Context-Aware Friend Recommendation for Location Based Social Networks using Random Walk

Hakan Bagci
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
hakan.bagci@ceng.metu.edu.tr

Pinar Karagoz
Department of Computer Engineering
Middle East Technical University
Ankara, Turkey
karagoz@ceng.metu.edu.tr

ABSTRACT

The location-based social networks (LBSN) facilitate users to check-in their current location and share it with other users. The accumulated check-in data can be employed for the benefit of users by providing personalized recommendations. In this paper, we propose a random walk based context-aware friend recommendation algorithm (RWCFR). RWCFR considers the current context (i.e. current social relations, personal preferences and current location) of the user to provide personalized recommendations. Our LBSN model is an undirected unweighted graph model that represents users, locations, and their relationships. We build a graph according to the current context of the user depending on this LBSN model. In order to rank the recommendation scores of the users for friend recommendation, a random walk with restart approach is employed. We compare RWCFR with popularity-based, friend-based and expert-based baseline approaches. According to the results, our friend recommendation algorithm outperforms these approaches in all the tests.

Keywords

Location-Based Social Networks; Friend Recommendation; Random Walk

1. INTRODUCTION

The recent advances in mobile communication and location-acquisition technologies encouraged mobile users to share their location data through LBSNs [18]. The datasets that are collected over LBSNs contain user and location items. They also include check-in history of users and friendship data. These datasets can be employed for personalized recommendations.

LBSNs provide a new way of friend recommendation based on user location histories [5]. User location histories provide valuable contextual information. Moreover, location histories and social behaviors are notably correlated [9]. Friend recommendation is extensively studied for traditional social networks. However, there are only a few number of friend recommendation algorithms that employ LBSN data in recommendation. Previous methods generally employ raw GPS data to find the similarity between users. When compared to raw GPS data, check-in data provides more contextual information. Moreover, most of the LBSNs collect check-in data rather than GPS trajectories. The majority of the proposed friend recommendation algorithms are not context-aware algorithms, hence they do not consider the current location of the user. A contextual friend recommendation algorithm can be more useful for finding new friends at visited locations.

In this paper, we propose a novel friend recommendation algorithm, RWCFR (**R**andom **W**alk based **C**ontext-aware **F**riend **R**ecommendation). Our friend recommendation algorithm considers social, personal and spatial context. It employs second degree friends (friends of friends). User's visited locations in recommendation region are also considered to identify place friends. Place friends are potential friends that have check-ins at the locations where the current user visited before. In addition to this, local experts and popular locations are employed in populating the context of the user.

In this paper, we use our LBSN model that is introduced in [2, 3]. This model represents an LBSN as an undirected unweighted graph. In [2] this model is employed for location recommendation and in [3] it is used for activity recommendation. The recommendation method that is employed in this study is similar to those in [2, 3], however, the main difference lies in the subgraph construction phase. RWCFR is a context-aware algorithm, hence it constructs a subgraph according to the current context of the user. Local experts and popular locations in region are added onto this subgraph. After constructing the subgraph, it is given as an input to our random walk algorithm, and it calculates the recommendation probabilities of users for friend recommendation. A list of potential friends is provided to user according to output of the random walk algorithm.

The main contribution of this work is that social connections, user's location history and current location are employed seamlessly in friend recommendation. Location experts and popular locations are extracted locally and employed in friend recommendation. Moreover, when new users and locations are inserted into database, unlike most of the collaborative filtering (CF) methods, our recommendation method does not need to update any existing model or structure, such as tensors. Our subgraph is constructed dynamically using efficient queries through a graph database.

Copyright is held by the International World Wide Web Conference Committee (IW3C2). IW3C2 reserves the right to provide a hyperlink to the author's site if the Material is used in electronic media.
WWW'16 Companion, April 11–15, 2016, Montréal, Québec, Canada.
ACM 978-1-4503-4144-8/16/04.
http://dx.doi.org/10.1145/2872518.2890466.

We compare the performance of our friend recommendation algorithm with popularity-based, friend-based and expert-based baselines. The results of the experiments show that RWCFR outperforms all the compared algorithms. This indicates that consideration of location, social and personal context together increases the friend recommendation accuracy.

