Journal of Universal Computer Science, vol. 22, no. 8 (2016), 1148-1174
submitted: 27/4/16, accepted: 29/7/16, appeared: 1/8/16 © J.UCS

Boosting Point-of-Interest Recommendation with Multigranular Time Representations

Gonzalo Rojas
(University of Concepción, Concepción, Chile
gonzalorojas@udec.cl)

Diego Seco
(University of Concepción, Concepción, Chile
dseco@udec.cl)

Francisco Serrano
(University of Concepción, Concepción, Chile
fserrano@udec.cl)

Abstract: Technologies of recommender systems are being increasingly adopted by Location Based Social Networks (LBSNs) with the purpose of recommending Points-of-Interest (POIs) to their users, and different contextual characteristics have been incorporated to enhance this process. Among these characteristics, the time at which users express their preferences (typically, by checking-in to different POIs) and ask for recommendations, is frequently referred as a first-order feature in this process. However, even when its influence on improving the accuracy of recommendations has been empirically demonstrated, time is still mainly considered through a monogranular representation (one-hour or one-day blocks). In this article, we introduce a POI recommendation approach based on a multigranular characterization of time, composed of hour, day-of-the-week, and month. Based on this concept, we propose two representations of user check-ins: one that directly extends a monogranular proposal of time for POI recommendations, and other based on a statistical representation of check-in distributions in time. For both representations, corresponding algorithms to compute user similarity and preference prediction are introduced. The experimental evaluation shows promising results in terms of accuracy and scalability.

Key Words: recommender systems, point-of-interest, time-aware recommendation, location-based social network

Category: H3.3, H4, L2.2

1 Introduction

The evolution of mobile technologies and communication networks has aroused interest in location-based services [Schiller and Voisard 2004], such as Location-Based Social Networks (LBSNs) [Kefalas et al. 2013]. These on-line social networks, mainly used from mobile devices, allow the users to share location-tagged content with their contacts. Despite their advantages, they also entail some computational challenges due to the huge amount of information they generate. Without proper processing, this overloading of information would prevent users from obtaining timely information and interacting appropriately with others. Recommender systems have proved useful in several domains to mitigate this problem by recommending the *items* that should be more interesting to users (based on their previous history), thus limiting the amount of delivered information. When applied in LBSNs, recommender systems have been shown useful to recommend Points-of-Interest (POIs) [Baltrunas et al. 2011], which is the name that receives the generic concept of item in this particular domain, augmented with descriptors of its physical location.

However, despite recommender systems allow users to effectively interact with location-based services, the quality of the recommendations must be improved in order to satisfy the user expectations. In [Kefalas et al. 2013], new perspectives aimed at increasing the effectiveness of these systems are addressed, and time stands out as a factor that may improve the effectiveness of the recommendation of POIs. Indeed, people tend to perform tasks or activities depending on the specific time. For example, people often have lunch between 1 pm and 3 pm. Therefore, if a user asks for a recommendation in such temporal range, a place to eat will have more chances of being a good recommendation.

The work by [Yuan et al. 2013a] was the first to incorporate the time variable in the recommendation of POIs. They use a collaborative filtering that computes the similarity of the users in terms of their history of visits, expressed as a set of *check-ins*. Thus, a user is more similar to those users who visited the same POIs at the same time. However, their work represents the time variable through one-hour-blocks, pending the use of other representations of time. This restriction may rise some poor recommendations. For example, there are some restaurants that close on specific days, farmer's markets that just open one or two days each week, or pubs that just open on weekends. Something similar happens for the seasons (there are some activities that are performed just on a specific season).

This article proposes to improve the accuracy of POI recommendation algorithms by considering multiple granules of time. Our proposal extends the monogranular notion of time introduced in [Yuan et al. 2013a], by considering time as multigranular variable composed of three granules: month, day-of-the-week, and hour. We adapt the definition and execution strategy of the proposed algorithms of similarity and prediction in order to support this multigranularity, and introduced some alternatives to perform these computations more efficiently. Our experimental evaluation shows the advantages of adopting a multigranular specification of time in the accuracy of POI recommendations, in comparison with the monogranular approach. Therefore, this work emphasizes the importance of preserving the multigranular nature of time in POI recommendations, by isolating the time variable due to its major influence on the accuracy of recommendations [Yuan et al. 2013a]. In this way, we aim at providing researchers and practitioners with a more complete specification of this variable prior to combining it with other contextual characteristics (location, social influence, etc.) or to analysing more complex temporal factors, such as the subjective perception of time.

The rest of the paper is organized as follows: Section 2 describes the state of the art in POI recommendation and the role of time in this process; Section 3 describes the application of the proposed multigranular representation of time in two alternative approaches to compute similarity and prediction values: (a) an adaptation of the algorithms proposed by [Yuan et al. 2013a] considering the three time granules; and (b) an approach based on a statistical characterization of POIs; Section 4 describes and discusses the experiments conducted to compare our proposal with the monogranular approach, through different combinations of the proposed similarity and prediction algorithms. Finally, Section 5 presents some conclusions and suggests some future work.

2 State of the art

Recommender systems [Resnick and Varian 1997, Ricci et al. 2011] have increasingly extended the set of characteristics on which recommendations are based. Context-aware recommender systems [Adomavicius and Tuzhilin 2011] compute recommendations according not only to the implicit or explicit user preferences for different items, but also considering contextual characteristics such as weather conditions, user emotions, location, and time. Besides, recommendation of geographic places (POIs) has taken advantage of these characteristics in better characterizing the actual user needs and thus providing more accurate recommendations. In this section, we present a literature review on the adoption of the temporal dimension in context-aware recommender systems, a more detailed description of the use of time in POI recommendations, and a review of the work that our proposal took as a baseline.

2.1 Time in context-aware recommender systems

Different proposals have faced the challenge of providing accurate context-aware recommendations in a scalable way. In [Rendle et al. 2011], *Factorization Machines* [Rendle 2010] are used to model a wide variety of contextual information and to provide rating predictions. This proposal is based on a general specification of contextual characteristics and aims at serving as a baseline for specialized models, but leaving the special treatment of the time variable open. In [Liang et al. 2012], authors explore the temporal dimension of microblogs (e.g., tweets from *Twitter*) as one of the features used to recommend topics. A measure of recency of contents, mixed with other descriptors of user behaviour, reaffirms the time-awareness as an improving factor of recommendation accuracy.

In [Koren 2009], authors introduce a factor model that describes the temporal dynamics of preferences in collaborative filtering recommender systems, improving the quality of predictions obtained with other factor models. Similarly, Xiong et al. [Xiong et al. 2010] introduce a method to describe the global time-evolving dynamics of all relevant users and items. Corresponding algorithms of both approaches describes the time variable through a day-of-the-week granule.

2.2 POI recommendation based on temporal information

As stated in [Ye et al. 2011], facilitating POI recommendations in LBSNs is a promising and interesting research problem, and also a feature well received by users of these systems [Zheng 2011]. These context-aware applications define a new social structure made up of individuals connected from their locations in the physical world. These connections can be either obtained from explicit interactions, such as checking-in or tagging media content at similar places; or being inferred from knowledge associated to preferred POIs, such as common interests, behavior, and activities. The rapid growing of LBSNs such as Foursquare [see [1]] or Facebook Places [see [2]] makes large volumes of user check-in data available to developers, thus raising the importance of the check-in as a first-order descriptor of user preferences (e.g., [Hsieh and Li 2014] approaches the insufficiency of check-in data for recommendation purposes). From these data, recommendations of unvisited POIs can be obtained by adapting traditional recommendation approaches, such as collaborative filtering [Zheng et al. 2010], or hybrid recommendations [Woerndl et al. 2007], to the specific requirements of geographic recommendations, by incorporating new content descriptors such as the geographic location [Horozov et al. 2006].

Time plays an important role in the POI recommendation process. Users are likely to prefer visiting a certain place at a specific time or slot (e.g., cafés at noon, pubs at night, etc.), so a good recommendation of a POI could be highly improved by adding a suggested slot that fits a user's preference or lifestyle. In this context, time increases the notion of collaborative filtering similarity: two users are similar not only because they visit the same places, but also because they prefer to do it at a similar time. Analogously, two places can be considered similar when they are visited by the same users at the same time slots.

