Extracting Geospatial Preferences Using Relational Neighbors

Leandro Balby Marinho, Thomas Sandholm[2], Cláudio de Souza Baptista,
Iury Nunes, Caio Nóbrega, Jordão Araújo

[1] Universidade Federal de Campina Grande, Brazil
{lbmarinho, baptista}@dsc.ufcg.edu.br, {iury.nunes,caio.nobrega,jordao.araujo}@ccc.ufcg.edu.br
[2] HP Labs, USA
thomas.e.sandholm@hp.com

Abstract. With the increasing popularity of location-based social media applications and devices that automatically tag generated content with locations, large repositories of collaborative geo-referenced data are appearing on-line. Efficiently extracting user preferences from these data to determine what information to recommend is challenging because of the sheer volume of data as well as the frequency of updates. Traditional recommender systems focus on the interplay between users and items, but ignore contextual parameters such as location. In this paper we take a geospatial approach to determine locational preferences and similarities between users. We propose to capture the geographic context of user preferences for items using a relational graph, through which we are able to derive many new and state-of-the-art recommendation algorithms, including combinations of them, requiring changes only in the definition of the edge weights. Furthermore, we discuss several solutions for cold-start scenarios. Finally, we conduct experiments on data collected from the Panoramio photo sharing site and provide empirical evidence that many of the proposed algorithms outperform existing location-aware recommender algorithms.

1. INTRODUCTION

With the affordable prices of GPS-enabled mobile devices and the success of social networks, location-based social media has become increasingly popular in recent years. Users can upload content, e.g., photos, videos, and text, and annotate that content with geographical identification metadata, typically known as *geotags*. Geotags act as geographic indexes helping users to organize and retrieve location-specific information. Foursquare[1], as an example, is a location-based service where users endorse and share tips about visited points of interest (POI). It reached 725 thousand registered users and 22 million check-ins (i.e. endorsed POIs) in 2010[2].

Recommender systems (RS) are among the best known techniques for helping users filter out and discover relevant information in large data sets. In the typical scenario, RS algorithms exploit user-item matrices representing user preferences for items, e.g., the rating history of purchased books in

---

[1] http://foursquare.com/
[2] http://mashable.com/2010/03/29/foursquare-growth-numbers/

Amazon, with the aim of recommending the items most likely to be relevant to the user. While most of the RS work to date has ignored the locations where users demonstrated interest for an item, there are many scenarios in which the geographic context of an item has a direct influence on the preferences of the user for that item. Ye et al. [Ye et al. 2011], for example, showed that Foursquare users prefer POIs that are nearby the POIs they already visited in the past, while [Matyas and Schlieder 2009] showed that Panoramio users who took pictures in nearby locations in the past, tend to share similar preferences of geographic regions in the future. Efficient extraction and representation of location-specific user preferences are thus essential to decide on what item to recommend.

Location-aware recommender systems suggest relevant geotagged items for a given user within a declared geographic area. Relevance here can assume different notions, depending on the geographic constraints imposed by the user. For example, a user may be interested in objects nearby his previous, current, or future location within a given radius. For example, a first time user visiting the Stanford University campus in Palo Alto, US, might be interested to know what is worth visiting inside the campus, while a second time visitor may want to know what else is worth visiting. Each of these scenarios can lead to various definitions of user preference, hence it is important to know which definition works best for each scenario. The literature concerning location-aware recommender system is still sparse, where the methods are ad-hoc and can not be easily changed to meet the different recommendation scenarios outlined above (e.g. [Matyas and Schlieder 2009; Sandholm and Ung 2011; Del Prete and Capra 2010]).

In this paper we propose a relational graph for capturing the geographic context of users that suits all the aforementioned recommendation scenarios. We introduce several strategies to represent location-specific user preferences in the graph and show how to derive many recommendation algorithms, including ensembles of them, by only changing the definition of the edge weights. Our contributions are as follows:

(1) We propose a new model for geotagged data that is able to capture both the geographic context of users and their preferences for objects within the declared geographic context.
(2) We introduce new similarity measures that take into account the spatial decisions of users.
(3) We propose a recommendation algorithm based on a relational neighbor graph, that derive many recommendation algorithms and ensembles of them, as special cases, by only requiring changes in the definition of the edge weights.
(4) Finally, we conduct experiments on data collected from Panoramio in various recommendation scenarios, including cold-start ones, and provide empirical evidence that many of the proposed algorithms outperform existing location-aware recommender algorithms.

The rest of the paper is organized as follows. Section 2 describes related work. Section 3 introduces the problem setting. Section 4 presents our relational graph representation of the data, weighting schemes for capturing user geographic preferences, and a recommendation algorithm based on relational neighbors. Section 5 presents the experimental setting and evaluation, and Section 6 concludes the paper and discusses future work.

2.   RELATED WORK

Related work falls into three broad categories: relational classification and similarity measures, location-aware methods, and methods focusing on improving the computational scalability of the model building and execution phases.