2. RELATED WORK

Friend recommendation is extensively studied for traditional social networks. LBSNs provide a new way of friend recommendation based on user location histories [5]. User location histories provide rich contextual information. Moreover, location histories and social behaviors are significantly correlated [9].

In [16], Symeonidis *et al.* propose a prototype system GeoSocial that is able to recommend locations, activities and friends to users. In friend recommendation they employ FriendLink algorithm [14]. In order to calculate the weights between links they use the average geographical distance between users' check-ins.

GeoLife2.0 is a social networking service that enables users to share their life experiences with other users. In [19], the authors say that GeoLife is capable of measuring the similarity between users based on the location histories. This similarity measure is employed for friend recommendation for individuals.

In [8], Chu *et al.* propose a friend recommendation approach based on the similar interests of the users. Moreover, they employ the real-life location and dwell time in friend recommendation. After gathering these data, the proposed method analyzes the data using weighted Voronoi diagram and interest similarity.

In [17], the authors propose a random walk based statistical framework for geo-friends recommendation (GEFR). It is a three-step approach and first, raw GPS data is analyzed and interesting and discriminative GPS patterns are extracted. The extracted geographical information and social network are combined in a heterogeneous information network. Random walk is applied on this network to provide friend recommendations. GEFR employs the patterns that are extracted from raw GPS data. Moreover, GEFR is not a context-aware algorithm and does not consider the current location of the user while recommending friends.

In [12], Li *et al.* introduce a three-layered friendship model that is used to evaluate the similarity between users in LBSNs. They employ social connections, user profiles and mobility patterns to find the correlation between users.

The work proposed in [15] is for recommending new friend links to the users. In this paper, the authors study to reduce the size of the prediction space. The aim of the study is to design a friend link prediction system which exploits data about user check-ins.

In [1], the authors introduce an algorithm for predicting and recommending links in social networks using supervised random walks. Their algorithm combines the data from the network structure with node and edge level attributes. These attributes are used to guide the random walk process on the graph.

3. PROPOSED APPROACH

In this section, we describe the details of our random walk based context-aware friend recommendation algorithm, namely RWCFR. RWCFR employs user's location history in friend recommendation. LBSN data is modeled using an undirected unweighted graph. Random walk with restart (RWR) is performed on the instance of this graph model to rank the recommendation scores. RWCFR is a context-aware algorithm, hence it employs local social relations, local experts, personal preferences and current location in friend recommendation.

3.1 LBSN Model

LBSN data essentially consists of users, friends, users' location history and friendship data. LBSN data can be modeled using an undirected unweighted graph. This graph defines the relationships between users and locations. Formally, this graph, G, is a tuple $G < V, E >$ where V is a set of nodes v and E is a set of edges e. $V = U \cup L$ where U and L are disjoint sets and U is a set of users, and L is a set of locations.

The users can be categorized into two, ordinary users and expert users. Experts are locally identified according to current location of the user and represent the users that have more knowledge about the region. Local experts are extracted using a HITS-based [6] [11] algorithm. The details of expert identification process is explained in Section 3.4.1.

The relationships between the elements of V are represented as a set of edges, E. There are two types of connection links between LBSN items. These relationships and their definitions are given below:

- **friend-of**: defines the relationship between two friends.
- **visit**: defines the relationship between a user and a location when that user has visited that location at least once.

3.2 Problem Definition

We formulate the friend recommendation problem as follows: G is a graph which is an instance of the LBSN data model that is described in Section 3.1. Given the current location and corresponding graph G, our aim is to predict the potential friends for a particular user u in U. Potential friends list is an ordered list of elements from U with size N, which is the desired number of recommendations. The goal is to generate potential friend list with highest accuracy, in other words, with minimal errors relative to user's future friends.

3.3 Random Walk

In order to rank the nodes of a graph using the information encoded by links, random walk can be employed [13]. The weights of the links define the transition probabilities. Random walk starts from a specific node and continues over the graph depending on this transition probabilities. In every transition, the target node's visit count is incremented. This count is used to rank the nodes of the graph. Random walk terminates when the process reaches a steady state.