In spite of this importance and previous consideration in context-aware recommender systems, temporal characteristics of user preferences have only recently being considered as a first-order actor in the recommendation process of POIs [Kefalas et al. 2013]. Works like [Hervás and Bravo 2011] rise the importance of temporal aspects in context-aware applications, but its specific role in the recommendation process is not clearly described. Some proposals introduced

[1] http://www.foursquare.com/about
[2] https://www.facebook.com/places/

statistical models to obtain the probability of a user checking-in to a location at a given time [Cho et al. 2011, Gao et al. 2013a], by combining periodic behavior and social relationships. In [Gao et al. 2013b], authors introduce a recommendation framework based on two temporal characteristics of check-ins: (1) non-uniformness: a user exhibits distinct check-in preferences at different hours of the day; and (2) consecutiveness: a user tends to have more similar check-in preferences in consecutive hours than in non-consecutive hours. In [Zhao et al. 2015], authors introduce a tensor factorization-based methodology in which POIs are recommended based on the time-varying behavior of users, described in terms of semantic categories of the places they checked-in. In this proposal, a day is divided into five intervals of distinct and fixed size, adopting this interval as the only time granule used to compute recommendations.

Some of these works, along with the proposal we took as a baseline (which we describe in the next section) are featured in the survey of POI recommendations presented in [Yu and Chen 2015], which summarizes the research on the area by classifying works according to the contextual characteristics incorporated into the recommendation process (geographic, social, and temporal). However, no consideration of time granularity is reported.

2.3 Description of our baseline

As an explicit attempt to adopt the temporal dimension in POI recommendations, the work introduced in [Yuan et al. 2013a] analyzes and describes the time variable of check-ins, and introduces the term *time-aware POI recommendation* to refer the challenge of recommending POIs for a given user at a specified time in a day. This proposal splits a day in multiple *hour slots* in which users check-in to different places. With this variable, the traditional user/item matrix to represent preferences is replaced by a user/time/POI (UTP) cube. Each $c_{u,t,l}$ element of this cube represents the user u check-in to a POI l during a time slot t. If the user u visited the POI l at time t, then $c_{u,t,l}$ =1, otherwise $c_{u,t,l} = 0$.

Recommendations are obtained by following a user-based collaborative filtering approach. However, in order to smooth the strict requisite of checking-in at the exact time-slot for two users to be considered similar, a previous similarity computation between pairs of hour slots is performed from the check-ins made by different users during those slots. Two slots are considered similar if many users have similar check-in behaviours (i.e., visit the same places) at corresponding hours. Calculation is performed by applying cosine similarity to check-in vectors of each user and obtaining the average of the results. Equation 1 shows how similarity between two time slots (t, t') is computed. The summand of the main summation corresponds to the *cosine similarity* between t and t' vectors of each user u, while the time similarity between t and t' (denoted by $\rho_{t,t'}$) is the average similarity for all users.

$$\rho_{t,t'} = \frac{1}{|U|} \sum_{u \, \epsilon \, U} \frac{\sum_l c_{u,t,l} \cdot c_{u,t',l}}{\sqrt{\sum_l c_{u,t,l}^2} \sqrt{\sum_l c_{u,t',l}^2}} \tag{1}$$

In order to calculate user similarities, binary values of check-in UTP cubes are updated by considering the computed time similarities. This update aims at smoothing the binary distinction of check-ins, not completely discarding the similarity between users that checked-in to the same place but during different time slots. These new values ($\widetilde{c}_{u,t,l}$) are computed with Equation 2.

$$\widetilde{c}_{u,t,l} = \sum_{t'=1}^{T} \frac{\rho_{t,t'}}{\sum_{t''=1}^{T} \rho_{t,t''}} c_{u,t',l} \tag{2}$$

In this way, similarity between users u and v (denoted by $w_{u,v}^{(te)}$, or *time enhanced* user similarity) is obtained by applying Equation 3, which is a traditional cosine similarity: if two users u and v have visited the same POIs during the same or similar slots, then they will be considered similar, because they will have a high similarity value.

$$w_{u,v}^{(te)} = \frac{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{u,t,l} \cdot \widetilde{c}_{v,t,l}}{\sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{u,t,l}^2} \sqrt{\sum_{t=1}^{T} \sum_{l=1}^{L} \widetilde{c}_{v,t,l}^2}} \tag{3}$$

Finally, and also similarly to a collaborative filtering approach, a prediction formula is applied to estimate the preference of a user for a POI at a certain time. This predicted value is denoted by $\widehat{c}_{u,t,l}^{(te)}$ and calculated according to Equation 4.

$$\widehat{c}_{u,t,l}^{(te)} = \frac{\sum_v w_{u,v}^{(te)} \sum_{t'} \widetilde{c}_{v,t',l} \cdot \rho_{t,t'}}{\sum_v w_{u,v}^{(te)}} \tag{4}$$

We claim that the adopted one-hour-blocks representation used in this pioneer work does not meet all the time-aware requirements of this process. As its authors admit, check-in behavior is highly influenced by the day-of-the-week or even the month of the year, so they propose as a future work to explore other temporal granules. However, their complementary work [Yuan et al. 2013b, Yuan et al. 2015] only adds a description to classify days into weekdays or weekend, or briefly explores the recommendation quality for different lengths of time slots (from 1 to 24 hours, [Yuan et al. 2014]).

In [Bannur and Alonso 2014], authors analyze check-in data from Facebook, and detect different and highly meaningful check-in patterns at different time granules, but the challenge of exploiting this information for POI recommendations remains open. We claim that considering check-in data at different time granules can highly influence the way that users perceive the recommendations provided. Our proposal faces this challenge by extending the monogranular temporal representation of [Yuan et al. 2013a] with a multigranular representation of time for POI recommendation, and presenting an experimental evaluation that compares both proposals.

3 Multigranular time representation for POI recommendation

This section introduces our proposals for POI recommendations based on a multigranular representation of the time variable, by considering three granules of time: hour, day-of-the-week and month. Our first proposal is a direct extension of the monogranular time specification of [Yuan et al. 2013a], by adapting its representation and algorithms to the extended specification. As we show in Section 4, this alternative improves the results of the state of the art in terms of recommendation accuracy. The second alternative proposes a *statistical* description of POI's check-ins from their three time granules. This variant reduces the accuracy of recommendations, but still competes with the state of the art and requires less memory and processing time.

3.1 Multigranular Extension

From now on, we refer to the monogranular treatment of the temporal variable of the proposal presented in [Yuan et al. 2013a] as the *Baseline*. In this section, we extend this *Baseline* to consider a multigranular representation of the time variable. We refer to this approach as $MultiGran$. In the experimental evaluation we consider three different granules: *Hour*, *Day-of-the-week* and *Month*. These granules were chosen because they can be efficiently obtained from the check-ins *timestamps*, are easily generalizable, and satisfy formal specifications of time, such as the one introduced in [Bettini et al. 1998]. Nevertheless, our proposal is generic and can be easily adapted to consider other time granules.

In order to make our proposal comparable with the *Baseline*, our recommender system also adopts a user-based collaborative filtering approach (described in Section 2.3), with its two main computations: (a) *user similarity computation*, in order to obtain those users that are similar to the user u that requests a recommendation, and (b) *prediction computation*, so as to sort the POIs visited by those similar users (and not visited by u) by an estimated preference score, obtaining a ranking of recommendations. Below, we explain our approach for these two steps. However, we need to introduce some notation first.

We extend the notation introduced by the *Baseline*. Recall from Section 2 that $c_{u,t,l}$ represents the binary entries of a user/time/POI (UTP) cube of check-ins, whose dimensions are users u, one hour slots t and POIs l. In our proposal, we decompose the timestamp of check-ins into the three chosen granules, obtaining three UTP cubes with entries denoted by $c^{(h)}_{u,t,l}$, $c^{(d)}_{u,t,l}$ and $c^{(m)}_{u,t,l}$, which represent the check-in time in hours, days-of-the-week and months, respectively.

The *Baseline* performs a time similarity computation as a basis for the further computation of user similarity, by calculating the cosine similarity between check-in vectors from its UTP cube (see Equation 1). In our proposal, we extend this approach to the three granules, by applying the same formula to the vectors from the three UTP cubes (i.e., obtaining the average cosine similarity between pairs of hour, day-of-the-week and month slots for all users). As a result, we obtain three similarity matrices, whose entries are denoted by $\rho^{(h)}_{t,t'}$, $\rho^{(d)}_{t,t'}$, $\rho^{(m)}_{t,t'}$.

