

A Deep Recurrent Collaborative Filtering Framework for Venue Recommendation

Jarana Manotumruksa
University of Glasgow
Glasgow, Scotland, UK
j.manotumruksa.1@research.gla.ac.uk

Craig Macdonald, Iadh Ounis
University of Glasgow
Glasgow, Scotland, UK
first.lastname@glasgow.ac.uk

ABSTRACT

Venue recommendation is an important application for Location-Based Social Networks (LBSNs), such as Yelp, and has been extensively studied in recent years. Matrix Factorisation (MF) is a popular Collaborative Filtering (CF) technique that can suggest relevant venues to users based on an assumption that similar users are likely to visit similar venues. In recent years, deep neural networks have been successfully applied to tasks such as speech recognition, computer vision and natural language processing. Building upon this momentum, various approaches for recommendation have been proposed in the literature to enhance the effectiveness of MF-based approaches by exploiting neural network models such as: word embeddings to incorporate auxiliary information (e.g. textual content of comments); and Recurrent Neural Networks (RNN) to capture sequential properties of observed user-venue interactions. However, such approaches rely on the traditional inner product of the latent factors of users and venues to capture the concept of collaborative filtering, which may not be sufficient to capture the complex structure of user-venue interactions. In this paper, we propose a Deep Recurrent Collaborative Filtering framework (DRCF) with a pairwise ranking function that aims to capture user-venue interactions in a CF manner from sequences of observed feedback by leveraging Multi-Layer Perception and Recurrent Neural Network architectures. Our proposed framework consists of two components: namely Generalised Recurrent Matrix Factorisation (GRMF) and Multi-Level Recurrent Perceptron (MLRP) models. In particular, GRMF and MLRP learn to model complex structures of user-venue interactions using element-wise and dot products as well as the concatenation of latent factors. In addition, we propose a novel sequence-based negative sampling approach that accounts for the sequential properties of observed feedback and geographical location of venues to enhance the quality of venue suggestions, as well as alleviate the cold-start users problem. Experiments on three large checkin and rating datasets show the effectiveness of our proposed framework by outperforming various state-of-the-art approaches.

1 INTRODUCTION

Location-Based Social Networks (LBSNs) such as Foursquare and Yelp have become popular platforms that allow users to find interesting venues to visit based on their preferences, share their

location to their friends (i.e. making a checkin) as well as leave comments on venues they have visited. Such implicit and explicit feedback can be leveraged to study user's behaviour in urban cities as well as to make personalised recommendation of venues to visit. Among various collaborative filtering techniques, Matrix Factorisation (MF) [12] is widely used to predict users' ratings on venues by leveraging explicit feedback (e.g. prior ratings of venues). Then, the venue recommendations are generated by ranking the venues based on their predicted ratings. However, users in LBSNs only focus on the top-K ranked venues for recommendation, and *explicit* feedback can be difficult to collect. For these reasons, Bayesian Personalisation Ranking (BPR) [20], which aims to generate accurate ranked lists of venues, and leverages *implicit* feedback, is more effective than rating prediction-based models (i.e. regression models).

A challenge of implicit feedback from observing checkins is that only positive feedback can be observed, and MF-based and BPR models trained on only positive feedback are likely to be biased to positive instances. To address this challenge, various negative sampling approaches have been proposed [7, 20, 29]. For example, a negative sampling approach proposed by Rendle *et al.* [20] in BPR uniformly and randomly selects items/venues that the users have not interacted with as negative instances. Moreover, users' preferences are not *static* and change dynamically over time (e.g. users may prefer to visit shopping malls at daytime but prefer to visit bars in evenings) [11]. However, both MF and BPR can only capture users' long-term (*static*) preferences and not their short-term (*dynamic*) preferences. Meanwhile, recent observed feedback can have more influence on users' likely checkin behaviour than historical feedback. For instance, consider a user who has recently visited several art museums and a restaurant, sequentially. Models that only capture the user's long-term preferences will recommend other museums to visit, whereas a model that can capture the user's short-term preferences might recommend a bar to visit instead.

Another challenge in recommendation systems is the problem of cold-start users (i.e. users who have little observed feedback), which can hinder the effectiveness of MF-based approaches. To alleviate the cold-start user problem, various MF-based approaches in literature have been proposed to leverage additional sources of information such as friendships, textual content of comments and geographical location of venues (e.g. [16–18, 29]). Previous studies [4, 11, 19, 25, 28, 32] have shown that the sequential properties of user's interactions (e.g. sequences of checkins or clicks) play an important role in improving the quality of recommendation for various tasks. To effectively capture users' *dynamic* preferences as well as take previous feedback into account, various approaches have been proposed to exploit Recurrent Neural Network models (RNN) for recommendation systems (e.g. [25, 28, 32]). Such approaches still rely on a dot product of latent factors of users and items to capture the user-item interactions in a Collaborative Filtering manner,

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than the author(s) must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'17, Singapore, Singapore

© 2017 Copyright held by the owner/author(s). Publication rights licensed to ACM. 978-1-4503-4918-5/17/11...\$15.00

DOI: 10.1145/3132847.3133036

where similar users will obtain similar recommendations. However, the dot product of latent factors may not be sufficient to capture the complex structures of user-item interactions. To address this challenge, He *et al.* [7] recently proposed a Neural Matrix Factorisation (NeuMF) framework that leverages a Multi-Layer Perceptron to learn the complex structures of user-item interactions. NeuMF can generalise MF-based approaches by replacing the dot product with a neural architecture that can learn an arbitrary function from implicit feedback. In this work, we propose a Deep Recurrent Collaborative Filtering framework that aims to learn the complex structure of users’ *static* and *dynamic* preferences. In particular, our contributions are summarised below:

- We propose a Deep Recurrent Collaborative Filtering framework (DRCF) with a pairwise ranking function for venue recommendation. Our proposed framework differs from the framework of He *et al.* [7] (NeuMF) in three aspects: (1) DRCF can capture both user’s *dynamic* and *static* preferences, while NeuMF can only capture user’s *static* preferences; (2) NeuMF ignores the dot product of latent factors when making the prediction. In contrast, DRCF framework exploits the dot products of latent factors to model the user-venue interactions; and (3) DRCF aims to generate accurate ranked list of venue, unlike NeuMF which aims to accurately predict user’s rating on items. To the best of our knowledge, our proposed framework (DRCF) is the first study that extends NeuMF framework to model both users’ *static* and *dynamic* preferences.
- Within the DRCF framework, we propose Generalised Recurrent Matrix Factorisation (GRMF) and Recurrent Multi-Level Recurrent Perceptron (MLRP) models that exploit Recurrent Neural Networks (RNN) models to capture users’ *dynamic* preferences. In contrast to existing RNN-based approaches [25, 28, 32] for recommendation systems, GRMF and MLRP models do not only rely on the dot products of latent factors but also element-wise product and concatenation of latent factors, which are weighted using the Deep Neural Networks architecture.
- We propose a novel sequence-based (*dynamic*) negative sampling approach that takes the sequential properties of checkins and geographical location of venues into account to enhance the effectiveness of the DRCF framework, as well as alleviate the cold-start user problem.
- We conduct comprehensive experiments on 3 large-scale real-world datasets from Foursquare, Yelp and Brightkite to demonstrate the recommendation accuracy of DRCF. The experimental results demonstrate that DRCF consistently and significantly outperforms various state-of-the-art venue recommendation approaches across three datasets.

The rest of this paper is organised as follows. We review related literature on venue recommendation in Section 2. Then, we briefly provide the problem statement, the MF-based and RNN-based approaches and the Neural Matrix Factorisation framework (NeuMF) in Section 3. Our proposed DRCF framework and its components are described in Section 4. The experimental setup for our experiments is detailed in Section 5, while comprehensive experimental results comparing the effectiveness of DRCF with various state-of-the-art approaches are reported in Section 6. Concluding remarks follow in Section 7.

