Recomendación Contextual II

Denis Parra
Sistemas Recomendadores
IIC 3633
2do semestre de 2015

Agenda Semestral

6 - 8 Oct	User centric evaluation + User interfaces	Prof. Denis Parra
13 - 15 Oct	Context-aware recommenders	Prof. Denis Parra
20 - 22 Oct	Context-aware recommenders II	Prof. Denis Parra
27 - 29 Oct	Active/Reinforcement Learning Recommender Systems	Gabriel della Maggiora y Javier Machin
3 - 5 Nov	Graph-based recommendation	Juan Pablo Salazar y Christopher Arenas
10 - 12 Nov	Applications: music	Miguel Fadic
17 - 19 Nov	Modelos graficos probabilisticos para sistemas recomendadores	Laura Cruz (invitada)

En esta clase

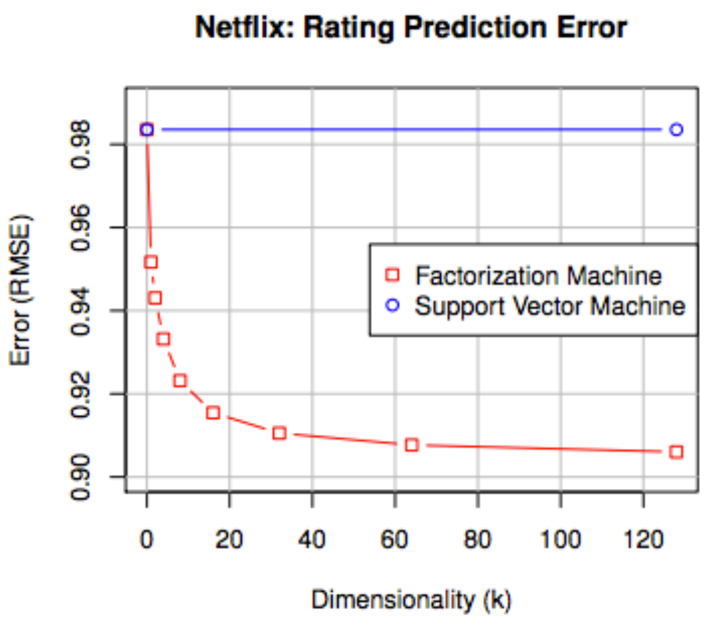
- Feedback sobre sus proyectos
- Factorization Machines
- Resultado de proyecto final, clase RecSys 2014 (usando MovieCity)
- Best paper RecSys 2015: Augusto Q. Macedo, Leandro B. Marinho, and Rodrygo L.T. Santos. 2015. Context-Aware Event Recommendation in Event-based Social Networks.

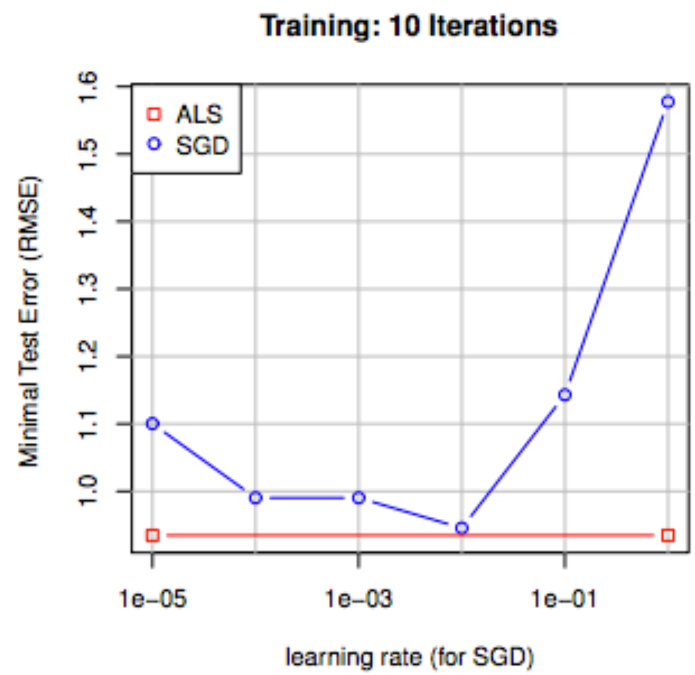
¿Cómo presento mis resultados en el proyecto final?

R: Usando como ejemplo los papers de Rendle et al.

Comparación con varios algoritmos

- Check parameters (learning rate, dimensionality, regularization, context)





Comparación con varios algoritmos

- Check different datasets/features

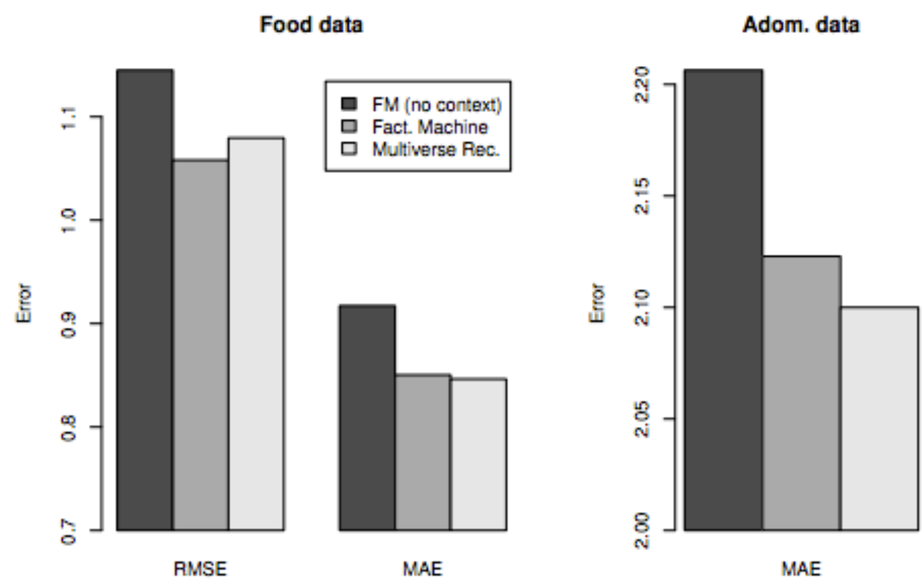


Figure 6: The context-aware methods *Multiverse Recommendation* [5] and our proposed context-aware *Factorization Machine* benefit from incoop-erating the context-information into the rating prediction.

Comparación con varios algoritmos

Steffen Rendle (2013): Scaling Factorization Machines to Relational Data, in Proceedings of the 39th international conference on Very Large Data Bases (VLDB 2013), Trento, Italy.

Table 2: Prediction error on the Netflix prize dataset. A star * indicates that this is the best value reported in the corresponding paper for this method. The methods are grouped by the information that they take into account. The RMSE results are measured on the Quiz dataset (leaderboard scores).

