Karchan Tomar,2023, 11:5
ISSN (Online): 2348-4098
ISSN (Print): 2395-4752

Friend Recommendation in Location-based Social Media Networks via Deep Pairwise Learning

M. Tech. Scholar Karchan Tomar, Prof. Virendra Verma
Dept. of Computer Science & Engineering
Patel College of Science & Technology, Indore, India

Abstract- This research focuses on the mobility management technique of network slicing in 5G networks. Network slicing involves dividing the physical network infrastructure into multiple virtual networks called network slices, each optimized for specific applications or use cases. Network slice handover enables seamless transfer of a user's connection between different network slices without interrupting the quality of service. The research aims to analyze and develop an algorithm for making network slice handover decisions in segmented 5G networks and evaluate its performance. The research proposes an algorithm for network slice handover delivery decisions based on an analytic model using Markov chain. The network model's architecture and the implementation of the vertical delivery decision-making algorithm are described. The performance of the algorithm is evaluated using key performance indicators (KPIs) related to quality of service at the connection level, such as the likelihood of new calls being blocked, or connections being lost. Simulations are conducted to assess how changes in metrics, such as contact arrival rate, capability, new call threshold, base bandwidth unit, and call departure rate, impact the quality-of-service measurements. The simulation results show that the developed algorithm generally provides good quality of service levels, with a lower likelihood of dropping a distribution call compared to new calls being blocked in all cases. The research contributes to the understanding and improvement of network slice handover in 5G networks by proposing an algorithm and evaluating its performance through simulations. The results demonstrate the algorithm's effectiveness in maintaining quality of service during network slice handover.

Keywords: 5G systems; Software Defined Networks; 5G- Network; Slice Handover; Machine learning; Network Function Virtualization.

I. INTRODUCTION

Location-based Social Media Networks (LBSNs) such as Yelp and Foursquare provide services to users that checkin at geo- graphical locations and share such experiences with friends. For instance, Foursquare reports more than 50 million users, 12 billion check-ins and 105 million places in 20181 . Such rich information of social and geographical context provide a unique opportunity for researchers to study user's social behavior [1]. Generating friend recommendations in LBSNs is a challenging task, exploiting the contextual information of users' check-in behavior and mobility patterns. For example, when users often check-in at common locations and visit the same places there is a higher chance of them to become friends based on their cooccurences at such close places [2]–[4]. The challenge is to differentiate between real social friends from strangers by observing their mobility patterns or, in other words, differentiate coincidences from real friends' meetings [5]. In addition, users that are active in close regions might be encouraged to become friends [6]. In an attempt to predict social relationships, several link prediction methods have been proposed considering the available contextual information at LBSNs [1], [2], [5]–[8]. However, these studies try to predict whether two users might become friends or not, and formulate the friend/link prediction task as a binary classification problem. In practice though, end-users are usually interested in the top-k friend recommendations, that is producing a ranked list of potential friends [9]–[12].

Instead of performing friend/link prediction, friend recommendation strategies are roughly divided into random walk-based models [13], [14] and collaborative filtering strategies [15], [16]. Random walk-based models perform random walks on the social graphs and rank users based on the computed scores. Collaborative filtering strategies are further divided into point wise and pairwise learning to rank strategies. Representative point wise models are matrix factorization strategies, such as SVD and non-negative matrix factorization. Pointwise models try to approximate the values for an observed friendship as close as they can based on a point wise loss function, that is a reconstruction error when factorizing the adjacency matrix with the social relationships.