Relational Classification and Similarity Measures. Relational classification has been applied to areas where entities are linked in an explicit manner, like hypertext documents, such that the class of a target instance only depends on the class of its related instances [Preisach and Schmidt-Thieme 2006]. In [Marinho et al. 2009] we presented a relational neighbor classification approach in the

ECML/PKDD Discovery Challenge 2009, which was about tag recommendations. Similarly to that work, in this paper we formulate the recommendation problem in a relational neighbor framework where geographic contexts link entities represented by user/location pairs.

In [Matyas and Schlieder 2009] a new similarity measure for computing location recommendations based on a non-overlapping hierarchical taxonomy of locations is presented. The key idea is that co-activity in locations can be better captured if you zoom out to larger and larger locations. Similar to our work they use a panoramio data set to evaluate their algorithm. However, their model assumes a cold-start user is making the query, i.e. a user who has no trace in the geographic context of the query. The reliance on a place taxonomy is also more restricting than our general model.

Jeh and Widom define SimRank in [Jeh and Widom 2002]. The SimRank similarity measure is as the name suggests highly influenced by the more famous PageRank [Brin and Page 1998] algorithm. The general idea is that "two objects are similar if they are referenced by similar objects" [Jeh and Widom 2002]. This definition recursively propagates similarity through a relational graph to leverage structural context in addition to the more traditional use of object content and attribute information. In our work akin to the model used in [Matyas and Schlieder 2009], we also have a recursive, relation-based definition of similarity but in the form of hierarchical geographic contexts. For faster computation we simplify the similarity model to not be recursive between leaf-nodes within the same geographic context in our graph. Our evaluations show that this can be done without any loss of accuracy if the user who requested the recommendation already has an activity trace within the geographic context for which the recommendation is sought. Minkov and Cohen [Minkov and Cohen 2010] also explore relational properties among objects to compute similarity scores. They focus on the problem of effectively searching in graphs comprised of interrelated objects of various types. They propose effective random-walk based procedures to evaluate the Personal PageRank measure [Brin and Page 1998], a measure that takes into account the scope of the query in addition to the well-known random surfer assumption of the original PageRank algorithm.

Location-aware Methods. In [Ye et al. 2011] a standard user-based collaborative filtering approach is extended with a novel mechanism of geographic influence based on a statistical model of how likely a user is to check-in to two places based on their distance. The geographic model is a distributional assumption (power law) fitted to real data sets from two popular check-in services. It is worth noticing that this model does not consider the geographic context of the user as we do.

In [Berjani and Strufe 2011] interactions between users and POIs are modeled in a matrix such that standard matrix factorization techniques can be applied. Similarly to [Ye et al. 2011], this work also does not consider the location where users are requesting recommendations.

A number of studies have looked at timestamped GPS traces to predict future locations within very restricted geographic regions [Cao et al. 2010; Zheng et al. 2010; Burbey and Martin 2008]. Markov models and tensor factorization models are fit to the data and non-personalized predictions of the most likely next location or the most likely activity given a location and time are produced. There is a lot of novel work on automatically detecting geographic context in these papers but the general approach cannot be replicated easily in our scenario since we do not have the same luxury of rich traces as we mainly focus on implicit feedback as input. Relying on GPS traces is furthermore a privacy concern and has scalability and power consumption implications. We produce more personalized recommendations given user-user similarities as opposed to just looking at the most popular or most frequent behavior. Furthermore large Markov models and tensor factorization algorithms tend to be very costly to compute, which would therefore need to be done in an off-line setting whereas we also target real-time recommendations.

Bayesian networks have also been studied to model and learn patterns in location, time and weather contexts for individual users in [Park et al. 2007]. However, compared to our model these models tend to be complex and require expert human knowledge to construct, and furthermore they are

not tractable. [Brunato and Battiti 2003] applies a center of mass model to detect and recommend locations and POIs. This work was before the check-in systems era so now we could just more easily query the check-in services for this information. In [Horozov et al. 2006] user-user based CF with location pre-filtering is employed in an explicit voting scenario. The cold start problem is solved by generating random recommendations using pseudo users. We address the problem by incorporating out-of geo context similarities for in-context recommendations which is less ad-hoc.

In our previous work in [Sandholm et al. 2010] we studied popularity inferred recommendations based on location, friends and tag prefiltering using both explicit and implicit feedback. In this work we extend that model to provide personalized recommendations similar to the work in [Sandholm and Ung 2011] but with a new model that incorporates location, and distance metrics directly in the evaluation graph as opposed to relying on ad-hoc and costly pre and post filtering.

The GeoFolk system [Sizov 2010] was designed to take both geographic context and text features into account for various information retrieval tasks such as tag recommendation, content classification and clustering. Experiments show that combining both textual and geographic relevance leads to more accurate results than using the two factors in isolation. Although our methods and use case targets are quite different from this work, the empirical evidence of the influence geographic context has on information retrieval is promising and serves as motivation for our work.

Computational Scalability. Popular and accurate recommender system methods such as those based on matrix factorization can incur very high model building as well as execution overhead, in particular as more contextual factors beyond users and items are taken into account. As a result there have been many attempts at improving the computational scalability of pre-existing methods.