2 RELATED WORK

Matrix Factorisation, proposed by Koren *et al.* [12], is a widely used collaborative filtering technique in recommendation systems. Traditional MF-based approaches leverage explicit and implicit feedback (e.g. rating of venues and checkins) to model user’s preferences and characteristic of venues. In particular, MF aims to find the latent factors of users and venues to accurately predict the rating of the user to unvisited venues. Various MF-based approaches have been proposed in the literature (e.g. [3, 7, 14, 16–18]) to generate personalised venue recommendation to the users by ranking the unvisited venues based on the predicted user-venue ratings. Such approaches can be identified as *pointwise* approaches [15]. However, as users in LBSNs only focus on the top-K suggestions, MF-based approaches that aim to optimise pointwise loss (e.g. Root Mean Square Error) are less effective than approaches that are directly optimised for ranking venues (i.e. focusing on the top-K ranked list of venues that are relevant to user’s preferences) [1, 2, 15]. Rendle *et al.* [20] proposed a *pairwise* optimisation criteria that can be applied with MF-based approaches, named Bayesian Personalised Ranking (BPR), in which an assumption is that for each user, the user’s previously visited venues are preferred over their non-visited ones. Although previous literature [1, 2, 15] have demonstrated that *listwise* approaches can be more effective than *pairwise* and *pointwise* approaches for general information retrieval tasks such as web search, such *listwise* approaches are not feasible in recommendation contexts, because of the need to rank a large set of non-observed item/venues for each user rather than small sets of candidate web documents.

The aforementioned MF-based approaches assume that users’ preferences are *static*, hence users who have visited similar sets of venues in different orders would get similar venue suggestions. Indeed, users’ preferences are not *static* and change dynamically over time [11]. To leverage sequences of implicit feedback (e.g. checkins) in order to model users’ *dynamic* preferences, existing approaches in literature (e.g. [4, 19]) have been proposed based on Markov Chains. However, such Markov Chains-based approaches have a similar deficiency as they can only model local sequential behaviour between each pair of adjacent feedback items [28]. With the success of Deep Neural Network models (DNN) in several domains such as on speech recognition, computer vision and natural language processing (e.g. [6, 9, 30]), various approaches (e.g. [25, 26, 28, 32]) have been proposed to exploit Recurrent Neural Networks (RNN) models to capture users’ *dynamic* preferences from the observed sequences of implicit feedback. For example, Tang *et al.* proposed a MF-based approach that leverages Bidirected Long-Short Term Memory, a recurrent neural network, to predict the rating of user for movie recommendation, while Yu *et al.* [28] extended BPR to incorporate RNN models to recommend a ranked list of items for the user to purchase, given his previous sequence of item baskets.

Apart from the recommendation systems that exploit recurrent-based models to leverage sequential properties of implicit feedback, DNN models that learns user-item interactions have been explored in the literature (e.g. [5, 7, 23]). Cheng *et al.* [5] proposed a Wide & Deep learning approach for mobile application recommendation system that exploits both linear models and DNNs to incorporate various features of users and items. Recently, He *et al.* [7] explored the advantages of DNNs in modelling user-item interactions in the collaborative filtering manner. In particular, they proposed a Neural Matrix Factorisation (NeuMF) framework that exploits Multi-Level Perception architecture to accurately predict the user’s item rating.

	u_1	u_2	u_3	u_4
u_1	1	0.5	0.4	0.6
u_2	0.5	1	0.6	0.2
u_3	0.4	0.6	1	0.4
u_4	0.6	0.2	0.4	1

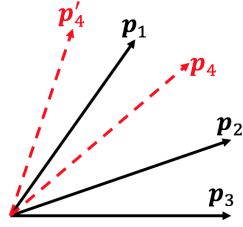


Figure 1: An illustration of the similarity between users and vectors representing the latent factors of each user in user latent space P . NB: Figure regenerated from [7].

3 MATRIX FACTORISATION APPROACHES FOR RECOMMENDATION SYSTEMS

In this section, we first formalise the problem statement as well as the notations used in this paper (Section 3.1). Then, we briefly describe the Matrix Factorisation approaches (Section 3.2), the Neural Collaborative Filtering framework (NeuMF) that exploits a deep neural networks architecture to generalise MF-based approaches (Section 3.3) and MF-based approach that exploits a Recurrent Neural Networks model to captures the *dynamic* preferences of users (Section 3.4). Note that these MF-based approaches and framework were not originally proposed for venue recommendation but are flexible to do so. For simplicity, we explain these approaches and framework in the context of venue recommendation. Finally, Section 3.5 summarises the elicited limitations of these MF-based approaches. Later, in Section 4, we describe our proposed framework that addresses these limitations.

3.1 Problem Statement

The task of venue recommendation is to generate a ranked list of venues $\in \mathcal{V}$ that a user might visit given his/her historical feedback (e.g. previously visited venues from checkin data). Let \mathcal{V}_u^+ denote the list of venues the user has previously visited, sorted by time and \mathcal{S}_u denote the list of sequence of visited venues (e.g. if $\mathcal{V}_u^+ = (v_1, v_2, v_3)$, then $\mathcal{S}_u = ((v_1), (v_1, v_2), (v_1, v_2, v_3))$). $s_t \in \mathcal{S}_u$ denotes the sequence of visited venues of user u at time t (e.g. $s_2 = (v_1, v_2)$). We can represent all checkins by all users as a matrix $C \in \mathbb{R}^{m \times n}$ where m and n are the number of users and venues, respectively. Let $c_{u,i} \in C$ denote a user $u \in \mathcal{U}$ visited venue $i \in \mathcal{V}$. \mathcal{N}_i is a set of neighbouring venues that are nearby venue i . Note that $c_{u,i} = 0$ means user u has neither left a rating nor made a checkin at venue i .

3.2 Matrix Factorisation

Traditional Matrix factorisation-based approaches (MF) [12] assume that users who share similar preferences (e.g. visit similar venues) can influence each other. MF techniques aim to approximate the matrix C by finding a decomposition of C , i.e. a dot product of latent factors of users $P \in \mathbb{R}^{m \times d}$ and venues $Q \in \mathbb{R}^{n \times d}$ where d is the number of latent dimensions, $C \approx P_u^T Q_i$. In particular, the predicted checkin of user u on venue i can be computed as follows:

$$\hat{c}_{u,i} = p_u^T q_i = p_u \odot q_i = \sum_{k=1}^d p_{u,k} \hat{q}_{i,k} \quad (1)$$

where \odot denotes the dot product and p_u and q_i are latent factor of user u and venue i , respectively. Indeed, MF behaves as a linear

model of latent factors by assuming that each dimension of the latent factor is independent and linearly combining those dimensions with the same weight [7]. The objective of MF is to minimise the pointwise loss between the predicted checkin $\hat{c}_{u,i}$ and the observed checkin $c_{u,i}$ and the loss function is defined as follows:

$$L(\Theta) = \min_{\Theta} \frac{1}{2} \sum_{u=1}^m \sum_{i=1}^n I_{i,j} \cdot (c_{u,i} - \hat{c}_{u,i})^2 + \frac{\lambda}{2} \|\Theta\|_F^2 \quad (2)$$

where $I_{i,j}$ is an indicator variable that is 1 if user i makes a checkin at venue j , otherwise 0. To avoid overfitting, a traditional regularisation technique is added into Equation (2), where $\lambda \geq 0$ is a regularisation parameter, $\Theta = \{P, Q\}$ denotes all the parameters to be learnt and $\|\cdot\|_F^2$ denotes the Frobenius norm.

Once the training process of MF is finished, the distance of the latent factors of users who have visited similar venues will be close in the latent factor space (i.e. similar users will get similar venue suggestions). He *et al.* [7] argued that the dot product of latent factors may not be sufficient to capture the complex structures of user-item interactions. Figure 1 illustrates a limitation of the dot product operation explored by He *et al.* [7], which can degrade the effectiveness of MF-based approaches. The table in the figure provides the cosine similarity between the latent factors of pairs of users, e.g. p_1 and p_2 are the latent factors of user u_1 and u_2 , respectively, and the cosine similarity between u_1 and u_2 is 0.5. Let us first consider users u_1, u_2 and u_3 . The vectors in Figure 1 present the geometric relative angles of p_1, p_2 and p_3 in the latent factor space. From the angles of the vectors, we can see that u_1 shares more common preferences with u_2 than u_3 . Next, let us consider u_4 , the highlighted row in the table. In fact, u_4 is most similar to u_1 , followed by u_3 and u_2 . Placing p_4 closest to p_1 (the two possible geometric positions between p_1 and p_4 in latent factor space are presented as red-dashed lines) results p_4 is closer to p_2 than p_3 . This scenario can lead to a large pointwise loss (i.e. u_4 gets similar venue suggestions to u_2 , rather than u_3). Modelling such complex structures of user-venue interactions is challenging.