In order to gain insight into the time similarity at different granules, let us consider the example of days-of-the-week, by assuming the following entries of a day-of-the-week similarity matrix: $\rho^{(d)}_{t,t'}[Sat][Sat] = 1$, $\rho^{(d)}_{t,t'}[Sat][Sun] = 0.5$, $\rho^{(d)}_{t,t'}[Sat][Fri] = 0.3$, and $\rho^{(d)}_{t,t'}[Sat][dow] = 0.1\,\forall dow \notin \{Fri, Sat, Sun\}$. From these values, we can conclude that Saturday and Sunday are more similar than Friday and Saturday (even when both pairs are consecutive days), and even more than Saturday and the rest of weekdays, which can be explained because people tend to visit places in weekdays other than those in weekend. Analogously, for month granule, months of summer vacation (July and August, in the North Hemisphere) are likely to be more similar than any of them and a working month.

3.1.1 User similarity computation

Intuitively, two users are similar if they have visited the same POIs at the same time. In our multigranular representation of time, binary check-ins let two users be similar only if they visited the same POIs at the same hour, day-of-the-week, and month. In order to relax this strict requirement, we extended the smoothing process of binary check-ins introduced by the *Baseline* (see Equation 2), by applying it to the three UTP cubes. In this way, two users are similar if they visited the same POIs at similar times (either hours, days of the week, and/or months). The notation of the smoothed check-ins for each granule is: $\widetilde{c}^{(h)}_{u,t,l}$, $\widetilde{c}^{(d)}_{u,t,l}$, and $\widetilde{c}^{(m)}_{u,t,l}$, respectively. In this way, we can adapt the *Baseline* formula of user similarity (see Equation 3) to calculate separate similarity values for each time granule. As an example, Equation 5 describes the user similarity calculation between two users in the day-of-the-week time granule.

$$w^{(d)}_{u,v} = \frac{\sum_{t=1}^{T}\sum_{l=1}^{L}\widetilde{c}^{(d)}_{u,t,l}\cdot\widetilde{c}^{(d)}_{v,t,l}}{\sqrt{\sum_{t=1}^{T}\sum_{l=1}^{L}\widetilde{c}^{(d)2}_{u,t,l}}\sqrt{\sum_{t=1}^{T}\sum_{l=1}^{L}\widetilde{c}^{(d)2}_{v,t,l}}} \tag{5}$$

Once values of user similarities for each granule are obtained, they are combined in order to obtain a unique time-based, user similarity value ($wmg_{u,v}$) for each (u, v) pair of users. Equation 6 shows an abstract specification of this combination, as a function f of the three similarities considered in our proposal:

$$wmg_{u,v} = f(w^{(h)}_{u,v},\ w^{(d)}_{u,v},\ w^{(m)}_{u,v}) \tag{6}$$

In the experimental evaluation, we compare the performance of different combinations of similarity functions. For example, if we define f as the product of the similarities calculated for each time granule, Equation 6 is implemented by Equation 7:

$$wmg_{u,v} = w^{(h)}_{u,v} \cdot w^{(d)}_{u,v} \cdot w^{(m)}_{u,v} \tag{7}$$

With this combination, for each time t and POI l we require both users u and v to visit the place at a similar month, day-of-the-week and hour, in order to obtain a high value of similarity. In other words, if both users visited the same POI in the same month (or at least in the same season) and during the weekend, but u visited the place during midday hours and v did it at night, the similarity value would be lower. Suppose users u_1 and u_2 usually have dinner in the same restaurants on weekends. Then their wmg_{u_1,u_2} similarity value would be high because the three terms of this formula are high. Now, consider another user u_3 that usually have lunch (and not dinner) in the same restaurants but on weekdays. Then, wmg_{u_1,u_3} would be lower because the first and second term are lower. Finally, wmg_{u_1,u_4} would be zero if user u_4 has not visited such restaurants. In Section 4 we explore some less restrictive variants for f that improve the accuracy of the recommendations.

3.1.2 Ranking of recommendations

In order to predict which are the POIs that the current user are likely to visit at a certain time, the system sorts the POIs visited by his/her similar users (obtained from the previous step), and sort them in a ranking. As a three-granule extension of the prediction formula from the *Baseline* (see Equation 4), Equation 8 shows how to calculate a preference value for a POI l, for a given user u at time t.

$$\widehat{cmg}_{u,t,l} = \frac{\sum_v wmg_{u,v} \sum_{t'} (\widetilde{c}^{(h)}_{v,t',l}\ \rho^{(h)}_{t,t'}) \cdot (\widetilde{c}^{(d)}_{v,t',l}\ \rho^{(d)}_{t,t'}) \cdot (\widetilde{c}^{(m)}_{v,t',l}\ \rho^{(m)}_{t,t'})}{\sum_v wmg_{u,v}} \tag{8}$$

Intuitively, a POI l will obtain a high predicted preference value if many users that are similar to u visited it at a time similar to t. The preference value is computed by comparing the check-ins of similar users v for l at times t'. By multiplying check-in values with the corresponding ρ time similarity, the formula gives more relevance to check-ins at those t' that are more similar to t in each granule. As resulting values for each granule are multiplied, we are prioritizing POIs visited at a time similar to the ones the user is requesting the recommendation, by considering all granules. In preliminary experiments, the multiplication of these values provided better recommendations than other alternatives (minimum, maximum, and bounded sum). Once predictions have been made for all candidate POIs, a list sorted in descending order of this value is delivered to the current user.

Continuing with the example, suppose u_1 is requesting a recommendation on Friday around dinner time. The POIs that would obtain a larger $\widehat{cmg}$ would be those visited by users similar to u_1 (i.e., POIs visited by u_2, and to a lesser extent by u_3). From these POIs, those visited at similar times (i.e., weekend and night) would be positioned first in the ranking because of the influence in the formula of the $\rho^{(h)}$ and $\rho^{(d)}$ matrices.

3.2 Statistical Representation

In this section, we propose a simpler statistical representation for each time granule, which replaces the UTP cubes from *MultiGran* approach. This makes the computation of recommendations more efficient in terms of memory consumption and processing time. We refer to this approach as *Stat*.

In this approach, each POI at each time granule is represented by a set of statistical values computed from the check-ins database. We first describe a simple approach in which we assume that these statistical values perfectly describe the user behavior for each POI, and then we show its potential problems and how to extend the representation.

These statistical values can be graphically represented in Box-and-Whisker charts, and correspond to a minimum value, a maximum value, and the limits of the three quartiles, Q_1, Q_2 and Q_3 of this representation. Figure 1 shows two examples obtained from the dataset used in our experimental evaluation. Recall that the box extends from Q_1 to Q_3 and contains 50% of the check-ins, and Q_2 represents the median. Thus, these five values describe the concentration of check-ins along time (where time can be any of the three time granules). A simple preprocessing of the check-in data, by grouping them by POIs and calculating these statistical values for each granule, is required to use this information in a recommender system.

This simplification presents a potential problem in POIs that are not well described by a single block at a specific time granule. Consider, for instance, a restaurant that receives many visits for lunch (around 2 pm) and dinner (around 9 pm), but just few in the afternoon (it may be even closed). If we compute the statistical values described above, the box would cover the range 1 pm to 9 pm and the median would be around 4 pm, which do not describe the actual check-in activity with accuracy. To overcome this problem, the number and size of *temporal blocks* that properly characterize the check-ins behavior must be computed first. This is a well-known problem in data mining as it can be thought as a special case of data clustering in one dimension, usually known as *data segmentation*. In this problem, an array of values must be partitioned into classes based on natural groups in the data distribution. In our experiments, as a

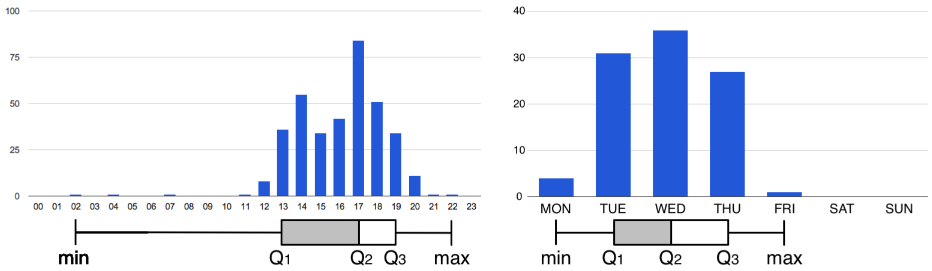

Figure 1: Equivalence between histograms and Box-and-Whisker charts. The histograms represent the number of check-ins per hour for the POI 31928 and the number of check-ins per day-of-the-week for the POI 1261444 of the GoWalla dataset [Cho et al. 2011], respectively.

proof of concept, we implemented an ad-hoc segmentation algorithm that either splits the distribution of the check-ins of each POI into two blocks (if it finds a valley in the distribution) or keeps it as a single block, leaving the consideration of more complex classification methods like *Jenks natural breaks* [Jenks 1967] or *Head/Tail breaks* [Jiang 2013] for future work. These segmentation algorithms aim to determine the best arrangement of one-dimensional values into different classes. We conjecture that the use of this segmentation techniques may improve the accuracy of the recommendations.