Method (Name)	Reference	Learning Method	k	Quiz RMSE
Models using user ID and item ID				
Probabilistic Matrix Factorization	[17, 16]	Batch GD	40	*0.9170
Probabilistic Matrix Factorization	[17, 16]	Batch GD	150	0.9211
Matrix Factorization	[7]	Variational Bayes	30	*0.9141
Matchbox	[18]	Variational Bayes	50	*0.9100
ALS-MF	[10]	ALS	100	0.9079
ALS-MF	[10]	ALS	1000	*0.9018
SVD/ MF	[3]	SGD	100	0.9025
SVD/ MF	[3]	SGD	200	*0.9009
Bayesian Probablistic Matrix Factorization (BPMF)	[16]	MCMC	150	0.8965
Bayesian Probablistic Matrix Factorization (BPMF)	[16]	MCMC	300	*0.8954
FM-BS, pred. var: user ID, movie ID	-	MCMC	128	0.8937
Models using implicit feedback				
Probabilistic Matrix Factorization with Constraints	[17]	Batch GD	30	*0.9016
SVD++	[3]	SGD	100	0.8924
SVD++	[3]	SGD	200	*0.8911
BSRM/F	[24]	MCMC	100	0.8926
BSRM/F	[24]	MCMC	400	*0.8874
FM-BS, pred. var: user ID, movie ID, impl.	-	MCMC	128	0.8865
Models using time information				
Bayesian Probabilistic Tensor Factorization (BPTF)	[21]	MCMC	30	*0.9044
FM-BS, pred. var: user ID, movie ID, day	-	MCMC	128	0.8873
Models using time and implicit feedback				
timeSVD++	[5]	SGD	100	0.8805
timeSVD++	[5]	SGD	200	*0.8799
FM-BS, pred. var: user ID, movie ID, day, impl.	-	MCMC	128	0.8809
FM-BS, pred. var: user ID, movie ID, day, impl.	-	MCMC	256	0.8794
Assorted models				
BRISMF/UM NB corrected	[19]	SGD	1000	*0.8904
BMFSI plus side information	[11]	MCMC	100	*0.8875
timeSVD++ plus frequencies	[4]	SGD	200	0.8777

Factorization Machines

- Rendle, S. (2010, December). **Factorization machines**. In Data Mining (ICDM), 2010 IEEE 10th International Conference on (pp. 995-1000). IEEE.
- Rendle, S., Gantner, Z., Freudenthaler, C., & Schmidt-Thieme, L. (2011, July). **Fast context-aware recommendations with factorization machines**. In Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval (pp. 635-644). ACM.
- Rendle, S. (2012). **Factorization machines with libFM**. ACM Transactions on Intelligent Systems and Technology (TIST), 3(3), 57.

Máquinas de Factorización (2010)

- Inspiradas en SVM, permiten agregar un número arbitrario de features (user, item, contexto) pero funcionan bien con "sparse data" al incorporar variables latentes factorizadas (inspiradas en Factorización Matricial). No se necesitan vectores de soporte para optimizar el modelo.
- Generalizan diversos métodos de factorización matricial
- Disminuyen la complejidad de aprendizaje del modelo de predicción respecto de métodos anteriores

Motivación

- Cada tarea de recomendación (implicit feedback, agregar tiempo, incorporar contexto) requiere rediseño del modelo de optimización y re-implementación del algoritmo de inferencia
- Lo ideal sería usar alguna herramienta como libSVM, Weka, … agregar los vectores de features
- Pero para manejar datos tan dispersos, se podrían mantener las factorizaciones!

Dado un modelo con interacción d=2

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{j,f} v_{j',f},$$

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad V \in \mathbb{R}^{p \times k}.$$

Dado un modelo con d=2

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{j=1}^{p} w_j x_j + \sum_{j=1}^{p} \sum_{j'=j+1}^{p} x_j x_{j'} \sum_{f=1}^{k} v_{j,f}\, v_{j',f},$$

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^p, \quad V \in \mathbb{R}^{p \times k}.$$

Sesgo (bias) global

Coeficientes de regresión de la j-ésima variable

Interacción de features

Factorización (variables latentes)

Reducción del modelo

1) Model Equation: The model equation for a factorization machine of degree $d = 2$ is defined as:

$$\hat{y}(\mathbf{x}) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle \mathbf{v}_i, \mathbf{v}_j \rangle \, x_i \, x_j \qquad (1)$$

where the model parameters that have to be estimated are:

$$w_0 \in \mathbb{R}, \quad \mathbf{w} \in \mathbb{R}^n, \quad \mathbf{V} \in \mathbb{R}^{n \times k} \qquad (2)$$

And $\langle \cdot, \cdot \rangle$ is the dot product of two vectors of size k:

$$\langle \mathbf{v}_i, \mathbf{v}_j \rangle := \sum_{f=1}^{k} v_{i,f} \cdot v_{j,f} \qquad (3)$$

Propiedades

- Expresividad* (cualquier matrix semi-definida positiva)
- Multilinearidad**
- **Complexity**

O(kn^2) -> O(kn)
Y debido a dispersión de los datos, O(km_D)

$$\sum_{i=1}^{n}\sum_{j=i+1}^{n}\langle \mathbf{v}_i,\mathbf{v}_j\rangle x_i x_j$$
$$=\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\langle \mathbf{v}_i,\mathbf{v}_j\rangle x_i x_j-\frac{1}{2}\sum_{i=1}^{n}\langle \mathbf{v}_i,\mathbf{v}_i\rangle x_i x_i$$
$$=\frac{1}{2}\left(\sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{j,f}\, x_i x_j-\sum_{i=1}^{n}\sum_{f=1}^{k} v_{i,f}\, v_{i,f}\, x_i x_i\right)$$
$$=\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)\left(\sum_{j=1}^{n} v_{j,f}\, x_j\right)-\sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$
$$=\frac{1}{2}\sum_{f=1}^{k}\left(\left(\sum_{i=1}^{n} v_{i,f}\, x_i\right)^2-\sum_{i=1}^{n} v_{i,f}^2\, x_i^2\right)$$

*, ** ver detalles en Rendle, S. (2010, December). **Factorization machines**.

Comparación con otros modelos

- En el paper Rendle, S. (2010, December). **Factorization machines**, se muestra como desde FM se puede derivar:
  - Matrix Factorization
  - SVD++
  - Pair-wise Interaction Tag-Factorization (PITF)
  - Factorized Personalized Markov Chains (FPMC)

