sum_{f=1}^{k} \vec{u_f} \vec{e_f}$$

Content-Aware

Content may help to capture similar and recurrent events.



TF-IDF with time decay

User profile is the aggregation of all her events' TF-IDFs weighted:

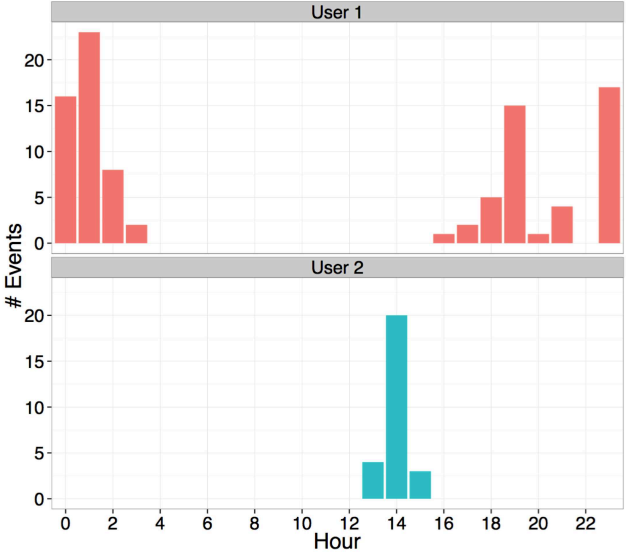
$$\vec{u} := \sum_{e \in E_u} \frac{1}{(1+\alpha)^{\tau}} \times \vec{e}.$$

- $\vec{e}$ ...TF-IDF of event e.
- α ...time decay factor.
- $\tau(e)$ ...days from the RSVP to e until the recommendation moment.

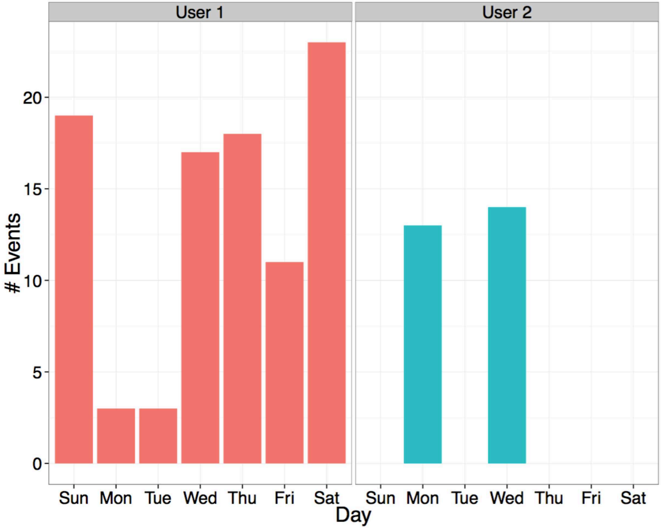
The recommendation score is given by:

$$\hat{s}(u,e) = cos(\vec{u}, \vec{e}).$$

Which time do users go to events?



Which day do users go to events?



Time-Aware

Each user is the centroid of the events she attended in the past:

$$\vec{u} := \frac{1}{E_u} \sum_{e \in E_u} \vec{e}.$$

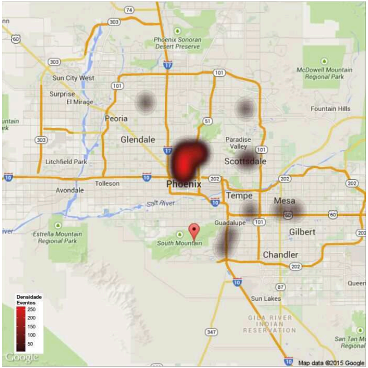
where $\vec{e}$ is a 24×7-dimensional vector in the space of all possible days of the week and hours of the day.

The recommendation score is now:

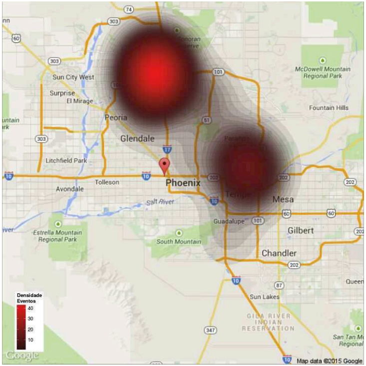
$$\hat{s}(u, e) := cos(\vec{u}, \vec{e})$$

Location-Aware

Assumption: Users tend to attend events close to the events they attended in the past.



(a) User 1



(b) User 2

Kernel Density Estimation

The recommendation score is given by

$$\hat{s}(u, e) = \frac{1}{|L_u|} \sum_{l' \in L_u} K_H(l_e - l').$$

where l_e is the lat-long coordinate of event e and K_H is the Gaussian kernel.

Learning to Rank

Let $\mathcal{D} := \{(x_1, y1), \ldots, (x_n, y_n)\}$ be the training set where $x_i := (\hat{s_1}(u, e), \ldots, \hat{s_m}(u, e), |U_e|)$ is a feature vector containing the scores for each recommender and $y_i = \{0, 1\}$ denote whether user u attended event e or not.

The goal is to learn a function $h(x)$ s.t. for any pair (x_i, y_i) and (x_j, y_j) the following holds:

$$h(x_i) > h(x_j) \Leftrightarrow y_i > y_j.$$

We have used **Coordinate Ascent** [Metzler, 2007], a state-of-the-art listwise learning to rank approach.

