Event Recommendation in Event-based Social Networks

Augusto Q. de Macedo
Federal University of Campina Grande
Aprigio Veloso 882
Campina Grande, Brazil
augusto@copin.ufcg.edu.br

Leandro B. Marinho
Federal University of Campina Grande
Aprigio Veloso 882
Campina Grande, Brazil
lbmarinho@computacao.ufcg.edu.br

ABSTRACT
With the large number of events published all the time in event-based social networks (EBSN), it has become increasingly difficult for users to find the events that best match their preferences. Recommender systems appear as a natural solution to this problem. However, the event recommendation scenario is quite different from typical recommendation domains (e.g. movies), since there is an intrinsic new item problem involved (i.e. events can not be "consumed" before their occurrence) and scarce collaborative information. Although some few works have appeared in this area, there is still lacking in the literature an extensive analysis of the different characteristics of EBSN data that can affect the design of event recommenders. In this paper we provide a contribution in this direction, where we investigate and discuss important features of EBSN such as sparsity, events life time, co-participation of users in events and geographic features. We also shed some light on the performance and limitations of several well known recommendation algorithms and combinations of them on real data collected from meetup.com.

Categories and Subject Descriptors
H.3.3 [**Information Search and Retrieval**]: Information filtering; H.2.8 [**Database Applications**]: Data mining

General Terms
Algorithms, Experimentation

Keywords
Recommender systems, Statistical Analysis, Social network, Cold-Start

1. INTRODUCTION
In the last few years the Event-Based Social Networks (EBSN), such as Meetup[1] and Plancast[2], have gained momentum due to their ability to connect people around the events they attended or are likely to attend in the future. In EBSN people can create events of any kind, for example, musical concerts and political manifestations, and share it with other users. With the large number of events available all the time, especially in large and touristic cities, it has become increasingly difficult for the users to find the events that best match his/her preferences. Recommender systems appear as a natural solution for this problem.

[1]www.meetup.com
[2]www.plancast.com

The event recommendation problem, however, is quite different from the classic recommendation scenarios (e.g., movie recommendation), where the items to be recommended have already been consumed/rated by other users. In EBSN, the events to be recommended can not be "consumed" or rated before its occurrence, so, in principle, there is a lack of collaborative data available for traditional collaborative filtering-based algorithms to operate upon, which raises the issue known as the new item cold-start problem. One way to alleviate this problem is to use the intention of users on going or not to events, through their RSVPs[3], as explicit feedback data. But as we will show along the paper, even this kind of data is very sparse.

Although some few works have appeared recently in this area, there is still a gap in the literature concerning an extensive analysis of the different characteristics of EBSN data that can affect the design of effective event recommenders. In this paper we try to fill in this gap by addressing the following questions:

- How sparse is the RSVP data and how it affects collaborative-filtering algorithms?
- In which point of the event life time users tend to provide RSVPs?
- How the geographic distance between the users home and active events affect their decision on attending these events?
- Are past RSVPs usefull for predicting future RSVPs?

We derive important insights from this investigation that we believe will pave the way to the design of more efficient and informed recommendation algorithms. Moreover,

[3]RSVP stands for the French expression "répondez s'il vous plaît", meaning "please respond"

we compare several well known recommendation algorithms and discuss their performances and limitations on real data collected from the Meetup platform, a popular EBSN that offers large portions of event data through their API.

The rest of this paper is organized as follows. In Section 2 we discuss related works. In Section 3 we present the data collection and analysis. In Section 4 compare several well known algorithms from the literature and discuss their performances and limitations. Section 5 concludes the work.

2. RELATED WORKS

In this section we summarize the most relevant related work on event recommendation.

Minkov et al. [4] approach the event recommendation problem through a ranking-based matrix factorization algorithm. For composing the training data, explicit feedback was required through a form where users had to indicate which events, in this case scientific seminars, they were likely to attend. The results of this paper show that this approach is superior to content-based filtering. Although they have conducted experiments with real users, it consisted of a small scale experiment where only 90 users over 15 weeks were considered. Moreover, it was required explicit feedback from the users. Our work focus on an offline large scale analysis and experimentation on data collected from a popular EBSN.