En la Práctica: LibFM

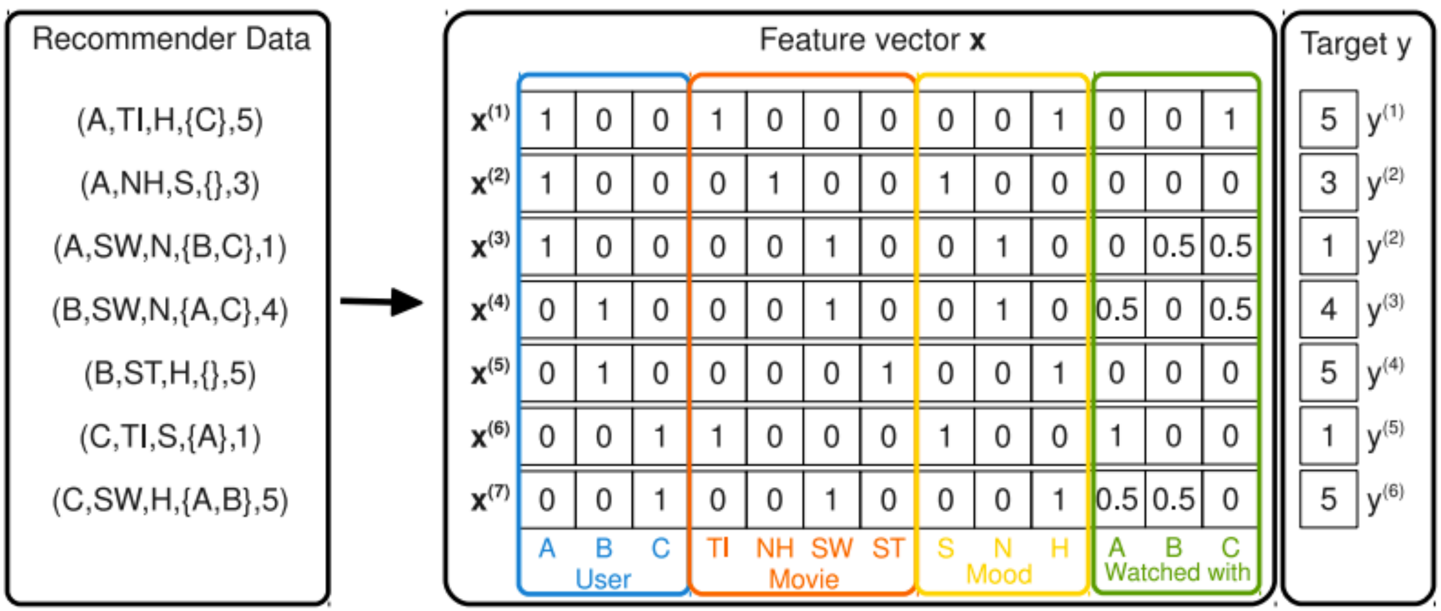


Figure 2: Context-aware recommendation data (left side) is transformed into a prediction problem from real-valued features (right side) by encoding the categorical and set categorical variables (left side) with indicator variables (right side). Here in the feature vector x, the first three values indicate the user, the next four ones the movie, the next three ones the mood and the last three ones the other users a movie has been watched with.

Modelo FM del caso anterior

$$\hat{y}(\mathbf{x}(u,i,c_3,c_4)) = w_0 + w_i + w_u + w_{c_3} + \sum_{t \in c_4} x_t\, w_t$$
$$+ \langle \mathbf{v}_i, \mathbf{v}_u \rangle + \langle \mathbf{v}_i, \mathbf{v}_{c_3} \rangle + \langle \mathbf{v}_i, \sum_{t \in c_4} x_t\, \mathbf{v}_t \rangle$$
$$+ \langle \mathbf{v}_u, \mathbf{v}_{c_3} \rangle + \langle \mathbf{v}_u, \sum_{t \in c_4} x_t\, \mathbf{v}_t \rangle + \langle \mathbf{v}_{c_3}, \sum_{t \in c_4} x_t\, \mathbf{v}_t \rangle$$

Sesgo (bias) global

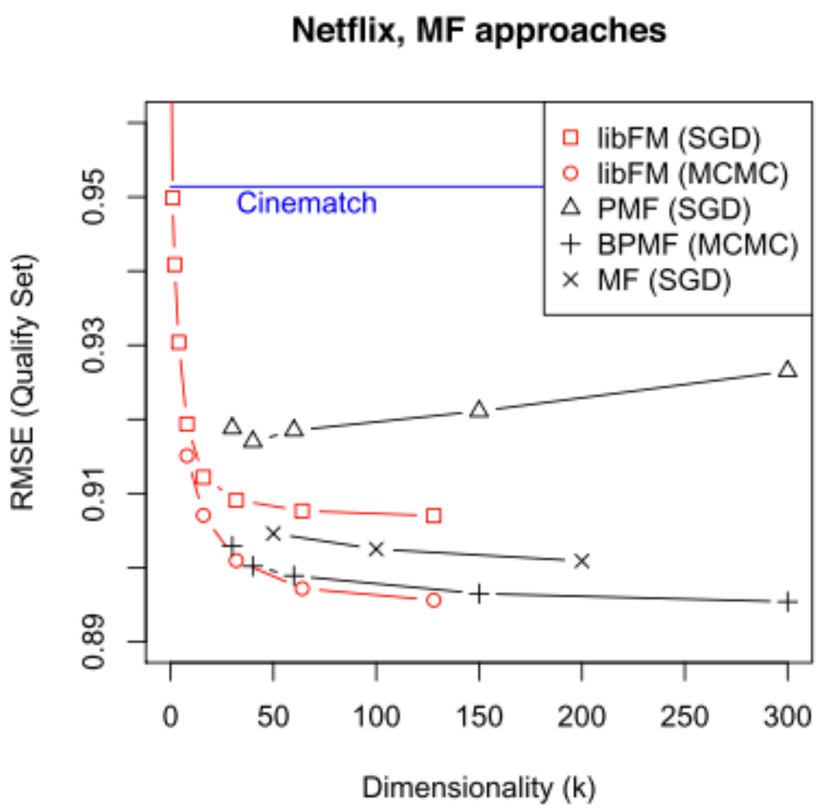
Parametros del Item

Parametros del Usuario

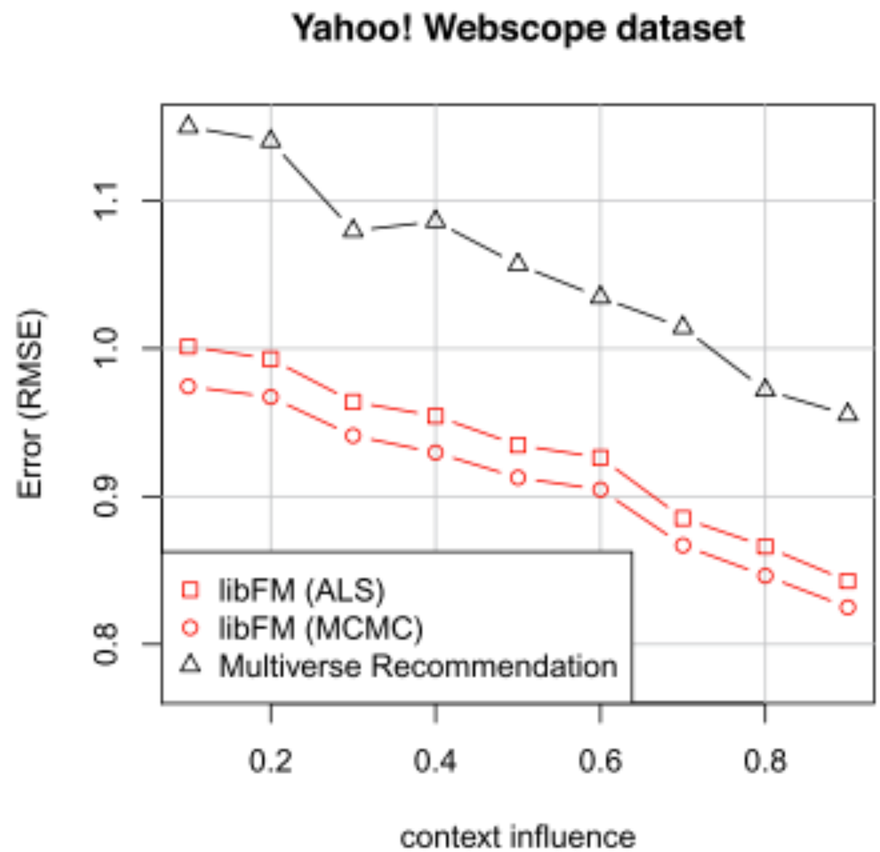
Sentimiento del usuario al ver la pelicula

Usuarios con quienes vio la película

Results



(a) Matrix factorization (MF).



(a) Context-aware recommendation.