3.3 Neural Matrix Factorisation (NeuMF)

To address the aforementioned challenge of MF-based approaches that rely on the dot products of latent factors, He *et al.* [7] proposed a Neural Matrix Factorisation Filtering framework (NeuMF) that consists of two components, namely: Generalised Matrix Factorisation (GMF) and Multi-Level Perceptron (MLP) models to capture complex structures of user-venue interactions from their implicit feedback using element-wise product and concatenation of latent factors where the dimensions of the latent factors are treated dependently.

3.3.1 General framework. Figure 2 illustrates the multiple layers of the NeuMF framework; the output of one layer serves as the input of the layer above. The connections between layers of NeuMF framework are presented using red-dotted lines, the layers that are not connected with the red-dotted lines are not part of NeuMF (i.e. the RMF and RNN layers). Starting at the bottom of the figure, the input layer consists of a binary sparse vector with one-hot encoding that represents user v_u^U and venue v_i^I , respectively. The sparse vectors of user and venue are fed into the embedding layer. The outputs of the embedding layer can be seen as the latent factors of user $p_u = P^T v_u^U$ and venue $q_i = Q^T v_i^I$ in the context of factorised model. Next, the latent factors are fed into the Neural Collaborative Filtering layers (i.e. hidden layers) to discover certain

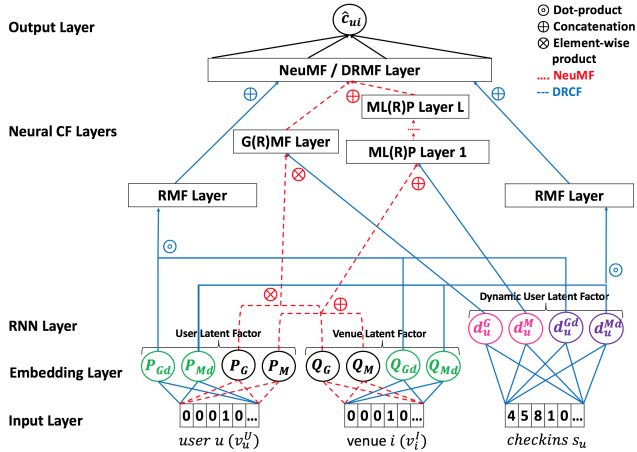


Figure 2: Deep Recurrent Collaborative Filtering Framework. The connections of each layer linked by the red-dotted lines illustrate the NeuMF framework.

latent structures of user-venue interactions. The final output layer provides the predicted checkin $\hat{c}_{u,i}$, which is defined as follows:

$$\hat{c}_{u,i} = a_{out}(h(\phi^{GMF} \oplus \phi^{MLP})) \quad (3)$$

where a_{out} denotes the activation function, \oplus denotes the concatenation of two layers, ϕ^{GMF} and ϕ^{MLP} , and $h(x) = (W^T x + b)$ is the hidden layer – W and b are the weight matrix and bias vector, respectively. Overall, $\theta_h = \{W, b\}$ denotes a set of parameters of the hidden layers. $h(x)$ ensures that each dimension of the latent factors from ϕ^{GMF} and ϕ^{MLP} are dependent. He *et al.* [7] proposed to use the sigmoid function $\sigma(x) = 1/(1 + e^{-x})$ as the activation function a_{out} . The combination of GMF and MLP models enables NeuMF to model user-venue interactions as non-linear latent factor models. In particular, GMF and MLP models capture user-venue interaction using element-wise product and concatenation of latent factors, respectively (further discussed in Section 3.3.2 and Section 3.3.3 below). Similar to MF’s loss function (Equation (2)), the framework aims to minimise the Root Mean Square Error (RMSE) between the predicted checkin $\hat{c}_{u,i}$ and the observed checkin $c_{u,i}$. However, as mentioned in Section 1, users in LBSNs only focus the top-K ranked list of venues for obtaining recommendations, and hence we argue that the training of NeuMF that aims to minimise a regression metric (RMSE) may not provide an effective top-K ranked list of venues (**Limitation 1**). Moreover, to alleviate the challenge of implicit feedback mentioned in Section 1, i.e. lack of negative instances, He *et al.* [7] proposed to apply traditional negative sampling, as defined in BPR, to randomly select unvisited venues as negative instances. However, we argue that more effective negative sampling approaches should take both the sequential properties of observed feedback as well as the geographical location of venues into account (**Limitation 2**).

3.3.2 Generalised Matrix Factorisation model (GMF). As discussed in Section 3.2, He *et al.* [7] argued that the dot-product may not be sufficient to capture the complex structures of user-venue interactions. They proposed a Generalised Matrix Factorisation model (GMF) to address the limitation of MF-based approaches that relies on dot product operation, which is defined as follows:

$$\phi^{GMF} = p_u^G \otimes q_i^G \quad (4)$$

where \otimes denotes the element-wise products of two latent factors and $p_u^G = P_G^T v_u^U$ (see red-dashed lines with \otimes operation in Figure 2) and $q_i^G = Q_G^T v_i^I$ are the latent factors of user u and venue i that are projected from the GMF user and venue embedding layers¹ (P_G and Q_G), resp. (see the black nodes in the embedding layer of Figure 2).

3.3.3 Multi-Level Perceptron model (MLP). Next, He *et al.* [7] proposed a Multi-Layer Perceptron model (MLP) that aims to capture complex structure of user-venue interactions using the concatenation of the latent factors, which is defined as follows:

$$\phi^{MLP} = a_L(h_L(\dots a_1(h_1(p_u^M \oplus q_i^M)))) \quad (5)$$

where L is the number of layers and $p_u^M = P_M^T v_u^U$ and $q_i^M = Q_M^T v_i^I$ are the latent factors of user u and venue i that are projected from MLP user and venue embedding layer, P_M and Q_M , respectively (see black nodes in the embedding layers of Figure 2). $\theta_e = \{P_G, P_M, \dots, Q_M\}$ denotes the set of parameters of the embedding layers. Although the activation function a_L can be a sigmoid, a hyperbolic tangent (tanh) or a Rectified Linear Unit (ReLU), they applied ReLU as the activation function for the hidden layer h_L to alleviate the saturation problem (i.e. neurons stop learning when their output is near either 0 or 1, a problem that can be suffered by the sigmoid and tanh functions). By concatenating p_u^M and q_i^M , MLP is more flexible than the GMF and the factorised models since both the dot-product and element-wise product operations require the dimension d of the latent factors to be identical. However, We argue that both GMF and MLP can only capture the users’ *static* preferences, while previous works [4, 11, 19, 25, 28, 32] have shown that users’ *dynamic* preferences also play an important role in effective recommendation systems (**Limitation 3**). In addition, although GMF and MLP can capture different structures of user-venue interactions by using both element-wise product and concatenation of latent factors, we argue that the NeuMF framework should not ignore the structure of user-venue interactions that can be captured by the dot-product of latent factors (**Limitation 4**).

3.4 Recurrent Neural Networks-based Approaches

Recently, various approaches have been proposed to enhance the effectiveness of MF-based approaches for recommendation systems by exploiting Recurrent Neural Networks (RNN) [28, 32] to leverage sequential properties of observed implicit feedback. For example, Zhang *et al.* [32] proposed a RNN-based approach that models users’ *dynamic* preferences from sequences of clicks as follows:

$$d_{u,t} = \sigma(Xq_{i,t} + Rd_{u,t-1}) \quad (6)$$

where q_i denotes the latent factor of item i the user visited at time t and h_{t-1}^u is the *dynamic* preferences of the user at previous time $t-1$. R is a recurrent connection weight matrix that captures sequential signals between every two adjacent hidden states $d_{u,t-1}$ and $d_{u,t}$ and X is a transition matrix between the latent factors of venues and users. We note that $\theta_r = \{R, X\}$ denotes a set of parameters of RNN layers. For example, if a model consists of two RNN layers, we will have $\theta_r = \{R_1, R_2, X_1, X_2\}$. $\sigma(x)$ is the sigmoid function. Then, similar to MF-based approaches, they apply the dot product to estimate the probability that user u will checkin at venue i given his recently click sequences, i.e. $\hat{c}_{u,i} = d_{u,t}^T q_i$. The approach of Yu *et al.* [28] is similar to that of Zhang *et al.*, but the application of BPR to train their proposed RNN-based model. We argue that

¹ The embedding layer is equivalent to latent factors of MF-based approaches.

there are two limitations that need to be addressed. First, their proposed approach does not take the user’s *static* preferences into account (**Limitation 5**). Although we can apply more sophisticated RNN-based models (e.g. Long-Short Term Memory model) that are capable of dealing with long sequences of observed feedback, such models are computationally expensive. Indeed, a venue that the user has visited a couple of months ago has less impact to user’s preference than a venue recently visited. Hence, an accurate model need to be capable to capture both the *static* and *dynamic* preference of users is more likely to generate better venue recommendation. Second, to model user-venue interactions in a collaborative filtering manner, their approach still relies on the dot products of latent factor of venues q_i and the user’s *dynamic* preference $d_{u,t}$. However, as mentioned above, previous work [7] has shown that the dot product of latent factors may not be sufficient to capture the complex structure of user-venue interactions (**Limitation 6**).