Research Questions

*Q*1. How **effective** is MCLRE for event recommendation?

*Q*2. How **robust** is MCLRE to **sparsity** in the RSVP data?

*Q*3. Which contextual **features** are **effective** recommenders?

Research Questions

*Q*1. How **effective** is MCLRE for event recommendation?

*Q*2. How **robust** is MCLRE to **sparsity** in the RSVP data?

*Q*3. Which contextual **features** are **effective** recommenders?

Research Questions

*Q*1. How **effective** is MCLRE for event recommendation?

*Q*2. How **robust** is MCLRE to **sparsity** in the RSVP data?

*Q*3. Which contextual **features** are **effective** recommenders?

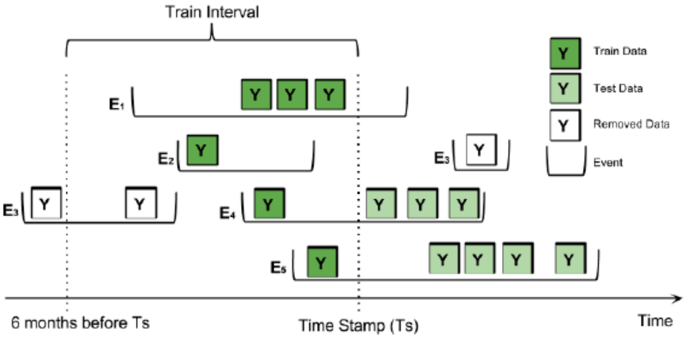
Evaluation Protocol

- Meetup.com data from January, 2010 to April, 2014
- Cities Collected: Phoenix, Chicago and San Jose

City	$\|G\|$	$\|U\|$	$\|E\|$	**RSVPs**	**Sparsity**
Chicago	2,321	207,649	190,927	1,375,154	99.99%
Phoenix	1,661	117,458	222,632	1,209,324	99.95%
San Jose	2,589	242,143	206,682	1,607,985	99.99%

Evaluation Protocol

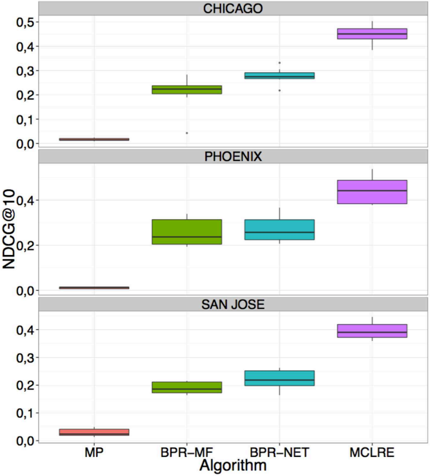
- 12 time stamps equally spaced in time over 52 months.
- Sliding training window.
- For each city, the four initial partitions are used as validation sets and the remaining partitions for evaluation.



Compared Algorithm

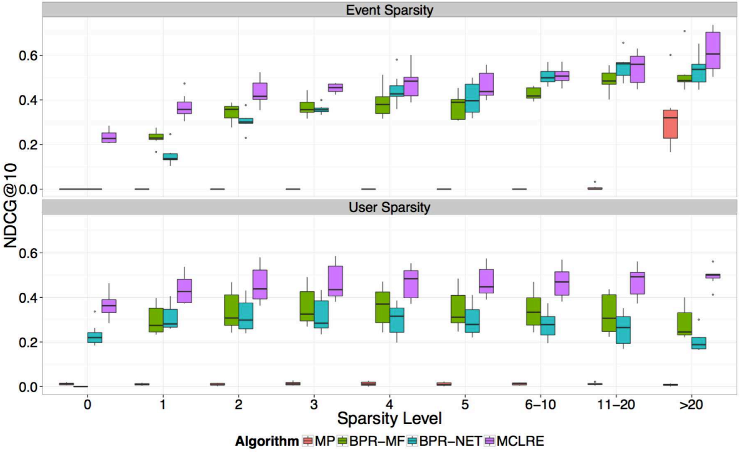
- Most Popular
- BPR-MF [Rendle, 2009]
- BPR-NET [Qiao, 2014]
- **MCLRE**
- Evaluation Metric: NDCG@10

Recommendation Effectiveness



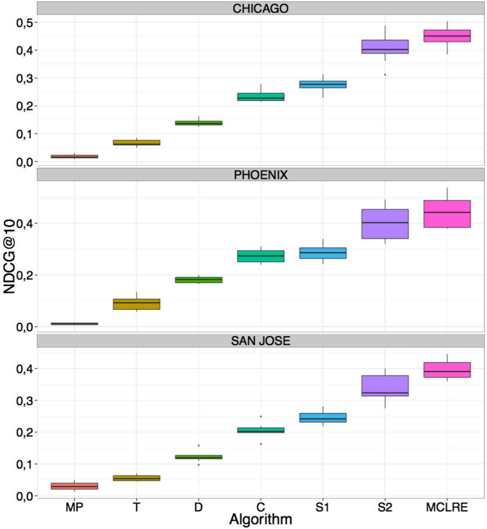
Improvement of up to 79%!

Robustness to Data Sparsity



Only method able to recommend in full cold-start!

Contextual Feature Analysis



Conclusions

- The use of multiple contexts can both lead to highly accurate recommendations and mitigate the cold-start problem.
- Events created by groups of which a user is a member are far more relevant than the content of the events or collaborative RSVP data.