However, matrix factorization strategies fail to deal with the severe data imbalance issue due to the sparseness of user data, as the amount of observed relationships is much smaller than the unobserved ones. This leads to models biased towards making low friendship probability [15]. Instead, pairwise learning to rank strategies focus on the ranking performance directly, by considering the relative order of friends in the ranked list. The Bayesian personalized ranking (BPR) model is widely used in the top-k recommendation task [17]. The pairwise ranking criterion of the BPR model is based on the assumption that a user prefers the observed social relationships over the unobserved ones. This idea results in a pairwise ranking loss function that tries to discriminate between a small set of observed relationships and a very large set of unobserved ones. Due to such imbalance between the user's observed relationships and unobserved ones, the BPR model uniformly samples negative examples from the set of unobserved relationships to reduce the training time. To produce top-k friend recommendations in LBSNs, both BPR-based [11] and random walk-based models [9], [12] have been proposed. However, these studies do not account for the fact that the contextual information of users' check-in behavior and mobility patterns are non-linearly correlated with users' social relationships. Recently, deep learning strategies have proved to be an effective means for capturing non-linear representations for deep network embeddings [18], [19] and Point-of-Interest (POI) recommendations [20].

Nonetheless, these studies do not focus on the top-k friend recommendation task. Contribution. To overcome the limitations of existing methods in this paper we introduce a Friend-based Deep Pairwise Learning strategy (FDPL), making the following contributions: (i) we propose a multi-view non-negative factorization strategy to capture the influence of the contextual information of users' co-occurrences and location-based user similarities on social relationships, generating users' low dimensional latent embeddings. (ii) We learn the non-linear representations of the latent embeddings with a deep learning strategy formulating the top-k friend recommendation task as a deep pairwise learning to rank task based on the BPR framework. Our experiments on three benchmark datasets show the superiority of our proposed FDPL model over several baseline models.

The remainder of the paper is organized as follows, Section II reviews the related study and in Section III we detail the proposed FDPL model. Finally, Section IV presents the experimental results and Section V concludes the study.

II. RELATED WORK

A. Friend/Link Prediction in LBSNs Brown et al. [6] study the influence of spatial proximity on users' social relationships, that is the correlation of the geographical closeness of users' check-ins and social relationships. Scellato et al. [1] study the link prediction space of two users to check whether they will become friends or not, and use a semi-supervised framework to predict new links among friends-of-friends and place-friends. Pham et al. [2] introduce an entropy-based model to first measure the diversity of co-occurrences and then utilize location entropy to weigh each co-occurrence differently, depending on the popularity of each location. Cheng et al. [7] formulate the friend prediction problem as a binary classification problem. They introduce two models, the first model focuses on predicting friend- ship of two individuals with only one of their co-occurred places' information. The second model proposes a solution for predicting friendship of two individuals based on all their cooccurred places. Bayrak et al. [8] study the influence of place categories on friend prediction, assuming that friends might visit locations that belong to the same type of location categories e.g., museums, cinemas and so on.

Instead of the top-k friend recommendation task, all the aforementioned methods focus on the link prediction task in LBSNs, that is a binary classification task trying to determine whether two users might become friends or not. Recently, there has been a surge of interest in representation learning for Social Media Networks. For example, DeepWalk [18] learns representations of nodes using a sequence of truncated random walks. The learned representations capture a linear combination of community membership at multiple scales. Gover et al. [19] introduce the node2vec model, to learn a mapping of nodes to a low-dimensional space of features that maximizes the likelihood of preserving network neighborhoods of nodes.

They define a flexible notion of a node's network neighborhood and design a biased random walk procedure, which efficiently explores diverse neighborhoods. However, both DeepWalk and node2vec try to learn non-linear representations of Social Media Networks for the multi-label network classification and link predictions tasks and do not produce top-k friend recommendations. B. Top-k Friend Recommendation in LBSNs The BPR model is a baseline model that considers a pairwise ranking loss function to generate top-k recommendations [17]. Ding et al. [15] extend the BPR model by first extracting deep features based on a convolutional neural network, and then using a deep neural network to produce friend recommendations. However, the available contextual information at LBSNs is ignored in both studies. Yu et al. [9] introduce a random walk process to find geographically related friends. Raw GPS data are analyzed to extract discriminate GPS patterns.