In [Sandholm and Ung 2011; Ye et al. 2010; Del Prete and Capra 2010] the general issue of complex and high-latency model building and execution for location-based recommendations is addressed. To achieve real-time performance [Sandholm and Ung 2011] and [Del Prete and Capra 2010] pre-filter based on location and [Ye et al. 2010] pre-filters based on friends to reduce the complexity of the models. As opposed to pre- or post-filtering context we make contextual parameters an inherent part of our graph model to allow interesting combinations of various types of context in an efficient way. Moreover, clustering photos into regions of interest, as we do, decrease drastically the number of recommendable items that need to be kept in memory, which enable real time recommendations. Furthermore, the number of discovered clusters is often small. Notice that even in top touristic places, such as Rio, there is usually only a few hundred regions of interest to be recommended.

Rendle et al. [Rendle et al. 2011] propose a more generic matrix factorization-based model for fast context-aware recommendation, although the geographic context is never tried. As pointed out before, matrix factorization-based models can provide fast recommendations after the model is trained, but the training phase can still be expensive, especially on large datasets. Our method, in contrast, achieves scalability by not utilizing any complex training phase.

3. PROBLEM SETTING

The recommendation scenario is as follows. A user specifies the geographic region of interest, e.g., a city or a region in that city, and the recommender engine suggests items within the declared geographic region that are likely to be relevant to the user.

So, let U be the set of users, G the set of regions denoting geographic contexts, and I the set of geotagged items. Notice that what is meant by a region is application dependent since regions can assume different geographic shapes, such as point coordinates, circles, lines, and polygons. In this paper we only consider implicit feedback data[3], i.e., the set $S \subseteq U \times G \times I$ of ternary relations between

---

[3]Although our framework can trivially support explicit feedback as well.

users, geographic contexts, and geotagged items. The task is then to find a prediction scoring function

$$\hat{s} : U \times G \times I \to \mathbb{R} \qquad (1)$$

that predicts a preference score for items within certain geographic regions, given a target user. Now, for a given user $u \in U$, and a given geographic context $g \in G$, the topN recommendations can be computed by

$$topN(u, g) := \operatorname*{argmax}_{i \in I_g}^{n} \hat{s}(u, g, i) \qquad (2)$$

where n denotes the topN items to be recommended and I_g the set of items within the geographic context g. For convenience, we also define $I_{u,g} := S \cap (\{u\} \times \{g\} \times I)$ as the set of items of user $u \in U$ in a given geographic context $g \in G$.

4.   A RELATIONAL APPROACH

Relational classification refers to an active area of machine learning where classifiers usually consider, additionally to the typical attribute-value data of objects, relational information. A scientific paper, for example, can be connected to another paper that has been written by the same author or because they share common citations. It has been shown that in many classification problems, relational classifiers outperform purely attribute-based classifiers [Chakrabarti et al. 1998; Lu and Getoor 2003; Preisach and Schmidt-Thieme 2006]. In particular, Macskassy and Provost [Macskassy and Provost 2003] showed that simple relational neighbor-based techniques, besides requiring low computational costs, perform competitively to, and in some cases even outperforms, more complex relational methods such as Probabilistic Relational Models and Relational Probability Trees. The basic idea is that the classification of a target instance solely depends on the class labels of related instances of the same type. Since geotagging data is inherently relational, we propose to capture the geographic context of users and items by a relational graph, which, as a side effect, gives us many tools from relational classification that can be directly applied to the location-aware recommendation problem.

In order to easily use neighborhood-based classification methods, similarly to [Macskassy and Provost 2003; Preisach and Schmidt-Thieme 2006], we adopt a homogeneous view of the relations in the data. In a homogeneous view we have only one entity type, such that, there is a set of target entities $x \in X$ and relations $R \subseteq X \times X$ between them. However, as we saw in Section 3, geotagging data forms a set S of ternary relations between three different types of entities. Therefore, we first need to convert these ternary relations into the desired homogeneous relations.[4] We do this as follows.

4.1   Graph Definition

First let $V := \{(u, g) \mid \exists i \in I : (u, g, i) \in S\}$ denote the set of all distinct user/geographic context combinations in S. We now propose to interpret S as a graph $\mathbb{G} := (V, E)$, where V is the set of vertices and $E \subseteq V \times V$ the set of edges. We assume that there is an edge between two vertices if they share the same geographic context, i.e.,

$$\{(v, v') \in E \mid g_v = g_{v'}\}$$

For convenience, let $g_v := g$ and $u_v := u$ denote the geographic context and the user of node $v \in V$ respectively. In other words, we assume that users who share the same geographic region are related to some extent.

Now suppose that John is a first time visitor in Rio de Janeiro, Brazil, and wants to know from other people who already have been in Rio which places are worth visiting there. To continue our

---

[4]Relational classification is possible with heterogeneous node types as well as has been demonstrated in [Jeh and Widom 2002] and [Minkov and Cohen 2010] but the added complexity defeats our goal of fast model building and evaluation.
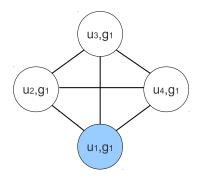
Fig. 1.   Example of relational graph for capturing geographic context.

example, we denote the pair John/Rio by a colored node in the graph of Fig. 1, i.e., the target node for which we want to compute recommendations. John is denoted as u_1 and Rio as g_1 respectively, and the other nodes connected to it contain the users who already have been in Rio. For computing recommendations, we just need to go through the items of the neighbors and define some selection criterion on which items to recommend.