Using Libfm

- Llamada 1:

```
./libFM -task r -train ml1m-train -test ml1m-test -dim '1,1,8'
```

- Llamada 2:

```
./libFM -task r -train ml1m-train.libfm -test ml1m-test.libfm -dim '1,1,8' -iter 1000
  -method sgd -learn_rate 0.01 -regular '0,0,0.01' -init_stdev 0.1
```

$$X = \begin{pmatrix} 1.5 & 0.0 & 0.0 & -7.9 & 0.0 & 0.0 & 0.0 \\ 0.0 & 10^{-5} & 0.0 & 2.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 1.0 \end{pmatrix}, \quad \mathbf{y} = \begin{pmatrix} 4 \\ 2 \\ -1 \end{pmatrix}$$

Example

```
4 0:1.5 3:-7.9
2 1:1e-5 3:2
-1 6:1
...
```

- Factorization Machines: Presentación inspirada en http://www.slideshare.net/hongliangjie1/libfm

Proyecto Final curso RecSys 2014

- Trade-offs Between Implicit Feedback and Context-Aware Recommendation
  - Santiago Larraín, PUC Chile
  - Nicolás Risso, PUC Chile
- Moviecity Dataset

Proyecto Final curso RecSys 2014

- Moviecity

Columna	Descripción
user_id	Identificador de usuario único
version_id	Identificador único de contenido
user_watchinglist_time_minutes_spent	Consumo acumulado en minutos
DURATION_MINUTES	Largo del contenido
account_country_code	Código de país del usuario
country_description	IdentificadorNombre del país
country_region	Regíon geográfica
Kids	Marca de si el contenido es para kids o no
Genre	Genero del contenido
Subgenre	Genero primario del contenido

Dataset Moviecity

Mes	Cantidad de usuarios	Cantidad de items	Rmin	Rmax	Ravg
Junio	95013	1679	0	33.94	0.26
Julio	83924	1612	0	38.20	0.34
Agosto	95013	1679	0	33.94	0.26
Total	191657	1918	0	38.20	0.32

Month	Row count	User count	Item count
June	407.078	95.013	1.679
July	482.772	83.924	1.612
August	548.419	95.013	1.668
Total	1.438.269	191.657	1.918

Table 2: Dataset statistics by month

Dataset MovieCity II

Géneros	Subgéneros	Países	Zonas geográficas
Kids	Animation	Mexico	N
Pelicula	Family	Argentina	S
Serie	Thriller	Peru	SA
Movies And Features	Comedy	Colombia	C
Anime	Adventure	Chile	
Documental	Drama	Uruguay	
	Documentary	Venezuela	
	Horror	Rep. Dominicana	
	Action	Honduras	
	Classics	Panama	
	Musical	El Salvador	
	Science Fiction	Guatemala	
	Western	Bolivia	
		Costa Rica	
		Nicaragua	
		Paraguay	

Métodos I

- Hu and Koren ~ Implicit Feedback

$$p_{u,i} = \begin{cases} 1 & \text{if } r_{u,i} > 0 \\ 0 & \text{other case} \end{cases} \qquad c_{u,i} = 1 + \alpha r_{u,i}$$

$$\min_{x*,y*} \sum_{u,i} c_{u,i}(p_{u,i} - \vec{x}_u^T \vec{y}_i)^2 + \lambda \Big(\sum_u ||\vec{x}_u||^2 + \sum_i ||\vec{y}_i||^2 \Big)$$

$$\vec{x}_u = (Y^T C^u Y + \lambda I)^{-1} Y^T C^u p(u) \tag{4}$$

$$\vec{y}_u = (X^T C^i X + \lambda I)^{-1} X^T C^i p(i) \tag{5}$$

Where $C^u \in \mathbb{R}^{U \times k} : C^u_{i,i} = c_{u,i}$ and $C^i \in \mathbb{R}^{I \times k} : C^i_{u,u} = c_{u,i}$

Métodos II

- Factorización Tensorial (usando HOSVD)

If we define $\mathbf{P} \in \mathbb{R}^{U \times I \times C_1 \ldots \times C_N}$, we can decompose the origin al data as follows:

$$\mathbf{P} = S \times_U U^{(U)} \times_I U^{(I)} \times_{C_1} \ldots \times_N U(C_N) \qquad (6)$$

Where $U^{(n)} \in \mathbb{R}^{n \times n}$ is a n dimension tensor and $\times_n$ is a tensor product in dimension n.

Métodos III

- Factorization Machines, Rendle (2010)

$$y(x) := w_0 + \sum_{i=1} w_i x_i + \sum_{i=1} \sum_{j=i+1} (\vec{v_i} \cdot \vec{v_j}) x_i x_j$$

$$y(x) := w_0 + \sum_{i=1} w_i x_i +$$
$$\sum_{l=1} \sum_{i_i=1} \cdots \sum_{i_l=i_{l-1}+1} \Big(\prod_{j=1} x_{i_j} \Big) \Big(\sum_{f=1} \prod_{j=1} v_{i_j} f \Big)$$

Métricas de Evaluación

- RMSE: Diferencia de tiempo entre programa visto y lo predicho

1. $RMSE = \sqrt{\sum_{(u,i)\in Train} (r_{u,i} - \hat{r}_{u,i})^2}$
2. $MAE = \sum_{(u,i)\in Train} \left| r_{u,i} - \hat{r}_{u,i} \right|$
3. $\overline{rank} = \frac{\sum_{(u,i)\in Train} r_{u,i} rank_{u,i}}{\sum_{(u,i)\in Test} r_{u,i}}$

Optimización de los modelos

Parameters k	α	λ	RMSE	MAE	rank
10	20	75	0.6869	0.4686	0.0500
10	20	150	0.7420	0.5172	0.0639
10	20	250	0.7687	0.5529	0.0882
40	20	150	0.7281	0.4936	0.0496
40	20	250	0.7844	0.5615	0.0872
40	40	75	0.5934	0.3567	0.0151
40	40	150	0.6331	0.4053	0.0252
40	40	250	0.6935	0.4681	0.0544
40	60	75	**0.5931**	**0.3392**	**0.0138**
40	60	150	0.5994	0.3674	0.0197
40	60	250	0.6443	0.4211	0.0363

Implicit Feedback

k	RMSE	MAE	rank
10	1.024	0.7787	0.5092
25	1.028	0.7807	0.5059
40	**1.014**	**0.7683**	**0.5003**

Tensor Factorization

k	RMSE	MAE	rank
10	**0.6019**	**0.3958**	**0.4181**
25	0.6862	0.4333	0.4398
40	0.6310	0.4396	0.4272

Factorization Machines

Comparación de los Modelos

Model	RMSE	MAE	rank
Matrix factorization	0.7404	0.4820	**0.1334**
Tensor factorization	1.0024	0.7553	0.5117
Factorization machine	**0.6105**	**0.4148**	0.4092

- Matrix Factorization: $k = 40$; $\lambda = 75$; $\alpha = 60$.
- Tensor Factorization: $k = 40$.
- Factorization Machine: $k = 10$.

Conclusiones

- Error de MAE entre 40% y 70%: diferencia promedio entre el tiempo predicho y el tiempo que el usuario realmente vio. Mejor método es Factorization Machines, indicando que para esta tarea el contexto ayuda.
- Ranking: el mejor método es Implicit Feedback recommender. Extrañamente, esto indica que para rankear, el mejor método no requiere contexto.

