# RecSys for Distributed Events: Investigating the influence of recommendations on visitor plans

Richard Schaller
Computer Science (AI Group)
Univ. of Erlangen-Nuremberg
richard.schaller@fau.de

Morgan Harvey
Faculty of Informatics
University of Lugano
morgan.harvey@usi.ch

David Elsweiler
I:IMSK
University of Regensburg
david@elsweiler.co.uk

## ABSTRACT

Distributed events are collections of events taking place within a small area over the same time period and relating to a single topic. There are often a large number of events on offer and the times in which they can be visited are heavily constrained, therefore the task of choosing events to visit and in which order can be very difficult. In this work we investigate how visitors can be assisted by means of a recommender system via 2 large-scale naturalistic studies ($n$=860 and $n$=1047). We show that a recommender system can influence users to select events that result in tighter and more compact routes, thus allowing users to spend less time travelling and more time visiting events.

## Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information search and retrieval—*Information filtering*

## Keywords

Mobile Assistance System, Distributed Events, Recommender

## 1. INTRODUCTION AND MOTIVATION

A distributed event is a collection of smaller, single events occurring at the same day and conforming to one overarching theme. Such events are growing in popularity all over the world. Well known examples include the Cannes International Film Festival and the Edinburgh Festival Fringe. Typically, the events are geographically (e.g. throughout a city) and temporally (e.g. during an evening) dispersed. While variety is the biggest selling point of these events, making the decision of which sub-events to visit is not straightforward. Visitors have to discover which events are on offer and select a small subset from a list of hundreds that they find appealing. This decision depends on factors such as user preferences, time constraints, the location of the event and transport connections [5]. As a result of the sheer choice, people often feel overloaded and rely on tips from friends and on serendipitous discovery [6].

Recommender systems (RS) present a possible solution for narrowing down the amount of events visitors have to consider whilst still taking personal preferences into account. Recommenders have been explored in several domains e.g. e-commerce or leisure [3, 1]. However to the best of our knowledge no evaluation of RS in this particular type of leisure event domain exists. In this work we examine how different RS perform in the context of 2 distributed events. Specifically, we investigate how RS affect the events and routes users choose. The main contributions are:

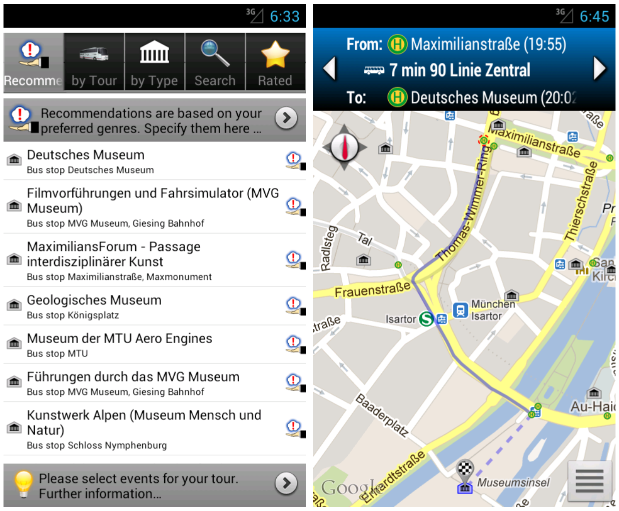
- An offline evaluation of different RS on rating data collected during two distributed events, showing how performance can be improved by combining these RS.
- An online study that investigates how RS, when used in practice, can influence user behaviour, specifically properties of routes visitors create in collaboration with our system.