A seminal and closely related work to ours is the one introduced in [3] where the authors analyze real data collected from Meetup all over USA and investigate EBSN properties, such as heavy-tailed degree distributions, strong geographic dependence of social interactions, and the interplay between online and offline interactions of users. They also propose a recommendation model of users in EBSN.

In [5] it is proposed a content-based recommender where cultural events metadata are enriched with open linked data available on the web. While this approach might work well for small scope event domains, it may find problems to cover the multitude of event types of EBSN. Another work from Pessemier et al. [1] presented a smartphone application, Outlife, to recommend events for users and users to invite for an event based on the users Facebook profiles. The event recommendation is addressed by selecting the most appropriate algorithm for each situation (with a decision tree) out of a set of recommender algorithms. If no ratings are available a content-based algorithm is used.

A recent work by Khrouf et al. [2] propose a hybrid event recommender that combines linked open data, social information and content features. While the authors focus their experiments on a small set of Last.fm users and events and a small set of event types (i.e. mostly concerts and festivals), we investigate large scale data on a multitude of event types. Furthermore, wile the authors of this work focus on the denser portions of the data, we investigate the performance of several recommenders under the true level of sparsity found on EBSN.

Thus, our work is complementary to the aforementioned works, where we investigate previously unexplored features of EBSN and how they can affect the performance of event recommendation algorithms.

3. DATA ANALYSIS

Meetup is one the world's largest EBSN nowadays[4]. It provides an on-line environment where people can meet both on-line and face-to-face. Events of all kinds are published all the time, ranging from simple get togethers to large concerts and conferences. Moreover, large portions of data are offered through the site on-line API[5], which turns Meetup into a good test bed for investigating new event recommendation approaches.

3.1 Data Collection

The cities chosen for our experiments were Phoenix, Chicago and San Jose, all from USA. These cities were selected because they (i) are among the top cities in number of users and events in Meetup and (ii) are located in different states, which represent eventual cultural differences and thus contribute to form a rich and diverse sample to work with.

Meetup is organized in on-line groups, where every group has a physical location. To collect the data, we passed the city names as seeds and retrieved all the groups located in a radius of 100 miles from a city location returned by Meetup. Then every user, event and RSVP (i.e. user-event pairs) of those groups were retrieved. The data collected comprise the period from January, 2010 to December, 2011. Table 1 presents the characteristics of the data collected. It is worth noting the extreme sparsity of RSVPs in all cities considered.

Table 1: Data Statistics

City	Users	Events	RSVPs	Sparsity
Phoenix	589,808	215,338	1,557,161	99.998%
Chicago	719,011	220,076	1,353,795	99.999%
San Jose	281,547	242,216	1,717,792	99.997%

In the following we investigate some characteristics of this data with respect to RSVPs, event life time, co-participation of users in events and the distances between the users home and event locations.

3.2 RSVP Analysis

When an event is created, users can provide RSVPs to it, i.e., provide (Yes or No) responses. We consider that a user who respond with ”Yes” has a higher probability to attend the event than the user who answer ”No” or provide no answer. Hence, we use this response as a proxy value to the event attendance rate, as the real attendance count is not available in Meetup.

Figure 1 shows the distribution of positive RSVPs per event for all cities. The numbers show that more than 45% of the events have at most 1 RSVP. Approximately 90% of the events have at most 10 RSVPs in all cities. The logarithm scale in the x-axis emphasizes the high skewness of the distribution leading one to conclude that the large majority of events, in all cities investigated, have low attendance.

[4]http://www.meetup.com/about/
[5]www.meetup.com/meetup_api/

This represents a major problem for most of the collaborative filtering-based recommendation algorithms which are well known to deteriorate under severe levels of sparsity.