This idea assumes that entities related to each other, in this case users sharing the same geographic context, are similar and tend to select the same items. Notice that the strength of the similarity depends on the size of the geographic region being considered. If the declared geographic context is a small region, say the Copacabana beach in Rio de Janeiro, then it is more likely that users within this area will be more strongly related to each other than users sharing larger geographic contexts, such as the whole country of Brazil. We can alleviate these effects by defining appropriate weights to each relation, as we will see next.

4.2   Weighting Schemes

In this section we present several weighting strategies for the edges of the relational graph. And as we will see in the next section, each weighting scheme leads to a specific recommendation algorithm, without, however, the need to change the overall algorithm.

Uniform Weighting. Here the same weight value is assigned to each edge (1 in our case), denoting that each neighbor is of the same importance to the target user. Therefore, the weight w for any edge $(v,' v) \in E$, is defined as $w^{\mathrm{uni}}(v, v') := 1$. The computational cost for assigning uniform weights to the relations of a target node v is in the order of $O(|N_v|)$, i.e., a linear scan in the neighbors of v, here denoted by N_v.

Correlation Weighting. For an edge $(v, v') \in E$ where the vertices v and v' represent profile vectors, each component of the vector is a geotagged item and the values denote the preference of a user for an item, i.e.,

$$\vec{m}_v := (i_1, ..., i_{|I|})$$

We can either construct this vector solely based on the items of the geographic context of interest, or we could consider the items of all geographic contexts. The assumption is that users who have selected the same items within the same geographic contexts are more similar than users who did not. If a user declared interest for some geographic context for which he has not yet selected any item, it will not be possible to compute any correlation similarity with other users, unless we build the profile vectors considering the items of other geographic contexts.

The edge weight between two nodes is finally computed by applying a correlation measure between the desired nodes' profile vectors. We have chosen the well known cosine similarity as correlation measure, which is typically used by implicit feedback-based collaborative filtering algorithms [Desrosiers
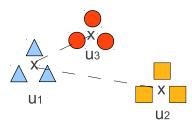
Fig. 2. Example of geographic similarity. The items of users u_1, u_2, u_3 are represented by triangles, circles and rectangles respectively. Centroids are represented by "X" and distances by dashed lines. Thus, u_3 would be regarded as more similar to u_1 than u_2.

and Karypis 2011]:

$$w^{\mathrm{cor}}(v, v') := \frac{\langle \vec{m}_v, \vec{m}_{v'} \rangle}{\|\vec{m}_v\| \|\vec{m}_{v'}\|} \qquad (3)$$

The computational cost for assigning cosine similarities to the relations of a target node v is in the order of $O(Z \cdot |N_v|)$ since we need to compute $|N_v|$ similarities, each requiring Z operations.

Geographic Similarity. We can use the geographic distances between users' items to define the strength of their relation. The assumption is that users who select items nearby the items of other users should be closely related. Therefore, for defining the weight between two nodes $(v, v') \in E$ through geographic similarity, we first calculate the geographic centroids of the set of items of users u_v and $u_{v'}$. For computing the geographic centroid of a given user $u \in U$, we sum up the coordinates of each geotagged item of user u within the declared geographic context g, i.e., the items in the set $I_{u,g}$, and divide the resulting sum by the number of items in the set

$$\frac{1}{|I_{u,g}|} \sum_{i \in I_{u,g}} p(i)$$

where $p(i)$ returns the latitude/longitude coordinate used to geotag item $i \in I$. Now, the geographic distance between $u(v)$ and $u(v')$ centroids is given by any of the many existing functions for calculating geodetic distances between latitude/longitude coordinates, e.g. the Haversine formula. Here we denote such a function by $d(x, y)$ where x and y are two coordinates. Finally, we only need to turn the distance into a similarity and bound it to the range $[0, 1]$, which is done as follows

$$w^{\mathrm{geo}}(v, v') := 1 - \frac{d\left(c_{u_v}, c_{u_{v'}}\right)}{d_{\max}} \qquad (4)$$

where d_{max} is the maximal possible distance between any points in the region of interest and c_u is the geographic centroid of the set of items of user u. In other words, when the distance between the items of two users is small, their similarity tend to 1, and is 0 when the distance equals $d_{\max}$. Fig. 2 illustrates this idea. Notice that for this to work, we are assuming that users tend to form geographic clusters among the selected items. This was empirically observed in [Ye et al. 2011] by showing that Foursquare users tend to check-in to POIs that are nearby the POIs they have already visited.

The computational cost of this weighting scheme is in the same order as the *correlation weighting* above.