## 2. DATA COLLECTION

Our test collection is derived from transaction logs from an Android application developed for the Long Night of Music 2012 (LNMusic) and the Long Night of Munich Museums 2012 (LNMuseum). These are large popular events which take place in the city of Munich, where the user has a vast choice of things they can do and limited time to see them. The app offers the user four ways to locate events of interest them (browsing by route or topic, free-text search and by way of recommendations). Each feature has a separate tab as depicted in Figure 1:left. We focus here on the first tab (*Recommender*) which offers personalized recommendations to the user, generated from a hybrid system combining content-based and collaborative filtering algorithms. Content-based algorithms require a user-profile, which can be specified via a slider interface to indicate interest in different music genres (LNMusic) or topics (LNMuseums). For the collaborative filtering algorithm, the user profile consists of all the events the user has already selected. The ratings of all users are transferred to a central server that calculates a user-specific model and sends it back to the client. The 15 (as yet unrated) events with the highest scores from the RS are shown to the user. These events are also highlighted if they appear in the other tabs. Events can be rated in 3 ways: “Don’t want to visit”; “would like to visit, if it fits into the route”; “want to visit in any case”. For the RS we reduced these options into a binary rating: dislike event and like event.

Once the user has indicated the events they would most like to visit, the system creates an itinerary for the evening

*SIGIR'13,* July 28–August 1, 2013, Dublin, Ireland.
Copyright 2013 ACM 978-1-4503-2034-4/13/07 ...$15.00.



Recommendations are based on your preferred genres. Specify them here ...

Deutsches Museum
Bus stop Deutsches Museum

Filmvorführungen und Fahrsimulator (MVG Museum)
Bus stop MVG Museum, Giesing Bahnhof

MaximiliansForum - Passage interdisziplinärer Kunst
Bus stop Maximilianstraße, Maxmonument

Geologisches Museum
Bus stop Königsplatz

Museum der MTU Aero Engines
Bus stop MTU

Führungen durch das MVG Museum
Bus stop MVG Museum, Giesing Bahnhof

Kunstwerk Alpen (Museum Mensch und Natur)
Bus stop Schloss Nymphenburg

Please select events for your tour.
Further information...

From: Maximilianstraße (19:55)
7 min 90 Linie Zentral
To: Deutsches Museum (20:0

**Figure 1: Event browser with recommender tab (left) and map screen with the planned route (right)**

containing a subset of the selected events, accounting for constraints such as start and end times of events and travel time between events based on public transport routes and schedules. This further processing of the selected events distinguishes our RS from typical RS problems.

Both apps were available for download from the Google Play Store and advertised on the official LNMusic and LNMuseums web page. We recorded all interactions with the app from 860 and 1,047 users. This included tab changes, tour/event genre/topic selection, click-throughs, generated tours, modifications to tours, as well as all event ratings submitted. In total we obtained 4,973 ratings (1,568 neg./3,405 pos.) for the LNMusic and 10,992 ratings (4,145 neg./6,847 pos.) for the LNMuseums. We use these ratings as a basis for the offline experiments described in the following section.

## 3. OFFLINE COMPARISON

To ascertain which RS performs best in recommending events to visitors, we compare 6 different systems against each other as well as various combinations of these:

1. *Random.* Simply assigns a uniformly distributed score between 0 and 1 to each event.

2. *Popularity.* Each event is assigned a score relative to the prior probability of it being chosen.

3. *Content-based (CB).* RS which uses the similarity between the genres/topics assigned to each event and the user profile defined via sliders.

4. *SVD*-based collaborative filtering algorithm which uses a gradient decent approach to calculate the matrix decomposition [4]. In training the model we used 5 dimensions, a learning rate of 0.001, a regularization constant of 0.001 for the matrix and a constant of 0.005 for the biases, these values were determined based on some preliminary experiments.

5. *BLITR.* A modern, state-of-the-art collaborative filtering algorithm [2]. We conducted experiments in advance to find optimal parameter settings: 5 dimensions for the user latent factors, 8 dimensions for the event latent factors, $\alpha$ and $\beta$ were both set to 5.0.