Random walk with restart (RWR) is a specialized version of random walk. It is generally used in graphs that have many nodes. When the number of nodes is high, it is possible to move out of the context during random walk. This may cause to visit less important nodes. RWR does not allow moving out of the context, because in each transition there is a constant probability to jump back to the starting node.

As a result of this constraint, nodes that are closer to the starting node tends to have more visit counts. RWR is a commonly used method that provides good relevance score between nodes in a graph.

$$Q = \alpha W + (1 - \alpha)R \quad (1)$$

$$p = pQ \quad (2)$$

Equation 1 defines the matrix that arranges the transition probabilities of a random walk. Here, W stores the transition probabilities of the graph nodes. R is used for modeling the random restart behavior of the random walk. In order to determine the weight between W and R, a predefined α is employed. We can adjust the random walk restart behavior by changing the value of α. Equation 2 must be solved for calculating the ranks of the nodes. p is a vector that represents the steady state probabilities. Here p_i denotes the i^{th} node's probability. In order to calculate the final p, we iterate over Equation 2 until it converges [13].

We employ RWR for making personalized friend recommendations for LBSNs. We construct a graph for the user that requests friend recommendation. The starting node of the random walk is the current user. In order to rank the nodes, we employ a specialized version of RWR in which all the transition probabilities are uniform. This means that the probability of moving to each neighbor from a particular node is the same. Since estimating the weights between connected nodes is a costly task, this assumption lowers our computation cost.

3.4 Friend Recommendation Algorithm

In this section, we explain the details of our proposed friend recommendation algorithm. RWCFR is a context-aware friend recommendation algorithm. It employs RWR to estimate the ranks of the potential friends. RWCFR considers personal, social and location context of the current user.

The graph representing the entire LBSN is very huge. Therefore, we construct a subgraph according to current context of the user requesting friend recommendation. Our friend recommendation algorithm consists of two phases. In the first phase, subgraph is constructed. Then, the actual recommendation is performed using RWR. The details of subgraph construction and recommendation using RWR are given in Section 3.4.1 and Section 3.4.2, respectively.

3.4.1 Subgraph Construction

In order to recommend friends to user, first we build a subgraph according to user's current context. This graph is constructed using the following items:

- Previously visited locations of user in vicinity (personal spatial context)
- Friends and their previously visited locations in vicinity (social spatial context)
- Experts and their previously visited popular locations in vicinity (social spatial context)
- Friends of friends (social context)
- Users that visited locations that the current user previously visited (social spatial context)

This subgraph is an instance of the LBSN data model introduced in Section 3.1. Vicinity is defined by a rectangular region based on a constant radius. Vicinity is also called as recommendation region. The subgraph construction procedure for friend recommendation is given in Algorithm 1.

GetUserLocationsInVicinity procedure retrieves the previously visited locations of the user in vicinity. Similarly, *GetFriendLocationsInVicinity* method is used for retrieval of the locations of friends in recommendation region.

GetExpertLocationsInVicinity procedure identifies the local experts and popular locations. In order to identify the local experts and popular locations, a HITS-based [6][11] algorithm is employed. Here, locations are authority and users are hub nodes. A particular user's hub score represents the knowledge of that user in vicinity, and authority score of a location denotes the popularity of that location in recommendation area. Users who visit many high quality locations tend to have high knowledge about the vicinity. In a similar manner, if a particular location is visited by many high quality users, namely experts, it is more probable for that location to be a quality location [4]. HITS is an iterative algorithm and it converges rapidly in our experiments because the size of the bi-partite graph is small.

GetFriendsOfFriends method is used for finding the second degree friends of the current user. Since friends of friends of the current user could be considered as potential future friends, we add these users to our subgraph.

The users visiting the same places are likely to become friends. Therefore, the users that have check-ins at the locations that the current user previously visited are also potential future friends of the current user. *GetUsersThatVisitedLocation* procedure retrieves the users that checked-in at a location that the current user previously visited. This procedure is called for each vicinity location that the current user checked-in before.