Best Paper RecSys 2015

- Augusto Q. Macedo, Leandro B. Marinho, and Rodrygo L.T. Santos. 2015. Context-Aware Event Recommendation in Event-based Social Networks.
- Original slides at:

http://www.slideshare.net/leandrobalby/recsys15-presentation

Presenting …

Context-Aware Event Recommendation in Event-Based Social Networks

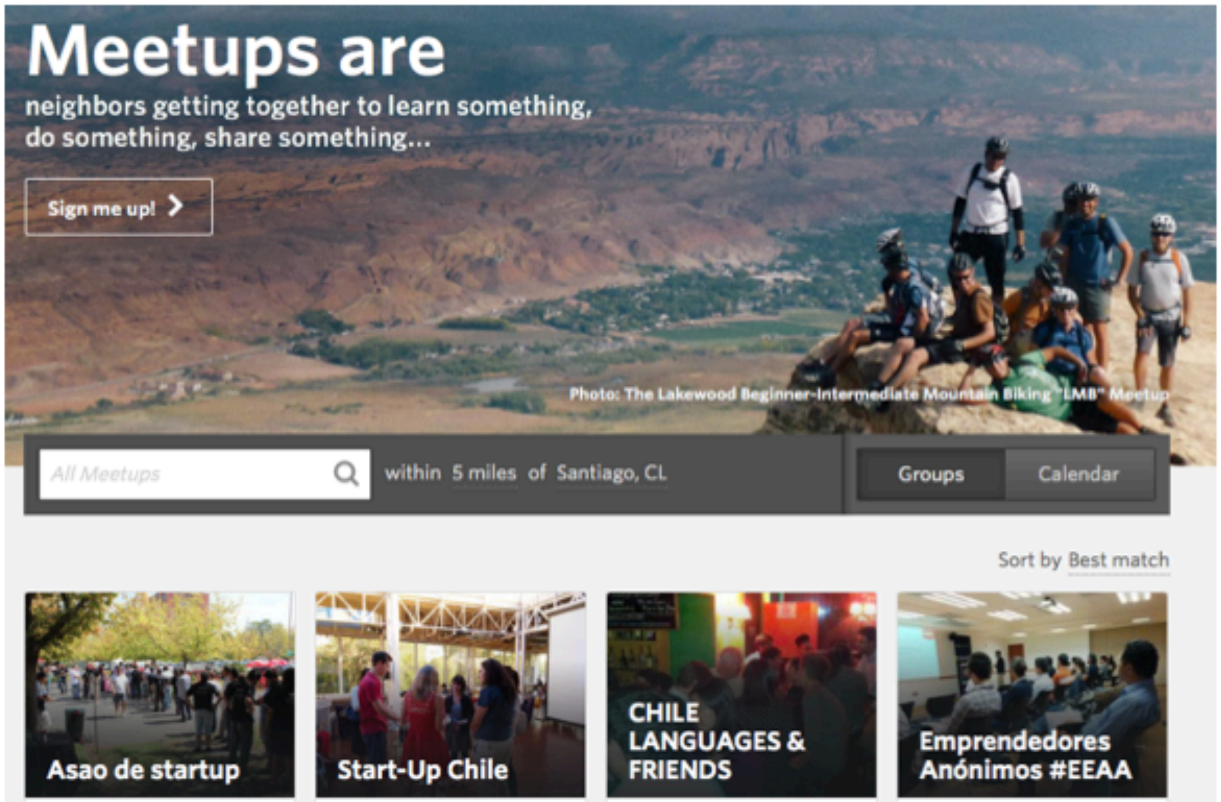
Augusto Q. Macedo, Leandro B. Marinho and Rodrygo L. T. Santos

Federal University of Campina Grande (UFCG)
Federal University of Minas Gerais (UFMG)

ACM Conference on Recommender Systems

Event-based Social Networks

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



Which events best match the user's preferences?

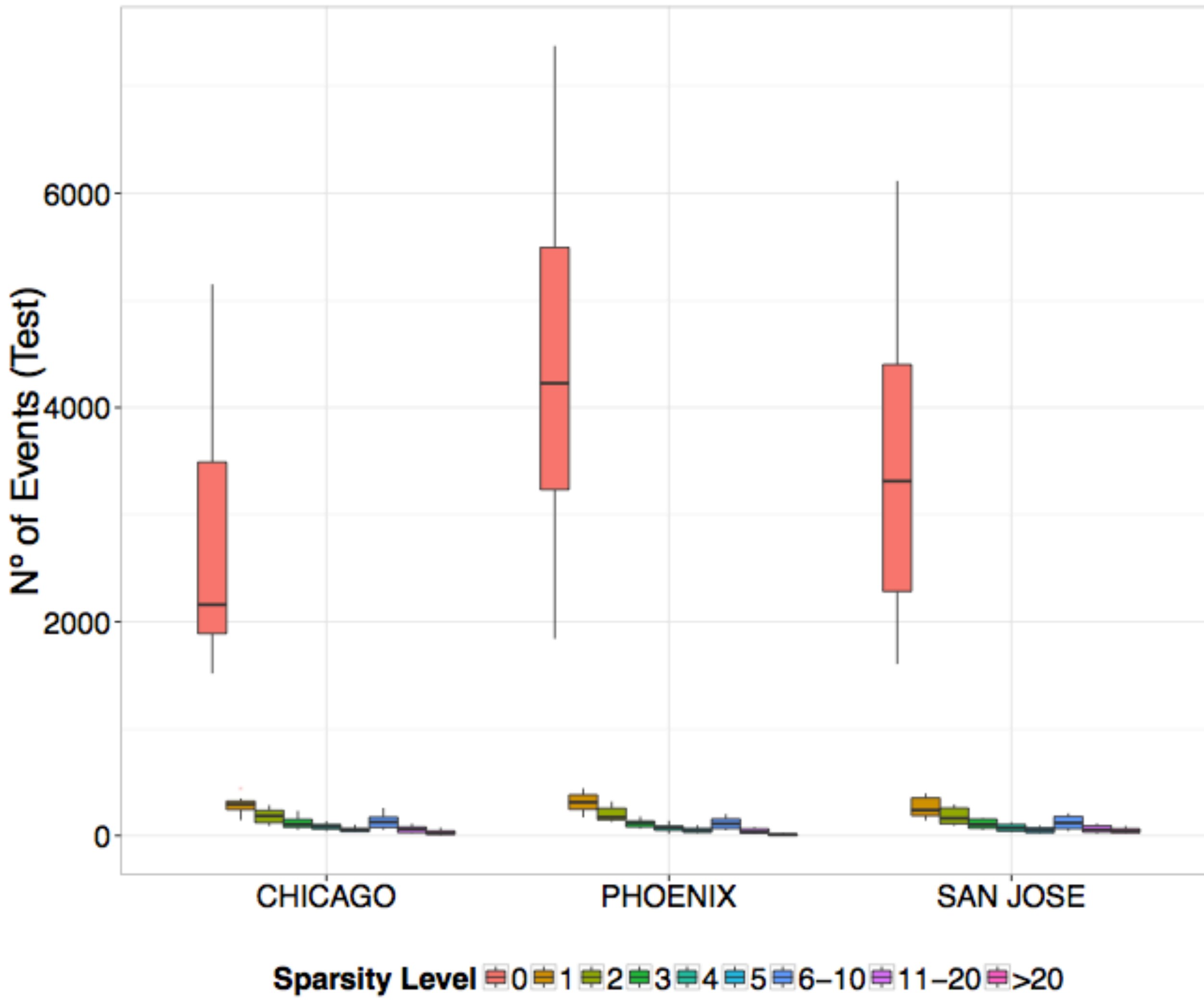
Event Recommendation is Cold-Start

Events are always in the future.

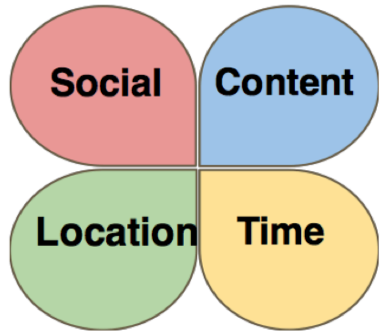
Idea 1: Use RSVP Data

However, RSVP data is very sparse!