6. *Temporal (Temp)* - derived from a metric developed in [5] where it was shown that distributed event visitors don't like to spend much time travelling[1], therefore it is important to choose events that form a compact route. A measurement for this is the so-called *temporal contiguity (TC)* which is defined over a set of events as the time needed for a route utilising all available transport links and visiting all events in that set[2].

| Recommender | LNMusic | LNMuseums |
|---|---|---|
| Random | 9.42% | 12.72% |
| Popularity | 27.22% | 38.08% |
| Content-based (CB) | 19.62% | 29.23% |
| SVD | 21.20% | 32.42% |
| BLITR | 16.04% | 22.38% |
| Temporal (Temp) | 19.19% | 17.33% |
| $0.75{\cdot}\text{CB} + 0.25{\cdot}\text{SVD}$ | 21.53% | 30.30% |
| $0.5{\cdot}\text{CB} + 0.5{\cdot}\text{SVD}$ | 21.99% | 30.70% |
| $0.25{\cdot}\text{CB} + 0.75{\cdot}\text{SVD}$ | 21.62% | 30.30% |
| Dyn(CB,SVD) | 24.12% | 36.18% |
| $0.75{\cdot}\text{Dyn(CB,SVD)} + 0.25{\cdot}\text{Temp}$ | 24.76% | 36.95% |
| $0.5{\cdot}\text{Dyn(CB,SVD)} + 0.5{\cdot}\text{Temp}$ | 24.95% | 36.61% |
| $0.25{\cdot}\text{Dyn(CB,SVD)} + 0.75{\cdot}\text{Temp}$ | 24.71% | 36.18% |
| $0.75{\cdot}\text{CB} + 0.25{\cdot}\text{BLITR}$ | 21.09% | 28.91% |
| $0.5{\cdot}\text{CB} + 0.5{\cdot}\text{BLITR}$ | 20.99% | 28.91% |
| $0.25{\cdot}\text{CB} + 0.75{\cdot}\text{BLITR}$ | 20.95% | 29.00% |
| Dyn(CB,BLITR) *SysA* | 21.00% | 28.98% |
| $0.75{\cdot}\text{Dyn(CB,BLITR)} + 0.25{\cdot}\text{Temp}$ *SysB* | 20.79% | 28.94% |
| $0.5{\cdot}\text{Dyn(CB,BLITR)} + 0.5{\cdot}\text{Temp}$ *SysC* | 21.23% | 28.95% |
| $0.25{\cdot}\text{Dyn(CB,BLITR)} + 0.75{\cdot}\text{Temp}$ | 21.36% | 28.25% |

**Table 1: Recall of different RS, offline evaluation.**

**Evaluation.** Our initial offline experiments are based on data from the LNMusic, which influenced our choice of recommender for the online study for the LNMuseums (Section 4). We also include offline experiments with the LNMuseums data to allow online and offline results to be fairly compared.

To mitigate against the cold-start problem, we consider only users with 10 or more ratings (132 users on the LNMusic, 274 users on the LNMuseums) and run a 5-fold cross-validation [2]. Ratings for a user are chosen as test data with 20% probability, the remaining being used as training data. As some of the chosen algorithms are non-deterministic, we ran each of the tests 30 times and took the mean performance. Of the recommendations generated, we took the 15 highest scoring events and calculated the recall of these with respect to the test data. Recall was chosen as the evaluation metric as it best fits our application domain: Binary ratings render RSME, MAE or similar measurements of closeness between predicted score and actual user rating unsuitable. Ranking-based evaluation approaches are also inappropriate in this setting as only 15 recommendations are shown, meaning that all listed events are likely to be viewed.

The first section of Table 1 shows a summary of the recall performance of the RS listed above. The results were generally (except for *Temp*) better on the LNMuseums than on the LNMusic. This might be caused by the lower number of events and event locations available on the former (174 vs. 204 events, about 100 vs. 120 event locations). *Popularity* outperforms the other systems by a large margin. This is likely a result of the Zipf-like distribution of event popularity (being measured by the number of positive ratings). Despite this high level of performance, we don't investigate this approach further because: 1) it does not deliver a personalized recommendation (meaning all users get the same

[1] Travel time is a much more crucial factor than distance [5].