Summarizing, in the *Stat* approach each POI at each time granule is characterized by one or more *statistical blocks*, which are defined by five statistical values (i.e., min, max, Q_1, Q_2 and Q_3). All these values are computed during a preprocessing step. Next, we describe how these values can be used in a recommender system to compute the similarity between users and a ranking of recommendations.

3.2.1 User similarity computation

The five statistical values described above are used to divide each *temporal block* at each time granule into four *zones*, as shown in Figure 2. With this representation, we say that two users are similar if they have visited the same places in the same zone of the temporal block for the three time granules. If two users have visited the same place but not exactly in the same zone of the temporal block, they are still similar (although to a lesser extent). In order to quantify a measure of similarity for users visiting the same place, we propose the *Similarity matrix* shown in Figure 2 (left). This matrix shows all the possible combinations of two users visiting the same POI. In the matrix, similarity is obtained from the distance in zones between two users. However, we separated the case of contiguous zones by considering whether both users are in the box (0.8) or one of them is in a whisker (0.6). With this five possibilities, we have five different values we distributed uniformly in [0..1]. This uniform distribution provided the best experimental results in comparison with other skewer distributions. As in $MultiGran$ approach, the similarity between two users is calculated as the sum of the similarities for all the POIs that they have both visited.

	Zone1	Zone2	Zone3	Zone4
Zone 1	1.0	0.6	0.4	0.2
Zone 2	0.6	1.0	0.8	0.4
Zone 3	0.4	0.8	1.0	0.6
Zone 4	0.2	0.4	0.6	1.0

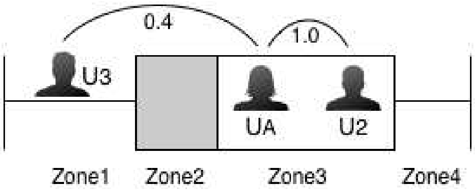

Figure 2: Similarities between 4 zones of a temporal Box-and-Whisker block.

To understand this, consider the example described in Figure 2 (right). In this scenario, the current user U_A and two other candidate users U_2 and U_3 have visited the same POI (characterized by the Box-and-Whisker chart in the figure). As they have visited the same place they are already similar, but check-ins of U_A and U_2 are located in the same zone of the temporal block, while U_3's check-in is located in a different zone, so U_A is more similar to U_2 than to U_3.

If there are some POIs represented by several temporal blocks in one or more granules, i.e., if their temporal check-in distribution is better represented by two or more box-and-whisker statistical blocks, we propose a similarity value of 0.1 between two users who have visited the same POI at different blocks. This value is lower than all the values in the Similarity matrix, but still not zero (because by definition, two users who have visited the same place should be similar). We remark that these similarity values are those that offered the best results in our experiments. However, for different applications/datasets it may be necessary to adjust them.

$$wst_{u,v} = \sum\nolimits_l \beta_{z_{u,l},z_{v,l}} \tag{9}$$

Formally, Equation 9 defines this similarity (denoted by $wst_{u,v}$) between two users, u and v. Here, the similarity is computed as the sum for all the POIs l that both u and v have visited. For each POI l, $z_{u,l}$ and $z_{v,l}$ correspond to the zone in a temporal block in which users u and v checked-in to l, respectively. Then, $\beta_{z_{u,l},z_{v,l}}$ can be obtained from the Similarity matrix shown in Figure 2. If $z_{u,l}$ and $z_{v,l}$ are in different temporal blocks $\beta_{z_{u,l},z_{v,l}} = 0.1$. Note that this is computed independently for each temporal granule. A global similarity value between two users can be obtained by combining the similarities for each time granule as in our $MultiGran$ approach.

3.2.2 Ranking of recommendations

Like in the *MultiGran* approach, the recommender system must estimate a predicted value of preference for the different POIs that are candidates to be recommended to the current user at a given time. Those candidates are obtained from POIs that have been visited by similar users, but not visited yet by the current user. Using the statistical representation of time, a prediction value ($\widehat{cst}_{u,t,l}$) is calculated using Equation 10.

$$\widehat{cst}_{u,t,l} = \sum\nolimits_{v} wst_{u,v} \cdot \sigma_{t_h} \cdot \sigma_{t_w} \cdot \sigma_{t_m} \tag{10}$$

In this equation, $\sigma_{t_h}, \sigma_{t_w}$ and σ_{t_m} are weights that indicate how good recommendation is the POI l for the current hour, day-of-the-week and month. The value will be higher if the current time is close to the median of the statistical characterization of l in the corresponding granule. Specifically, the weight takes the value 1.0 if the current time coincides with the median, 0.8 if it falls in the box, and 0.4 if it falls in any of the whiskers. Recall that the box represents the 50% of the data and each whisker the 25%, so weights for corresponding coincidences were assigned with the ratio 2:1. In our experiments, these values provided the best experimental results in comparison with other values with the same ratio, but they can be modified according to the characteristics of other datasets.

4 Experimental Evaluation

We evaluate our proposal by comparing it with the *Baseline* (temporal variable treatment from [Yuan et al. 2013a]). Aimed at obtaining evidence of its influence on the quality of POI recommendations, we compare the results obtained by analyzing the hour, day-of-the-week, and month when users checked-in, with those obtained by considering only the hour.

The evaluation consisted on two sets of experiments. The first set (see Section 4.2) compares five variants of our *Multigran* approach with the *Baseline*, by measuring the accuracy of the recommendations obtained from both approaches. The second set (see Section 4.3) compares our *Stat* approach with the *Baseline* and the variants of *Multigran* best evaluated by the first set.

In these experiments, the *Baseline* computes the user similarity and prediction based on the hour-based representation of time. Both sets of experiments are presented with measures that we consider relevant to evaluate the quality of recommendations, such as the number of times the first recommended POI was actually visited by the user at the time of recommendation ($TOP1$). In order to gain more insight about the algorithms, in Section 4.4 we present a summary of the results using more standard evaluation metrics, which are variants of the classical *precision* and *recall* metrics.

4.1 Description and preparation of datasets

In our experiments, we use the *GoWalla* dataset available at the Stanford Network Analysis Project [Cho et al. 2011]. This dataset contains 6,442,890 check-ins over the period of February 2009 to October 2010 in the *GoWalla* location based social network, which closed in 2012. Note that this is a superset of the dataset used by [Yuan et al. 2013a], which restricted their dataset to those check-ins made within California and Nevada. As the *Baseline*, we preprocessed the dataset to remove users who have checked-in less than 5 times, and then POIs with less than 5 check-ins. Figure 3 shows the distribution of the check-ins in the dataset at the three time granules we considered in our experiments.

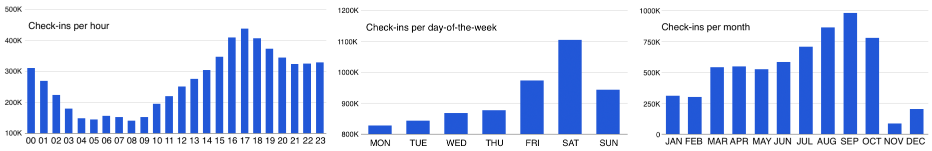

Figure 3: Distribution of check-ins in GoWalla dataset at different time granules.

It is interesting to notice that the distribution of check-ins tend to decrease between 4 am and 9 am (see Figure 3, left), when people usually sleep. In contrast, we can see more activity in the evening, when people usually end their work day. Usually, LBSNs like GoWalla are oriented to share information of places visited in social activities. This idea is also supported by the distribution of check-ins during the days-of-the-week (Figure 3, center), where the number of check-ins increases significantly on weekends. Finally, Figure 3 (right) shows an increase of check-ins between July and October, which may occur because it is summer season in the Northern hemisphere and most check-ins correspond to that area. There is also a significant decrease in November and part of December. This is probably because the dataset covers until October 2010. Thus, for November, December (and also January) the dataset just contains the check-ins of one year, whereas for the other months it contains the check-ins of two years.