Then, the extracted geographical information and the social network of friends are combined in a heterogeneous information network, performing random walks to provide friend recommendations. Yu et al. [10] infer social relations based on users' preferences on POIs' categories, by evaluating the degree to the preference coverage. Lu et al. [11] present the GIB-FR model, a Bayesian latent model that combines geographical information and user behaviour for friend recommendation. In particular, they investigate whether users who share common areas when they participate in social events will have a tendency to associate with each other. Finally, they formulate the recommendation task as a pairwise ranking problem, using the BPR framework. Bagci et al.

[12] build a graph based on users' context, that is social relation, personal preferences and current location. To rank the recommendation scores of friends, a random walk with restart approach is employed. User's visited locations in recommendation region are also considered to identify places related to friends, that is potential friends that have check-ins at common locations. In addition, local experts and popular locations are employed in populating the context of the user. To identify the local experts and popular locations in a certain region, a HITS-based algorithm is introduced.

C. Social-based Point-of-Interest Recommendation Social-based recommendations account for the fact that people tend to rely more on recommendations from friends than on recommendations of anonymous people similar to them [21]. The challenge of social-based recommendations is to learn the influence of friends' selections on users' preferences, as friends do not share necessarily the same preferences [22]. Representative social-based models are Social Bayesian Personalized Ranking (SBPR) [23] and social matrix + u u xy + factorization [24]. In a similar spirit, in POI recommendation, Location-based user similarity matrix (L). Let l ∗ and l ∗ x y Ye et al. [25] also consider users' social correlation for POI recommendation, following a friend-based collaborative filtering strategy. In particular, they produce POI recommendations based on similar friends, where the similarity between friends is calculated based on their common check-in POIs and common friends.

In [26], a friend-based collaborative filtering be the locations in which users x and y are more active, that is the locations where they check-in most often. To calculate the location-based user similarity matrix, we account for the fact that when users are often active in common regions they might become friends. Given the geographical coordinates (lat,long), we first compute the angular distance δ(l ∗,l ∗) x y strategy is also used to leverage friends' check-ins, where the between locations l ∗ and l ∗ based on the Haversine formula2 . x y similarity between friends is computed based on the distance of their home locations. Each element of the location-based user similarity matrix L ∈ R n×n is computed as follows:

III. THE PROPOSED FDPL MODEL

1 L(x, y) = ∗ ∗ (3)

A. Input Let N and L be the sets of users and locations, where n = |N| and m = |L| are the numbers of users and locations, respectively. Users' check-in data are tuples in the form of (user, location, time). Each user u has a set of friends Au, and each location is associated with a pair of geographical latitude and longitude coordinates in the form of (lat, long). In the following we present the three input matrices of the proposed FDPL model, that is (i) the social adjacency matrix A, (ii) the co-occurrence matrix C and (iii) the location-based user similarity matrix L. Social adjacency matrix (A). In our evaluation datasets users' social relationships are undirected and unweighted (Section IV-A). According to each user u's social relationships in Au, we compute a binary adjacency matrix A ∈ {0, 1} n×n . Co-occurrence matrix (C). Let Lx,y ⊆ L be the set of common locations that user x and y have visited, with x, y ∈ N. First we compute the frequencies of visits that users x and y have checked-in at each common location l ∈ Lx,y separately, denoted by f (l) and f (l) . Let c (l) be the number x y xy 1 + (δ(lx , ly) × r) where r is the earth radius. As some locations l ∗ and l ∗ x y might have long distance, thus being uncorrelated, in our implementation we filter out locations with distance more than 200 km and set L(x, y) = 0. B. The Pairwise Ranking Problem Having computed matrices A, C and L, the goal of our model is to generate top-k friend recommendations for a user u ∈ N . In our FDPL model we formulate the friend recommendation problem as a pairwise ranking task [17].