Sparsity level per User



The Model: MCLRE



Multi-Contextual Learning to Rank Events (MCLRE)

Q1. How effective is MCLRE for event recommendation?

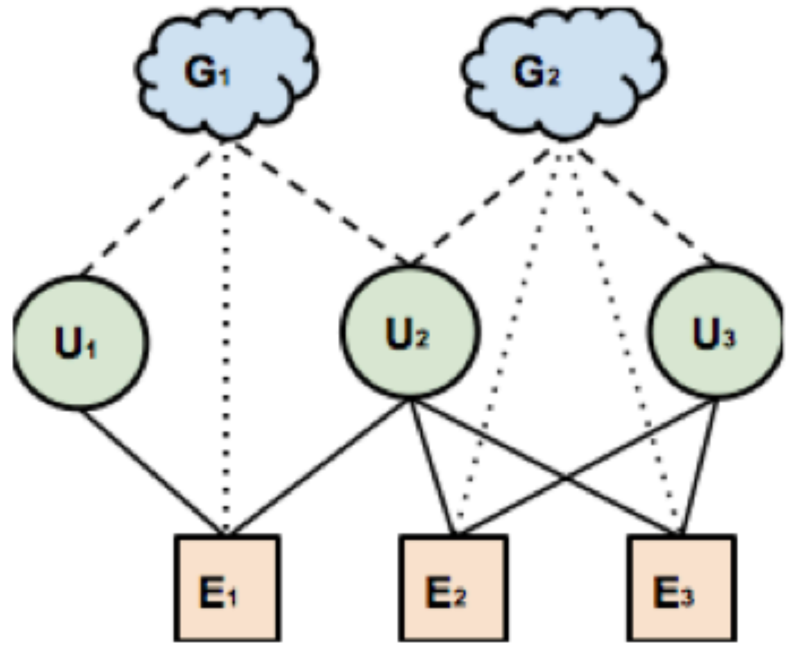
Q2. How robust is MCLRE to sparsity in the RSVP data?

Q3. Which contextual features are effective recommenders?

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 MF (Drumond, 2012)

$$\underset{\Theta}{\operatorname{argmin}} \; \underbrace{\alpha L(R_{UE}, \mathbf{U}\mathbf{E}^T) + \beta L(R_{UG}, \mathbf{U}\mathbf{G}^T) + \gamma L(R_{GE}, \mathbf{G}\mathbf{E}^T)}_{\text{sum of weighted losses}}$$
$$+ \underbrace{\lambda_U \|\mathbf{U}\|^2 + \lambda_E \|\mathbf{E}\|^2 + \lambda_G \|\mathbf{G}\|^2}_{\text{regularization term}}$$

- $R_{XY} \ldots$ Relation between X and Y.
- $\Theta := \{\mathbf{U}, \mathbf{E}, \mathbf{G}\} \ldots$ 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 of Events

TF-IDF with time decay

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

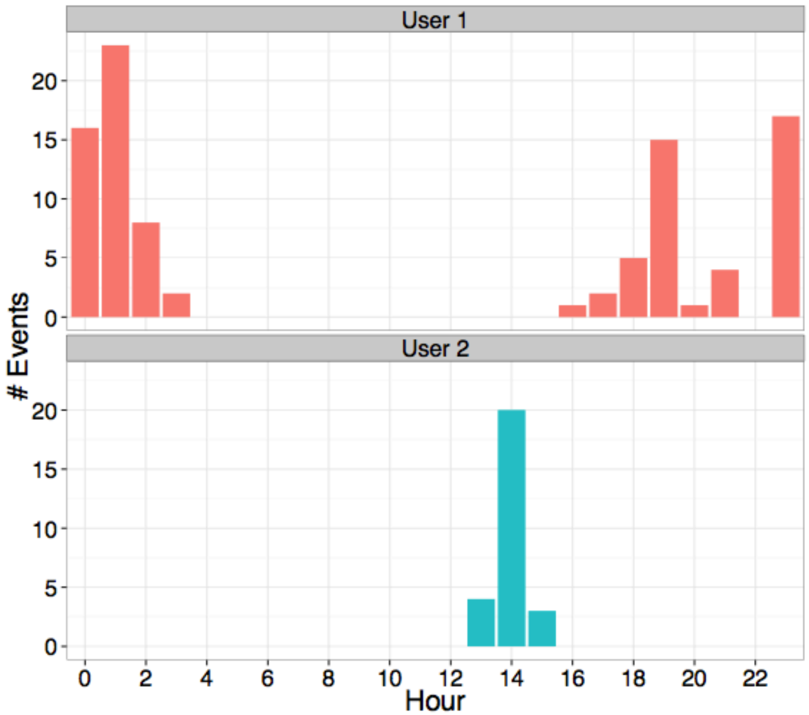
$$\vec{u} := \sum_{e \in E_u} \frac{1}{(1+\alpha)^{\tau(e)}} \times \vec{e},$$

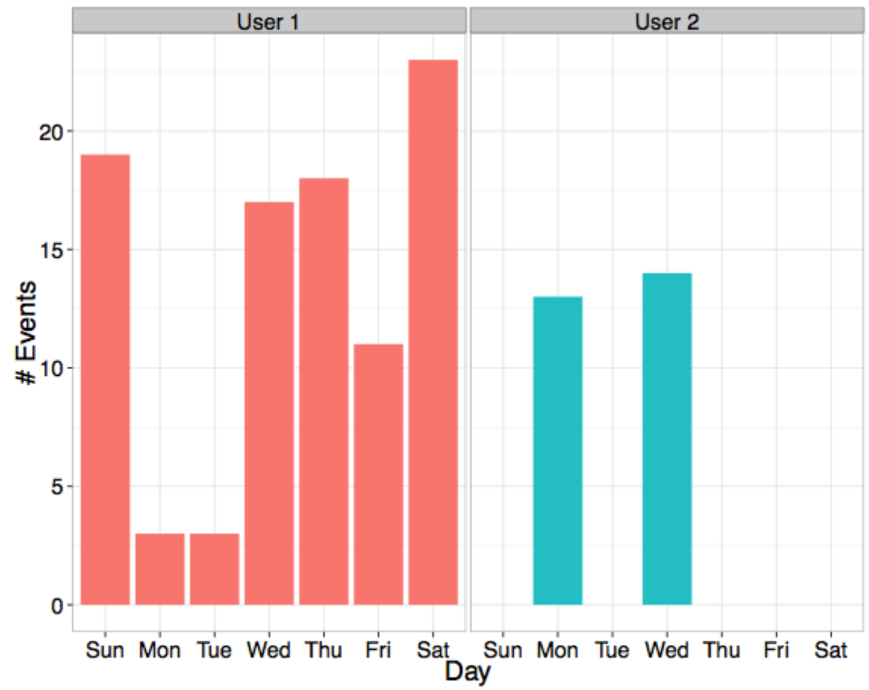
- $\vec{e}$ ... TF-IDF of event e.
- α ... time decay factor.
- $\tau(e)$ ... days from the RSVP to e until the recommendation moment.

The recommendation score is given by:

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

Which time/day users go to Events?