As it is usual in the evaluation of recommender systems [Campos et al. 2013], the GoWalla dataset was divided in a training and a testing file. By adopting a common practice in Data Mining [Witten and Frank 2005], we assigned two-thirds of the collected check-ins to the *training file*, and the remaining third to the *test file*. The training file was used to compute user similarity, while the test file was used to simulate the actual visits of users and compare them with the provided recommendations. Check-ins were randomly distributed between both files. We also evaluated our proposal by using a time-splitting for the dataset division (all check-ins from the test file are more recent than any training check-in). Although the obtained results were slightly different (all the algorithms performed a bit worse), the trends between random and time splitting alternatives were the same. Hence, time-splitting results were omitted for briefness purposes.

The experiments were run on a Ubuntu Server (Kernel 3.13.0-35) with 32GB RAM and an Intel Core processor i7-3820@3.60GHz. The algorithms were compiled with Java version 1.6.0_32 and executed with 25GB of RAM limit. We implemented the code of the two approaches described in this document. The source code of the *Baseline* was provided by its authors, and adapted to work with the *GoWalla* dataset.

4.2 Evaluation of *MultiGran* approach

The evaluation of our *MultiGran* approach consisted of the following steps:

1. **Computation of time similarity:** time similarity matrices ($\rho^{(h)}$, $\rho^{(d)}$, $\rho^{(m)}$) are generated from the training file, containing similarity values for each pair of time slots. Resulting matrices indicate, for example, how similar is 9:00 am to 3:00 pm, Monday to Thursday, or June to December, respectively, based on the user check-ins in those slots.

2. **Generation of smoothed UTP cubes:** the UTP cube obtained directly from the training file is decomposed into the three UTP cubes corresponding to each time granule. By using the time similarity computed in the previous step, these cubes are smoothed by applying Equation 2, in order to obtain less restrictive user similarity values.

3. **Generation of POIs recommendations:** to recommend POIs to a particular user at a specific time, the following steps are followed:
   (a) **Extraction of similar users:** a user similarity computation is made by executing our multigranular user similarity function (Equation 6). Five alternatives of similarity f function (explained below) were evaluated. From computed user similarity values, a set of the most similar users was obtained.
   (b) **Generation of the POIs recommendations ranking:** from a set of candidate POIs to which similar users checked-in, a predictive preference value is obtained by our multigranular prediction formula (Equation 8), by which the list of candidate POIs are sorted in descending order.

The combination of the five user similarity alternatives considered in the step 3a with the mutigranular prediction formula of the step 3b, results in corresponding five variants of the recommendation process. These variants are named with the *MultiGran_* prefix, representing the computation of prediction based on the three granules, and a suffix that refers the user similarity approach evaluated:

- **MultiGran_BL:** this alternative uses the same user similarity formula as the *Baseline*, i.e., it implements the function f as the hour-based user similarity. In this variant, two users are similar if both visited the same POI at the same or similar hour.
- **MultiGran_MULTI:** this is the most strict variant, which defines the function f as a multiplication of the three user similarity values of each granule (see Equation 7). Here, two users are similar if they have checked-in to the same place at the same or similar hour, day-of-the-week, *and* month.
- **MultiGran_MIN:** it implements the function f as the minimum of the three values of user similarity. Less strict than the previous one, this variant adopts a pessimistic view of the similarity between two users, by assigning it the value of the time granule when they less coincided.
- **MultiGran_MAX:** it defines the function f as the maximum of the three user similarity values. In contrast to the previous one, this approach adopts an optimistic view of user similarity, by assigning it the value of the time granule when two users mostly coincided.
- **MultiGran_SUM:** function f is implemented as a bounded sum of the three user similarity values. Unlike $MultiGran\_MULTI$, it increases (instead of decreasing) the user similarity by jointly considering the three granules.

All these variants were compared with the *Baseline*, by following the same evaluation steps and using the same dataset. Three evaluations were performed and their results are detailed below. The difference between these evaluations resides in what it is assumed as *ground truth*. This is, in the evaluation process we compare the recommendation with what the user actually did (the ground truth). However, there are several definitions of this concept and all of them may be meaningful in some applications. For example, as we know the POI that the user visited at the time he/she asked for the recommendation, we can use it as ground truth. This is our first evaluation, which is the most restrictive (it is very difficult for the system to predict that exact POI). Other definitions of ground truth could be what the user visited at any time, or in the same hour/day-of-week/month, etc. Among those we selected two that we consider more relevant. The second evaluation marks the recommendation as a true positive if the user visited the recommended POI at any time and the last evaluation accepts a recommendation if the user visited the POI any day, but at the same hour (we include this evaluation as it is similar to the one used by the *Baseline*).

4.2.1 Evaluation 1

For a given user at a given time, the 20 POIs with the highest predictive preference value are recommended. Then, we compare this list with the list of POIs that the user actually visited at that time (which is obtained from the test file

Experiment	TOP1	Total	%TOP1
Baseline	595	4,051	14.7%
MultiGran_BL	1,557	4,377	35.6%
MultiGran_MULT	1,424	4,116	34.6%
MultiGran_MIN	1,538	4,361	35.3%
MultiGran_SUM	1,452	4,254	34.1%
MultiGran_MAX	1,340	4,138	32.4%

Table 1: Summary of the results of the Evaluation 1.

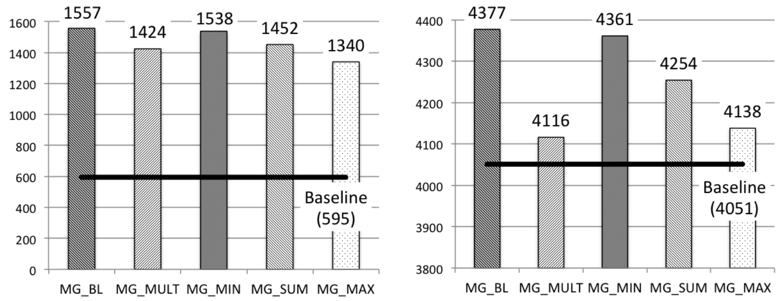

Figure 4: Number of times the POI visited by the user was matched by the first recommendation (TOP1, left) and by any of the 20 recommendations (right)

and, most of the times, contains a single POI or two at most). As the list of recommended POIs is sorted by the predictive value, its first POI should be the one preferred by the user.

In Table 1, *TOP1 column* shows the number of times the first recommended POI was actually visited by the user at the time of recommendation; *Total column* shows how many times any of the 20 POIs recommended were actually visited by the user; and *%TOP1 column* shows the percentage of times the first recommendation was visited (first column) in relation to the number of times that any recommendation was visited (second column).

As it can be seen in Table 1, all the multigranular variants that we propose in this work outperform the *Baseline*. The differences between the alternatives become clearer in the graphs of Figure 4, where the *Baseline* is represented as an horizontal line across the graph. Figure 4 (left) shows that the *Baseline* recommends in the TOP1 the actual visited POI about 600 times, whereas our best proposal, *MultiGran_BL*, guesses the TOP1 correctly around 1,600 times (i.e., our approach outperforms in almost 2.7 times the *Baseline*). Our other proposals also outperform the *Baseline*, but *MultiGran_BL* prevails over them.

If we recall from the previous subsection, *MultiGran_BL* variant uses the same equation to obtain similar users than the *Baseline*. Therefore, the difference between them resides in the computation of the prediction value. This result reinforces our hypothesis that the quality of the recommendations may increase

Experiment	TOP1 visited at any time	Evaluation 1	Increment
Baseline	959	595	364
MultiGran_BL	1,768	1,557	211
MultiGran_MULT	1,574	1,424	150
MultiGran_MIN	1,727	1,538	189
MultiGran_SUM	1,634	1,452	182
MultiGran_MAX	1,530	1,340	190

Table 2: Summary of the results of Evaluation 2.

with the consideration of a multigranular representation of the time variable.

In Figure 4 (right), we show the number of times that any of the 20 recommendations matched the actual POI visited by the user. In this case, differences between our alternatives are larger and only $MultiGran\_MIN$ results are close to those from $MultiGran\_BL$. This is interesting because $MultiGran\_MIN$ implements a rather pessimistic definition of user similarity, in comparison to $MultiGran\_MAX$, which considers the maximum user similarity and obtained worse results (the comparatively low results of $MultiGran\_MULT$ could be explained by the influence of the maximum user similarity in its computation). One possible explanation of this is a bigger influence of user similarity at hour granule in the quality of recommendations, and that the minimum user similarity value could frequently occur in this granule. This is a relevant aspect that deserves further examination.