We define a friendship probability xui, where xui = A(u, i) denotes that users u and i are friends. Thus, we can define two disjoint sets, a set A + of observed relationships of user u, and a set A − of unobserved relationships. For the task of friend recommendation, we build a pairwise ranking model that is able to rank the observed friends before the unobserved ones. For any pair of friends i and j, with i ∈ A + and j ∈ A −, u u the friendship probability xui should be greater than xuj. To describe this relation we define a partial relation i >u j. For each useru∈ N the set of all partial elationshipsis computed of co-occurrences of users x and y at a common location l. Provided that check-ins with high time interval might be weakly correlated or not correlated at all, if any co-occurrences as follows: Ru = {i >u j|i ∈ A + −

happen in a long interval e.g., more than a month, then these co-occurrences are omitted when computing c (l) . Each element u ,j ∈ Au } (4) We define our friend recommendation task as the following of the co-occurrence matrix C ∈ R n×n follows: (l) xy is calculated as ranking problem: Definition (Problem). "Given the set of all partial relationships Ru for each user u ∈ N, the goal of FDPL is to C(x,y) = wl× (l) (l) (1) maximize the ranking likelihood probability as follows:" l∈Lx,y min{fx ,fy } max P(i >u j) (5) where wl is the weight of each location l expressing the importance of co-occurrences at l. To lower the chance of two users being at the same location by coincidence when C. Model Overview u∈N (i,j)∈Ru computing the weight wl we consider the location's popularity pl which is the check-in frequency of all users at location l. c Thus, the weight wl is calculated as follows: z∈Lx,y pz An overview of the proposed FDPL model is presented in Figure 1. The inputs are the social adjacency matrix A, the co-occurrence matrix C and the location-based user similarity matrix L (Section III-A). Following a multi-view non-negative matrix factorization strategy, first the goal is to wl = l (2) jointly learn the influence of co-occurrences and locationbased user similarities on social relationships and compute a According to

Eq. (2) popular locations with high values of pl are downweighted, expressed by low weights wl when computing matrix C(x, y) in

Eq. (1). user latent matrix U ∈ Rn×d of the social adjacency matrix A, 2https://en.wikipedia.org/wiki/Haversine formula p u F F F we formulate the multi-view joint factorization problem as the following joint loss function: minL = LA + λCLC + λLLL (6) Θe where the three loss functions LA, LC and LL correspond to the joint factorizations of the input matrices A, C, and L. Θe is the parameter set of the joint loss function L, and parameters λC and λL regularize the respective loss functions in Eq. (6). Note that in Eq. (6) a regularization parameter for LA is omitted, as matrix A is the main adjacency matrix with users' relationships in the friend recommendation task.

The problem of the joint loss function in Eq. (6) is similar to the Multi-View Non-negative Matrix Factorization (Multi-NMF) problem of [27]. Multi-NMF tries to bring the latent matrices of different views/matrices as close as possible to a common consensus matrix. As the three input matrices are symmetric and coupled at the user dimension, we have a consensus matrix U∗ ∈ Rn×d, with d being the low-dimensional latent Fig. 1. Overview of FDPL. For each user u, friends i ∈ A +, and j ∈ embeddings. While jointly factorizing the three input matrices, the goal of Multi-NMF is to minimize the three reconstruction − Au denote an observed and an unobserved relationship, respectively. The dimensions of the bottom layer H0 are equal to the d dimensional embeddings errors ||U(v) − U∗||2 of the consensus matrix U∗ and the of multi-view NMF. respective latent matrices U(v) ∈ Rn×d , with v = 1,... ,3 in our setting. We calculate the loss functions LA, LC and LL of Eq. (6) as follows: with d being the low dimensional embeddings. Then, in our • LA = ||A−UU⊤||2 +γA||U−U ∗||2 , whereU ∈Rn×d is architecture presented in Figure 1, we perform Deep Pairwise Learning to generate friend recommendations. As defined in our pairwise ranking task in Eq. (5), for each user u we have both the left and right user latent matrix, when factorizing the symmetric matrix A. The second term of LA denotes the reconstruction error of U and the consensus matrix pairs of partial relations (i,j) ∈ Ru. We consider the user U∗ ∈Rn×d ⊤ 2 ∗ 2 latent vectors Ui ∈ Rd , Uu ∈ Rd and Uj ∈ Rd , that is the • LC = ||C − UCUC ||F + γC||UC − U ||F . Since C is i-th, u-th and j-th rows of U. Then, we design three neural networks, where each latent vector Ui, Uu and Uj is provided to the respective neural network.