[2] To allow comparison this value is normalized by the number of events

recommendations), 2) because our app offers other means of accessing popular events (e.g. the search tab) and 3) it is best to remove bias towards popular items [7]. While such bias is typical of RS, it is not favorable since it lessens serendipity and recommendations from the "Long Tail" are considered more valuable.

Of the two collaborative filtering algorithms *SVD* has a strong lead over *BLITR*. This contradicts what is reported in [2] where *BLITR* achieved a superior level of performance for the GroupLens dataset. We think that this might depend heavily on the amount of data as the GroupLens dataset used in [2] is a few orders of magnitude larger (about 10,000 vs. 10 million ratings). Lack of data can cause issues with the convergence of the Gibbs sampler.

*CB* aligns itself between the two collaborative filtering algorithms. Since this RS uses an explicitly stated user profile, it depends on the number of people entering their interests. On the LNMusic and on the LNMuseums this was only the case for 62.6% and 41.1% of the users.

Contrary to the other RS, *Temp* achieved better performance on the LNMusic than on the LNMuseums. This might be because on the LNMuseums more events take place at the same location, thus making location a less discriminative feature.

**Hybrid RS.** To increase the performance of RS a widely used technique (especially for event recommendations [1]) is to combine different approaches. We therefore ran a study with pairwise combinations of the above RS with the same evaluation method as before. There are many possible techniques for combining RS of which we chose the following: A weighted combination of the scores of two RS and a dynamic weighting of the two RS based on the richness of the user-profile available to each RS. This dynamic weighting was chosen for combinations of *CB* and the collaborative filtering algorithms (*SVD* and *BLITR* ). The weight of *CB* is determined by looking at how diverse the different dimensions of the profile vector are. The weight of the collaborative filtering algorithms is based on how many events the user has already selected.

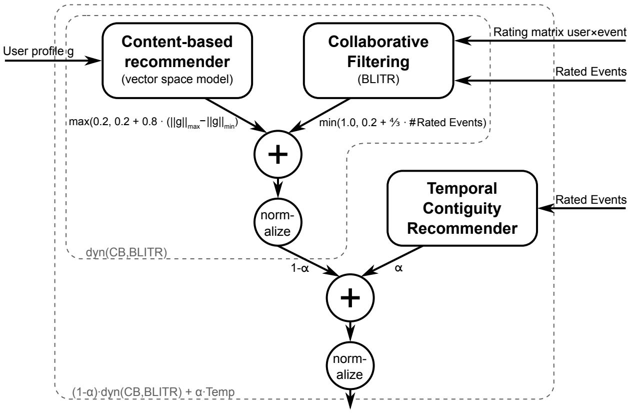

User profile g
Content-based recommender
(vector space model)
Collaborative Filtering
(BLITR)
Rating matrix user×event
Rated Events
max(0.2, 0.2 + 0.8 · (||g||max−||g||min)
min(1.0, 0.2 + ⅓ · #Rated Events)
+
norm-alize
Temporal Contiguity Recommender
Rated Events
dyn(CB,BLITR)
1-α
α
+
norm-alize
(1-α)·dyn(CB,BLITR) + α·Temp

**Figure 2: Structure of the hybrid recommender**

Table 1:bottom shows the most interesting combinations of RS and their performance. Additionally we added a combination of three RS (see Figure 2): *CB*, one of the collaborative filtering approaches and *Temp*. These results show that, in almost all cases, a combination of RS delivers better recall performance than those RS on their own.

Again, the difference between the two nights with respect to recall performance is obvious. However, when comparing different weights for a certain RS combination no large differences were found. The dynamic weighting increases the performance when combining *CB* with *SVD* but not when combining *CB* and *BLITR*. The dynamic weighting of *CB* and *BLITR* is still useful as it serves to alleviate the cold-start problem of new users having only a few ratings. Combining a RS with *Temp* has little effect, only a slight performance loss, if any, is noticeable. This is also true when comparing *CB* on its own with a combination of *CB*, one of the collaborative filtering algorithms and *Temp*. This suggests that while the TC of visited events is a major concern for visitors [5], our users do not seem to consider it when selecting events.

