3-2024

A Social-aware Gaussian Pre-trained model for effective cold-start recommendation

Siwei Liu
Mohamed Bin Zayed University of Artificial Intelligence

Xi Wang
University College London

Craig Macdonald
University of Glasgow

Iadh Ounis
University of Glasgow

Recommended Citation

S. Liu, X Wang, C Macdonald and I Ounis, "A Social-aware Gaussian Pre-trained model for effective cold-start recommendation," *Information Processing and Management*, vol. 61, no. 2, Mar 2024. doi: 10.1016/j.ipm.2023.103601

Highlights

A Social-aware Gaussian Pre-trained Model for Effective Cold-start Recommendation

Siwei Liu,Xi Wang,Craig Macdonald,Iadh Ounis

- Social graph pre-training does improve recommendation performance
- A Gaussian Mixture Model can effectively extract meaningful relations from the pre-trained embeddings
- Experimental results show significant improvements using the proposed Social-aware Gaussian Pre-trained model, especially for *cold-start* users

A Social-aware Gaussian Pre-trained Model for Effective Cold-start Recommendation

Siwei Liu[a], Xi Wang[b], Craig Macdonald[c] and Iadh Ounis[c]

[a]*Mohamed bin Zayed University of Artificial Intelligence , Department of Machine Learning, Masdar, Adu Dhabi, 44737, United Arab Emirates*
[b]*University College London, School of Computer Science, Gower Street, London, WC1E 6BT, United Kingdom*
[c]*University of Glasgow, School of Computing Science, Lilybank Gardens, Glasgow, G12 8QQ, United Kingdom*

ARTICLE INFO

Keywords:
Gaussian Mixture Model
Social Network
Graph Neural Networks
Recommender Systems

ABSTRACT

The use of pre-training is an emerging technique to enhance a neural model's performance, which has been shown to be effective for many neural language models such as BERT. This technique has also been used to enhance the performance of recommender systems. In such recommender systems, pre-training models are used to learn a better initialisation for both users and items. However, recent existing pre-trained recommender systems tend to only incorporate the user interaction data at the pre-training stage, making it difficult to deliver good recommendations, especially when the interaction data is sparse. To alleviate this common data sparsity issue, we propose to *pre-train* the recommendation model not only with the interaction data but also with other available information such as the social relations among users, thereby providing the recommender system with a better initialisation compared with solely relying on the user interaction data. We propose a novel recommendation model, the *Social-aware Gaussian Pre-trained* model (SGP), which encodes the user social relations and interaction data at the *pre-training* stage in a Graph Neural Network (GNN). Afterwards, in the subsequent fine-tuning stage, our SGP model adopts a Gaussian Mixture Model (GMM) to factorise these pre-trained embeddings for further training, thereby benefiting the *cold-start* users from these pre-built social relations. Our extensive experiments on three public datasets show that, in comparison to 16 competitive baselines, our SGP model significantly outperforms the best baseline by upto 7.7% in terms of NDCG@10. In addition, we show that SGP permits to effectively alleviate the *cold-start* problem, especially when users newly register to the system through their friends' suggestions.

1. Introduction

Deep learning-based models have achieved a remarkable success in different domains (Litjens, Kooi, Bejnordi, Setio, Ciompi, Ghafoorian, Van Der Laak, Van Ginneken and Sánchez, 2017; Zhang, Yang, Chen and Li, 2018). However, although these deep models have a strong expressiveness power, they cannot easily reach the maximal optimised solution during the training stage without an effective initialisation (Erhan, Courville, Bengio and Vincent, 2010; Van Engelen and Hoos, 2020). Therefore, the pre-training technique has been commonly used to optimise the deep models by providing them with an effective initialisation (Erhan et al., 2010; Xin, Pimentel, Karatzoglou, Ren, Christakopoulou and Ren, 2022). Such a pre-training technique has been shown to lead to state-of-the-art performances when the pre-trained model is further *fine-tuned* to address *downstream* Natural Language Processing (NLP) (González, Hurtado and Pla, 2020; Devlin, Chang, Lee and Toutanova, 2019) or information retrieval tasks (Ma, Guo, Zhang, Fan, Ji and Cheng, 2021; Zheng, Hui, He, Han, Sun and Yates, 2021). However, this effective technique has been less studied in recommender systems possibly due to the limitations in the existing datasets. For example, in the NLP tasks, one unsupervised deep language model can be pre-trained from unlabelled texts (e.g. Wikipedia) and fine-tuned for a supervised downstream task (Devlin et al., 2019; Brown, Mann, Ryder, Subbiah, Kaplan, Dhariwal, Neelakantan, Shyam, Sastry, Askell, Agarwal, Herbert-Voss, Krueger, Henighan, Child, Ramesh, Ziegler, Wu, Winter, Hesse, Chen, Sigler, Litwin, Gray, Chess, Clark, Berner, McCandlish, Radford, Sutskever and Amodei, 2020). In contrast, in the recommendation scenario, each dataset contains its specific information about the corresponding users

siwei.liu@mbzuai.ac.ae (S. Liu); xi-wang@ucl.ac.uk (X. Wang); craig.macdonald@glasgow.ac.uk (C. Macdonald); iadh.ounis@glasgow.ac.uk (I. Ounis)
ORCID(s): 0000-0002-7326-2883 (S. Liu); 0000-0001-5936-9919 (X. Wang); 0000-0003-3143-279X (C. Macdonald); 0000-0003-4701-3223 (I. Ounis)

and items, but no other ground truth knowledge such as Wikipedia could be leveraged from outside the dataset to help estimate the users' preferences and items' attributes.

An existing Neural Collaborative Filtering (NCF) recommendation approach (He, Liao, Zhang, Nie, Hu and Chua, 2017) has proposed to pre-train the recommendation model with a Multi-Layer Perceptron (MLP) (Ramchoun, Idrissi, Ghanou and Ettaouil, 2016). Although effective, the MLP module does not consider other available auxiliary side information, such as the social relations among users (Seyedhoseinzadeh, Rahmani, Afsharchi and Aliannejadi, 2022; Elahi, Kholgh, Kiarostami, Saghari, Rad and Tkalčič, 2021) or the items' timestamps (Li, Wang and McAuley, 2020), therefore the applied pre-training technique of NCF is limited in providing the *cold-start* users with a better initialisation. Since the social relations among users have been shown to be essential in enhancing the recommendation performance and alleviating the *cold-start* problem (Liu, Ounis, Macdonald and Meng, 2020; Camacho and Alves-Souza, 2018), we propose to incorporate the social relations and the interaction data at the pre-training stage so that a better initialisation can be obtained for those users who have fewer interactions.