Partonomy-Based Similarity. The different granularities between geographic contexts can be represented by a hierarchical *partonomy*, i.e., a graph $\mathbb{G} := (G, E)$ where vertices are represented by geographic contexts and each edge $e \in E$ represents a part-of relation between two geographic re-

gions [Matyas and Schlieder 2009]. For example, a POI is a part of a city, a city is a part of a state, a state is a part of a country, etc. Matyas et al. [Matyas and Schlieder 2009] proposed several similarity measures for capturing user similarities with respect to specific levels of a weighted geographic partonomy. The idea is that if two users are not found to be similar in a lower level of the hierarchy, they may, eventually, be found to be similar in a higher level. For example, even if two users did not visit the same places in some city, they still can be considered to be similar if they have visited the same city. We can use these measures, together with a partonomy, to weigh the relations in our relational graph. Here we just present the best performing measure reported in [Matyas and Schlieder 2009].

The similarity of two users $u, u' \in U$ with respect to node $g \in G$ is calculated as follows

$$sim^{inf}(u, u', g) := \frac{\sum_{(c \in A_g \cap B_g)} information(c)}{\sum_{(c \in A_g \cup B_g)} information(c)} \tag{5}$$

where A_g and B_g denote the sets of children nodes of g in which user u and u' selected items respectively. The similarity is weighted by the function *information*, which can be seen as the inverse of the popularity of a node. The assumption is that users sharing less popular nodes are more similar than users sharing more popular ones [Matyas and Schlieder 2009].

Now, for computing the similarity between two users $u, u' \in U$ with respect to a certain partonomy layer l, the following formula is used

$$sim^{two\text{-}layer}(u, u') := \frac{\sum_{(g \in G_l)} sim^{inf}(u, u', g) \cdot w^{node}(g)}{\sum_{(g \in G_l)} w^{node}(g)} \tag{6}$$

where G_l is the set of nodes in the l-th layer of the partonomy and the weight $w^{node}(g)$ of node g reflects the importance of node g across its siblings in the partonomy. This measure was named *two-layer similarity* in [Matyas and Schlieder 2009] because when computing the similarity between two users in a certain layer, the measure uses the layer immediately below.

Returning the discussion to the weighting of the edges in our relational graph, we can now define the weight of a certain edge $(v, v') \in E$ in terms of the *two-layer similarity*, i.e.,

$$w(v, v') := sim^{two\text{-}layer}(u_v, u_{v'}) \tag{7}$$

This similarity can be quite expensive to compute, given all the necessary steps described above for weighting the partonomy. Even assuming that the weighted partonomy is given, the complexity for computing the weight between v and all its neighbors is still higher than for the other similarities. The computational cost is in the order of

$$O(|N_v||G_l||C_l| \cdot Z)$$

where C_l is the set of children nodes of the l-th layer, since for each neighbor, we need to go through all the locations g in the l-th layer of the partonomy, for each location go through all its children, and finally for each child we need to perform Z operations.

4.3 Recommendation Algorithm

Now, we have all the components for deriving a location-aware recommendation procedure. Algorithm 1 describes the overall recommendation process. It receives as input a weighted graph $\mathbb{G}$, a user/geographic context pair denoted by v, for which we want to generate recommendations, and the number n of recommendations to be returned. The algorithm iterates through the neighbors of v (line 3), denoted by N_v, and for each neighbor, it iterates through the items within the geographic context

of interest (line 4) and accumulates weights in the array *scores*, that is indexed by items. Finally, it sorts the scores in descending order of weights and presents the top-n geotagged items that the target user has not already selected.

Note that if we define all weights to 1, we end up recommending the most popular items within the geographic context of interest. Or if we decide to weigh edges according to the similarities presented in Section 4.2, we end up with many flavors of collaborative filtering-based algorithms.

Assuming that the weighted relational graph is given, the complexity of this algorithm only depends on the computation of a weighted sum of geotagged items, which means $|N_v|$ passes in the set of items I. Hence, the complexity is given by $O\left(|I||N_v|\right)$.

Algorithm 1 Graph-based Location Recommendations

1: **Input:** $\mathbb{G}(V, E)$, $v \in V$, n
2: **Output:** list of topN recommendations
3: **for all** $v' \in N_v$ **do**
4:   **for all** $i \in I_{u_{v'}, g_{v'}}$ **do**
5:     $scores[i] \leftarrow scores[i] + w(v, v')$
6:   **end for**
7: **end for**
8: topN $\leftarrow \underset{i \in I \setminus I_{u_v, g_v}}{\overset{n}{\arg\max}} \; scores[i]$

---

5. EXPERIMENTS

In this section we describe the dataset we have used; the evaluation protocol adopted; and the results for each dataset. We considered three different recommendation scenarios.

In the first scenario, we hide all the geotagged items of each test user in a geographic context, and use the remaining data for trying to predict the removed items. This corresponds to a cold-start scenario where a user has not selected any item in the geographic context of interest. We will refer to this scenario as *leave-all-out*.