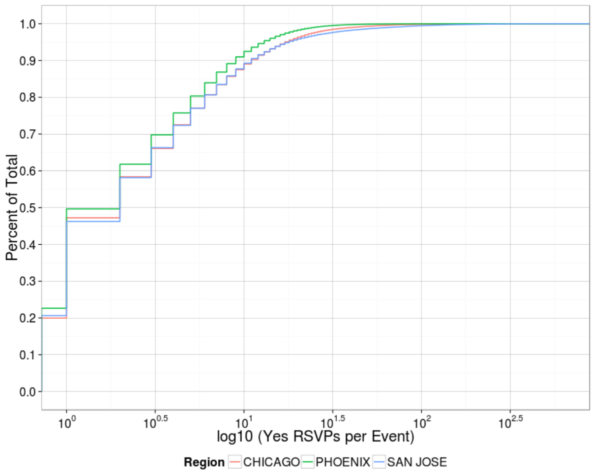

Figure 1: Cumulative Distribution of RSVPs per Event

3.3 Event Life Time

We consider the life time of an event as the period between its creation in Meetup and its occurrence. In Figure 2 we can see that most of the events have a life time ranging from 5 to 100 days. This means that while a small percentage of events have a very short life time (1 day), most of the events are active long enough to be discovered by the users or brought to their attention.

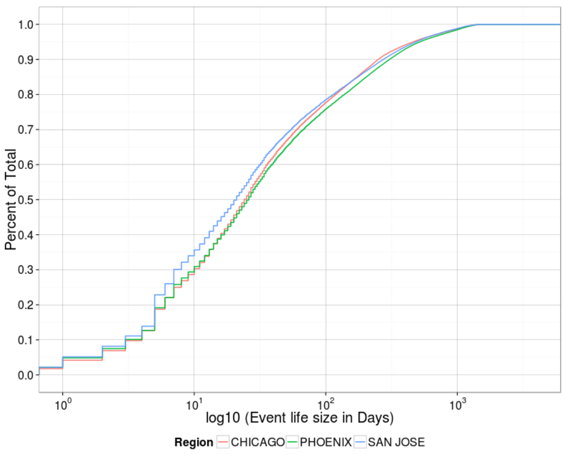

Figure 2: Cumulative Distribution of Event Life

3.4 When do RSVPs occur?

Here we investigate when exactly the positive RSVPs occur during the life time of events. Figure 3 shows in the x-axis the 21 first positive RSVPs[6], in chronological order, regarding all the events of the three cities considered. The y-axis ranges from 0, when the event is created, to 1, when it happens. Notice that the more positive RSVPs events receive, the closer to the events occurrence the RSVPs are given. Although the cities investigated present small variations in this respect, they follow the same overall pattern, i.e., most of the RSVPs are provided close to the occurrence of the event. This is even more visible in the events with a life time greater than 100 days, for example, which we noticed to receive more than 80% of all positive RSVPs (among the 21 considered) in the last 20% of the their life times.

This observation bears several implications to the design of effective event recommenders. For example, after the creation of the event there will be scarce collaborative (in terms of RSVPs) information to be used, leaving room to content-based approaches. As the occurrence of the event approaches, more RSVPs are provided which favours collaborative filtering-based methods and hybrid approaches.

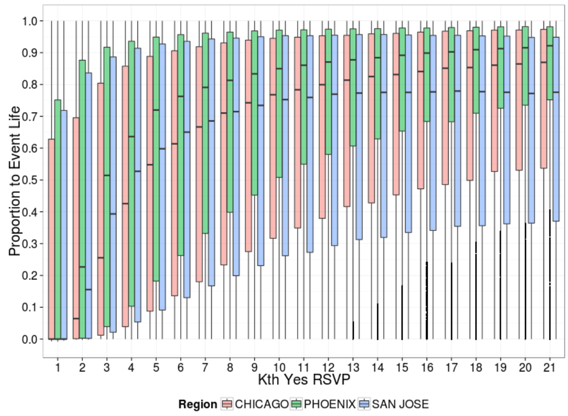

Figure 3: Cumulative Distribution of the time to the k-th "Yes" RSVP relative to the event life time

3.5 Collaborative Analysis

The collaborative aspect of the data was investigated by the distribution of events co-participation by two different users (in terms of positive RSVPs). Our analysis suggests that approximately only 30% of the users co-participated in two or more events in all cities considered. This observation represents an empirical bound to the effectiveness of collaborative filtering-based recommenders.