3.5 Summary of Limitations

To conclude, in the above analysis, we have identified four limitations of NeuMF and two limitations of RNN-based models:

Limitation 1: There is an inherent disadvantage in the NeuMF for identifying the top-ranked venues to present to users.

Limitation 2: Negative sampling approaches for which this limitation applies (BPR, DREAM, NeuMF) do not account for the sequential properties of checkins and the geolocation of venues.

Limitation 3: MF-based approaches for which this limitation applies (GMF, MLP, NeuMF) assume that the users’ preferences are *static* and do not account for the sequential properties of checkins.

Limitation 4: MF-based approaches for which this limitation applies (GMF, MLP, NeuMF) ignore the dot product of latent factors that capture user-venue interactions.

Limitation 5: RNN-based approaches (RNN, DREAM) that exploit recurrent models to capture the users’ *dynamic* preference but do not take the users’ *static* preferences into account.

Limitation 6: RNN-based approaches that exploit recurrent models to capture user’s *dynamic* preferences for which this limitation applies (RNN, DREAM) model user-venue interaction using the dot product of latent factors to generate the venue recommendations.

In summary, there is no previous work that exploits RNN to capture both the *dynamic* and *static* preferences of users, and capture user-venue interactions with three operations: namely dot and element-wise products and concatenation. In the next section, we propose a framework that addresses these limitations.

4 DEEP RECURRENT COLLABORATIVE FILTERING FRAMEWORK

In this section, we propose a novel Deep Recurrent Collaborative Filtering framework (DRCF) with a *pairwise* ranking function and a novel sequence-based (*dynamic*) geo-based negative sampling approach for venue recommendation (Section 4.1). The proposed framework consists of three components: namely Generalised Recurrent Matrix Factorisation (GRMF), Recurrent Multi-Level Perceptron (MLRP) and Recurrent Matrix Factorisation (RMF) models, which will be discussed in Section 4.2, Section 4.3 and Section 4.4, respectively. In particular, our overall framework aims to address **Limitations 1, 2 & 4** in Section 4.1, while the GRMF and MLRP models aim to address **Limitations 3** and the RMF model aims to address **Limitations 5 & 6**. Later, in Section 6, we demonstrate

the effectiveness of DRCF framework and its components in comparison with state-of-the-art venue recommendation systems.

4.1 Unified Framework

Our proposed DRCF framework is illustrated in Figure 2. The DRCF framework consists of multiple layers and the connections between layers are presented using both blue-dashed and red-dotted lines. DRCF differs from NeuMF framework in various aspects. Starting at the bottom of the figure, at the input layer, we extend the NeuMF framework to leverage sequences of checkins of each user u , s_u . In embedding layers, there are four additional embedding layers that are highlighted in green in Figure 2: namely GRMF and MLRP user (venue) embedding layers that are used in the RMF model, P_{Gd} (Q_{Gd}) and P_{Md} (Q_{Md}), respectively. Later in Section 4.4, we explain why we need these four additional layers. Recurrent Neural Networks (RNN) layers are included in our framework (pink and purple in Figure 2) to encapsulate *dynamic* user preferences. In the Neural CF layers, we include RMF layers to discover certain latent structures of user-venue interactions. The final output layer is the predicted checkin $\hat{c}_{u,i}$, which is defined as follows:

$$\hat{c}_{u,i} = a_{out}(h(\phi^{GRMF} \oplus \phi^{MLRP} \oplus \phi^{RMF})) \quad (7)$$

where a_{out} is the activation function, h is the hidden layer and ϕ^{GRMF} , ϕ^{MLRP} and ϕ^{RMF} denote the GRMF, MLRP and RMF models that are described in Sections 4.2, 4.3 and 4.4, respectively. To address **Limitation 4**, our proposed DRCF framework seamlessly integrates the RMF model that captures user-venue interactions using the dot-product of latent factors, discussed in Section 4.4.

Next, instead of training the DRCF framework to minimise the *pointwise* loss between predicted checkin $\hat{c}_{u,i}$ and observed checkin $c_{u,i}$, as in Equation (2), we address **Limitation 1** by proposing to apply Bayesian Personalised Ranking (BPR) to learn the parameters $\Theta = \{\theta_r, \theta_e, \theta_h\}$, as follows:

$$\mathcal{J}(\Theta) = \sum_{u \in \mathcal{U}} \sum_{s_t \in \mathcal{S}_u} \sum_{k \in \mathcal{N}_i - s_t} \sum_{j \in \mathcal{V} - s_t} \left[\log(\sigma(\hat{c}_{u,i} - \hat{c}_{u,k})) - \log(\sigma(\hat{c}_{u,k} - \hat{c}_{u,j})) \right] \quad (8)$$

where i is a venue most recently visited in s_t , k is an unvisited venue that is nearby to venue i and \mathcal{N}_i is a set of venues that are nearby to venue i . Note that venues j and k are negative instances that we sample using our proposed *dynamic* geo-based negative sampling approach. This contrasts with the traditional negative sampling approach that is widely applied in previous literature [7, 20], which randomly selects negative instances from a *static* pool of negative venues $\mathcal{V}_u^- = \mathcal{V} - \mathcal{V}_u^+$. Instead, we address **Limitation 2** by proposing a novel *dynamic* geo-based negative sampling approach, DRCF_{geo}, that enhances the effectiveness of the DRCF framework and alleviate the cold-start user problem by taking sequences of checkins s_t at time t and the geographical location of venue i , i.e. its neighbour venues \mathcal{N}_i , into account. In particular, our proposed *dynamic* negative sampling approach (see lines 8-11 in Algorithm 1) samples an unvisited neighbouring venue k and an unvisited distance venue j from a *dynamic* pool of negative venues $\mathcal{N}_i - s_t$ and $\mathcal{V} - s_t$ not visited by the user in the current sequence of checkins s_t , respectively, rather than a *static* pool of negative venues as in the traditional negative sampling approach ($\mathcal{V} - \mathcal{V}_u^+$). Moreover, we also propose a *static* geo-based negative sampling approach, DRCF_{sgo},

Algorithm 1 An Optimisation Algorithm of DRCF

```
1: Input:  $\mathcal{U}, \mathcal{V}$  and sequences of visited venues  $S$ .
2: Output:  $\Theta = \{\theta_r, \theta_e, \theta_h\}$ 
3: initial  $\theta_r, \theta_e, \theta_h$ 
4:  $\mathcal{N} \leftarrow 0$  // iteration number
5: repeat
6:   for  $\mathcal{N} \leftarrow 1$  to  $|\mathcal{U}|$  do
7:      $u \leftarrow$  draw a random user from  $\mathcal{U}$ 
8:     for  $s_t$  in  $S_u$  do
9:        $i \leftarrow$  a venue most recently visited in  $s_t$ 
10:       $k \leftarrow$  draw a random unvisited venue from  $\mathcal{N}_i - s_t$ 
11:       $j \leftarrow$  draw a random unvisited venue from  $\mathcal{V} - s_t$ 
12:      Compute gradients of  $\theta_r, \theta_e, \theta_h$ 
13:      Update the above parameters
14:     end for
15:   end for
16: until convergence
```

that samples an unvisited neighbouring venue k and a distant negative venue j from a *static* pool of negative venues $\mathcal{N}_i - \mathcal{V}_u^+$ and $\mathcal{V} - s_t$ not visited by the user in the current sequence of checkins s_t , respectively. Later in Section 6, we compare the effectiveness of the proposed *dynamic* and *static* geo-based negative sampling approaches, DRCF_{dgeo} and DRCF_{sgeo}, respectively, with DRCF alone, which does not consider any k (neighbouring unvisited venues) during training. The gradients of $\theta_r, \theta_e, \theta_h$ can be estimated by the back propagation through time algorithm proposed by Rumelhart *et al.* [22], which we omit due to the space constraints. The optimisation algorithm of DRCF and the proposed *dynamic* geo-based negative sampling approach are described in Algorithm 1.