Given h hidden layers, we symmetric, the left and right user latent matrices UC ∈ Rn×d are equal. • LL = ||L − ULU ⊤ 2 ∗ 2 L ||F + γL||UL − U ||F. The locationfirst try to capture the non-linear representations H (q) , H (q) based user similarity matrix L is also symmetric, hence i u Rn×d and H (q) of Ui, Uu and Uj in each neural network separately, we keep only the user latent matrix UL ∈ . j with q = 1,... ,h. We calculate the friendship probabilities xui and xuj by combining the last hidden layers H (h) , H (h)

The regularization parameters γA, γC and γL control the reconstruction errors of the respective user latent matrices i u ∗ and H (h) with a sigmoid function σ(x) = 1/(1+e −x). Finally, of each loss function and the user consensus matrix U . To j in the output layer we predict the probability of the partial relation P (i >u j). In the remainder of the Section, we present a multi-view non-negative matrix factorization strategy in Section III-D, and a Deep Pairwise Learning strategy in Section III-E. D. Users' Low Dimensional Latent Embeddings To learn the influence of users' co-occurrences and locationbased user similarities on social

International Journal of Science, Engineering and Technology

An Open Access Journal

relationships, we formulate a multi-view joint factorization problem. In particular, by factorizing the social adjacency matrix A we try to capture the friends-of-friends relationships. At the same time, by factorizing the co-occurrence matrix C and the location-based similarity matrix L we try to learn the users' associations based on their co-occurrences at common locations and their activities in geographical close regions, respectively. Hence, reduce the complexity of our model, in our implementation we set the regularization parameters γA, γC and γL to 0.01. Summarizing, the parametersetΘe ofthe joint loss function L in Eq. (6) is set to Θe = {U, UC , UL, U ∗}. However, the minimization problem of Eq. (6) is not convex with respect to all the variables of the parameter set Θe.

To solve this problem, we follow an alternating optimization strategy, that is update one variable while fixing the remaining variables of Θe. According to the learning strategy of multiplicative rules of [27], we compute the update rules of each variable for the alternating optimization algorithm. Due to lack of space we omit here the computations of the update rules, as they can be computed in a similar way as in [27]. By solving the minimization problem of Eq. (6), we compute the user latent matrix U of the social adjacency matrix A, by also accounting for the auxiliary information of users' co-occurrences and activities in geographical close regions. u u u E.

Deep Pairwise Learning Next, in our architecture of Figure 1 we adopt a Deep Pairwise Learning to rank technique to produce top-k friend recommendations. Having computed the user latent matrix U with the low d-dimensional embeddings, for each user u ∈ N we consider the partial relations (i,j) ∈ Ru based on Eq. (4). Then, the low d-dimensional embeddings, that is the latent vectors Ui, Uu and Uj, are provided to the respective three Output. At the output layer of Figure 1, we use the hidden representations and the biases of the last hidden layers, that is the h-th layers of the three neural networks, which are then combined to compute the friendship probabilities xui and xuj (Section III-B). At the output layer we use the sigmoid function σ to ensure that the friendship probabilities xui and xuj are in the range of [0, 1]. The friendship probabilities xui and xuj are calculated as follows: neural networks, as shown in Figure 1. (h) ⊤ H (h) (h) h) Hidden layers. When training the FDPL model we aim to xui = σ(Hi u + bi + b ( ) ⊤ (9) maximize the likelihood in Eq. (5), hence the loss function of xuj = σ(H (h) H(h) (h) FDPL becomes: j u + bj + b (h)) minL = − Θb P(i >u j) + λ||Θb||2 (7) Given that xui and xuj ∈ [0,1], at the output layer the partial relation between xui and xuj is computed as P(i >u j) = (xui−xuj)/2+0.5.Then, based on the computed probability u∈N (i,j)∈Ru Θb is the parameter set, with Θb = P(i >u j), the prediction of an unobserved relationship (u,i) is calculated by forwarding its low d-dimensional embedding {W(q) (q) (q) (q) (q) (q) i ,Wu ,Wj ,bi ,bu ,bj }, ∀q = 1,... , h, where h is the number of hidden layers used in the three neural Ui on the respective neural network as shown in