In the second scenario, we remove some geotagged items for each test user, 4 photos in this case, and use the remaining data for trying to predict the removed items. This scenario represents those users who already selected some items in a given location but want to know what other items are worth selecting in this location. We will refer to this scenario as *leave-some-out*

In the third scenario, we have a mix of both scenarios, i.e., some fraction of the users are first time users and the other fraction already have selected some items in the location of interest. This corresponds to a more realistic scenario, and to the best of our knowledge, this is the first time location-aware algorithms are evaluated in this kind of scenario. We will refer to this scenario as *leave-some/all-out*

5.1 Data Collection and Preparation

Panoramio is a photo-sharing website from Google where users can upload, geotag, and retrieve photos of landmarks. Each photo in Panoramio is georeferenced using latitude and longitude information. Similarly to [Matyas and Schlieder 2009], we assume that if a user takes a picture at some specific location, then he has some interest in that location. As geographic context of interest we have chosen the city of Rio de Janeiro, which is one of the top touristic places in the world, so, it has a large set of photos.

Table I.   Panoramio data for Rio de Janeiro

	$|U|$	$|I|$
Training	1,062	4,906
Test	186	3,590

In order to retrieve the set of photos taken in Rio, we did a spatial search for photos inside the bounding box of Rio using the data access API of Panoramio. We then iterated through each photo in the result set, and retrieved the users who took these photos. For each of these users, we then retrieved the other locations where they took photos, as well as the users and photos in those locations. Then we removed the cities with too little activity from users who also visited Rio. Approximately the top three cities in all the crawled states were kept in the evaluation set. We used the gazetteer of HP Gloe[5] to obtain the place names.

In order to use the *two-layer-similarity* weighting scheme presented in Section 4.2, we built a geographic partonomy, and similarly to [Matyas and Schlieder 2009], we worked with three countries (Brazil, Chile, and USA) as high level nodes, states and cities as intermediary nodes, and geographic clusters as leaf nodes. Our data set contains 35,920 photos (4,906 from Rio) and 7,048 users in total.

In Panoramio the geographic items are represented by latitude and longitude coordinates, hence, it is very difficult for two users to take a picture in the exact same location. Therefore, we adopted the same approach as in [Matyas and Schlieder 2009], where the authors used geographic clusters to represent geographic items. So, instead of recommending individual point coordinates, we recommend regions where users may be interested in taking photos. For computing the clusters, we used the DBSCAN [Ester et al. 1996] algorithm with the following parameters: *MaxRadius* = 1 Km and *MinPoints* = 3. The *MaxRadius* parameter was set to 1 Km, as we assume that a radius of 1 Km from a given photo is sufficient to establish a geographic similarity between photos, e.g., photos taken at the University of Stanford campus. We tested to set *MaxRadius* = 500 meters without any significant differences in performance, so we do not show those results here. Also, the minimum number of points to form a cluster was set to 3, in order to establish popularity of a given POI. We computed 1,187 clusters, 221 of which are in Rio.

5.2 Evaluation Protocol

For testing the algorithms, we considered the dense part of the data, i.e., only the users who have taken at least 5 photos in Rio (see Table I). For the *leave-some-out* and *leave-some/all-out* scenarios, we generated 5 random splits of training/test sets and took the average precision and recall on top-10 recommendation lists over all splits. We have a hit every time a hidden photo is found to belong to some of the recommended geographic clusters. Whenever CF is not able to fill the recommendation list up to 10, we fill up the list with the most popular items that are not already in the recommendation list.

5.3 Algorithms

We have used several weighting schemes for our relational graph, which resulted in the following recommendation algorithms:

—**Most popular (MP):** Recommends the most popular geographic clusters in Rio. For doing that, we just apply Uniform Weighting (see Section 4.2) to the graph in Algorithm 1.
—**Intra-Cluster (IC):** Uses the geographic similarity defined in Section 4.2 for defining intra-cluster similarities between users. The idea is as follows. The weight of any edge $(v, v') \in E$ is now

---

[5]http://www.hpgloe.com/

calculated by summing up $w^{\text{geo}}(v, v')$ for all clusters where u_v and $u_{v'}$ have photos, and dividing the resulting sum by the total number of clusters. After that, we again just need to plug the weighted graph into Algorithm 1 for computing recommendations.

—**Collaborative Filtering (CF):** This is the geographic version of the classic user-based collaborative filtering algorithm. We use the correlation weighting scheme defined in Section 4.2 where profile vectors' components are the geographic clusters where users took pictures, within and out of the context of interest.

—**Two-Layer (TL):** This is the algorithm originally introduced in [Matyas and Schlieder 2009] but recast as a relational neighbor recommender. We weigh the relational graph with the *two-layer similarity* and plug it into the recommendation algorithm.

—**Correlation + Two-Layer (CF-TL)**: We found empirically that TL works best for cold-start scenarios, while CF outperforms the other algorithms for non cold-start scenarios (see Figures 3 and 4). This gave us the insight to propose a weighting combination strategy where we weigh each relation differently according to the case presented. If the target user is a cold-start user, we weigh his relations with the *two-layer similarity*, if not, we use the *correlation weighting*. The flexibility for combining different weighting schemes in such an easy way is one of the main advantages of our approach.