Time-Aware Recommender

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

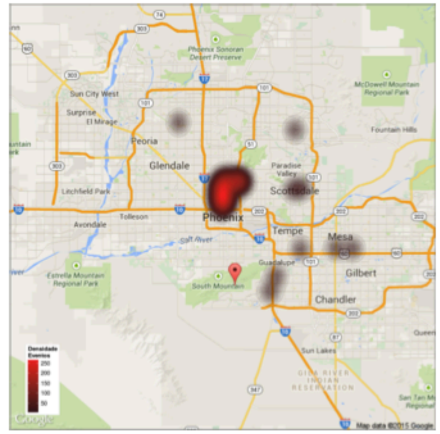
$$\vec{u} := \frac{1}{|E_u|} \sum_{e \in E_u} \vec{e}$$

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

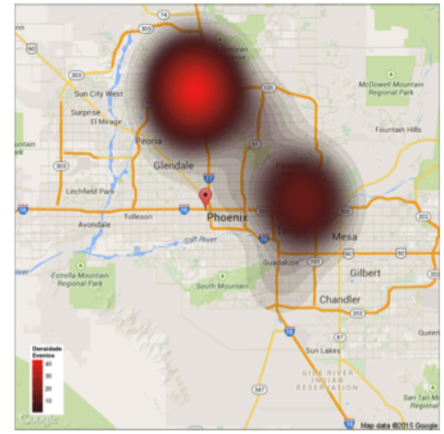
The recommendation score is now:

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

Location-Aware



User 1



User 2

The recommendation score is given by

$$\hat{s}(u,e) := \frac{1}{|L_u|} \sum_{l' \in L_u} K_{\mathbf{H}}(l_e - l'),$$

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

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

Learning to Rank Events

Let $\mathcal{D} := \{(x_1, y_1), \ldots, (x_n, y_n)\}$ 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.

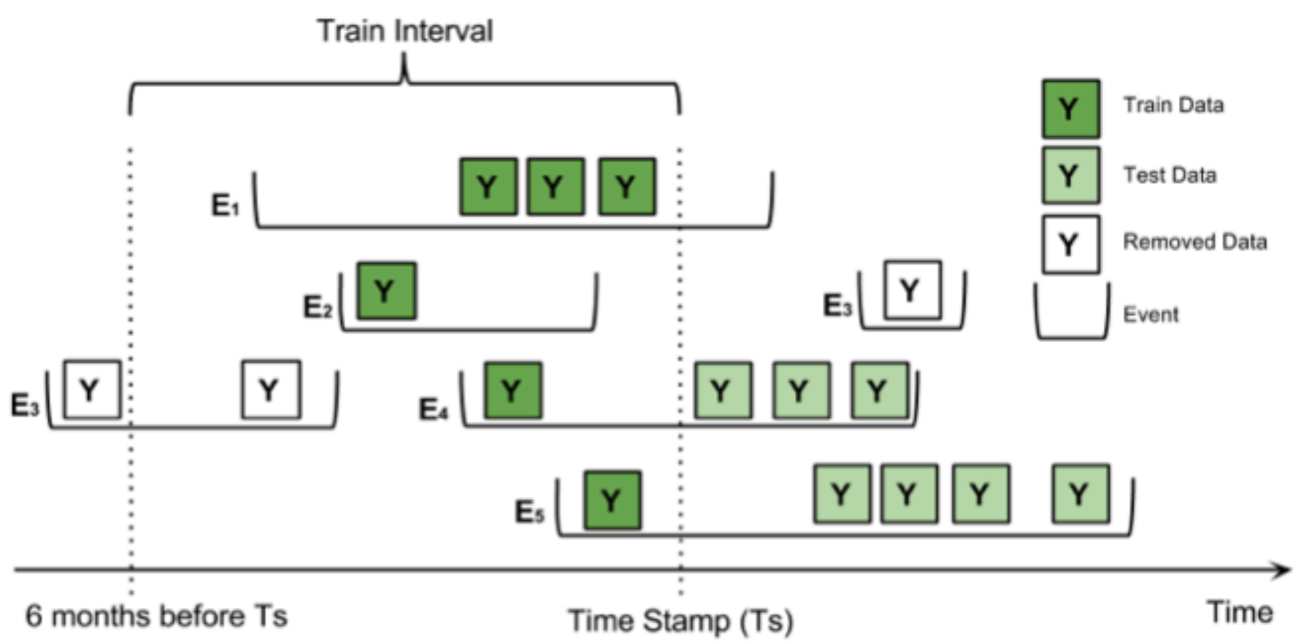
Data

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

City	$\lvert G \rvert$	$\lvert U \rvert$	$\lvert E \rvert$	**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 Protocal

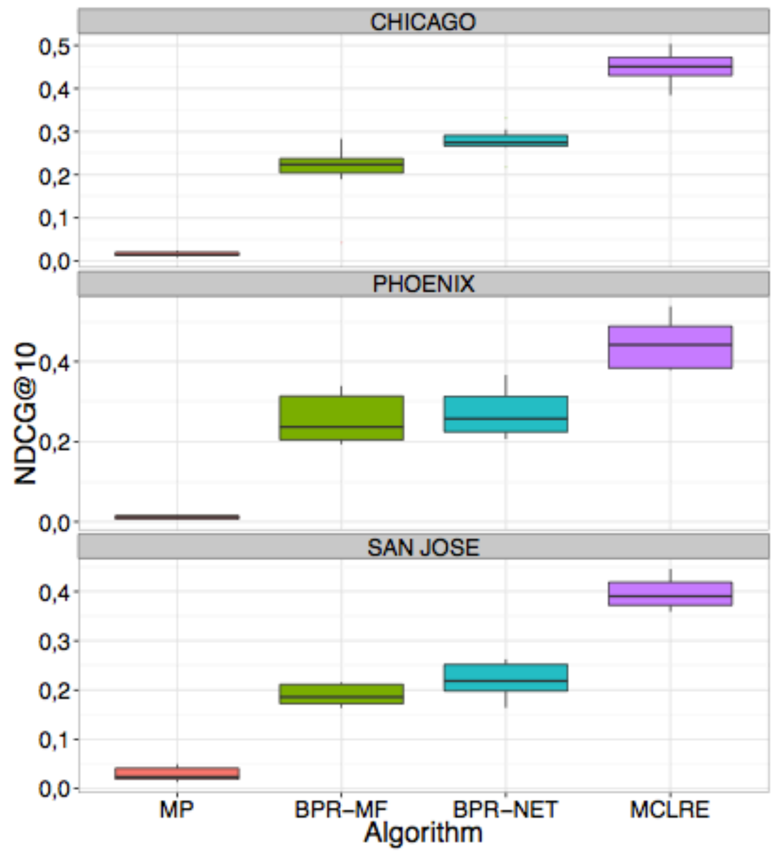
- 12 timestamps equally spaced in time over 52 months.
- Sliding training window.
- For each city, the four initial partitions are used as validation sets and the remaining partitions for evaluation.