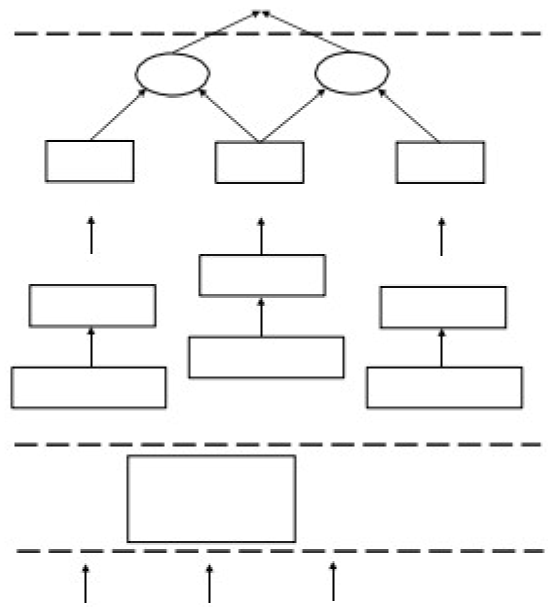

Fig. 1. Overview of FDPL. For each user u, friends i ∈ A +, and j ∈

Figure 1 and then computing the friendship probability xui. The final top-k networks of Figure 1. Matrices W (q) , W (q) and W (q) are friend recommendations are generated by ranking the unobi u j the weighting matrices of the q-th hidden layers to produce the deep learning representations of the latent vectors Ui, Uu served social relationships based on the probability P(i >u j). Model training. In our implementation we used Tensorand Uj. Variables b (q) b (q) ,b (q) denote the respective biases flow4 .

We computed the model parameters Θb via backpropi u j of the q-th hidden layers of each neural network. As the size of hidden layers is important, in our architecture the bottom layer is the widest and each successive layer has a smaller number of hidden units. This way it learns more abstractive features of the d-dimensional embeddings and consequently better captures the non-linear correlations of users' social relationships with the auxiliary information of users' co-occurrences and activities in geographically close regions. For each neural network we implement the tower structure, halving the layer size for each successive layer. Hence, to implement the tower architecture we add the constraint of 2 h ≤ d for the number of hidden layers h and the low d-dimensional embeddings of Multi-NMF. For the hidden layers there are several choices of activation functions, like sigmoid, hyperbolic tangent tanh(x) and rectifier linear unit function ReLU (x). In our implementation, we used ReLU activation functions, with ReLU(x) = max(0, x), as they are non-saturated3 , well-suited for sparse data and making the model less likely to be overfitting [28]. Using ReLU activation functions, ∀q = 1,... ,h, the q-th hidden layers of the three neural networks produce the respective representations as follows: agation with stochastic gradient descent. In particular, we employed mini-batch Adam [29], which adapts the learning rate for each parameter by performing smaller updates for frequent and larger updates for infrequent parameters.