3.4.2 Recommendation using RWR

After constructing the subgraph for friend recommendation, we employ RWR to rank potential friends. The procedure of RWR is given in Algorithm 2. This algorithm performs the actual friend recommendation using random walk. Here, $recCount$ defines the requested number of recommendations. $iterCount$ represents the random walk iteration count. $restartProb$ parameter determines the restart behavior of the random walk. $curNode$ is the node currently being visited. *SelectNextNode* procedure selects one of the neighbors of $curNode$ and that node becomes the new $curNode$. $curNode$ changes in every transition and the visit count of the node is incremented by 1. When the number of iterations reach to $iterCount$, random walk algorithm terminates. Then, the nodes are sorted according to visit counts. It is important to note that this algorithm runs on the local contextual graph of the user. Therefore, the number of vertices in this graph is expected to have much smaller values when compared to the entire graph.

An example friend recommendation subgraph is given in Figure 1. Here, *User_1* requests friend recommendation at a particular location. This subgraph is constructed using Algortihm 1. *Expert_1* and *Expert_2* are identified as local experts. *PopularLoc_1* and *PopularLoc_2* are the popular locations. Friends and friends of friends such as *Friend_4* and *Friend_5* are also depicted on the graph. Moreover, users that have check-ins at the locations that *User_1* visited

Algorithm 1 Subgraph Construction Algorithm

Initialize $G<V,E>$ {Subgraph of the user}
$userLocs \leftarrow GetUserLocationsInVicinity()$
$friendLocs \leftarrow GetFriendLocationsInVicinity()$
$expertLocs \leftarrow GetExpertLocationsInVicinity()$
Create new user vertex u for the current user
$V \leftarrow V \cup u$
for all loc in $userLocs$ **do**
  $v_l \leftarrow$ new location vertex for loc
  $V \leftarrow V \cup v_l$
end for
for all $friendLoc$ in $friendLocs$ **do**
  $v_f \leftarrow$ new user vertex for $friendLoc.user$
  $v_l \leftarrow$ new location vertex for $friendLoc.location$
  $V \leftarrow V \cup \{v_f, v_l\}$
  $e_f \leftarrow$ new edge between u and v_f
  $e_l \leftarrow$ new edge between v_f and v_l
  $E \leftarrow E \cup \{e_f, e_l\}$
end for
for all $expertLoc$ in $expertLocs$ **do**
  $v_f \leftarrow$ new user vertex for $expertLoc.user$
  $v_l \leftarrow$ new location vertex for $expertLoc.location$
  $V \leftarrow V \cup \{v_f, v_l\}$
  $e_f \leftarrow$ new edge between u and v_f
  $e_l \leftarrow$ new edge between v_f and v_l
  $E \leftarrow E \cup \{e_f, e_l\}$
end for
$friendsOfFriends \leftarrow GetFriendsOfFriends(V_f)$
for all $friendOfFriends$ in $friendsOfFriends$ **do**
  for all ff in $friendOfFriends.friends$ **do**
    $v_{ff} \leftarrow$ new user vertex for ff
    $V \leftarrow V \cup v_{ff}$
  end for
end for
for all loc in $userLocs$ **do**
  $v_l \leftarrow$ location vertex for loc
  $locUsers \leftarrow GetUsersThatVisitedLocation(loc)$
  for all $locUser$ in $locUsers$ **do**
    $v_{lu} \leftarrow$ new user vertex for $locUser$
    $V \leftarrow V \cup v_{lu}$
    $e_{lu} \leftarrow$ new edge between v_{lu} and v_l
    $E \leftarrow E \cup e_{lu}$
  end for
end for
$V_f \leftarrow$ friends and experts vertices
$friendships \leftarrow GetFriendRelationships(V_f)$
for all $friendship$ in $friendships$ **do**
  $e_f \leftarrow$ new edge between $v_{friendship.f_1}$ and $v_{friendship.f_2}$
  $E \leftarrow E \cup \{e_f\}$
end for

Table 1: Friend Recommendation Results

Friend	Visit Count
Friend_3	104
Friend_5	65
Friend_7	51
Friend_4	47
Friend_6	45

before (e.g. *Friend_7*) are also added onto the graph.