3.6 Distance Analysis

Figure 4 depicts the distance distribution between the users home and events locations, also investigated by other works [2, 6]. We can see that around 50% for the users provided positive RSVPs to events within 10 Km from their homes, while users do not provide RSVPs to events farther then 100 Km to their homes. A recommendation algorithm could use this observation to weigh events nearby the users home higher than farther events.

4. EVALUATION

In this section we compare some well known top-n recommendation algorithms for the event recommendation task. We also evaluate the algorithms in different levels of sparsity in order to investigate their limitations.

[6]Note that approximately 95% of all events have 21 or less RSVPs

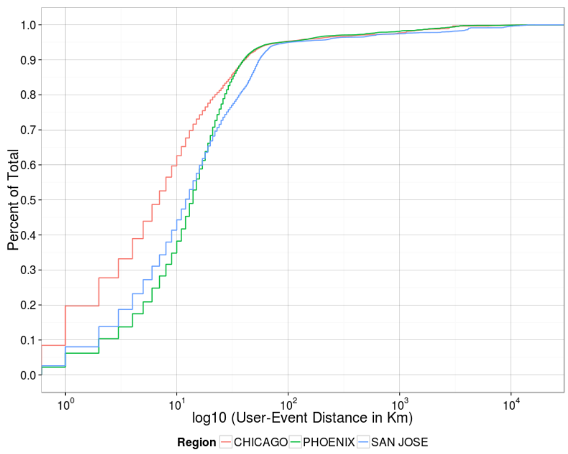

Figure 4: Cumulative Distribution of the Distance between the User and Event Location

4.1 Data Preparation

The data sets of each city were time split in order to resemble a real world setting. We selected 6 time stamps, equally spaced in time, for splitting training and test. For each partition time stamp, we used the previous 6 months for training and the events created during these 6 months but occurring after the partition time stamp for test. The average number of users, events and user-events pairs (RSVPs) after these partitions are displayed in Table 2 .

Table 2: Average number of susers, events and user-events pairs after paritions

City	# Users	# Events	# User-Events
Phoenix	2,176.8	4,483.3	9,870
Chicago	2,814.3	2,955.7	8,703.7
San Jose	3694.7	3,052.2	11,025.5

4.2 Sparsity Analysis

Here we investigate the sparsity of the recommendable events, in all partitions, in the following levels:

$$\{0, 1, 2, 3, 4, 5, 6 - 10, 11 - 20, > 20\}$$

where each level denotes the number of positive RSVPs received per event. Figure 5 shows the event sparsity level plot. The y-axis counts the number of events in the test set that has the given sparsity level in the train. This plot tell us that regardless of when we partition the data set, there will be always a large number of events with no RSVPs. Therefore, cold-start appear as an inherent problem of the event recommendation domain.

4.3 Evaluation Metric

In this paper we are considering top-n item recommendations, which are usually related to the generation of a personalized ranking recommendation list. In our case, the task of the recommender is to correctly predict which events a given test user will provide positive RSVPs in the future (test set). We have used the well known Normalized Discounted Cumulative Gain (NDCG) metric truncated to 20