In each backpropagation iteration we performed negative sampling to randomly select a subset of unobserved social relationships as negative instances j ∈ A −. In our implementation we sampled five negative samples for each positive/observed sample, and set the batch size of mini-batch Adam to 512 with a learning rate of 1e-4. Finally, to account for the fact that the initialization of the model parameters Θb plays an important role for the convergence and performance of our model, we followed a pretraining strategy [30]. By applying single-view factorization of A and producing the respective latent matrix U, we first trained our model only using the social relationships in A with random initializations until convergence - ignoring the auxiliary information in matrices C and L. Then, we used the trained parameters as the initialization of our model with the auxiliary information of users' co-occurrences and location based user similarities.

IV. EXPERIMENTS

Evaluation Setup H (q) (q) (q−1) (q−1) i = ReLU(Wi Hi +bi) Datasets. In our experiments, we use three publicly avail- H(q) (q) (q−1) (q−1) able datasets from Brightkite5 , Gowalla6 and Foursquare7 . The u = ReLU(Wu Hu +bu) (8) Brightkite dataset was collected during April 2008-October H (q) (q) (q−1) (q−1) j = ReLU(Wj Hj +bj) 2010 and consists of 58,228 users, 214,078 social relations, 4,491,144 check-ins and 772,788 locations. The Gowalla with H (0) = Ui, H (0) = U and H (0) = Uj. i u u j 3The saturation problems occurs when neurons stop learning and their output is near either 0 or 1, a problem that can be suffered by the sigmoid and tanh functions [28]. 4http://www.tensorflow.org 5http://snap.stanford.edu/data/loc-brightkite.html 6http://snap.stanford.edu/data/loc-gowalla.html 7http://www.public.asu.edu/~hgao16/Publications.html

REFERENCES

[1] S. Scellato, A. Noulas, and C. Mascolo, "Exploiting place features in link prediction on location-based Social Media Networks," in Proc. of ACM KDD, 2011, pp. 1046–1054.

[2] H. Pham, C. Shahabi, and Y. Liu, "Inferring social strength from spatiotemporal data," ACM Trans. Database Syst., vol. 41 , no. 1, pp. 7:1–7:47, 2016.

[3] H. Yin, B. Cui, L. Chen, Z. Hu, and Z. Huang, "A temporal context- aware model for user behavior modeling in social media systems," in Proc. of ACM SIGMOD, 2014, pp. 1543–1554.

[4] H. Yin, Y. Sun, B. Cui, Z. Hu, and L. Chen, "LCARS: a location- content-aware recommender system," in Proc. of ACM KDD, 2013, pp. 221–229.

[5] G. S. Njoo, M. Kao, K. Hsu, and W. Peng, "Exploring check-in data to infer social ties in location based Social Media Networks," in Proc. of PAKDD, 2017, pp. 460–471.

[6] C. Brown, V. Nicosia, S. Scellato, A. Noulas, and C. Mascolo, "Where online friends meet: Social communities in location-based networks," in Proc. of AAAI ICWSM, 2012.

[7] R. Cheng, J. Pang, and Y. Zhang, "Inferring friendship from check- in data of location-based Social Media Networks," in Proc. of IEEE/ACM ASONAM, 2015, pp. 1284–1291.

[8] A. E. Bayrak and F. Polat, "Examining place categories for link prediction in location based Social Media Networks," in Proc. of IEEE/ACM ASONAM, 2016, pp. 976–979.

[9] X. Yu, A. Pan, L. A. Tang, Z. Li, and J. Han, "Geo-friends recommenda- tion in gps-based cyber-physical social network," in Proc. of IEEE/ACM ASONAM, 2011, pp. 361–368.

[10] F. Yu, N. Che, Z. Li, K. Li, and S. Jiang, "Friend recommendation considering preference coverage in location-based Social Media Networks," in Proc. of PAKDD, 2017, pp. 91–105.

[11] Y. Lu, Z. Qiao, C. Zhou, Y. Hu, and L. Guo, "Location-aware friend recommendation in event-based Social Media Networks: A bayesian latent factor approach," in Proc. of ACM CIKM, 2016, pp. 1957–1960.