## 4. ONLINE EVALUATION

Helping users find events of interest to them is one part of our system, but our app goes beyond that and assists in the creation of routes throughout the evening. Thus, predicting whether or not a user will find an event interesting, as we did in Section 3, is only modelling a part of the problem. Arguably more important measures of system performance are whether or not recommended events appear in routes generated for the user and the properties of these routes. In this section we describe a classical A/B-Test with three different RS on the LNMuseums to evaluate these aspects: *SysA* - a classical RS[3], *SysB* - which considers Temporal Contiguity (TC) and *SysC* - which is strongly influenced by TC (see Table 2). Users were randomly assigned to one of the three test groups resulting in 321, 355 and 371 users being allocated to the 3 systems.

We evaluate three aspects of performance in the online study:

**User acceptance** of recommendations was determined by counting how many events users selected (from all tabs) that had been recommended by the system (and thus highlighted in the GUI). Users of *SysA* selected 35.6% (586 of 1,645) of highlighted events, whereas users of *SysB* and *SysC* selected only 33.9% (616 of 1818 events) and 32.1% (682 of 2123 events) respectively. These results show a significant difference[4] ($p$=0.024) between *SysA* and *SysC*, indicating that weighting temporal contiguity higher resulted in lower user acceptance in the online study. A possible explanation is that users of SysA had more trust in the system, somewhat contradicting the results of the offline study where all 3 systems had similar performance.

**Probability of appearing in route** is a means to investigate if and how recommendations were actually used. As users could initiate multiple route generations we considered the set $E_{route}$ of events which appeared in any of the generated routes. We then calculate the ratio of events selected by a user $E_{sel}$ that are in his/her $E_{route}$ set. The results of the three RS evaluated by this metric is depicted in the second column of Table 2.

Based on this metric we can see that the systems which include TC return better performance, comparing *SysA* and *SysB* this difference is significant ($p$=0.029). Overall, considering TC improves the chance of a selected event being included in the generated route by more than 3%. This finding seems to contradict the previous result where *SysA* had higher levels of acceptance.

[3] Due to technical reasons we used BLITR instead of SVD though the later performed better in the offline evaluation.

[4] Using a two-tailed test of population proportion

|  |  | All route generations |  |  | Route generations based on >50% recommended events |  |  |
|---|---|---|---|---|---|---|---|
| Recommender | $E_{Sel} \in E_{route}$ | TC [min] | #events in plan | event time | TC [min] | #events in plan | event time |
| SysA | 939 (57.1%) | 22.71 | 4.57 | 73.6% | 27.59 | 4.32 | 70.6% |
| SysB | **1104 (60.7%)** | 22.89 | 4.53 | 73.6% | **20.86** | **4.69** | **75.1%** |
| SysC | 1260 (59.3%) | **20.80** | **4.69** | **74.5%** | 22.03 | 4.53 | 73.8% |

**Table 2: Online evaluation: Chance of an event appearing in route and properties of generated routes**

**Examining Route properties** shows how routes vary with different RS, for this we investigate 2 sets of generated routes. The first set considers all routes generated containing at least 2 events ($n$=960). The second looks at all routes that are based on a set consisting of more than 50% recommended events, as these likely reflect users who trust the RS ($n$=419). We use the following route metrics which best represent what visitors reported as important criteria to ensure a positive experience for distributed events [5]:

- *Temporal Contiguity*: A lower TC of events in the route often results in less travel time between the events.

- *Number of events*: As visitors reported the desire to visit multiple events, we consider a higher number of events to better meet these expectations.

- *Ratio of event visiting time*: Visitors don't like to spend their time on buses when they could be visiting events instead. Thus a higher ratio between event visiting time and the total time on the evening should reflect an enhance visitor experience.