RWR algorithm operates on the constructed subgraph and the recommendation results are populated. The recommendation results that are sorted by visit count are shown in Table 1. Here, *Friend_3* has the highest visit count and it is recommended in the first place.

Algorithm 2 Random Walk Recommendation Algorithm

$recCount \leftarrow$ number of requested recommendations
$iterCount \leftarrow$ iteration count of random walk
$restartProb \leftarrow$ probability of restart in each move
$G < V, E > \leftarrow$ subgraph of the user
$u \leftarrow$ vertex of the current user in V
$curNode \leftarrow u$
for $i < iterCount$ **do**
  $p \leftarrow$ rand(0,1)
  if $p < restartProb$ **then**
    $curNode \leftarrow u$
  else
    $curNode \leftarrow SelectNextNode(curNode)$
    $curNode.visitCount \leftarrow curNode.visitCount + 1$
  end if
  $i \leftarrow i + 1$
end for
$sortedNodes \leftarrow SortNodesByVisitCount(G<V,E>)$
$result \leftarrow SelectFirstKNodes(sortedNodes, recCount)$

Figure 1: Sample Graph for Friend Recommendation

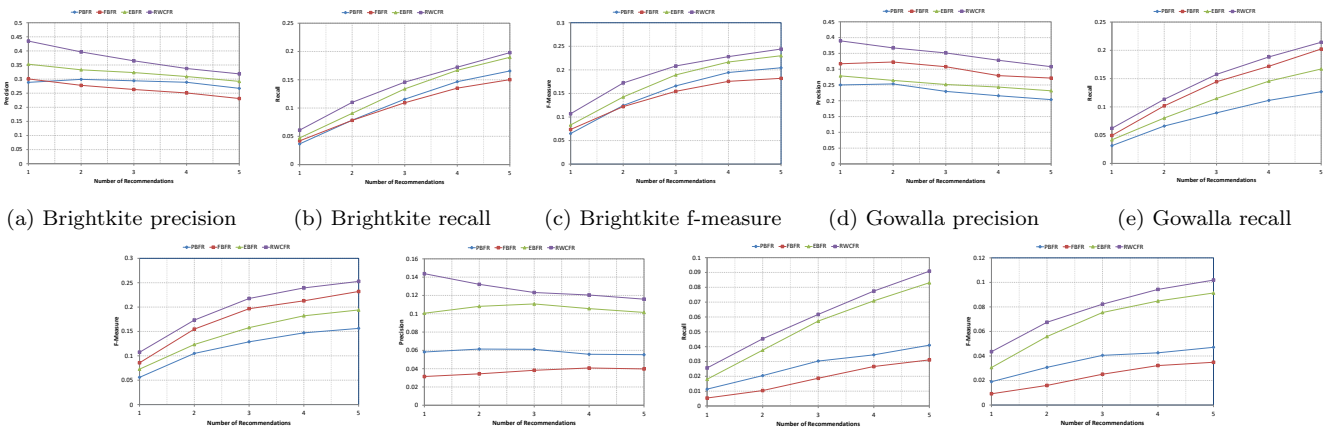
4. EVALUATION

Our friend recommendation algorithm is not directly comparable to the approaches that are explained in Section 2. Therefore, we identified the common approaches that are used in friend recommendation for LBSNs. We compare the performance of RWCFR with these baseline algorithms. The definitions of these approaches are given below:

Popularity-Based Friend Recommendation (PBFR): Recommends the users that have check-ins in recommendation region and have the highest number of friends.

Friend-Based Friend Recommendation (FBFR): Recommends the second degree friends (i.e. friends of friends) sorted by the number of friends.

Expert-Based Friend Recommendation (EBFR):



(a) Brightkite precision (b) Brightkite recall (c) Brightkite f-measure (d) Gowalla precision (e) Gowalla recall

(f) Gowalla f-measure (g) Foursquare precision (h) Foursquare recall (i) Foursquare f-measure

Figure 2: Performance comparison of algorithms in terms of precision, recall and f-measure for each dataset

Recommends the friends of local experts that have the highest number of friends.