[12] H. Bagci and P. Karagoz, "Context-aware friend recommendation for location based Social Media Networks using random walk," in Proc. of ACM WWW, Companion Volume, 2016, pp. 531–536.

[13] Z. Yin, M. Gupta, T. Weninger, and J. Han, "A unified framework for link recommendation using random walks," in Proc. of IEEE/ACM ASONAM, 2010, pp. 152–159.

[14] L. Backstrom and J. Leskovec, "Supervised random walks: predicting and recommending links in Social Media Networks," in Proc. of ACM WSDM, 2011, pp. 635–644.

[15] A. K. Menon and C. Elkan, "Link prediction via matrix factorization," in Proc. of ECML PKDD, 2011, pp. 437–452.

[16] D. Ding, M. Zhang, S. Li, J. Tang, X. Chen, and Z. Zhou, "Baydnn: Friend recommendation with bayesian personalized ranking deep neural network," in Proc. of ACM CIKM, 2017, pp. 1479–1488.

[17] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: bayesian personalized ranking from implicit feedback," in Proc. of UAI, 2009, pp. 452–461.

[18] B. Perozzi, R. Al-Rfou, and S. Skiena, "Deepwalk: online learning of social representations," in Proc. of ACM KDD, 2014, pp. 701–710.

[19] A. Grover and J. Leskovec, "node2vec: Scalable feature learning for networks," in Proc. of ACM KDD, 2016, pp. 855–864.

[20] C. Yang, L. Bai, C. Zhang, Q. Yuan, and J. Han, "Bridging collaborative filtering and semi-supervised learning: A neural approach for POI recommendation," in Proc. of ACM KDD, 2017, pp. 1245–1254.

[21] D. Rafailidis and F. Crestani, "Collaborative ranking with social rela- tionships for top-n recommendations," in Proc. of ACM SIGIR, 2016, pp. 785–788.

[22] ——, "Joint collaborative ranking with social relationships in top-n recommendation," in Proc. of ACM CIKM, 2016, pp. 1393–1402.

[23] T. Zhao, J. McAuley, and I. King, "Leveraging social connections to improve personalized ranking for collaborative filtering," in Proc. of ACM CIKM, 2014, pp. 261–270.

[24] M. Jamali and M. Ester, "A matrix factorization technique with trust propagation for recommendation in Social Media Networks," in Proc. of ACM RecSys, 2010, pp. 135–142.

[25] M. Ye, P. Yin, W. Lee, and D. L. Lee, "Exploiting geographical influence for collaborative point-of-interest recommendation," in Proc. of ACM SIGIR, 2011, pp. 325–334.

[26] J. Zhang and C. Chow, "igslr: personalized geo-social location recom- mendation: a kernel density estimation approach," in Proc. of ACM SIGSPATIAL, 2013, pp. 324–333.

[27] J. Gao, J. Han, J. Liu, and C. Wang, "Multi-view clustering via joint nonnegative matrix factorization," in Proc. of SIAM SDM, 2013, pp. 252–260.

[28] L. Xu, C. Choy, and Y. Li, "Deep sparse rectifier neural networks for speech denoising," in Proc. of IEEE IWAENC, 2016, pp. 1–5.

[29] D. P. Kingma and L. J. Ba, "Adam: A method for stochastic optimiza- tion," in Proc of ICLR, 2015.

[30] X. He, L. Liao, H. Zhang, L. Nie, X. Hu, and T. Chua, "Neural collaborative filtering," in Proc. of ACM WWW, 2017, pp. 173–182.

[31] H. Li, Y. Ge, R. Hong, and H. Zhu, "Point-of-interest recommendations: Learning potential check-ins from friends," in Proc. of ACM KDD, 2016, pp. 975–984.

[32] D. Rafailidis and F. Crestani, "A collaborative ranking model for cross- domain recommendations," in Proc. of ACM CIKM, 2017, pp. 2263– 2266