4.2 Generalised Recurrent Matrix Factorisation (GRMF)

In this section, we explain how we exploit Recurrent Neural Network models (RNN) to capture users' *dynamic* preferences in the collaborative filtering manner as well as how to integrate the dot products of latent factors into the GMF model. In particular, we aim to address **Limitation 3** by extending the GMF model to leverage sequential properties of checkins. As mentioned in Section 2, users' *dynamic* preferences play an important role to enhance the effectiveness of factorised models [4, 11, 19, 25, 28, 32]. For example, users in the evening are more likely to visit a bar directly after they have visited restaurant. However, such behaviour cannot be captured by the GMF model because it does not take the sequential properties of checkins into account during the training process (**Limitation 3**). We propose GRMF, an extension of the GMF model mentioned in Section 3.3.2, to exploit an RNN model as follows:

$$\phi^{GRMF} = [d_{u,t}^G \otimes p_u^G \otimes q_i^G] \quad (9)$$

where d^G is the user's *dynamic* preferences of user u at time t that are projected from the RNN layer, p_u^G and q_i^G are the latent factors of user u and venue i that are projected from the GRMF embedding layers, respectively. Indeed, the proposed GRMF model is the element-wise product of latent factors (see the connection between red-dashed and blue lines under the GRMF layer with \otimes operation in Figure 2). Note that we do not consider the choice of recurrent

models and RNN settings in this work, which have already been explored in previous literature (e.g. [24, 25]).

4.3 Multi-Level Recurrent Perceptron (MLRP)

As mentioned in Section 3.3.3, we argue that the effectiveness of the MLP model can be enhanced by leveraging the sequential properties of checkins. In particular, similar to previous section, we propose to extend the MLP model to exploit RNN-based model to capture users' *dynamic* preferences to address **Limitation 3** as follows:

$$\phi^{MLRP} = [a_L(h_L(\dots a_1(h_1(d_{u,t}^M \oplus p_u^M \oplus q_i^M))))] \quad (10)$$

where d^M is the user's *dynamic* preferences of user u at time t that is projected from the RNN layer, p_u^M and q_i^M are the latent factor of user u and venue i that are projected from the MLRP Embedding layer, respectively (see the connection between red-dashed and blue lines under the MLRP layer with \oplus operation in Figure 2). Note differences between $p_u^G(q_i^G)(d_{u,t}^G)$ and $p_u^M(q_i^M)(d_{u,t}^M)$ in Equations (9) & (10), we exploit different embedding and RNN layers for each model in order to independently learn the complex structures of both *dynamic* and *static* user-venue interactions from different models (i.e. the GRMF captures the interactions using the element-wise product operation, while the MLRP capture the interactions using the concatenation). Again, the effectiveness of the framework that allows different models to learn from different set of embedding layers have been explore in previous literature [7]. Although our proposed DRCF framework allows different models to learn independently, in the output layer (Equation (7), we exploit a hidden layer h to seamlessly and dependently integrate those models to generate the ranked-list of venues to the users. In the next section, we explain RMF, the last component of the DRCF framework, which incorporates the dot products of latent factors.

4.4 Recurrent Matrix Factorisation (RMF)

In this section, we propose a Recurrent Matrix Factorisation (RMF) model that captures both the users' *static* and *dynamic* preferences using the dot products of latent factors as follows:

$$\phi^{RMF} = [(\underbrace{d_{u,t}^{Gd}}_{dynamic} + \underbrace{p_u^{Gd}}_{static}) \otimes q_i^{Gd}] \oplus [(\underbrace{d_{u,t}^{Md}}_{dynamic} + \underbrace{p_u^{Md}}_{static}) \otimes q_i^{Md}] \quad (11)$$

where $d_{u,t}^{Gd}(d_{u,t}^{Md})$ are the latent factors of the *dynamic* preferences of user u at time t that are projected from the GRMF (MLRP) RNN layer. $p_u^{Gd}(p_u^{Md})$ and $q_i^{Gd}(q_i^{Md})$ are the latent factor of user u and venue i that are projected from the GRMF (MLRP) embedding layers, respectively. Therefore, the RMF model addresses **Limitation 5** by incorporating latent factors, $d_{u,t}^{Gd}(d_{u,t}^{Md})$ and $p_u^{Gd}(p_u^{Md})$, that represent both users' *dynamic* and *static* preferences. Indeed, in order to allow GRMF, MLRP and RMF models to learn independently, we follow He *et al.* [7] to train the RMF models by using different sets of embedding and RNN layers (see green nodes in the embedding layer and purple nodes in the RNN layer of Figure 2). Each of the GRMF, MLRP and RMF models use different operations (i.e. GRMF and MLRP use element-wise product and concatenation, while RMF uses dot product). In doing so the complex structures of user-venue interactions are independently captured by these three models using the different operations.

Moreover, to address **Limitation 6**, unlike previous RNN-based approaches [28, 32] mentioned in Section 3.4, we do not directly

use the dot product of latent factors to generate the venue recommendations. Instead, we concatenate the RMF model with GRMF and MLRP models and exploit a hidden layer h to determine the influence of each model (see Equation (7)). In summary, our proposed DRCF framework that consists of GRMF, MLRP and RMF models can comprehensively capture the complex structures of *dynamic* and *static* user-venue interactions by leveraging the sequential properties of checkins using dot products, element-wise products and concatenation of latent factors. To the best of our knowledge, the proposed DRCF framework is the first that exploits those three operations to capture the *dynamic* and *static* user-venue interactions.

5 EXPERIMENTAL SETUP

In this section, we evaluate the effectiveness of our proposed DRCF framework and its components by comparing with state-of-the-art venue recommendation approaches. In particular, to address **Limitations 1 - 6**, we answer the following research questions:

RQ1 *Can we enhance the effectiveness of the components of the DRCF framework for venue recommendation systems, namely the GRMF and MLRP models, by (a) leveraging the sequential properties of checkins to capture the users' dynamic and static preferences, (b) incorporating the dot product of latent factors into the models and (c) training those models to generate accurate ranked lists of venues for users?*

RQ2 *Are the MF-based models that capture both users' dynamic and static preferences using either the element-wise product or concatenation of latent factors more effective than state-of-the-art RNN-based approaches that only model the users' dynamic preference using a dot product of latent factors?*

Furthermore, as discussed in Section 3.3.1, no previous attempt has proposed negative sampling approaches that take the geographical location of venues into accounts to address the cold-start problem. Hence, our third research question:

RQ3 *Can our proposed dynamic geo-based negative sampling approach that leverages both sequential properties of checkins and geographical location of venues enhance the effectiveness of DRCF and alleviate the cold-start problems?*

In the remainder of this section, we describe the experimental setup in terms of datasets and measures (Section 5.1), baselines (Section 5.2) and algorithm parameters (Section 5.3). The experimental results and analysis follow in Section 6.

5.1 Datasets & Measures

We conduct experiments using publicly available large-scale LBSN datasets. In particular, to show the generalisation of our proposed framework across multiple LBSN platforms and sources of feedback evidence, we use two checkin datasets from Brightkite² and Foursquare³, and a rating dataset from Yelp⁴. We follow the common practice from previous works [8, 13, 20, 29, 31] to remove venues with less than 10 checkins/ratings. Table 1 summarises the statistics of the filtered datasets. To evaluate the effectiveness of our proposed framework, following previous works [7, 8, 20], we adopt a *leave-one-out* evaluation methodology: for each user, we select her most recent checkin/rating as a ground truth and randomly select 100 venues that she has not visited before as the testing set, where the remaining checkins/ratings are used as the training set.

² <https://snap.stanford.edu/data/> ³ https://archive.org/details/201309.foursquare-dataset_umn ⁴ https://www.yelp.com/dataset_challenge

Table 1: Statistics of the three used datasets.