4.1 Evaluation Methodology and Metrics

We evaluate our algorithm on three different datasets, which are Brightkite [7], Gowalla [7] and Foursquare [10] datasets. These are the filtered versions of the original datasets for New York City.

RWCFR is a context-aware friend recommendation algorithm and it recommends friends to users based on the current location. We know the current location of the user in real life. However, in order to employ these datasets in our experiments, we need to determine the recommendation points for each user. In order to obtain these points, we employ DBSCAN clustering algorithm. We cluster each user's check-in data, and resultant clusters' centers are employed as current locations of the users in corresponding runs of the algorithm.

RWCFR tries to predict the potential friends of the users. Therefore, we need to partition each user's check-in history and friendship data into two, training and test datasets. RWCFR algorithm operates on training datasets and produces the friend recommendation results for each user. These results are validated using the corresponding test datasets of the users. In order to evaluate the results, we employ two well-known metrics, which are precision and recall. We also employ f-measure to evaluate precision and recall together.

4.2 Experiment Results

In this section, we compare RWCFR with baseline friend recommendation algorithms. The experiments are conducted on three different datasets, which are Brightkite, Gowalla and Foursquare datasets. The results of the experiments are given in Figure 2.

As reported by the results, it is clear that RWCFR outperforms PBFR, FBFR and EBFR considering precision, recall and f-measure metrics. RWCFR is a multi-criteria algorithm and it considers popularity, friendships and local experts together. The proposed LBSN model fuses these data seamlessly. In addition to this, our friend recommendation algorithm also takes user's personal preference into consideration. However, other algorithms are based on a single consideration and they do not provide a data fusion model.

In Brighkite and Foursquare experiments, FBFR has the lowest performance among all the algorithms. In Gowalla experiments, it is the second algorithm in terms of recommendation accuracy. However, it is still behind the performance of RWCFR. If a user has few friends, it is difficult for FBFR to recommend friend of friends. Moreover, in the cases FBFR produces enough number of recommendations, the accuracy of it is very low. Hence, we can conclude that friend of friends are useful for friend recommendation but it is not sufficient for friend recommendation.

PBFR has the lowest performance in Gowalla experiments. In other experiments it is slightly better than FBFR but it still has lower performance than EBFR and RWCFR. Popularity-based techniques are simple and widely used in recommendation. Although it is a simple approach, the results produced by popularity-based approaches are reasonable because popular items target the majority of the audience. PBFR recommends the users that have check-ins in vicinity and have the highest number of friends. However, it does not consider social context (i.e. friendship links) of the user. Moreover, it does not ask the opinion of local experts. It also does not consider the personal preferences of the user. All these reasons put PBFR behind RWCFR. Popularity is still a reasonable friend recommendation approach, but it should be combined with other approaches to produce more accurate results, as in RWCFR.

EBFR asks the opinions of the local experts for friend recommendation. Local experts are very useful in location and activity recommendation [2, 3]. The friend recommendation results show that experts are also good at recommending friends to the users. It has the highest performance after RWCFR in Brightkite and Foursquare experiments. In Gowalla experiments, it is the third algorithm in terms of recommendation accuracy. EBFR is a better choice in recommendation compared to FBFR and PBFR. However, EBFR is still not able to span enough users for recommendation.

RWCFR considers popularity, local experts and second degree friends in friend recommendation. Moreover, it also takes local history and place friends into consideration. All these data are combined with the help of proposed LBSN

model. In addition to this, RWCFR produces more accurate results than all the baselines with the power of random walk. RWCFR has the highest performance for each of the datasets. These results clearly indicate that RWCFR is a stable friend recommendation algorithm for LBSNs.

5. CONCLUSION

In this paper, we propose a random walk based context-aware friend recommendation algorithm for LBSNs. Our friend recommendation algorithm considers social, personal and spatial context. RWCFR constructs a subgraph according to the current context of the user. This graph is given as an input to the random walk algorithm to rank the users for friend recommendation.