4.2.2 Evaluation 2

In this evaluation, we check whether the TOP1 recommended POI was actually visited by the user at any time, not only at the time of recommendation (which is the case of Evaluation 1). In other words, we relax the ground truth and, thus, the potential list of true positives is larger. The results are shown in Table 2.

These results are similar to those from the first evaluation. *MultiGran_BL* variant is positioned over the others. Given this, we can see that *MultiGran_BL* proposal increased by 211 check-ins with respect to the first evaluation, meaning that on 211 occasions the user visited another place at the time that recommendation was required, but did visit the recommended POI at another time.

Once again, our proposal outperforms the *Baseline*. Results from $MultiGran\_BL$ variant are better than the *Baseline* by a factor of two. In this second evaluation, the *Baseline* technique was the one with a greater increment when compared to the first evaluation. However, this is not necessarily better, as it may mean that our proposals better match the prediction considering the time.

4.2.3 Evaluation 3

This evaluation is similar to the one used by the *Baseline*, which grouped the places visited by the user at a specific hour and compared with the recommendation that was generated for that specific hour. Obviously, this evaluation is more advantageous for the *Baseline*, which generates predictions for blocks of hours (unlike our proposals that generate predictions for three granules of the registered timestamps). The results of this experiment are shown in Figure 5.

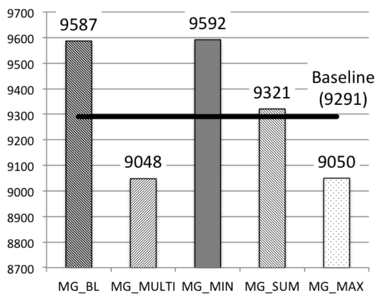

Figure 5: Number of times the user visited one of the 20 recommended POIs in the same hour for which the recommendation was generated.

Even though this evaluation was designed for the *Baseline*, three of our proposals outperform the *Baseline*. In this evaluation, for the first time *MultiGran_MIN* proposal is better than *MultiGran_BL*, although by a small margin. In this evaluation we considered that, if the user visited any of the 20 recommended POIs, then it was a successful prediction. That is why we see in the graph values around the 9,600 check-ins (a much larger value than in previous evaluations). Note also that the differences in this evaluation are very small (around 300 check-ins from a universe of more than 9,000 predictions well made).

4.3 Evaluation of the statistical proposal

In previous evaluations, *MultiGran* proposal proved to be more accurate than the *Baseline*. However both the *Baseline* and *MultiGran* proposal have an elevated computational cost, with respect to processing time and memory consumption. This was the motivation of the *Stat* approach proposed in Section 3.2: a lighter technique that characterizes POIs check-ins by using statistical data. In this section, we evaluate this proposal and compare it in terms of accuracy and computational cost with the *Baseline* and *MultiGran_BL* approach, which obtained the best results in the experiments described above.

Experiment	Running Time (s.)	TOP1	Total	%TOP1
Baseline	6,291	595	4,051	14.7%
Baseline_Stat	1,442	494	3,649	13.5%
MultiGran_BL	6,816	1,557	4,377	35.6%
MultiGran_Stat	1,311	1,378	3,926	35.1%

Table 3: Execution times of the proposed methods.

4.3.1 Processing time

The *Stat* approach defines how to compute the similarity between users by using a few statistical values. This provides an alternative to the UTP cubes of the *Baseline* and *MultiGran* approaches. In order to evaluate the efficiency of this approach, we incorporated the user similarity equation of this proposal in the *Baseline* and *MultiGran_BL* proposal. Table 3 shows running times of the proposals when using their original equation to obtain users similarity and then using the equation proposed in the statistical proposal (we use the suffix *_Stat* to name these new variants). For both proposals, running times do not consider the pre-processing time required to represent POIs through UTP cubes and Box-and-Whisker charts, respectively. Furthermore, this table shows the accuracy of the methods.

Table 3 shows that the results for accuracy decreased in both methods when using the *Stat* representation. However, the decrease in accuracy is low when compared with the savings in execution time of both proposals, which reaches almost an 80% (i.e., the running time when using *Stat* is about 20% of the original time). This shows that the characterization of the POIs through a few statistical data implies that the equations must check few data to deliver a similarity value. Thus, the *Stat* representation provides significant savings in execution time while keeping good quality recommendations. Note that our *MultiGran_Stat* variant is still better than the *Baseline* and it takes less running time.

4.3.2 Memory consumption

To measure the efficiency of the proposals, another important factor is the memory required to run the algorithms. In this section we study the space required by the proposed techniques. We distinguish two kinds of techniques: those using UTP cubes to train the system (i.e., *Baseline* and *MultiGran_BL*) and those that characterize POIs with statistical values (i.e., *_Stat* variants). In order to estimate the space required by each proposal, we use the same training file that contains 4,257,572 check-ins to 1,074,704 different POIs from 102,589 users.

In the first kind of techniques, the UTP cubes demand the largest amount of memory. The *Baseline* stores one UTP cube, whereas *MultiGran_BL* approach uses three cubes, one for each time granule. We assume that each smoothed UTP

Cube	Detail	Total(MB)
Hour	$102,589 \cdot 24 \cdot 27 \cdot 4$ Bytes	> 253.59
Day-of-the-Week	$102,589 \cdot 7 \cdot 27 \cdot 4$ Bytes	> 73.96
Month	$102,589 \cdot 12 \cdot 27 \cdot 4$ Bytes	> 126.79

Table 4: Memory required by each of the UTP cubes.

cube stores $U \cdot T \cdot P_x$ values, where U is the total number of users, T represents the number of time units at a specific time granule, and P_x is the average number of POIs contained in the training file for each user. Assuming that these cubes store integers that require 4 bytes, the amount of memory consumed by each cube is detailed in Table 4. Thus, the *Baseline* requires about 250 MB and $MultiGran\_BL$ requires about 450 MB.

On the other hand, the memory required by the statistical proposal depends on the characterization of the POIs. Each time block is characterized by 5 values, however, each POI at a specific time granule may be characterized by one or more blocks. Then, the memory consumption of the statistical proposal is given by the equation $5 \cdot B \cdot G \cdot P$, where B is the number of blocks that characterize a POI, G is the number of time granules (3 in these experiments), and P is the number of POIs contained in the training file. Assuming 4 bytes integers, the amount of memory consumed by the statistical proposal is about 60 MB per block.

These results show that the amount of memory used by the statistical proposal is significantly lower than the used by approaches based on UTP cubes. In these experiments, we used one or two blocks for each time granule, so the worst case is when all the time granules use two blocks. In this case, the system requires about 123 MB of memory, which is about half the space of the *Baseline*.

Finally, it is interesting to notice that the memory required by the statistical proposal depends on the number of POIs and not on the number of check-ins, which is the case of the proposals based on UTP cubes. Thus, as the number of check-ins is expected to grow much faster than the number of POIs, the memory saving of the *Stat* approach would be even larger over time.

4.4 Discussion

To summarize, let us focus on the best evaluated approaches ($MultiGran\_BL$ and $MultiGran\_Stat$) and compare them with the *Baseline* in terms of more standard measures [Herlocker et al. 2004]. For this purpose, we use precision@N (denoted as $pre@N$) and recall@N ($rec@N$), where N is the length of the recommendation list [Ye et al. 2011, Yuan et al. 2013a]. $prec@n$ measures the number of correctly predicted POIs in the top-N, whereas $rec@N$ is defined as the ratio of relevant POIs in the top-N recommendations to total number of relevant POIs. Note that these metrics depend on what we consider relevant (i.e., the ground truth) and that the $TOP1$ measure presented above is a sort of $prec@1$.

In order to contextualize these results, as stated in [Yuan et al. 2013a]: *we focus on the relative improvements achieved, instead of on the absolute values.* Absolute values are relatively low, because the used dataset has low density, which usually results in low evaluation values [Ye et al. 2011], and also because just few POIs in the dataset may represent the real interests of each user.

The following figures summarize the results of this comparison in terms of precision and recall. As in previous sections, we show the results of three different definitions of ground truth: only the POI visited by the user at the exact time of recommendation (Figure 6), all the POIs visited by the user (Figure 7), and all the POIs visited by the user at the same hour but any day (Figure 8). The first definition is the most restrictive, which results in lower values on precision and recall, but is also the most realistic as it evaluates the recommendation considering only the precise POI visited by the user.