	Brightkite	Foursquare	Yelp
Number of normal users	14,374	10,766	38,945
Number of venues	5,050	10,695	34,245
Number of ratings or checkins	681,024	1,336,278	981,379
Number of cold-start users	5,578	154	6903
% density of User-Venue matrix	0.93	1.16	0.07

The venue recommendation task is thus to rank those 101 venues for each user, aiming to rank highest the recent, ground truth checkin/rating. We conduct two separate experiments, namely: *Normal Users* (those with ≥ 10 checkins) and *Cold-start Users* (< 10 checkins) to evaluate the effectiveness of our proposed DRCF framework and its components in the general and cold-start settings. Recommendation effectiveness is measured in terms of Hit Ratio (HR) and Normalized Discounted Cumulative Gain (NDCG) on the ranked lists of venues – as applied in previous literature [7, 8, 28, 29]. In particular, HR considers the ranking nature of the task, by taking into account the rank(s) of the venues that each user has previously visited/rated in the produced ranking, while NDCG goes further by considering the checkin frequency/rating value of the user as the graded relevance label. Lastly, significance tests use a paired t-test.

5.2 Baselines

Our proposed Deep Recurrent Collaborative Filtering (DRCF) framework consists of three components, namely: Generalised Recurrent Matrix Factorisation (GRMF), Multi-Level Recurrent Perceptron (MLRP) and Recurrent Matrix Factorisation (RMF) models. We first compare our proposed DRCF framework with the NeuMF framework as well as state-of-the-art MF-based and RNN-based approaches, which can be categorised as the traditional MF-based approaches, RNN-based approaches and Deep Neural Network-based approaches. Note that such approaches may not be originally proposed for venue recommendation but are flexible to do so without any disadvantages. Then, we compare the components of our proposed framework with the components of state-of-the-art Neural Matrix Factorisation framework (NeuMF) [7], the GMF and MLP models. Our baselines are summarised below:

5.2.1 Traditional MF-based approaches.

MF. The traditional matrix factorisation proposed by Koren *et al.* [12] that aims to accurately predict the users' checkin on the unvisited venues.

BPR. The classical pairwise ranking approach, coupled with matrix factorisation for user-venue checkin frequency prediction, proposed by Rendle *et al.* [20].

GeoBPR. A state-of-the-art BPR model that incorporates geographical influence proposed by Yuan *et al.* [29]. Their model assumes that neighbourhood venues of venues previously visited by users should be ranked higher than the distant ones. This model uses a static negative sampling approach that incorporates the geographical location of venues.

5.2.2 RNN-based approaches.

RNN. A sequential click prediction with recurrent neural networks approach proposed by Zhang *et al.* [32] (see Section 3.4).

DREAM [28]. A state-of-the-art RNN model that incorporates BPR for ranking optimisation. As DREAM is originally proposed for

next shopping-basket recommendation, to permit a fare comparison with our proposed DRNN, we reimplement DREAM to treat a single checkin as the shopping-basket purchase.

5.2.3 Deep Neural Network (DNN)-based approaches.

NeuMF. A state-of-the-art Neural Matrix Factorisation framework⁵ proposed by He *et al.* [7], which consists of two components: namely Generalised Matrix Factorisation (GMF) and Multi-Level Perceptron (MLP) (see Section 3.3).

GMF. A component of the NeuMF framework that models the user-venue interaction using the element-wise product of latent factors (see Section 3.3.2).

MLP. A component of the NeuMF framework that models the user-venue interaction using the concatenation of latent factors (see Section 3.3.3).

We implement all baselines and our proposed approach using Keras⁶, a deep learning framework built on top of Theano⁷. The choice of recurrent models is fixed to the RNN model proposed by Zhang *et al.* [32], as used by Yu *et al.* [28]. Finally, we omit the Markov Chain-based baselines (e.g. [21]), as experimental results in [28] showed that RNN-based models are more effective than Markov Chain-based ones. Similarly, we omit state-of-the-art MF-based approaches (e.g. eALS [8]) as He *et al.* [7] showed that the NeuMF framework significantly outperforms such approaches.

5.3 Recommendation Parameter Setup

Following [16, 17, 27, 29], we set the dimension of the latent factors d of our proposed DRCF framework and all of the MF-based, RNN-based and DNN-based baselines to be identical: $d = 10$ across three datasets. Following He *et al.* [7], we randomly initialise all hidden, embedding and RNN layers’ parameters, $\theta_r, \theta_e, \theta_h$, with a Gaussian distribution (with a mean of 0 and standard deviation of 0.01) and apply the mini-batch Adam optimiser [10] to optimise those parameters, which yields faster convergence than SGD and automatically adjusts the learning rate for each iteration. We initially set learning rate to 0.001⁸ and set the batch size to 256. Finally, to permit a fair comparison between MF-based approaches that exploit a Multi-Level Perceptron architecture to capture the user-venue interactions using the concatenation of latent factors (i.e. the MLP and MLRP models), we employ three hidden layers, $L = 3$. As the impact of the hidden layer’s size L and dimension size d have been explored in previous work [7, 8], we omit varying the size of the hidden layers and the dimension of the latent factors in this work. Indeed, the larger size of hidden layers and dimension may cause overfitting and degrade the generalisation of the models [7, 8].

6 EXPERIMENTAL RESULTS

Tables 2 & 3 report the effectiveness of various approaches in term of the HR@10 and NDCG@10 on the three used datasets. In particular, Table 2 reports effectiveness for all users, while Table 3 reports results for Cold-Start users. We focus first on Table 2, which contains four groups of rows: The first group, denoted vs. Baselines, reports the effectiveness of our proposed DRCF framework compared to baselines, including the NeuMF framework as well as MF-based and RNN-based approaches. The second and third groups report

the results of the components of our proposed DRCF framework (the GRMF and MLRP models) in comparison with components of the NeuMF framework (the GMF and MLP models) as well as state-of-the-art RNN-based approaches (Component Ablation of GRMF and MLRP row, respectively). The fourth group reports the improvement of DRCF framework when incorporating either our proposed *dynamic* or *static* geo-based negative sampling approach that takes the geographical location of venues into account⁹.

Firstly, on inspection of the first group of rows of Table 2, we note that the relative venue recommendation quality of the baselines on the three datasets in terms of the two measures are consistent with the results reported for the various baselines in the corresponding literature [7, 28, 32]. For instance, NeuMF outperforms MF and BPR across three datasets. Similarly, DREAM outperforms RNN and BPR across the three datasets. Note that previous works [7, 28, 32] used different datasets, while our reimplementations of their proposed approaches obtain similar relative improvements.

Comparing DRCF with the various baselines in the first group of rows of Table 2, we observe that DRCF consistently and significantly outperforms NeuMF, DREAM, BPR, RNN and MF, for HR and NDCG, across all datasets. For instance, DRCF improves NDCG by 23% over NeuMF for the Foursquare dataset. These results imply that our proposed framework that takes both users’ *dynamic* and *static* preferences as well as the dot products of latent factors into account is more effective than the NeuMF framework [7] using the same source of information.

Next, we note that unlike the Brightkite and Foursquare checkin datasets, the Yelp dataset consists of only user-venue ratings, and hence the sequential properties of visits to venues cannot be observed. We observe that the RNN-based approaches (RNN and DREAM) that consider the users’ *dynamic* preferences are more effective than the traditional MF-based approaches (MF and BPR) across the checkin datasets, while are outperformed by BPR for Yelp dataset since those RNN-based approaches cannot leverage the sequential properties of rating data. However, our proposed DRCF, which considers both the users’ *dynamic* and *static* preferences, is still the most effective across the different type of datasets. In addition, we observe that DRCF consistently and significantly outperforms its two components GRMF_{*rd*b} and MLRP_{*rd*b} for both measures across the three datasets, except for HR on the Brightkite dataset, where GRMF_{*rd*b} is statistically indistinguishable from DRCF (difference in HR < 1%).

Within the second and third groups of rows in Table 2, we further analyse the effectiveness of our proposed DRCF framework by comparing its components (GRMF and MLRP) with the components of NeuMF framework (GMF and MLP) as well as RNN-based approaches (RNN and DREAM). Since our proposed GRMF and MLRP models both consists of three components: the RMF layer that incorporates the dot product of latent factors; the RNN layer that models the users’ dynamic preferences; and BPR for pairwise ranking optimisation (instead of using a pointwise loss function). To determine the importance of the GRMF’s and MLRP’s components, we follow an ablation methodology, by recording the effectiveness of the GRMF and MLRP when each of those three components is removed in turn. For simplicity, we denote d as the RMF layer, r as the RNN layer and b as the BPR optimiser. For example, GRMF_{*rb*} denotes that RMF layer is removed from the model.