Graph Neural Networks (GNNs), a class of deep learning models (Zhou, Cui, Hu, Zhang, Yang, Liu, Wang, Li and Sun, 2020; Wu, Pan, Chen, Long, Zhang and Philip, 2020b), have been used to aggregate the nodes' information from their neighbourhoods so as to learn an overall structure from a given graph's type of data. Indeed, while GNNs have been previously exploited to enhance general recommender systems (He, Deng, Wang, Li, Zhang and Wang, 2020; Yang, Wu, Wu, Zhang, Hong, Zhang, Zhou and Wang, 2023; Yi, Ounis and Macdonald, 2023), they have only been recently studied as pre-training schemes (Hao, Zhang, Yin, Li and Chen, 2021). In this work, we devise a novel Social-aware Gaussian Pre-trained model (SGP), which incorporates the users' social relations in the pre-training stage and attempts to search for a relative optimised solution based on the learned social-aware initialisation during the fine-tuning stage. At the first stage, we pre-train a light GNN model with additional social information to give users/items meaningful initialised embeddings. Given the neighbourhood aggregation property of the GNN model, incorporating the social relations enables socially-connected users to become closer in this latent space through the aggregation process.

In the fine-tuning stage, we load the obtained pre-trained embeddings and re-train the model for further recommendations. The most straightforward approach for leveraging these pre-trained embeddings and decoding the social information is to directly reload them. However, it is essential to note that the interaction data, which will be used in the second stage, has already been exploited at the pre-training stage. Therefore, the direct reuse of the interaction data might cause the overfitting problem. To tackle the problem of data reuse, the relational knowledge distillation technique (Park, Kim, Lu and Cho, 2019; Gou, Yu, Maybank and Tao, 2021) has been proposed to distil relations from a pre-trained model. The underlying intuition of the relational knowledge distillation technique is that the distillation model is encouraged to extract the essential relations from the pre-trained model, thereby avoiding a deficient performance as well as overfitting (Turc, Chang, Lee and Toutanova, 2019). Motivated by this technique of relational knowledge distillation, we propose to distil the information from the pre-trained GNN model so that we can later reconstruct meaningful embeddings. Since all embeddings can be viewed as probability distributions, an intuitive solution for distilling information from those pre-trained embeddings is to follow existing works (Rendle, Krichene, Zhang and Anderson, 2020; He et al., 2020) and use a normal distribution to model the embeddings. However, those pre-trained embeddings contain prior knowledge and complex latent relations between users and items, which can hardly be modelled with a normal distribution without information loss. Therefore, during the intialisation of the fine-tuning stage, we propose to apply the Gaussian Mixture Model (GMM) (Reynolds, 2009), which assumes that all the data points are sampled from a mixture of a finite number of Gaussian distributions. By leveraging this well-developed GMM, our proposed method is devised to factorise the pre-trained embeddings into a finite number of Gaussian distributions, where this number is pre-defined and each distribution could be viewed as a specific interest of a group of users or a particular characteristic of a set of items.

To summarise, in this work, we make the following contributions:

• We devise a two-stage end-to-end social pre-trained recommendation model, SGP, which uses the GNN model to leverage social information. We show that SGP can achieve state-of-the-art performance on three real-world datasets of user-item interactions and social relations.

• We leverage the Gaussian Mixture Model to effectively distil information from the pre-trained embeddings for the downstream recommendation task.

• Our proposed SGP model is shown to significantly outperform 16 strong baselines from the literature, while being particularly useful for *cold-start* and extreme *cold-start* users (newly registered users).

The remainder of this paper is organised as follows. In Section 2, we position our work in the literature. Section 3 introduces all relevant notions used in this paper and formally defines the detailed architecture of our SGP model. The experimental setup and the results of our empirical experiments are presented in Sections 4 and 5, respectively, followed by some concluding remarks in Section 6.

2. Related Work

In the following, we overview the related work from three perspectives: pre-trained models that learn general representations for various downstream tasks (Section 2.1), graph-based recommendation models that leverage the graph structure of user-item interactions (Section 2.2), and social-aware recommendation systems, which incorporate social information into the recommendation process (Section 2.3).

2.1. Pre-trained Models

The pre-training technique has become an emerging research topic especially in the field of NLP. Pre-trained language models such as the BERT (Devlin et al., 2019) and the more recent GPT-3 (Brown et al., 2020) models have demonstrated their robust performance on different downstream NLP tasks. Through pre-training, a language model can learn contextualised embeddings for tokens from a large corpus of texts, so that these tokens can be reused for subsequent tasks with enhanced performances. Such models can then be later fine-tuned for a new downstream task, thereby enhancing the overall performance of the corresponding model and outperforming other handcrafted models. The pre-training technique was also adopted in recommendation models. For example, He et al. (2017) proposed the Neural Collaborative Filtering (NCF) model to introduce a novel deep learning-based method to the recommender systems community, which has attracted a substantial attention from researchers since then. The most remarkable contribution of the NCF model is that it successfully incorporates the multi-layer perceptron (MLP) module, which can in theory effectively approximate various types of prediction functions. However, it is noticeable that this NCF model also uses a generalised matrix factorisation (GMF) module to generate pre-trained embeddings, which limits the NCF model from incorporating auxiliary information at the pre-training stage. To this end, we propose to use instead the GNN technique to replace the MLP pre-trained module due to the former's ability of supporting the incorporation of heterogeneous relations such as the relations among users as well as the users' interaction data. Moreover, the GNN technique, which has been initially devised to implement the node classification and link prediction tasks, would naturally perform better than the GMF module on aggregating similar users and items (Wang, Samari and Siddiqi, 2018). Hence, compared with the GMF module, when the GNN model is used for the pre-training stage, the embeddings of the socially related users can be better aggregated in closer proximity in the latent space. Apart from using the GMF module to pre-train on the interaction data, Wen, Guo, Chen and Ma (2018) introduced a linear pre-trained recommender using the network embedding method. However, their proposed model failed to leverage the multi-hops social relations (i.e. a friend's friends), which can be seamlessly addressed by the GNN methods. The study by Hao et al. (2021) is a more recent related work to ours, which tried to tackle the cold-start problem by pre-training the recommendation model in a meta-learning setting. However, the contribution of their work is to use the underlying structure of the user-item interaction graph, which is different from our research goal of using social information to obtain better initialised users and items' representations. Besides, existing works have leveraged other GNN pre-training and contrastive pre-training techniques for sequential (Li, Liu, Guo, Liu, Peng, Philip and Achan, 2021; Xiao, Xie, Yao, Liu, Sun, Zhang and Lin, 2021; Xie, Sun, Liu, Gao, Ding and Cui, 2020) and conversational (Wong, Feng, Zhang, Vong, Chen, Zhang, He, Chen, Zhao and Chen, 2021) recommendations, respectively, which are distinct from our proposed general recommender SGP.

2.2. Graph-based Recommendation

Various graph-based recommenders (He et al., 2020; Chen, Wu, Hong, Zhang and Wang, 2020; Yu, Yin, Xia, Chen, Li and Huang, 2022a; Yu, Yin, Xia, Cui and Nguyen, 2022b) have been shown to achieve state-of-the-art performances through the development of the GNN technique and its variants (Kipf and Welling, 2017; Zhang, Ling, Shen, Wang, Lei, Shi, Wu and Li, 2021). Since the user-item interaction data can be intrinsically depicted as an interaction graph, the GNN technique and its variants have been seamlessly applied in various recommender systems and achieved good performances. For example, the proposed graph-based recommender model, NGCF (Wang, He, Wang, Feng and Chua, 2019), has been shown to outperform many competitive baselines by incorporating a Graph Convolutional Network (GCN) to encode the collaborative signals and to model the users' and items' embeddings.