5.4 Results

Fig. 3 depicts the results for the *leave-some-out* scenario. Notice that when there is enough data available, CF outperforms all the other methods. The Intra-Cluster recommender, although worse than CF and TL, is better than MP in all cases. This indicates that the geographic similarity indeed is able to capture some preferences of the user, under the assumption that users tend to like the items of other users that are nearby the items they have selected in the past.
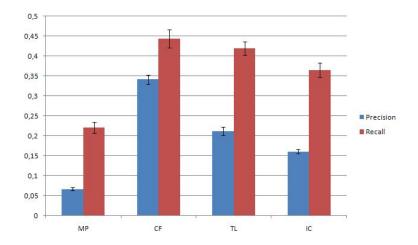

Fig. 3. Evaluation on *leave-some-out* scenario.

Fig. 4 shows the results for the *leave-all-out*. Notice that TL is the winner in this scenario. This is in line with the results of [Matyas and Schlieder 2009], where they showed that geographic partonomies can help to improve the recommendations in cold-start scenarios. Notice that since we remove all the photos of all test users in the context of interest, we just have one possible split of training/test, and thus cannot compute standard deviations and plot error bars.

In Fig. 5 we show the results of the *leave-some/all-out* scenario when 50% of the test users are cold-start users. In this case, the combination method CF-TL is slightly better, both in precision
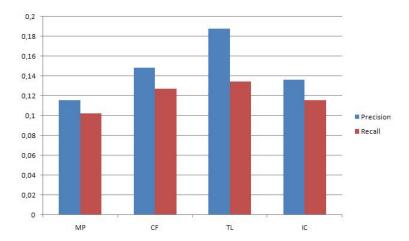
Fig. 4.   Evaluation on *leave-all-out* scenario.

and recall, than the other methods. But when 70% of ther users are cold-start users (see Fig. 6) the superiority of CF-TL becomes more evident. We also evaluated this scenario when 30% of the users are cold-start users, but since there was no significant differences in performance in comparison to 50% of cold-start users, we do not show those results here.
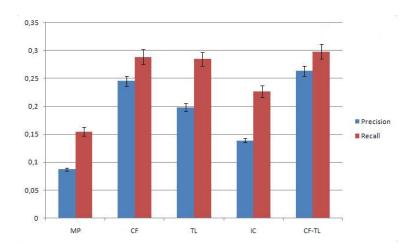

Fig. 5.   Evaluation on the *leave-some/all-out* scenario when half of the users are cold-start users.

6.   CONCLUSIONS AND FUTURE WORK

In this paper we introduced a relational graph for capturing the geographic preferences of users with the purpose of generating personalized recommendations in services with geotagged content. We also presented several weighting schemes for representing different types of user preferences in the proposed graph. Furthermore, we propose a recommendation algorithm template that is sufficiently generic to derive many traditional and new location-aware recommendation algorithms, including combinations of them, by only requiring changes in the definition of the edge weights. Assuming the graph is given, the algorithm requires modest computational effort since it runs linearly in the number of neighbors. We have tested the proposed algorithms on a geotagged photo collection collected from Panoramio and showed how our model easily suits many different recommendation scenarios.
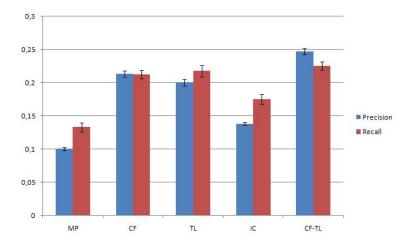
Fig. 6.   Evaluation on the *leave-some/all-out* scenario when 70% of the users are cold-start users.

We also gained insights about which notion of similarity works best for a set of scenarios. In cold-start scenarios a geographic partonomy seems to be a good alternative, whereas when there is enough data available the plain location-aware collaborative filtering algorithm yields the best result. In response to this finding, we proposed to combine a partonomy-based similarity measure with the cosine similarity by weighting individual relations in the graph according to the type of the user, i.e., cold-start versus non cold-start. By doing this, we achieved better recall and precision in particular in scenarios where there are many cold-start users.

As future work, we plan to incorporate temporal aspects in the model, such that the items to be recommended match the temporal context of the user. For example, it may not make as much sense to recommend ski resorts in New York during the summer as it would to make the same recommendation during the peak winter season. Another natural extension of our work would be to assign the weights in our graph based on the strength of the social ties between the users, e.g. based on their declared or implied social networks. Finally, we plan to investigate machine learning approaches for learning optimal weights based on the location-aware recommendation task at hand.

REFERENCES

BERJANI, B. AND STRUFE, T. A recommendation system for spots in location-based online social networks. In *Proceedings of the 4th Workshop on Social Network Systems*. Salzburg, Austria, pp. 1–6, 2011.

BRIN, S. AND PAGE, L. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the seventh international conference on World Wide Web 7*. Brisbane, Australia, pp. 107–117, 1998.

BRUNATO, M. AND BATTITI, R. Pilgrim: A location broker and mobility-aware recommendation system. In *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*. Fort Worth, USA, pp. 265–272, 2003.