⁵ https://github.com/hexiangnan/neural_collaborative_filtering

⁶ <https://github.com/fchollet/keras> ⁷ <http://deeplearning.net/software/theano/>

⁸ The default learning rate setting of the Adam optimiser in Keras.

⁹ For the first three groups of rows, recall from Section 4.1 that DRCF does not sample unvisited neighbouring venues during sampling.

Overall, we observe similar trends as for DRCF framework, where GRMF and MLRP consistently and significantly outperform the RNN, DREAM, GMF and MLP models for the two measure across the three datasets. In terms of the specific impacts of the three components of GRMF and MLRP (c.f. RQ1), we observe that all three play important roles in the effectiveness of GRMF and MLRP models as significant decreases are often observed when each component is removed. More specifically, for RQ1(a), w.r.t. the sequential properties of checkins as modelled by the r RNN layers, significant decreases of both GRMF_{rdb} and MLRP_{rdb} are consistently observed compared to GRMF_{db} and MLRP_{db} , respectively. These results imply that the user’s dynamic preference can significantly improve the effectiveness of the GRMF and MLRP models. In particular, we observe the largest decreases when the r RNN layers are removed from the MLRP model (see MLRP_{db} in the third group).

For RQ1(b), concerning the dot product (d), similar trends are observed as in RQ1(a) where significant decreases for both GRMF_{rdb} and MLRP_{rdb} are consistently observed compared to GRMF_{rb} and MLRP_{rb} . More specifically, we observe the largest decrease when RMF layer (d) is removed from the GRMF model for the checkin datasets. These results implies that RMF layers play an important role to the GRMF model and combination of element-wise product used in GRMF and dot-product used in the RMF layer can significantly improve the effectiveness of the GRMF model.

For RQ1(c), viz. the BPR optimiser (b), similar trends are observed as in RQ1(a) and RQ1(b) where significant decreases of both GRMF_{rdb} and MLRP_{rdb} are consistently observed across three datasets when these two models are degraded into *pointwise*-based approaches (c.f. GRMF_{dr} and MLRP_{dr}). This can be explained by the benefits of *pairwise* approaches in generating more effective suggestions than *pointwise* approaches, discussed in Section 3.3.1.

Overall, the strong results for GRMF and MLRP demonstrate the effectiveness of GRMF and MLRP models in comparison with the state-of-the-art approaches. Moreover, our DRCF framework combined with the GRMF and MLRP components provides a significant benefit across various datasets and measures, compared to various existing state-of-the-art approaches.

With respect to research question RQ2, we observe that the GRMF model that uses the element-wise products of latent factors to capture both users’ *dynamic* and *static* preferences consistently and significantly outperforms DREAM across three datasets, while significant differences between MLRP models and DREAM are observed in Brightkite and Foursquare. However, there is no significant difference between MLRP models and DREAM in the Yelp dataset because, as discussed above, these models cannot leverage sequential properties of ratings from the Yelp dataset.

From the fourth group of rows in Table 2, we demonstrate that the effectiveness of DRCF can be further enhanced with our proposed *dynamic* geo-based negative sampling approach¹⁰. In particular, we observe that our proposed *dynamic* negative sampling approach can significantly improve the effectiveness of DRCF in term of NDCG for the Brightkite and Foursquare datasets. In addition, DRCF_{dgeo} consistently and significantly outperforms GeoBPR, the state-of-the-art BPR model that considers the geographical location of venues, across all three datasets. However, for the Yelp dataset, where the sequential properties of checkins cannot be obtained, the *static* sampling approach (DRCF_{sgeo}) significantly outperforms

¹⁰ Recall that DRCF results in the first three groups of rows do not use the *dynamic* geo-based negative sampling approach proposed in Section 4.1.

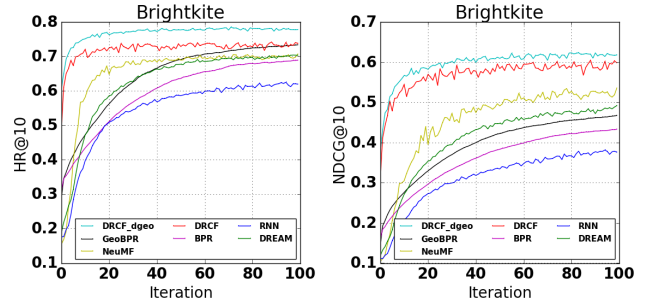


Figure 3: Test recommendation HR & NDCG of various approaches with respect to the number of iterations.

the *dynamic* sampling approach (DRCF_{dgeo}). Overall, the proposed *dynamic* geo-based sampling approach can significantly improve the effectiveness of DRCF across three datasets. In addition, Figure 3 reports the test performance of DRCF_{dgeo} and the baselines on Brightkite dataset with all users, with respect to the number of iterations. From the figure, we observe that DRCF_{dgeo} outperform the baselines at every iteration and converges faster than others¹¹.

Finally, we address the third research question, by reporting the effectiveness of DRCF and state-of-the-art approaches for cold-start users. Indeed, within the first group of rows in Table 3, we observe that DRCF consistently outperforms all of the baselines for NDCG across three datasets (significantly so for all baselines except RNN, and DREAM for Foursquare). In addition, by incorporating our proposed *dynamic* geo-based sampling approach into the DRCF framework (DRCF_{dgeo}), DRCF can be significantly improved (for both measures, across all datasets). Therefore, in response to research question RQ3, we find that our proposed *dynamic* geo-based sampling approach that takes the geographical location of venues into account can enhance the effectiveness of DRCF framework as well as alleviate the cold-start problems.

7 CONCLUSIONS

In this paper, we proposed a novel Deep Recurrent Collaborative Filtering (DRCF) framework with a pairwise ranking function for venue recommendation, positioned within six elicited limitations in the state-of-the-art approaches. In particular, the proposed framework consists of two components (GRMF and MLRP models) that aim to capture both the users’ *dynamic* and *static* preferences from their sequences of their checkins. In addition, we proposed a novel sequential-based negative sampling approach that takes the geographical location of venues into account to alleviate the cold-start problem. Our comprehensive experiments on three large-scale datasets from the Brightkite, Foursquare and Yelp LBSNs demonstrate the significant improvements of our proposed DRCF framework and its components as well as the sequential-based sampling approaches for venue recommendation in comparison with various state-of-the-art venue recommendation approaches in both normal and cold-start settings. Indeed, on the Foursquare dataset, DRCF improves NDCG by 23% over the recent NeuMF framework [7]. For future work, we plan to extend DRCF framework to incorporate additional neural network layers such as Convolutional Neural Networks which would allow to capture the semantic properties

¹¹ We observed similar trends for the other datasets – we omit those figures for brevity.

Table 2: Performance in terms of HR@10 and NDCG@10 between various approaches. The best performing result is highlighted in bold; – and * denote a significant difference compared to the best performing result, according to the paired t-test for $p < 0.05$ and $p < 0.01$, respectively.