Building on NGCF, the LightGCN model (He et al., 2020) further enhanced its recommendation performance by eliminating redundant neural components from NGCF. Recently, GF-CF (Shen, Wu, Zhang, Shan, Zhang, Letaief and Li, 2021) was proposed to further enhance the performance by incorporating a graph filtering method. However, GF-CF is not an embedding-based model, hence it cannot be adapted to normal pre-training methods. Other variants of LightGCN, including SGL (Wu, Wang, Feng, He, Chen, Lian and Xie, 2021) and UltraGCN (Mao, Zhu, Xiao, Lu, Wang and He, 2021) have also achieved competitive performances. However, they incorporate memory-consuming data augmentation methods, which will be more challenging if side information is also considered. Inspired by the generalisability of LightGCN and its good trade-off between effectiveness and efficiency, we also adopt the simplified GCN (Wu, Souza, Zhang, Fifty, Yu and Weinberger, 2019). Moreover, we incorporate the social information into the embedding generation and updating process, which enables our proposed SGP model to encode the social relations into the users' embeddings. We will show how this auxiliary social information benefits our model by allowing it to obtain a better model initialisation thereby alleviating the *cold-start* problem.

2.3. Social-aware Recommendation

Since the early work of SoReg (Ma, Zhou, Liu, Lyu and King, 2011), social relations have been shown to be a valuable source of side information for recommendation. Indeed, SoReg uses social relations as a regularisation in order to enhance the recommendation performance. As deep learning-based recommendation systems have developed, researchers have also focused on how to use social relations more effectively in advanced recommendation scenarios. This has led to the evolution of social relations usage from traditional regularisation methods to more sophisticated relation encoding methods (Wu, Sun, Hong, Fu, Wang and Wang, 2018). In particular, with the emergence of graph-based recommendation, social relations have become a natural choice of side information since, within a user-item interactions graph, relations between users can be encoded as additional edges between user nodes instead of being treated as attributes of entities. For example, Diffnet++ (Wu, Li, Sun, Hong, Ge and Wang, 2020a) incorporated the additional social relations by adding the user-user edges into the original user-item bipartite graph. In addition, S^2-MHCN (Yu, Yin, Li, Wang, Hung and Zhang, 2021b) has been proposed to capture the high-order information among the users' social relations using a hypergraph neural network augmented by the self-supervised learning technique. Similarly, SCDSR (Luo, Yang, Zhang, Sun, Wu and Hong, 2022) is another self-supervised graph recommendation model, which built a heterogeneous graph using the social and information domains. By doing so, SCDSR aims to leverage the high-order correlations between non-bridge users in the social domain and items in the information domain. Although overall effective, these existing models typically neglect the situation when new users register to the system through their friends' suggestions. Compared with the existing models, our proposed SGP model not only benefits normal *cold-start* users with fewer than 5 interactions, but also provides effective recommendations to those extreme *cold-start* users who have no interactions at all.

3. Model Architecture

Our proposed SGP model consists of two main stages: 1) a social-aware pre-training stage, where a multi-layer GNN is employed to generate the pre-trained embeddings and 2) an information distillation stage, where we incorporate the Gaussian Mixture Model (GMM) to distil information from the pre-trained embeddings for the subsequent model's training and generation of recommendations. In the following, we first define the tasks and some preliminaries in Section 3.1. Next, Section 3.2 describes how to incorporate the social relations and a light GNN model to propagate the social information into users' and items' embeddings. Finally, in Section 3.3, we demonstrate how to employ the GMM to distil the social information from those pre-trained embeddings for the subsequent training and the production of final recommendations. To clearly illustrate our model, Figure 1 depicts the overall structure of SGP, where the upper and bottom regions describe the pre-training and fine-tuning stages, respectively. We conclude the presentation of the SGP model with a discussion of its benefits for extreme cold-start users (Section 3.4) as well as an analysis of its time complexity (Section 3.5).

3.1. Preliminaries

In this section, we introduce the notations used across the whole article and formally define our research task. Throughout this paper, we use calligraphy typeface alphabets to denote sets (e.g. $\mathcal{U}$ is the set of users). Besides, matrices and vectors are denoted by bold letters with uppercase letters representing matrices and lowercase letters representing vectors. In Table 1, we summarise main notations used in this article for a fast reference.