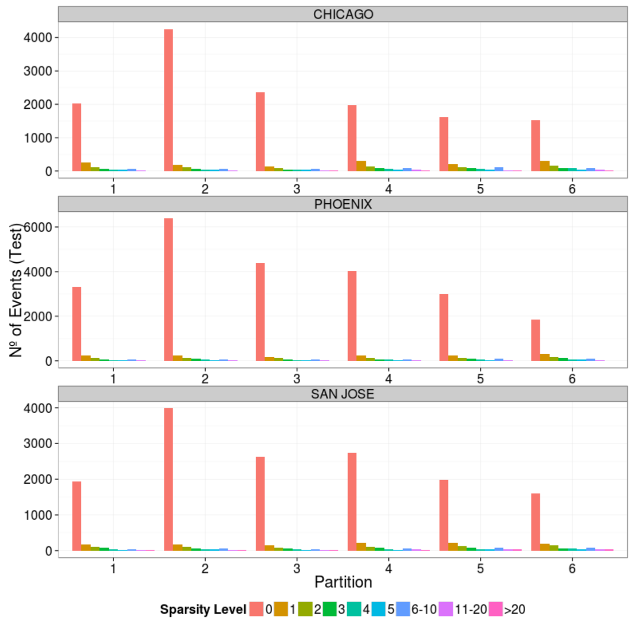

Figure 5: Event Sparsity Level per Partition

recommendations. So, the $NDCG@20$ for a given user u is defined as follows.

$$DCG@20 := \sum_{i=1}^{k} \frac{2^{rel_i} - 1}{log_2(i+1)} \tag{1}$$

$$NDCG@20 := \frac{DCG@20(u)}{IDCG@20(u)} \tag{2}$$

In Equation 1 above, rel_i is 1 or 0 if the event at position i is relevant or not respectively, and the function $IDCG_p(u)$ returns the perfect ranking value, acting as a normalization term.

4.4 Experimental Results

In this section we compare the following well know top-n item recommendation algorithms from the literature:

- *Random*: The recommendation list is randomly generated.
- *Most-Popular*: The candidate events are ranked in descending order of popularity. We define popularity of an event as the number of positive RSVPs received.
- *Location-Aware*: This algorithm ranks the events based on their distances to the users home, assuming that nearby events are more likely to be attended by the user. This algorithm does not rely on RSVP data.
- *BPR-MF*: The Bayesian Personalized Ranking [7] is a state-of-the-art matrix factorization-based algorithm for top-n item recommendation. Its hyper-parameters were defined by grid-search where the best results were achieved with 50 latent factors, 0.1 for the gradient descent learning rate and 500 iterations.

- *User-KNN* and *Item-KNN*: Correspond to the classic k-nearest neighbor collaborative filtering based on users or items. The Collaborative Analysis of Section 3.5 have an important role in these algorithms. After a grid-search, the neighborhood size was set to 100 for both algorithms.

- *Logistic-Regression*: We also tested an hybrid algorithm where the event scores of all aforementioned algorithms (except the *Random*) are fed into a logistic regression model.

Figure 6 displays the recommendation performances of each algorithm in each city considered. In spite of the high sparsity levels, the KNN based algorithms attain the best performances in comparison to the other individual algorithms. One possible explanation to this result is that, in many cases, users who will attend the same event are already friends or acquaintances and therefore may have mutual influence on the selection of future events. The *Location-Aware* algorithm is comparable to the *Most-Popular*, leading one to conclude that the geographic distance, although carrying some signal, is not among the main reasons affecting the decision of a user in attending or not an event. Another potential reason for this result is the inaccuracy of the users home position that is approximated from its IP address. Nonetheless, since this algorithm does not rely on RSVP data, it represents a good alternative for full cold-start scenarios. Although BPR-MF is usually better than simpler KNN based recommenders in other domains, this is not the case here. This might be related to the extreme level of sparsity of EBSN, which is not observed in other papers that concentrate their experiments on denser regions of collaborative data.

The *Logistic-Regression* approach is at least as good as the *Item-KNN*, attaining slightly better results in San Jose. Nonetheless, it is worth noticing that the overall $NDCG@20$ values are very low, achieving at most 0.3 in the best cases.