Results for the three studied RS are shown in Table 2. When comparing the RS based on all route generations *SysC* performs best. Comparing generated routes based on more than 50% recommended events the two RS which consider TC consistently perform better, with *SysB* performing best over all three metrics. Regarding TC, *SysB* significantly ($p \ll 0.01$) outperforms *SysA* (by nearly 25%), reducing the necessary travelling time between events from 28 min to 21 min. Users who were allocated to *SysB* also had on average an additional third of an event more in their generated routes, which is significant ($p$=0.038). When looking at the proportion of time between visiting events and travelling, use of *SysB* results in nearly 5% more time spent at events, which is also highly significant ($p \ll 0.01$). Comparing TC and event time ratio between *SysA* and *SysC* also reveals significant differences ($p$=0.011 and $p$=0.014), although no significant difference is observed for the number of events in the plan.

Based on these results, it seems that users have less faith in RS which account for TC. Nevertheless, users who accept recommendations from such systems are rewarded in that they can select fewer events, as their probability to be included in the generated routes increases. Furthermore, they will be given compact routes that contain more events, decreased travelling times and increased time spent at events.

# 5. CONCLUSION & FUTURE WORK

In this work we have compared a number of different recommender algorithms for use within apps designed to assist users during distributed leisure events. We first ran an offline comparison based on previously acquired event ratings and tested a number of combinations of these recommenders. We were particularly interested in the impact of temporal contiguity on the results, since this was determined to be an important factor for visitors to such events. The results of the offline study showed that combining recommenders produces good results, however that the level of TC appears to have little impact. Out of these we selected 3 promising approaches for an online study, two of which include the TC-based RS as a component, but with different weights. In contrast to the offline study, where all 3 systems performed similarly, in the online study the user acceptance decreased when TC was given a higher weight in recommendations. This also contradicts the usefulness of the recommended items: In the online study the greatest probability of an event being included in a route was when it was recommended by a RS accounting for TC. Considering the generated routes, both RS factoring in TC resulted in better routes as measured by the three introduced metrics: TC, number of events in the generated route and time spent at events. The third RS performed best when considering all generated routes, although not significantly. When concentrating on route generations with more than 50% recommended events, the second RS performed best, with a significant lead compared to the RS not considering TC.

In conclusion, it is indeed possible to influence users in a way that helps them to achieve a goal they state as being important (see [5]) but perhaps don't consider yet when rating events: Compactness of the generated route. This might be caused by the different objectives users want to achieve when visiting a distributed event: Events should be interesting, diverse and also make a tight plan feasible. The last objective might be too complex to consider when browsing the available events. This is also expressed in some of the comments about our app, where users asked for a small map to be shown along with the description of each event. We interpret this as a wish for assistance to help them finding contiguous events. However, we doubt such a map would be helpful in this case as spatial contiguity and TC are two very different concepts for people visiting distributed events (see [5]).

In future work we want to further study the optimal balance between an interest-based RS and an approach considering TC. The metrics used in this paper could be further improved by considering not just the routes generated, but also the actual events visited.

# 6. REFERENCES

[1] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of rec. algo. for an event rec. system. In *RecSys*, 2011.

[2] M. Harvey, I. Ruthven, F. Crestani, and M. Carman. Bayesian latent var. models for collab. item rating prediction. *CIKM*, 2011.

[3] D. H. Lee. PITTCULT: trust-based cultural event recommender. In *Proceedings of RecSys*, 2008.

[4] A. Paterek. Improving regularized singular value decomposition for collaborative filtering. In *KDDCup.07*, 2007.

[5] R. Schaller, M. Harvey, and D. Elsweiler. Entertainment on the go: Finding things to do and see while visiting distributed events. In *Proceedings of IIiX*, 2012.

[6] R. Schaller, M. Harvey, and D. Elsweiler. Out and about on museums night: Investigating mobile search behaviour for leisure events. In *Proc. of Searching4Fun Wksp, ECIR*, 2012.

[7] H. Steck. Item popularity and recommendation accuracy. In *5th ACM Recommender Systems*, pages 125–132. ACM, 2011.