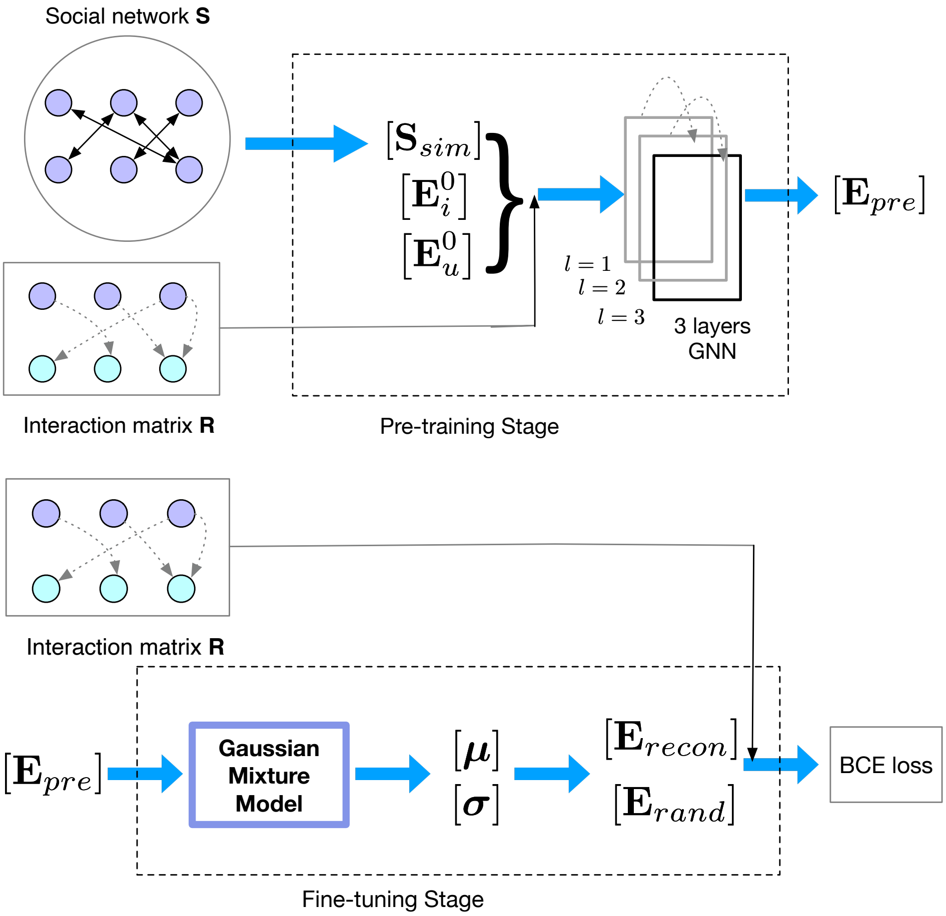

Figure 1: An illustration of our SGP model, where the pre-training stage and the fine-tuning stage are located above and below, respectively. In this figure, $\mathbf{E}_u^0$ and $\mathbf{E}_i^0$ are the initial embedding matrices of the users and items, which are randomly generated. $\mathbf{S}_{sim}$ is the social similarity graph, which can be computed using Equation (1). Furthermore, $\boldsymbol{\mu}$ and $\boldsymbol{\sigma}$ contain the mean and standard deviation of the pre-trained embeddings, which are defined in Section 3.3.1.

Our task is to highly rank relevant items for each user given their historical interactions and their available social relations. We consider a recommender system with a user set $\mathcal{U}$ ($|\mathcal{U}| = M$) and an item set $\mathcal{I}$ ($|\mathcal{I}| = N$). Let $\mathbf{R} \in \mathbb{R}^{M\times N}$ be the user-item interaction matrix, where the content of the matrix $\mathbf{R}^{M\times N}$ corresponds to either explicit user ratings (Koren, Bell and Volinsky, 2009) or implicit feedback (Rendle, Freudenthaler, Gantner and Schmidt-Thieme, 2009). We consider implicit feedback here because it is more abundant, therefore, $\mathbf{R}_{ui} = 1$ if the user u has interacted with the item i, otherwise $\mathbf{R}_{ui} = 0$. As mentioned before, social network information is also important for improving the recommendation performance, especially when a user does not have enough interactions with items. Let $\mathbf{S} \in \{0,1\}^{M\times M}$ be the user-user social network matrix, where the content of $\mathbf{S}$ represents the social connections between each pair of users.

3.2. The Pre-training Stage

A GNN model can leverage the nodes' information and their corresponding relational information in a graph by effectively aggregating information from each node's neighbours. In the recommendation scenario, each node represents either a user or an item. Therefore, suppose that only the interaction information is given, then each user node's neighbours could correspond to those items that have been interacted with by this user (and vice-versa for an item node). On the other hand, when the social network information is also available, then the user's neighbours can be his/her interacted items or friends. This friendship information is also important for the recommender system, because users are more likely to interact with those items that have been previously interacted with by their social

Table 1
Main Notations Used in this Article.

Notation	Description
$\boldsymbol{R}$	the matrix of implicit feedback data
$\mathcal{U}$, $\mathcal{I}$	the sets of users and items
$\mathbf{S}$	the social network matrix of all users
$\mathbf{S}_{sim}$	the social similarity matrix of all users
$\mathbf{E}$	the embedding matrix
$\mathbf{E}_{pre}$	the pre-trained embedding matrix
$\mathbf{E}_{recon}$	the reconstructed embedding matrix
$\mathbf{e}$	the embedding vector
$\mathcal{L}$	the Laplacian matrix
$\mathbf{W}$	the learnable weight matrix
μ, σ	the mean and standard deviation of an embedding
$\boldsymbol{\mu}$, $\boldsymbol{\sigma}$	the mean and standard deviation matrices of an embedding matrix
ζ	the loss
M, N	the number of users and the number of items
k	the number of Gaussian distribution
Θ	the trainable model parameters
λ	the controlling factor of the L_2 regularisation
y_{ui}, $\hat{y}_{ui}$	the observed interaction and the predicted interaction
$\mathcal{N}$	the normal distribution
$\mathbf{I}$	the identity matrix

neighbours (Yang, Guo, Liu and Steck, 2014). Hence, the users' available social relations provide useful insights for inferring their interests and predicting the items that they will interact with.

To effectively propagate the friends' information into each user who is socially connected, we firstly initialise each user with a randomised embedding vector $\mathbf{e}_u$ to represent his/her interests. Similarly, we can assign each item with a randomised embedding vector $\mathbf{e}_i$. We set $\mathbf{E}_u$ to be the embedding matrix containing all latent vectors of users and $\mathbf{E}_i$ to be the embedding matrix for all items. A graph neural network can be employed here to aggregate the users' social information for each user node. By stacking multi-layers GNNs, we can propagate high-order connectivities of social relations from multi-hop neighbours. In our case, we use the most commonly used 3-layers GNNs to capture reasonable depths in the social connectivities while avoiding the possible over-smoothing effect of the GNN.

To use multi-layers GNNs (Wu et al., 2019), we rely on a well-defined Laplacian matrix $\mathcal{L}$ for the specific GNN, so that the information propagation and convolution functions can be executed effectively in a matrix multiplication form. Different from the NGCF (Wang et al., 2019) model, which only tries to encode the interaction signal into both the users' and items' embeddings, our model focuses on the social information propagation in the pre-training stage. Furthermore, to further improve the Diffnet++ model (Wu et al., 2020a), which only incorporates the plain social relation links, our model pre-computes the cosine similarity between each user's social relation vector (Ma et al., 2011)[1]. These vectors constitute one-dimensional binarised vectors indicating the social links between users. Therefore, building on this advanced user-user social similarity graph $\mathbf{S}_{sim}$, the GNN function can better classify similar users. Given the user-user social graph $\boldsymbol{S} \in \{0,1\}^{M\times M}$, we can pre-compute the social similarity graph $\boldsymbol{S}_{sim}$ as follows:

$$\mathbf{S}_{sim} = \Big(\frac{\mathbf{S}\cdot\mathbf{S}^{\mathrm{T}}}{\sqrt{\mathrm{diag}(\mathbf{S}\cdot\mathbf{S}^{\mathrm{T}})}}\Big)^{\mathrm{T}} \cdot \frac{1}{\sqrt{\mathrm{diag}(\mathbf{S}\cdot\mathbf{S}^{\mathrm{T}})}}, \tag{1}$$

where T represents the transpose of a matrix and diag() computes the diagonal matrix of the corresponding matrix. Entries of $\mathbf{S}_{sim}$ are set to 0 if the corresponding user has no social relationships in $\mathbf{S}$. Given the similarity graph $\mathbf{S}_{sim}$,

[1]We use the cosine similarity because it can be efficiently computed for our sparse user-user matrix. In addition, the similarities between social relation vectors do not need estimating the magnitude, hence other similarity measures - e.g. the dot product or the Euclidean similarities, may not be appropriate.

we can derive its corresponding Laplacian matrix $\mathcal{L}_{sim}$ as follows:

$$\mathcal{L}_{sim} = \text{diag}(\mathbf{S}_{sim})^{-\frac{1}{2}} \cdot \mathbf{S}_{sim} \cdot \text{diag}(\mathbf{S}_{sim})^{\frac{1}{2}}. \tag{2}$$