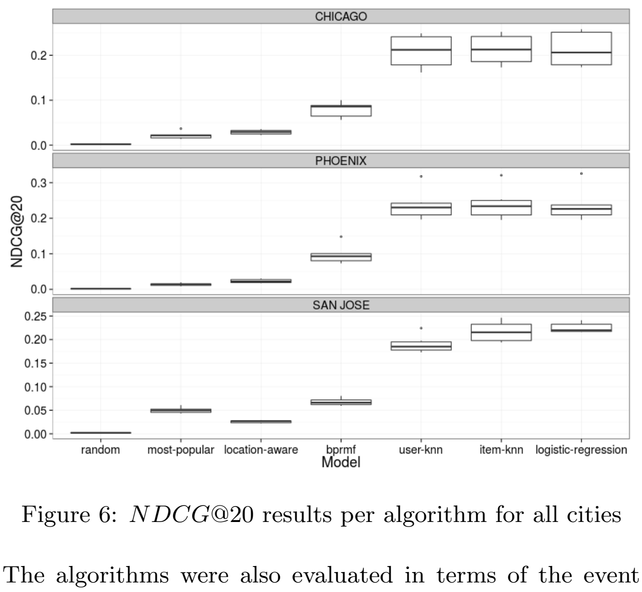

Figure 6: $NDCG@20$ results per algorithm for all cities

The algorithms were also evaluated in terms of the event sparsity level. Here we want to investigate which events are more likely to be correctly recommended according to their sparsity levels. We want to answer questions like: events having 20 or more positive RSVPs are more likely to be correctly recommended than events having 10 or less RSVPs? Figure 7 shows the results of this analysis for all algorithms in each sparsity level considered. The x-axis encode the algorithms and the colors encode the sparsity levels.

As expected, the more positive RSVPs an event has, the more likely it is to be correctly recommended by all recommendation algorithms, except the *Location-Aware* that does not use RSVP information, the *Item-KNN* and the *Logistic-Regression* that seems to deteriorate in Phoenix with the decrease of sparsity.

5. CONCLUSIONS AND OUTLOOK

In this paper we approached the problem of event recommendations in EBSN. We showed that this task is more challenging than typical recommendation domains investigated by the literature since EBSN data is inherently cold-start. One alternative to alleviate this problem is to use RSVP data, although this data is still very sparse.

We analysed important features of EBSN that can affect the design of effective event recommenders and compared well known algorithms on real data collected from the popular EBSN Meetup. Our main findings are summarized below:

- RSVPs tend to be given close to the occurrence of the event.
- The largest majority of events are cold-start.
- Despite the high sparsity of RSVP data, KNN-based algorithms appear as the best single alternative.
- Matrix-factorization does not perform as well in this domain as it does in other more typical domains.

In future work we intend to investigate the influence of group membership on event attendance and more sophisticated context-aware models to exploit the contextual data of events, such as time, tags and events descriptions.

6. REFERENCES

[1] T. De Pessemier, J. Minnaert, K. Vanhecke, S. Dooms, and L. Martens. Social recommendations for events. In *CEUR workshop proceedings*, volume 1066, page 4, 2013.

[2] H. Khrouf and R. Troncy. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the 7th ACM Conference on Recommender Systems*, RecSys '13, pages 185–192, New York, NY, USA, 2013.

[3] X. Liu, Q. He, Y. Tian, W.-C. Lee, J. McPherson, and J. Han. Event-based social networks: linking the online and offline social worlds. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '12, pages 1032–1040, New York, NY, USA, 2012. ACM.