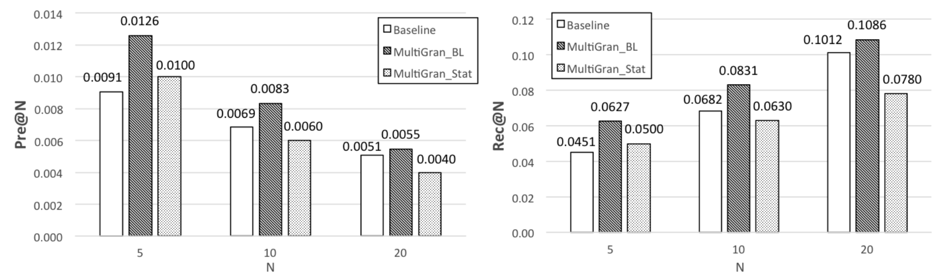

Figure 6: Precision and recall @N graphs when the ground truth contains only the POI actually visited by the user at the time s/he asked for a recommendation.

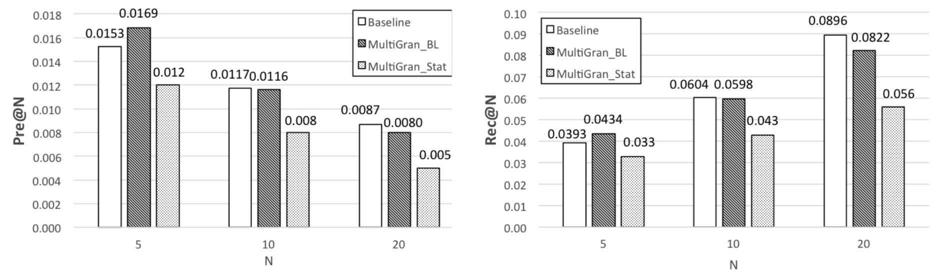

Figure 7: Precision and recall @N graphs when the ground truth contains all the POIs visited by the user (at any time in the future).

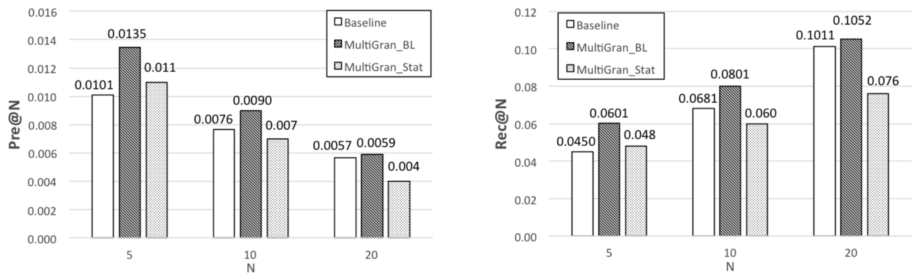

Figure 8: Precision and recall @N graphs when the ground truth contains all the POIs visited by the user at the same hour (but any day in the future).

The $MultiGran\_BL$ proposal systematically outperforms all the other approaches both in terms of precision and recall. Although we show prec@N and rec@N for $N = \{5, 10, 20\}$, we must emphasize the outstanding results for $N = 5$, as the first recommendations get more attention from the users (in the case of web search engines, for example, the first three links get about 75% of the clicks). If we restrict our comparison to this case, the $MultiGran\_BL$ approach always outperforms the other approaches. For prec@5, this improvement is more than 44% in the first evaluation, a bit more than 10% in the second evaluation, and a 30% in the last evaluation. Recall from [Yuan et al. 2013a] that, by considering the time variable, they got an improvement of about 40% over the state of the art (a recommendation algorithm that did not consider the time variable). Now, by considering a multigranular representation of the time variable, we get a similar improvement over their proposal.

The $MultiGran\_Stat$ approach is competitive with the $Baseline$ and even better in some evaluations. This is a remarkable result because, as shown above, it requires considerable less computational resources than the other approaches.

Finally, let us mention the spatial variable as another relevant contextual variable in POI recommendations. Yuan et al. [Yuan et al. 2013a] complement its monogranular specification of time with this variable, showing that the impact (in terms of precision and recall) of the time is much higher than the impact of the spatial variable. For example, in their experiments for the Gowalla dataset the prec@5 considering only the best temporal method is about 0.027, whereas prec@5 considering only the best spatial method is about 0.013. Even more important, prec@5 considering both time and space is 0.028 (i.e., a marginal improvement over the results achieved when considering only the temporal variable). In addition, we can also combine our algorithm with the proposal of [Yuan et al. 2013a] or any other proposal that considers the space variable and one would expect a similar improvement. However, this deserves additional research and experimentation.

5 Conclusions and Future Work

In this article, we focused on the importance of the time variable for the recommendation of POIs. Specifically, we proposed two novel solutions to the so called Time-aware POI recommendation problem, which consider multiple time granules (e.g., hour, day-of-the-week and month). In this way, certain POIs may be a good recommendation if we only consider a particular time granule, but they might be excluded from the recommendations if we consider the three of them, thus affecting the accuracy of the final recommendation. For example, a Pub may be a good recommendation at midnight but probably not on Monday.

We presented two proposals of POI recommendation algorithms that consider the time granularity. The first one ($MultiGran$) extends the monogranular time description from [Yuan et al. 2013a], adopting their representation based on check-in UTP cubes, but for each of the three aforementioned time granules. The second one ($Stat$) is based on a statistical representation of each POI at each time granule. Both representations are used to compute similarity between users and prediction values, the main steps of the collaborative filtering approach.

We conducted extensive experiments over a real-world LBSN dataset. The experimental results show the importance of a multigranular representation of the time variable because, by including the three time granules, we improved the accuracy of the recommendations by a factor of two (in terms of $TOP1$) and about a 40% (in terms of prec@5) with respect to the monogranular approach of the $Baseline$ (this is for the case of $MultiGran$). In addition, the $Stat$ proposal uses less memory than the UTP cubes, drastically reduces the running time and still obtains an accuracy comparable with the state of the art.

Several interesting directions are open for further exploration. First, we plan to explore space-efficient data structures for implementing the $MultiGran$ approach. It would be interesting to keep its good accuracy, while improving its scalability (especially in terms of memory consumption). There exist in the bibliography data structures that represent sparse matrices in very compact space, while supporting efficient access [Brisaboa et al. 2014]. An alternative may be to improve the accuracy of the $Stat$ approach, which already provides good scalability. In this regard, the use of segmentation algorithms to determine the best characterization (in terms of number of blocks) of each POI seems promising. In addition, each block can be divided into more zones (for example, we can consider deciles instead of quartiles) and this may impact the accuracy of the recommendations. Having shown the importance of considering the multigranular nature of the time variable for POI recommendations, further work is necessary to characterize which time granules have a greater impact in the accuracy of the recommendations for different domains.

Frequency with which users check-in the same place deserves special attention. In the scenario proposed by our multigranular specification, if a user checks-in the same restaurant several times a day, the frequency of these check-ins can be obtained from the UTP cube of the hour granule; if she checks-in the same place everyday, this daily routine will be identifiable from the day-of-the-week UTP cube; and if she checks-in it in different months, this will be registered in the month UTP cube. We claim that data of some recurring check-in behavior may be lost or difficult to obtain (e.g., two check-ins of the same place at the same hour in different days are registered only once in the hour UTP cube). Even when the similarity computation is intended to be computed periodically, and other check-ins can compensate the missing data, we state that the identification and representation of frequent check-in behavior in different time granules, and its effects on the quality of recommendation, deserves a deeper analysis.

As a final remark, we claim that a multigranular approach, once proven to be effective in terms of recommendation quality, can be better combined with other contextual characteristics, either isolating or combining different time granules with, for example, location characteristics. Hence, this work must be complemented with the inclusion of additional context variables (such as age range or weather conditions). In this line, the inclusion of the spatial variable is promising as space and time have been successfully exploited together in many domains. In the results of [Yuan et al. 2013a], the influence of the spatial variable is much lower than that of the time variable. However, a deeper study of its representation and combination with our approach may have a greater impact.

Acknowledgements

This work was supported by the National Council for Scientific and Technological Research (CONICYT, Chile) under grants Fondecyt ref. 11080277 (Gonzalo Rojas), Fondecyt ref. 11130377 (Diego Seco), and CONICYT-PCHA/ Magíster Nacional/2013-221320252 (Francisco Serrano).

References

[Adomavicius and Tuzhilin 2011] Adomavicius, G., Tuzhilin, A.: "Context-aware recommender systems"; Recommender Systems Handbook; 217–253; Springer, 2011.