Next, using the Laplacian matrix $\mathcal{L}_{sim}$, we can present the embedding updating function of our proposed SGP model as follows:

$$\mathbf{E}_u^{(l)} = \mathcal{L}_{sim} \cdot \mathbf{E}_u^{(l-1)}. \tag{3}$$

Starting from a randomly initialised $\mathbf{E}_u^0$, we stack 3 layers of the GNNs given in Equation (3) and update the embeddings for each user correspondingly. Following the LightGCN model (He et al., 2020), we discard those redundant neural components from the variant of GCN used in NGCF (Wang et al., 2019). Indeed, the self-connection setup, i.e. adding the dot product of an embedding with itself into Equation (3), was initially proposed in (Wang et al., 2019) to keep each node's original information and to avoid being possibly overloaded with information from the nodes' neighbours. However, this self-connection was later demonstrated in (He et al., 2020) to bring no benefit to recommendation performance; instead, it will reduce the training efficiency. Hence, we choose to also remove this redundant part in our embedding updating function following the existing work (He et al., 2020).

Algorithm 1: The pre-training stage

Input: Interaction matrix **R**; Social network matrix **S**
Output: Pre-trained embedding $\mathbf{E}_{pre}$.
Initialise embeddings $\mathbf{E}^0$ and other learnable parameters Θ;
Compute $\mathcal{L}_{sim}$ according to Equation 1 & 2;
while *not early-stopped* **do**
  $\zeta_{BCE} = 0$;
  for *each training instance in* **S do**
    Propagate social information according to Equation 3;
  end
  for *each training instance in* **R do**
    Compute epoch loss $\nabla\zeta$ according to Equation 4;
  end
  $\zeta_{BCE} \leftarrow \zeta_{BCE} + \nabla\zeta$;
  Update Θ, **E**;
end

We follow the GNN technique proposed in the aforementioned LightGCN model to update and aggregate the users' embeddings. However, different from LightGCN, which uses the GNN to incorporate the interaction data, we only incorporate the social information propagation. Similar with other graph-based recommendation models (Hamilton, Ying and Leskovec, 2017; Wang et al., 2019; Wu et al., 2020a; Mao et al., 2021), we keep the interaction data as the ground truth for supervising the pre-training of our model. At each training epoch, Equation (3) is invoked to perform the social aggregation, after which we use the binary cross-entropy (BCE) loss (He et al., 2017) as the objective function:

$$\zeta_{BCE} = -\sum_{(u,i)\in\mathbf{R}} y_{ui} \cdot \log\left(\hat{y}_{ui}\right) + \left(1 - y_{ui}\right) \cdot \log\left(1 - \hat{y}_{ui}\right) + \lambda \left\|\Theta\right\|^2, \tag{4}$$

where y_{ui} is the observed interaction, $\hat{y}_{ui}$ is the predicted interaction, which is a dot product of the item embedding and the user embedding obtained from Equation (3), while $\Theta = \{\{\mathbf{E}_u^l, \mathbf{E}_i^l, \mathbf{W}^l\}_{l=1}^3\}$ denotes all trainable model parameters and λ controls the L_2 regularisation strength to prevent overfitting.

As a result, our pre-trained embeddings $\mathbf{E}_{pre}$ can be obtained by minimising the objective function in Equation (4). For a better understanding, the training framework of our pre-training stage is summarised in Algorithm 1.

3.3. The Fine-tuning Stage

After detailing the pre-training stage, we first present the information distillation stage, where we describe how to use the Gaussian Mixture Model to distil hierarchical relations from the pre-trained embeddings $\mathbf{E}_{pre}$. Finally, we demonstrate how to use the reconstructed embeddings $\mathbf{E}_{recon}$ for the final recommendations. Similar with the previous section, we summarise the training framework of the fine-tuning stage in Algorithm 2.

3.3.1. Information Distillation Stage

By using Equation (4) for the pre-training and Equation (3) for the social aggregation, we aim to encode social information into our pre-trained embeddings $\mathbf{E}_{pre}$. The latter constitutes the obtained embedding matrix from optimising Equation (4). However, it is not obvious how these pre-trained embeddings can be reloaded. Since we have already used the interaction data as the ground truth during the pre-training stage , directly reloading these embeddings is likely to cause either an overfitting or a marginal improvement. Therefore, in the information distillation stage, we propose to distil information from these pre-trained embeddings. Next, we concatenate these distilled information at the tail of a randomly initialised embedding to add more generalisation to the final embeddings (Wang et al., 2019).

Before extracting useful information from the pre-trained embeddings, we propose to model each user's or item's latent vector as a multi-Gaussian distribution. This is consistent with the implementation details of existing works (Rendle et al., 2009, 2020; Wang et al., 2019; He et al., 2020). Indeed, the matrix factorisation technique can be interpreted as the search for the best fitted distribution for the users and items in a latent space. This is why in most cases (He et al., 2020; Rendle et al., 2020), the embeddings are initialised with a Gaussian distribution with a given mean (μ) and standard deviation (σ) e.g. $\mu = 0$ and $\sigma^2 = 0.1$. As discussed in Section 1, we expect the pre-training stage to capture hierarchical relations between users and items (Devlin et al., 2019). However, a standard Gaussian distribution cannot represent these learned complex relations from the pre-trained embeddings $\mathbf{E}_{pre}$, because its low representational power (Reynolds, 2009) limits its ability to convey the users' different preferences and their complex social relations, thereby potentially leading to an information loss. Hence, a mixture model is needed to leverage the possible multivariate Gaussian distributions learned from the pre-training stage and to avoid any possible information loss.

With the aforementioned proposal, we employ a well-developed statistical analysis tool, the Gaussian Mixture Model (GMM) (Reynolds, 2009), which can effectively decompose a multivariate Gaussian distribution into multiple (i.e. k) Gaussian distributions, where k is pre-defined. Specifically, GMM assumes that an observed data point $\mathbf{x}$ can be represented as a weighted sum of k Gaussian densities (Reynolds, 2009), calculated as follows:

$$p(\mathbf{x}|\lambda) = \sum_{i=1}^{k} \mathbf{w}_i \cdot g\left(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right), \tag{5}$$

where $\mathbf{x}$ is a continuous-valued feature vector, λ represents all the learnable parameters of GMM, i.e. $\lambda = \left\{w_i, \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right\}$, $\mathbf{w}$ is a k-dimensional vector containing the weight of each Gaussian density and the sum of $\mathbf{w}_i$ is equal to 1, i.e. $\sum_{i=1}^{k} \mathbf{w}_i = 1$, while $\boldsymbol{\mu}_i$ and $\boldsymbol{\Sigma}_i$ denote the mean vector and the covariance matrix, respectively.