[4] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola. Collaborative future event recommendation. In *Proceedings of the 19th ACM international conference on Information and knowledge management*, CIKM '10, pages 819–828, New York,

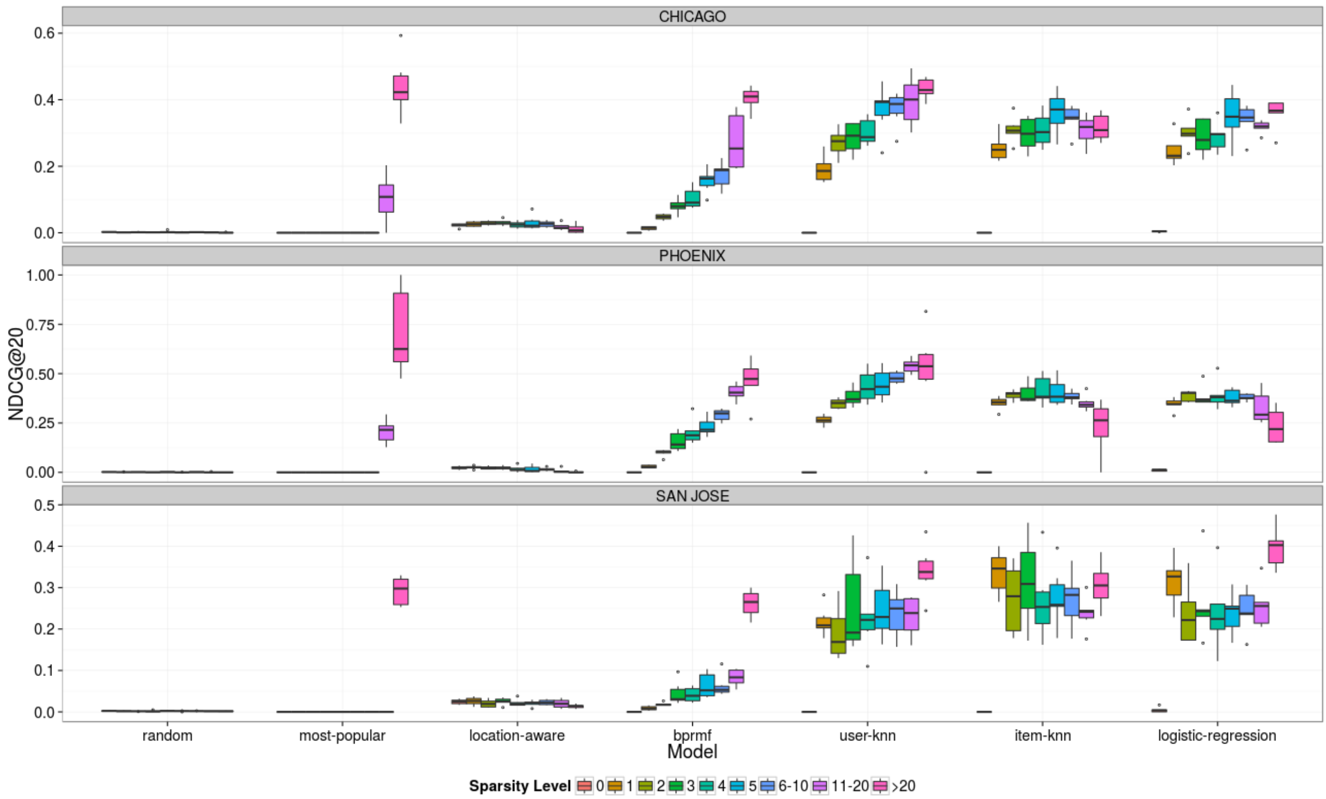

Figure 7: $NDCG@20$ results per algorithms and event sparsity level for all cities

NY, USA, 2010. ACM.

[5] T. D. Pessemier, S. Coppens, E. Mannens, S. Dooms, L. Martens, and K. Geebelen. An event distribution platform for recommending cultural activities. In *WEBIST*, pages 231–236, 2011.

[6] D. Quercia, N. Lathia, F. Calabrese, G. Di Lorenzo, and J. Crowcroft. Recommending social events from mobile phone location data. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 971–976, Washington, DC, USA, 2010. IEEE Computer Society.

[7] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, UAI '09, pages 452–461, Arlington, Virginia, United States, 2009. AUAI Press.