Normal Users Experiments						
	Brightkite		Foursquare		Yelp	
Model	HR	NDCG	HR	NDCG	HR	NDCG
vs. Baselines						
MF	0.6206*	0.3470*	0.6656*	0.3818*	0.3539*	0.1734*
RNN	0.6368*	0.3824*	0.8040*	0.5459*	0.3814*	0.1891*
BPR	0.6890*	0.4333*	0.7550*	0.4834*	0.4963*	0.2676*
DREAM	0.7041*	0.4839*	0.8147*	0.6081*	0.4349*	0.2235*
NeuMF	0.7073*	0.5358*	0.8361*	0.5842*	0.4934*	0.2729*
GRMF _{rdb}	0.7363	0.5670*	0.8805*	0.6814*	0.5209*	0.2890*
MLRP _{rdb}	0.7291*	0.5790*	0.8873*	0.7046*	0.4771-	0.2652*
DRCF	0.7419	0.6048	0.8952	0.7223	0.5162	0.2963
Component Ablation of GRMF						
RNN	0.6368*	0.3824*	0.8040*	0.5459*	0.3814*	0.1891*
DREAM	0.7041*	0.4839*	0.8147*	0.6081*	0.4349*	0.2235*
GMF	0.7072*	0.4500*	0.7753*	0.4874*	0.4809*	0.2570*
GRMF _{dr}	0.7380*	0.5199*	0.8523*	0.6126*	0.4383*	0.2232*
GRMF _{db}	0.7460	0.5326*	0.8281*	0.5765*	0.5164-	0.2864-
GRMF _{rb}	0.6704*	0.4772*	0.8273*	0.5984*	0.5210	0.2841*
GRMF _{rdb}	0.7363*	0.5670	0.8805	0.6814	0.5209	0.2890
Component Ablation of MLRP						
RNN	0.6368*	0.3824*	0.8040*	0.5459*	0.3814*	0.1891*
DREAM	0.7041*	0.4839*	0.8147*	0.6081*	0.4349*	0.2235*
MLP	0.6780*	0.4805*	0.7638*	0.4846*	0.4656*	0.2492*
MLRP _{dr}	0.7185*	0.4536*	0.8755*	0.5627*	0.4121*	0.2131*
MLRP _{db}	0.6851*	0.4923*	0.8012*	0.5326*	0.4740*	0.2604*
MLRP _{rb}	0.6985*	0.5390*	0.8709*	0.6737*	0.4917	0.2705
MLRP _{rdb}	0.7291	0.5790	0.8873	0.7046	0.4771*	0.2652*
Geographic Negative Sampling						
DRCF	0.7419*	0.6048*	0.8952*	0.7223	0.5162*	0.2963*
GeoBPR	0.7339*	0.4672*	0.8216*	0.5395-	0.5570*	0.3032*
DRCF _{sgeo}	0.7847	0.6047*	0.9086	0.7217	0.5682	0.3134
DRCF _{dgeo}	0.7852	0.6210	0.9095	0.7214	0.5618*	0.3064*

Table 3: As per Table 2, but only for Cold-Start users.

	Brightkite		Foursquare		Yelp	
Model	HR	NDCG	HR	NDCG	HR	NDCG
vs. Baselines						
MF	0.6768*	0.3913*	0.6623*	0.3650*	0.3748*	0.1868*
BPR	0.7519	0.4907*	0.7792-	0.4961*	0.5273-	0.2946*
RNN	0.6486*	0.3694	0.5909	0.4041*	0.3856*	0.1901*
DREAM	0.7452	0.4969*	0.7987	0.5379*	0.4523*	0.2239*
NeuMF	0.7160*	0.5894-	0.7922	0.6227-	0.5102*	0.2734*
GRMF _{rdb}	0.7409-	0.5618*	0.8442	0.6542	0.5399	0.3083
MLRP _{rdb}	0.7418-	0.5779*	0.8377	0.6138*	0.4928*	0.2788*
DRCF	0.7526	0.5980	0.8377	0.6645	0.5330	0.3136
Geographic Negative Sampling						
DRCF	0.7526*	0.5980*	0.8377-	0.6645-	0.5330*	0.3136*
GeoBPR	0.8093	0.5262*	0.7468-	0.4717*	0.5802-	0.3202*
DRCF _{sgeo}	0.8041	0.6009*	0.8636	0.6748-	0.5948	0.3410
DRCF _{dgeo}	0.8094	0.6199	0.8896	0.7074	0.5877	0.3318-

of textual content of comments and thereby further enhance the quality of venue recommendations.

REFERENCES

- [1] Christopher J. Burges, Robert Ragno, and Quoc V. Le. 2007. Learning to Rank with Nonsmooth Cost Functions. In *Proc. of NIPS*. 193–200.
- [2] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: from pairwise approach to listwise approach. In *Proc. of ICML*. 129–136.
- [3] Chen Cheng, Haiqin Yang, Irwin King, and Michael R Lyu. 2012. Fused Matrix Factorization with Geographical and Social Influence in Location-Based Social Networks. In *Proc. of AAAI*. 17–23.
- [4] Chen Cheng, Haiqin Yang, Michael R Lyu, and Irwin King. 2013. Where You Like to Go Next: Successive Point-of-Interest Recommendation. In *Proc. of IJCAL*. 2605–2611.
- [5] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, , and others. 2016. Wide & deep learning for recommender systems. In *Proc. of DLRS*. 7–10.
- [6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep residual learning for image recognition. In *Proc. of CVPR*. 770–778.
- [7] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural Collaborative Filtering. In *Proc. of WWW*. 173–182.
- [8] Xiangnan He, Hanwang Zhang, Min-Yen Kan, and Tat-Seng Chua. 2016. Fast matrix factorization for online recommendation with implicit feedback. In *Proc. of SIGIR*. 549–558.
- [9] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882* (2014).
- [10] Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
- [11] Yehuda Koren. 2010. Collaborative filtering with temporal dynamics. *Commun. ACM* 53, 4 (2010), 89–97.
- [12] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.
- [13] Huayu Li, Yong Ge, and Hengshu Zhu. 2016. Point-of-Interest Recommendations: Learning Potential Check-ins from Friends. In *Proc. of KDD*. 975–984.
- [14] Defu Lian, Cong Zhao, Xing Xie, Guangzhong Sun, Enhong Chen, and Yong Rui. 2014. GeoMf: Joint geographical modeling and matrix factorization for point-of-interest recommendation. In *Proc. of SIGKDD*. 831–840.
- [15] Tie-Yan Liu. 2009. Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval* 3, 3 (2009), 225–331.
- [16] Hao Ma, Dengyong Zhou, Chao Liu, Michael R Lyu, and Irwin King. 2011. Recommender systems with social regularization. In *Proc. of WSDM*. 287–296.
- [17] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2016. Regularising factorised models for venue recommendation using friends and their comments. In *Proc. of CIKM*. 1981–1984.
- [18] Jarana Manotumruksa, Craig Macdonald, and Iadh Ounis. 2017. Matrix Factorisation with Word Embeddings for Rating Prediction on Location-Based Social Networks. In *Proc. of ECIR*. Springer, 647–654.
- [19] Steffen Rendle. 2012. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology (TIST)* 3, 3 (2012), 57.
- [20] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proc. of UAI*. 452–461.
- [21] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. 2010. Factorizing personalized markov chains for next-basket recommendation. In *Proc. of WWW*. 811–820.
- [22] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1988. Learning representations by back-propagating errors. *Cognitive modeling* 5, 3 (1988), 1.
- [23] Ruslan Salakhutdinov, Andriy Mnih, and Geoffrey Hinton. 2007. Restricted Boltzmann machines for collaborative filtering. In *Proc. of ICML*. 791–798.
- [24] Yong Kiam Tan, Xinxing Xu, and Yong Liu. 2016. Improved recurrent neural networks for session-based recommendations. In *Proc. of DLRS*. 17–22.
- [25] Song Tang, Zhiyong Wu, and Kang Chen. 2017. Movie Recommendation via BLSTM. In *Proc. of ICMM*. 269–279.
- [26] Bart lomiej Twardowski. 2016. Modelling Contextual Information in Session-Aware Recommender Systems with Neural Networks. In *Proc. of RecSys*. 273–276.
- [27] Xin Wang, Wei Lu, Martin Ester, Can Wang, and Chun Chen. 2016. Social Recommendation with Strong and Weak Ties. In *Proc. of CIKM*. 5–14.
- [28] Feng Yu, Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan. 2016. A dynamic recurrent model for next basket recommendation. In *Proc. of SIGIR*. 729–732.
- [29] Fajie Yuan, Guibing Guo, Joemon Jose, Long Chen, and Haitao Yu. 2016. Joint Geo-Spatial Preference and Pairwise Ranking for Point-of-Interest Recommendation. In *Proc. of ICTAL*. 46–53.
- [30] Hanwang Zhang, Yang Yang, Huanbo Luan, Shuicheng Yang, and Tat-Seng Chua. 2014. Start from scratch: Towards automatically identifying, modeling, and naming visual attributes. In *Proc. of MM*. 187–196.
- [31] Jia-Dong Zhang, Chi-Yin Chow, and Yu Zheng. 2015. ORec: An opinion-based point-of-interest recommendation framework. In *Proc. of CIKM*. 1641–1650.
- [32] Yuyu Zhang, Hanjun Dai, Chang Xu, Jun Feng, Taifeng Wang, Jiang Bian, Bin Wang, and Tie-Yan Liu. 2014. Sequential Click Prediction for Sponsored Search with Recurrent Neural Networks. In *Proc. of AAAI*. 1369–1375.