Given the feature vector $\mathbf{x}$ conditioned on the mean vector $\boldsymbol{\mu}_i$ and the covariance matrix $\boldsymbol{\Sigma}_i$, we can calculate the corresponding Gaussian density as follows:

$$g\left(\mathbf{x} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right) = \frac{\exp\left\{-\frac{1}{2}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)' \Sigma_i^{-1}\left(\mathbf{x}-\boldsymbol{\mu}_i\right)\right\}}{(2\pi)^{D/2}\left|\Sigma_i\right|^{1/2}}, \tag{6}$$

where D is the dimension of the vector $\mathbf{x}$. In particular, the numerator is related to the Mahalanobis distance (McLachlan, 1999) (i.e. $\sqrt{\mathbf{x}-\boldsymbol{\mu}_i)'\Sigma_i^{-1}(\mathbf{x}-\boldsymbol{\mu}_i}$)), which represents the distance between $\mathbf{x}$ and $\boldsymbol{\sigma}$.

Since we assume that all the users' and items' embeddings correspond to combinations of Gaussian densities, we can extract meaningful information from these embeddings by analysing each pair of ($\boldsymbol{\mu}_i$, $\boldsymbol{\Sigma}_i$), as each pair represents each user's most important preferences or each item's most important characteristics. In order to extract such preferences and characteristics, we decompose the embeddings obtained from the pre-training stage in Section 3.2 as follows:

$$p(e_u|\lambda) = \sum_{i=1}^{k} \mathbf{w}_i \cdot g\left(\mathbf{e_u} \mid \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i\right), \tag{7}$$

where $\mathbf{e}_u$ is the pre-trained user embedding of the user u from $\mathbf{E}_{pre}$; k is the number of Gaussian distributions as defined above in Equation (5), which determines how many Gaussian densities should be obtained by decomposing the pre-trained embedding. A similar equation can also be applied to an item's embedding vector.

Algorithm 2: The fine-tuning process

Input: Rating matrix $\mathbf{R}$; Pre-trained embedding $\mathbf{E}_{pre}$; Pre-defined integer k.
Output: The recommended list for each user.
Inherit $\mathbf{E}_{pre}$ for initialising embeddings;
Use GMM to factorise $\mathbf{E}_{pre}$ according to Equation (8).
Randomly sample from k Gaussian distributions;
Generate reconstructed embeddings $\mathbf{E}_{recon}$ according to Equation (9).
Initialise other learnable parameters $\hat{\Theta}$;
while *not early-stopped* **do**
  $\zeta_{BCE} = 0$;
  for *each training instance in* $\mathbf{R}$ **do**
    Compute epoch loss $\nabla\zeta$ according to Equation (4);
  end
  $\zeta_{BCE} \leftarrow \zeta_{BCE} + \nabla\zeta$;
  Update $\hat{\Theta}$, $\mathbf{E}_{recon}$;
end
Do recommendation to find the recommended list based on the trained embeddings according to Equation (10);

Therefore, given the overall pre-trained embeddings $\mathbf{E}_{pre}$, we calculate all pairs of μ_i and Σ_i as follows:

$$\boldsymbol{\mu}, \boldsymbol{\Sigma} = \mathrm{GMM}(\mathbf{E}_{pre}, k), \tag{8}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are two matrices consisting of all (μ_i ,Σ_i) pairs for each user and item and the used GMM function is optimised by the EM algorithm.

After employing the GMM to those pre-trained embeddings, we obtain k pairs of μ and Σ for each user and item, from which we have enough statistical information to reconstruct the embeddings containing the social information encoded at the pre-training stage. For each user or item, we use the obtained (μ ,Σ) pairs to generate k Gaussian distributions, where we randomly sample the same number of elements from each distribution to reconstruct socially-aware embeddings. After that, we obtain the reconstructed embeddings $\mathbf{E}_{recon}$ as follows:

$$\mathbf{E}_{recon} = \mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\Sigma}), \tag{9}$$

where $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ are both obtained from Equation (8).

3.3.2. Model Training and Recommendation

After obtaining the reconstructed embeddings $\mathbf{E}_{recon}$, we use again the BCE loss function (He et al., 2017) to train the model but, this time, the model is initialised with $\mathbf{E}_{recon}$ instead of a random matrix. We concatenate the reconstructed embeddings $\mathbf{E}_{recon}$ with the randomly initialised embeddings to represent the users' preferences and items' characteristics. Therefore, the model is less likely to fall into the same relative optimised solution within the pre-training stage. To recommend items of interests to a user, we compute the dot product of the concatenated trained embeddings of this user with the trained embeddings of all items in the corpus. Hence, our proposed *Social-aware Gaussian Pre-trained* model (SGP) is devised to predict the interaction $\hat{y}_{ui}$ between user u and item i as follows:

$$\hat{y}_{ui} = (\mathbf{e}_{u-recon} \parallel \mathbf{e}_{u-rand}) \odot (\mathbf{e}_{i-recon} \parallel \mathbf{e}_{i-rand}), \tag{10}$$

where $\parallel$ denotes the concatenation and $\odot$ is the dot product. The obtained list of scores are then used to identify the items that a given user will be interested to interact with.

3.4. Discussion

Pre-training embeddings for the recommendation task has already been investigated in the literature. For example, to avoid the poorly performing local minima, both NCF (He et al., 2017) and CMN (Ebesu, Shen and Fang, 2018) apply the Generalised Matrix Factorisation (GMF) as a pre-training model to initialise the embedding of users and

Table 2
Statistics of datasets.

	Librarything	Epinions	Yelp
Users	60,243	114,738	215,471
Items	200,422	34,577	93,379
Interactions	930,053	110,671	1,506,039
Social edges	110,637	150,859	1,397,180
Interaction density (%)	0.008	0.003	0.007
Social density (%)	0.003	0.001	0.003

items. Specifically, to obtain the representations of users and items, a GMF model is trained using the weighted output of the embedding dot product:

$$\hat{\mathbf{R}}_{u,v} = \mathbf{W}\sigma\left(\mathbf{U}_u \odot \mathbf{V}_V\right), \tag{11}$$

where $\odot$ denotes the element-wise product of vectors, σ is an activation function and $\mathbf{W}$ is the trainable parameter.

However, existing models cannot leverage social relations during the pre-training stage. Hence, they may not improve the satisfaction of *cold-start* users. In comparison, Our SGP model can leverage social relations using a graph-based pre-training method such that *cold-start* users can benefit from friends with more interactions. In addition, SGP can particularly tackle the *extreme cold-start* problem by inferring the preferences of those *extreme cold-start* users as a combination of the preferences of their socially related friends.

3.5. Time Complexity Analysis

For the pre-training stage, we need to pre-compute the social similarity graph $\mathbf{S}_{sim}$ (see Equation (1) and the Laplacian matrix (see Equation (2)), where their complexities are $O(M^3)$ and $O(M^2)$, respectively. In addition, the complexity of the embedding updating function (see Equation (4)) is $O(M^2 \times |e|)$, where $|e|$ is the embedding dimension.