According to the results of the experiments, RWCFR performs better than all the baselines for all of the datasets. This is due to the fact that RWCFR is a multi-criteria algorithm, and it considers personal, spatial and social context together and fuses this data using our LBSN model. In addition to this, our recommendation approach does not need to update any existing model or structure, such as tensors, as in the case of CF-based approaches. Our subgraph is constructed dynamically using efficient queries through a graph database.

6. REFERENCES

[1] L. Backstrom and J. Leskovec. Supervised random walks: Predicting and recommending links in social networks. In *Proceedings of the Fourth ACM International Conference on Web Search and Data Mining*, WSDM '11, pages 635–644, New York, NY, USA, 2011. ACM.

[2] H. Bagci and P. Karagoz. Context-aware location recommendation by using a random walk-based approach. *Knowledge and Information Systems*, pages 1–20, 2015.

[3] H. Bagci and P. Karagoz. Random walk based context-aware activity recommendation for location based social networks. In *Data Science and Advanced Analytics (DSAA), 2015. 36678 2015. IEEE International Conference on*, pages 1–9. IEEE, 2015.

[4] J. Bao, Y. Zheng, and M. F. Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 199–208. ACM, 2012.

[5] J. Bao, Y. Zheng, D. Wilkie, and M. Mokbel. Recommendations in location-based social networks: a survey. *GeoInformatica*, 19(3):525–565, 2015.

[6] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. *Computer Networks and ISDN Systems*, 30(1):65–74, 1998.

[7] E. Cho, S. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1082–1090. ACM, 2011.

[8] C.-H. Chu, W.-C. Wu, C.-C. Wang, T.-S. Chen, and J.-J. Chen. Friend recommendation for location-based mobile social networks. In *Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on*, pages 365–370. IEEE, 2013.

[9] J. Cranshaw, E. Toch, J. Hong, A. Kittur, and N. Sadeh. Bridging the gap between physical location and online social networks. In *Proceedings of the 12th ACM international conference on Ubiquitous computing*, pages 119–128. ACM, 2010.

[10] H. Gao, J. Tang, and H. Liu. gscorr: Modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1582–1586. ACM, 2012.

[11] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM)*, 46(5):604–632, 1999.

[12] N. Li and G. Chen. Multi-layered friendship modeling for location-based mobile social networks. In *Mobile and Ubiquitous Systems: Networking & Services, MobiQuitous, 2009. MobiQuitous' 09. 6th Annual International*, pages 1–10. IEEE, 2009.

[13] A. Noulas, S. Scellato, N. Lathia, and C. Mascolo. A random walk around the city: New venue recommendation in location-based social networks. In *Privacy, Security, Risk and Trust (PASSAT), 2012 International Conference on and 2012 International Confernece on Social Computing (SocialCom)*, pages 144–153. IEEE, 2012.

[14] A. Papadimitriou, P. Symeonidis, and Y. Manolopoulos. Friendlink: Link prediction in social networks via bounded local path traversal. In *Computational Aspects of Social Networks (CASoN), 2011 International Conference on*, pages 66–71. IEEE, 2011.

[15] S. Scellato, A. Noulas, and C. Mascolo. Exploiting place features in link prediction on location-based social networks. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1046–1054. ACM, 2011.

[16] P. Symeonidis, A. Papadimitriou, Y. Manolopoulos, P. Senkul, and I. Toroslu. Geo-social recommendations based on incremental tensor reduction and local path traversal. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, pages 89–96. ACM, 2011.

[17] X. Yu, A. Pan, L.-A. Tang, Z. Li, and J. Han. Geo-friends recommendation in gps-based cyber-physical social network. In *Advances in Social Networks Analysis and Mining (ASONAM), 2011 International Conference on*, pages 361–368. IEEE, 2011.

[18] Y. Zheng. Tutorial on location-based social networks. In *21st World Wide Web Conference (WWW 2012)*, Lyon, 2012. ACM.

[19] Y. Zheng, Y. Chen, X. Xie, and W. Ma. Geolife 2.0: a location-based social networking service. In *Mobile Data Management: Systems, Services and Middleware, 2009. MDM'09. 10th International Conference on*, pages 357–358, Taipei, 2009. IEEE.