Compared Algorithms

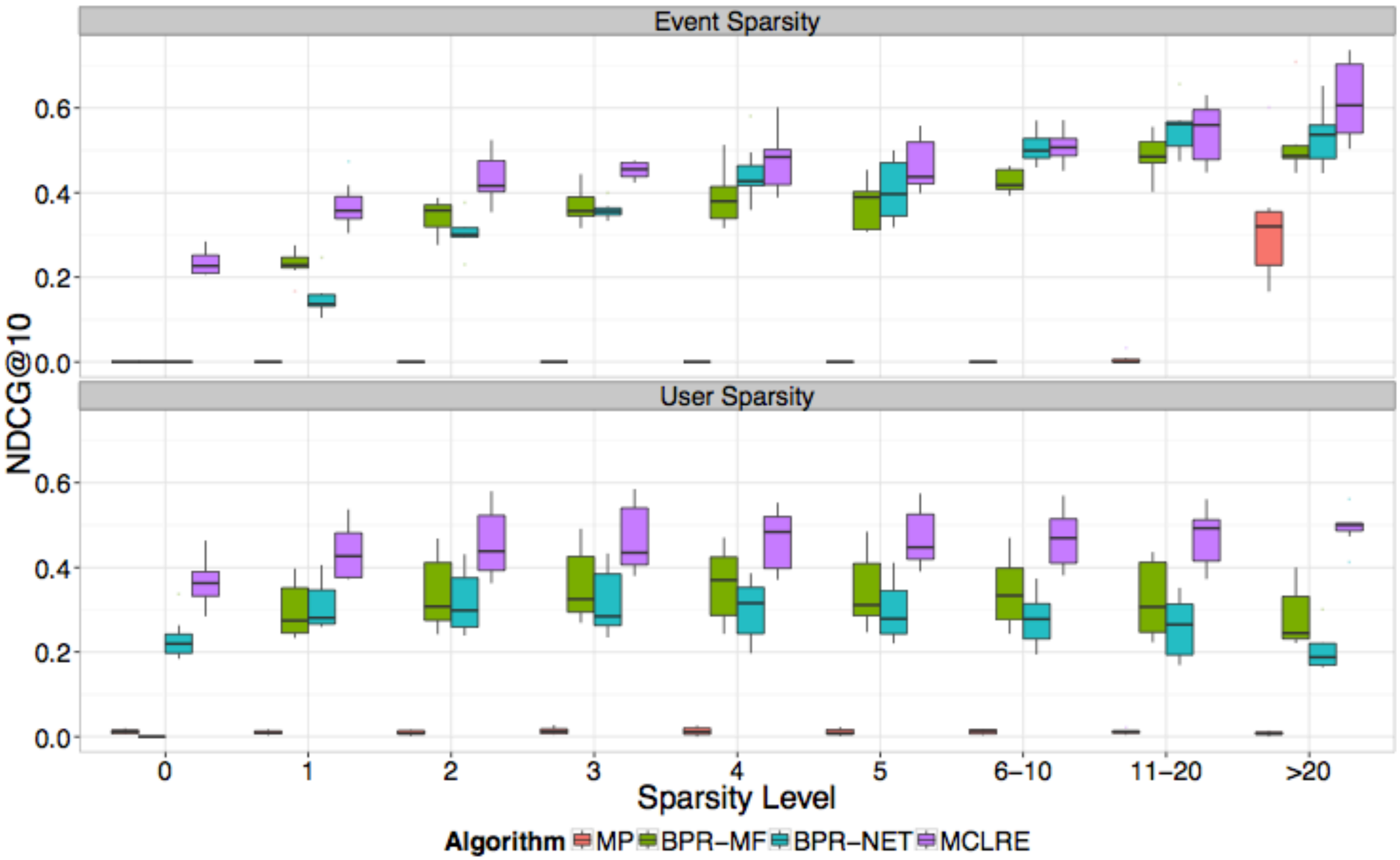
- Most Popular
- BPRMF [Rendle, 2009]
- BPR-NET [Qiao, 2014]
- **MCLRE**
- Evaluation Metric: NDCG@10

nDCG@10



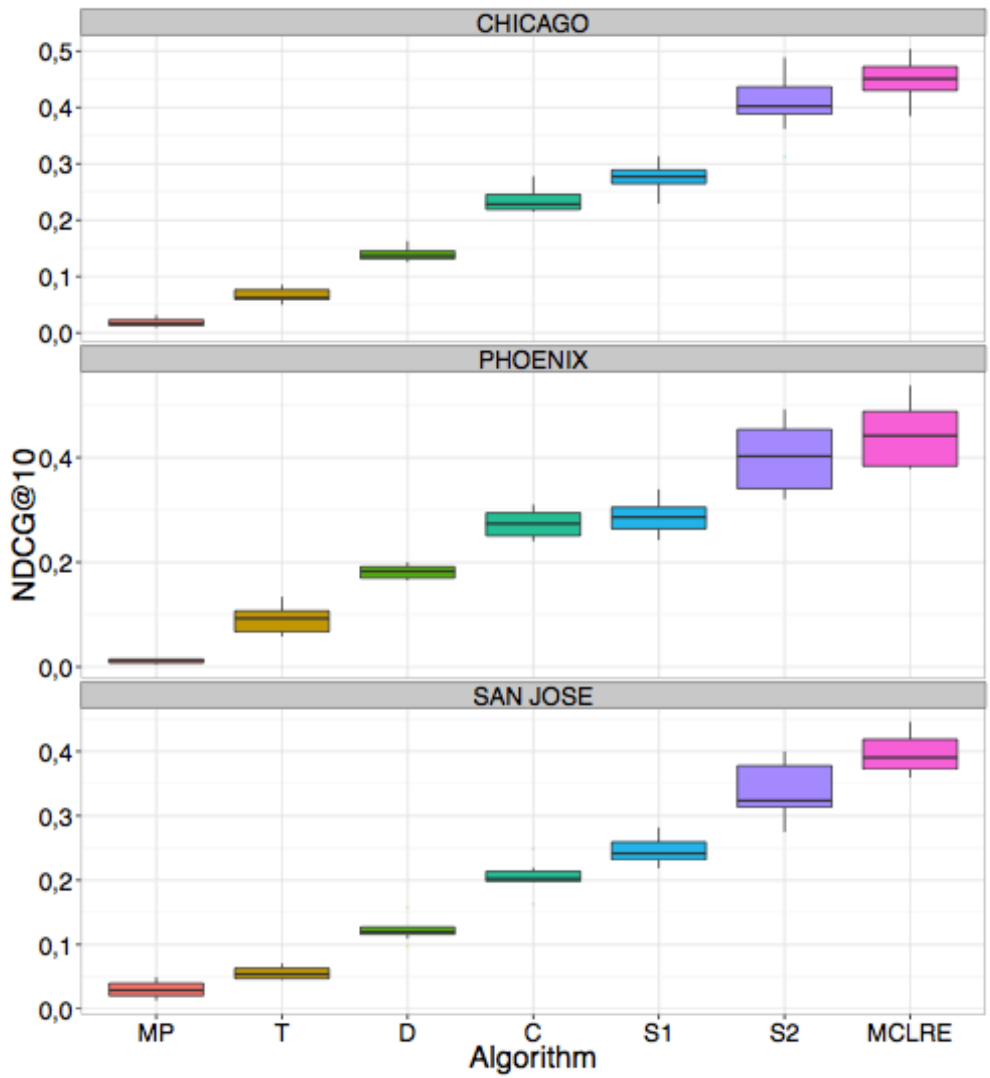
Improvement of up to 79%!

Robustnes to Data Spasity



Only method able to recommend in full cold-start!

Contextual Features Importance



Conclusions and Future Work

- 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.
- Easy to include/remove contexts and highly parallelizable.
- For future work: more cities, more contexts, more EBSNs.

References

- Multi-relational matrix factorization using bayesian personalized ranking for social network data. Artus Krohn-Grimberghe, Lucas Drumond, Christoph Freudenthaler, and Lars Schmidt-Thieme. In Proc. of WSDM '12.
- BPR: Bayesian Personalized Ranking from Implicit Feedback. Steffen Rendle, Christoph Freudenthaler, Zeno Gantner and Lars Schmidt-Thieme. In Proc. of UAI'09.
- Event Recommendation in Event-Based Social Networks. Zhi Qiao and Peng Zhang and Chuan Zhou and Yanan Cao and Li Guo and Yanchuan Zhang.In Proc. of AAAI'14.
- Linear Feature-based Models for Information Retrieval. Donald Metzler and W. Bruce Croft. Inf. Retr. 2007.