For the fine-tuning stage, we need to factorise $\mathbf{E}_{pre}$ using the GMM, which has a complexity of $O(t \times k \times L \times |e|)$, where t is the number of iterations for optimising the GMM function, k is the number of Gaussian distributions and L is the number of training samples. Finally, the recommendation step has a complexity of $O(M \times N)$, where M and N are the numbers of users and items, respectively.

Although the pre-training stage is more time-consuming than the fine-tuning stage, it produces reusable pre-trained embeddings. For a new user who joins the system through a friend's recommendation, the initial embedding can be generated based on the social relations. Since the pre-training stage does not need to be repeated as often as the fine-tuning stage, its time complexity is not a major concern for a practical deployment.

4. Datasets and Experimental Setup

To evaluate our proposed SGP model, we perform experiments on three public datasets: Librarything[2], Epinions[3] and Yelp[3]. These datasets are widely used in the recommender systems community. Librarything is a book review dataset, Epinions is a general customer review dataset, while Yelp is a venue check-in dataset. Table 2 provides the statistics of the three used datasets. In the following, we aim to address the following research questions:

RQ1. Can we use the GNN model to leverage the social information and generate pre-trained embeddings for both users and items, thereby improving the overall recommendation performance?

RQ2. Can we employ the Gaussian Mixture Model to distil information from the pre-trained embeddings and further enhance the recommendation performance?

RQ3. Does our SGP model help in alleviating the cold-start problem, especially for those extreme cold-start users?

RQ4. What is the impact of using the social relations on the pre-training stage of our SGP model?

RQ5. How do the embeddings dimension and different ranking cut-offs affect the recommendation performances of the pre-trained recommenders?

Below, we describe the 16 baselines used to evaluate the performance of SGP, the used evaluation methodology, and their corresponding experimental setup.

[2]http://cseweb.ucsd.edu/~jmcauley/datasets.html
[3]https://www.yelp.com/dataset

4.1. Baselines

We compare the performance of our SGP model to classical strong non-neural baselines as suggested by Dacrema, Cremonesi and Jannach (2019), as well as existing state-of-the-art neural models:

- **MF** (Rendle et al., 2020). This is the conventional matrix factorisation model, which can be optimised by the Bayesian personalised ranking (BPR (Rendle et al., 2009)) or the BCE losses. The regularisation includes the user bias, the item bias and the global bias.
- **SBPR** (Zhao, McAuley and King, 2014). SBPR is a classic model, which adds the social regularisation to the matrix factorisation method.
- **UserKNN** and **ItemKNN** (Sarwar, Karypis, Konstan and Riedl, 2001). Two neighbourhood-based models using collaborative user-user or item-item similarities.
- **SLIM** (Ning and Karypis, 2011). This is an effective and efficient linear model with a sparse aggregation method.
- **NCF** (He et al., 2017). The method is a CF model, which uses a generalised matrix factorisation method to generate pre-trained embeddings. An MLP module is also used in NCF to capture the nonlinear features from the interactions.
- **NGCF** (Wang et al., 2019). NGCF is devised to employ a multi-layer GCN on top of the user-item interaction graph to propagate the collaborative signal across multi-hops user-item neighbourhoods.
- **LightGCN** (He et al., 2020). Building on NGCF, LightGCN has fewer redundant neural components compared with the original NGCF model, which makes it more efficient and effective.
- **UltraGCN** (Mao et al., 2021). UltraGCN is a more efficient GNN-based recommender. It gains higher efficiency than LightGCN by skipping the message passing via a constraint loss.
- **SGL** (Wu et al., 2021). SGL leverages the self-supervised learning method to generate augmented views for nodes to enhance the model's robustness and accuracy.
- **VAE-CF** (Liang, Krishnan, Hoffman and Jebara, 2018). A state-of-the-art variational autoencoder-based collaborative filtering recommender system.
- **GraphRec** (Fan, Ma, Li, He, Zhao, Tang and Yin, 2019). This is the first GNN-based social recommendation method, which models both user-item and user-user interactions.
- **Diffnet++** (Wu et al., 2020a). This method is a graph-based deep learning recommender system, which uses the additional social links to enrich the user-item bipartite graph and improve the recommendation performance.
- **MPSR** (Liu, Zheng, Li, Zhang, Lin, Shen, Xiong and Wang, 2022). This is a recent model that uses GNN to construct hierarchical user preferences and assign friends' influences with different levels of trust from different perspectives.
- S^2**-MHCN** (Yu, Yin, Gao, Xia, Zhang and Viet Hung, 2021a). This is a self-supervised recommender system, which uses a hypergraph neural network to leverage the social relations between users.
- **SCDSR** (Luo et al., 2022). SCDSR is another self-supervised graph recommendation model that builds a heterogeneous graph using the social and interaction domains.

4.2. Evaluation Methodology

Following a common setup (Rendle et al., 2009; Xue, Dai, Zhang, Huang and Chen, 2017), we use a leave-one-out evaluation strategy to split the interactions of each dataset into training, validation and testing sets. To speed up the evaluation, we adopt the sampled metrics (He et al., 2017; Wang et al., 2019; Rendle et al., 2020), which randomly sample a small set of the non-interactive items as negative items (rather than take all the non-interactive items as negatives) of the validation and testing sets, and evaluate the metric performance on this smaller set. Here, we sample

100 negative items for each user in the testing sets for evaluation (He et al., 2017; Rendle et al., 2020). However, different from prior works (He et al., 2017; Rendle et al., 2020) that only use one oracle testing set per dataset with the sampled negative items, we construct 10 different testing sets with different sampled negative items for each dataset using different random seeds, in order to reduce the evaluation bias on some specific testing negatives (Krichene and Rendle, 2020). Hence, the reported performance of each run is based on the average of the 10 testing sets[4]. In order to answer **RQ1**, we compare our SGP model with all baselines in terms of Normalised Discounted Cumulative Gain@10 (NDCG), Recall@10 and Mean Average Precision@10 (MAP). We also compare the SGP model with both its pre-training and fine-tuning stages to a variant where only the pre-training stage is used (called SGP (Pre-training)), so as to address **RQ2**. All models are implemented with PyTorch using the Beta-RecSys open source framework (Meng, McCreadie, Macdonald, Ounis, Liu, Wu, Wang, Liang, Liang, Zeng et al., 2020). We use the Adam (Kingma and Ba, 2015) optimiser for all the neural network models' optimisations. To tune all hyper-parameters, we apply a grid search on the validation set, where the learning rate is tuned in $\left\{10^{-2}, 10^{-3}, 10^{-4}\right\}$; the latent dimension in $\{32, 64, 128\}$ and the L_2 normalisation in $\left\{10^{-2}, ..., 10^{-5}\right\}$. The node dropout technique is used in the NGCF, LightGCN, UltraGCN, MPSR and GraphRec models as well as our proposed SGP model. The dropout ratios vary amongst $\{0.3, 0.4, ..., 0.8\}$ as suggested in (van den Berg, Kipf and Welling, 2018). To control how many Gaussian distributions are extracted from the pre-trained embeddings, we vary the number of pre-defined multivariate Gaussian distributions k in Equation (7) in $\{2, 4, 6, 8, 10\}$. Note that due to the limit of the latent dimension, further increases in the k value might result in less data extracted from each pre-trained embedding. For each k value, we run our SGP model for 50 times with different random seeds and we plot the results on the three datasets as a box plot, where we illustrate not only the mean values but also the variance across different random seeds. For a fair comparison with (He et al., 2017, 2020; Wang et al., 2019; Wu et al., 2020a), we set the number of neural network layers of the models including NCF, NGCF, Diffnet++, LightGCN, UltraGCN, SGL, GraphRec and SGP to three. For the non-neural models, namely SBPR, MF, UserKNN, ItemKNN and SLIM, we tune them within the same range of learning rates and L_2 normalisations used for the neural baselines, while for the rest of parameters we follow the same implementation details as suggested in (Dacrema et al., 2019).