BURBEY, I. AND MARTIN, T. L. Predicting future locations using prediction-by-partial-match. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*. San Francisco, USA, pp. 1–6, 2008.

CAO, X., CONG, G., AND JENSEN, C. S. Mining significant semantic locations from gps data. *Proceedings of VLDB Endowment* vol. 3, pp. 1009–1020, September, 2010.

CHAKRABARTI, S., DOM, B., AND INDYK, P. Enhanced hypertext categorization using hyperlinks. *SIGMOD Rec.* 27 (2): 307–318, 1998.

DEL PRETE, L. AND CAPRA, L. differs: A mobile recommender service. In *Proceedings of the 2010 Eleventh International Conference on Mobile Data Management*. IEEE Computer Society, Kansas City, USA, pp. 21–26, 2010.

DESROSIERS, C. AND KARYPIS, G. A comprehensive survey of neighborhood-based recommendation methods. In *Recommender Systems Handbook*, F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor (Eds.). Springer, pp. 107–144, 2011.

ESTER, M., KRIEGEL, H.-P., SANDER, J., AND XU, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the second ACM SIGKDD international conference on Knowledge discovery and data mining*. Portland, USA, pp. 226–231, 1996.

HOROZOV, T., NARASIMHAN, N., AND VASUDEVAN, V. Using location for personalized poi recommendations in mobile environments. In *Proceedings of the International Symposium on Applications on Internet*. Phoenix, USA, pp. 124–129, 2006.

JEH, G. AND WIDOM, J. Simrank: a measure of structural-context similarity. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*. Edmonton, Canada, pp. 538–543, 2002.

LU, Q. AND GETOOR, L. Link-based classification using labeled and unlabeled data. In *Proceedings of the ICML Workshop on The Continuum from Labeled to Unlabeled Data in Machine Learning and Data Mining*, 2003.

MACSKASSY, S. A. AND PROVOST, F. A simple relational classifier. In *Proceedings of the Second Workshop on Multi-Relational Data Mining*. Washington DC, USA, pp. 64–76, 2003.

MARINHO, L. B., PREISACH, C., AND SCHMIDT-THIEME, L. Relational classification for personalized tag recommendations. In *Proceedings of the ECML/PKDD Discovery Challenge*. Vol. 497. Bled, Slovenia, 2009.

MATYAS, C. AND SCHLIEDER, C. A spatial user similarity measure for geographic recommender systems. In *GeoSpatial Semantics*, K. Janowicz, M. Raubal, and S. Levashkin (Eds.). Lecture Notes in Computer Science, vol. 5892. Springer Berlin / Heidelberg, pp. 122–139, 2009.

MINKOV, E. AND COHEN, W. W. Improving graph-walk-based similarity with reranking: Case studies for personal information management. *ACM Transactions on Information Systems* vol. 29, pp. 4:1–4:52, December, 2010.

PARK, M.-H., HONG, J.-H., AND CHO, S.-B. Location-based recommendation system using bayesian user's preference model in mobile devices. In *Ubiquitous Intelligence and Computing*, J. Indulska, J. Ma, L. Yang, T. Ungerer, and J. Cao (Eds.). Lecture Notes in Computer Science, vol. 4611. Springer Berlin / Heidelberg, pp. 1130–1139, 2007.

PREISACH, C. AND SCHMIDT-THIEME, L. Relational ensemble classification. In *Proceedings of the 6th International Conference on Data Mining*. Beijing, China, pp. 499–509, 2006.

RENDLE, S., GANTNER, Z., FREUDENTHALER, C., AND SCHMIDT-THIEME, L. Fast context-aware recommendations with factorization machines. In *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information*. Beijing, China, pp. 635–644, 2011.

SANDHOLM, T. AND UNG, H. Real-time location-aware collaborative filtering of web content. In *CaRR '11: Proc. of the 2011 Workshop on Context-awareness in Retrieval and Recommendation*. ACM, Palo Alto, California, pp. 14–18, 2011.

SANDHOLM, T., UNG, H., APERJIS, C., AND HUBERMAN, B. A. Global budgets for local recommendations. In *Proceedings of the fourth ACM conference on Recommender systems*. Barcelona, Spain, pp. 13–20, 2010.

SIZOV, S. Geofolk: latent spatial semantics in web 2.0 social media. In *Proceedings of the third ACM international conference on Web search and data mining*. New York, USA, pp. 281–290, 2010.

YE, M., YI, P., LEE, W.-C., AND LEE, D.-L. Exploiting geographical influence for collaborative point-of-interests recommendation. In *Proceedings of the 34th ACM SIGIR Conference*. Beijing, China, pp. 325–334, 2011.

YE, M., YIN, P., AND LEE, W. C. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*. San Jose, California, pp. 458–461, 2010.

ZHENG, V. W., ZHENG, Y., XIE, X., AND YANG, Q. Collaborative location and activity recommendations with GPS history data. In *Proceedings of the 19th international conference on World Wide Web*. Raleigh, North Carolina, USA, pp. 1029–1038, 2010.