[Baltrunas et al. 2011] Baltrunas, L., Ludwig, B., Peer, S., Ricci, F.: "Context-aware places of interest recommendations for mobile users"; A. Marcus (ed.), Design, User Experience, and Usability. Theory, Methods, Tools and Practice; Lecture Notes in Computer Science 6769, Springer Berlin Heidelberg (2011), 531–540.

[Bannur and Alonso 2014] Bannur, S., Alonso, O.: "Analyzing temporal characteristics of check-in data"; Proc. 23rd International Conference on World Wide Web, WWW'14 Companion, ACM, New York (2014), 827–832.

[Bettini et al. 1998] Bettini, C., Wang, X. S., Jajodia, S.: "A general framework for time granularity and its application to temporal reasoning"; Annals of Mathematics and Artificial Intelligence, 22, 1-2 (1998), 29–58.

[Brisaboa et al. 2014] Brisaboa, N. R., Ladra, S., Navarro, G.: "Compact representation of web graphs with extended functionality"; Inf. Syst., 39 (2014), 152–174.

[Campos et al. 2013] Campos, P. G., Díez, F., Cantador, I.: "Time-aware recommender systems: a comprehensive survey and analysis of existing evaluation protocols"; User Modeling and User-Adapted Interaction, 24, 1-2 (2014), 67–119.

[Cho et al. 2011] Cho, E., Myers, S. A., Leskovec, J.: "Friendship and mobility: user movement in location-based social networks"; Proc. 17th ACM SIGKDD Int. Conf. on Knowl. Disc. and Data Mining, ACM, New York (2011), 1082–1090.

[Gao et al. 2013a] Gao, H., Tang, J., Hu, X., Liu, H.: "Exploring temporal effects for location recommendation on location-based social networks"; Proc. 7th ACM Conference on Recommender Systems, RecSys '13, ACM, New York (2013), 93–100.

[Gao et al. 2013b] Gao, H., Tang, J., Hu, X., Liu, H.: "Modeling temporal effects of human mobile behavior on location-based social networks"; Proc. 22nd ACM Int. Conf. on Information and Knowledge Management, New York (2013), 1673–1678.

[Herlocker et al. 2004] Herlocker, J. L., Konstan, J. A., Terveen, L. G., Riedl, J. T.: "Evaluating collaborative filtering recommender systems"; ACM Trans. Inf. Syst., 22, 1 (2004), 5–53.

[Hervás and Bravo 2011] Hervás, R., Bravo, J.: "Coiva: context-aware and ontology-powered information visualization architecture"; Software Practice and Experience, 41, 4 (2011), 403–426.

[Horozov et al. 2006] Horozov, T., Narasimhan, N., Vasudevan, V.: "Using location for personalized POI recommendations in mobile environments"; Proc. Int. Symposium on Applications on Internet, SAINT, IEEE CS, Washington (2006), 124–129.

[Hsieh and Li 2014] Hsieh, H.-P., Li, C.-T.: "Inferring visiting time distributions of locations from incomplete check-in data"; Proc. 23rd Int. Conf. on World Wide Web, WWW '14 Companion, ACM, New York (2014), 295–296.

[Jenks 1967] Jenks, G. F.: "The data model concept in statistical mapping"; International Yearbook of Cartography, 7, 1 (1967), 186–190.

[Jiang 2013] Jiang, B.: "Head/tail breaks: A new classification scheme for data with a heavy-tailed distribution"; The Professional Geographer, 65, 3 (2013), 482–494.

[Kefalas et al. 2013] Kefalas, P., Symeonidis, P., Manolopoulos, Y.: "New perspectives for recommendations in location-based social networks: time, privacy and explainability"; Proc. Fifth Int. Conf. on Management of Emergent Digital EcoSystems, MEDES '13, ACM, New York (2013), 1–8.

[Koren 2009] Koren, Y.: "Collaborative filtering with temporal dynamics"; Proc. 15th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining, KDD '09, ACM, New York (2009), 447–456.

[Liang et al. 2012] Liang, H., Xu, Y., Tjondronegoro, D., Christen, P.: "Time-aware topic recommendation based on micro-blogs"; Proc. 21st Int. Conf. on Information and Knowledge Management, CIKM '12, ACM, New York (2012), 1657–1661.

[Rendle 2010] Rendle, S.: "Factorization machines"; Proc. 2010 IEEE Int. Conf. on Data Mining, ICDM '10, IEEE Computer Society, Washington (2010), 995–1000.

[Rendle et al. 2011] Rendle, S., Gantner, Z., Freudenthaler, C., Schmidt-Thieme, L.: "Fast context-aware recommendations with factorization machines"; Proc. 34th Int. Conf. on R&D in Inf. Retrieval, SIGIR '11, ACM, New York (2011), 635–644.

[Resnick and Varian 1997] Resnick, P., Varian, H. R.: "Recommender systems"; Communications of the ACM, 40, 3 (1997), 56–58.

[Ricci et al. 2011] Ricci, F., Rockach, L., Shapira, B., Kantor, P. B. (eds.): "Recommender systems handbook"; Springer, New York (2011).

[Schiller and Voisard 2004] Schiller, J., Voisard, A.: "Location Based Services"; Morgan Kaufmann Publishers Inc., San Francisco (2004).

[Witten and Frank 2005] Witten, I.H., Frank, E. "Data Mining: Practical machine learning tools and techniques"; Morgan Kaufmann, San Francisco (2005).

[Woerndl et al. 2007] Woerndl, W., Schueller, C., Wojtech, R.: "A hybrid recommender system for context-aware recommendations of mobile applications"; Proc. IEEE 23rd Int. Conf. on Data Engineering Workshop, ICDEW '07, IEEE Computer Society, Washington (2007), 871–878.

[Xiong et al. 2010] Xiong, L., Chen, X., Huang, T., Schneider, J. G., Carbonell, J. G.: "Temporal collaborative filtering with bayesian probabilistic tensor factorization"; Proc. 2010 Int. Conf. on Data Mining, SIAM, Philadelphia (2010), 211–222.

[Ye et al. 2011] Ye, M., Yin, P., Lee, W.-C., Lee, D.-L.: "Exploiting geographical influence for collaborative point-of-interest recommendation"; Proc. 34th Int. Conf. on R&D in Inf. Retrieval, SIGIR '11, ACM, New York (2011), 325–334.

[Yu and Chen 2015] Yu, Y., Chen, X.: "A Survey of Point-of-Interest Recommendation in Location-Based Social Networks"; Workshops at the 29th AAAI Conf. on Artificial Intelligence, Austin, TX (2015).

[Yuan et al. 2013a] Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N. M.: "Time-aware point-of-interest recommendation"; Proc. 36th Int. Conf. on R&D in Inf. Retrieval, SIGIR, ACM, New York (2013), 363–372.

[Yuan et al. 2013b] Yuan, Q., Cong, G., Ma, Z., Sun, A., Thalmann, N. M.: "Who, where, when and what: Discover spatio-temporal topics for twitter users"; Proc. 19th Int. Conf. on Know. Disc. and Data Mining, ACM, New York (2013), 605–613.

[Yuan et al. 2014] Yuan, Q., Cong, G., Sun, A.: "Graph-based point-of-interest recommendation with geographical and temporal influences"; Proc. 23rd Int. Conf. on Information and Knowledge Management, CIKM, ACM, New York (2014), 659–668.

[Yuan et al. 2015] Yuan, Q., Cong, G., Zhao, K., Ma, Z., Sun, A.: "Who, where, when, and what: A nonparametric bayesian approach to context-aware recommendation and search for twitter users"; ACM Trans. on Inf. Sys., 33, 1 (2015), 1, 2:1–2:33.

[Zhao et al. 2015] Zhao, X., Li, X., Liao, L., Song, D., Cheung, W.K.:"Crafting a Time-Aware Point-of-Interest Recommendation via Pairwise Interaction Tensor Factorization"; Proc. 8th Int. Conf. on Know. Science, Engineering and Management, Chongqing, China (2015), 458–470.

[Zheng et al. 2010] Zheng, V. W., Cao, B., Zheng, Y., Xie, X., Yang, Q.: "Collaborative filtering meets mobile recommendation: A user-centered approach"; Proc. 24th AAAI Conf. on Artificial Intelligence, AAAI 2010, AAAI Press (2010), 236–241.

[Zheng 2011] Zheng, Y.: "Location-based social networks: Users"; Computing with Spatial Trajectories, Springer (2011), 243–276.