To answer **RQ3**, we further examine the ability of our proposed SGP model to alleviate the *cold-start* problem, especially for those users who newly registered on the sites. In particular, we first compare the performances of our SGP model to the best performing baseline across different groups of users who have less than {5, 10, 15, 20} interactions, respectively. Second, to simulate the *extreme cold-start* situation when a user starts using an app that was suggested by his/her friends, we select those users who have social relations but less than five interactions. We define these users as the *extreme cold-start*[5] users and we remove all their interactions, so that the situation of newly registered users (no historical interaction) is simulated. Hence, through this defined *extreme cold-start* setup, we aim to recommend relevant items to those newly registered users solely based on their social relations.

In order to tackle **RQ4,** we conduct an ablation study to determine the effect of the social relations in our proposed SGP model and the Diffnet++ model. In this ablation study, we randomly drop {20%,40%,60%,80%} of social relations from both models and measure the resulting recommendation performance across the three used datasets, in order to determine if the performance improvements are indeed gained from the social-aware pre-training. To answer **RQ5**, we provide a detailed analysis on the largest dataset (i.e. Yelp) to evaluate the performances of SGP and LightGCN on different embedding dimensions and different cut-offs for the recommended items. Additionally, in order to directly observe the effect of social relations in the latent space, we use the t-distributed stochastic neighbour embedding (t-SNE) technique (Van der Maaten and Hinton, 2008) to visualise the final embeddings obtained by our SGP model, in comparison to the embeddings obtained by a classic MF model.

5. Results Analysis

In this section, we report the experimental results and answer the five research questions in turn.

[4]We have also employed an evaluation methodology where any potential bias is avoided. In this evaluation, we make use of the full set of negative items. We observed similar experimental results and conclusions to those shown in Table 3. Hence, our use of 10 different testing sets enhances evaluation efficiency but also sufficiently mitigates against any potential evaluation bias stemming from the negative item sampling.

[5]We sampled users with less than 5 interactions for the simulation because at least 3 interactions are needed for the train/valid/test set, and in order to keep enough users in the evaluation pool.

Table 3
Performances of SGP and other baselines on the three used datasets. All metrics are computed at rank cutoff 10. The best and second best performances are highlighted in boldface and underlined, respectively; * denotes a significant difference between the performance of SGP and that of the baselines according to the paired t-test with the Holm-Bonferroni correction for $p < 0.01$. The 'Social' column indicates whether a model uses social relations or not.

	Social	Epinions			Librarything			Yelp		
		NDCG	Recall	MAP	NDCG	Recall	MAP	NDCG	Recall	MAP
NCF	✗	0.0819*	0.1662*	0.0585*	0.3132*	0.4971*	0.2304*	0.2504*	0.4109*	0.1908*
NGCF	✗	0.0816*	0.1668*	0.0589*	0.2974*	0.4894*	0.2498*	0.2378*	0.3904*	0.1794*
LightGCN	✗	0.0830*	0.1723*	<u>0.0615*</u>	0.3310*	0.5081*	0.2484*	0.2735*	0.4304*	0.2194*
UltraGCN	✗	0.0825*	0.1700*	0.0603*	<u>0.3313*</u>	<u>0.5083*</u>	0.2480*	0.2598*	0.4002*	0.2011*
SGL	✗	0.0831*	0.1720*	0.0610*	0.3216*	0.4982*	0.2417*	0.2632*	0.4100*	0.2098*
VAE-CF	✗	0.0710*	0.1424*	0.0475*	0.3003*	0.4934*	0.2302*	0.2100*	0.3715*	0.1639*
Diffnet++	✓	0.0819*	0.1678*	0.0549*	0.3011*	0.4873*	0.2259*	0.2589*	0.4184*	0.1988*
GraphRec	✓	0.0810*	0.1661*	0.0527*	0.2997*	0.4807*	0.2248*	0.2478*	0.4097*	0.1901*
MPSR	✓	0.0821*	0.1692*	0.0589*	0.3203*	0.4927*	0.2345*	0.2597*	0.4032*	0.1925*
SBPR	✓	0.0791*	0.1571*	0.0509*	0.2997*	0.4931*	0.2300*	0.2398*	0.3937*	0.1808*
S^2-MHCN	✓	0.0824	0.1709	0.0550	0.3099	0.5185	0.2356	0.2600	0.4303	0.2223
SCDSR	✓	<u>0.0835</u>	<u>0.1733</u>	0.0604	0.3276	0.5020	<u>0.2501</u>	<u>0.2801</u>	<u>0.4406</u>	<u>0.2208</u>
MF	✗	0.0720*	0.1481*	0.0484*	0.2903*	0.4893*	0.2291*	0.2011*	0.3348*	0.1698*
UserKNN	✗	0.0752*	0.1678*	0.0497*	0.3123*	0.4987*	0.2345*	0.2297*	0.3797*	0.1758*
ItemKNN	✗	0.0743*	0.1667*	0.0486*	0.2977*	0.4872*	0.2139*	0.2238*	0.3709*	0.1712*
SLIM	✗	0.0719*	0.1522*	0.0497*	0.2918*	0.4821*	0.2298*	0.2098*	0.3407*	0.1766*
SGP (Pre-training)	✓	0.0725	0.1498	0.0497	0.2953	0.4913	0.2284	0.2201	0.3897	0.1798
SGP	✓	**0.0876**	**0.1794**	**0.0657**	**0.3569**	**0.5431**	**0.2647**	**0.2972**	**0.4631**	**0.2347**
%Improv.		4.9	3.5	6.8	7.7	6.9	5.8	6.1	5.1	6.3

Table 4
NDCG@10 performances of our SGP model in comparison to the LightGCN baseline across different user groups where n is the number of users' interactions; * denotes a significant difference versus LightGCN (paired t-test, p<0.01).

Dataset	Model	NDCG@10 n=5	n=10	n=15	n=20	n=all
Epinions	SGP	0.0763*	0.0812*	0.0857*	0.0864*	0.0876*
	LightGCN	0.0708	0.0751	0.0810	0.0818	0.0830
	%Improv.	7.77	8.12	5.80	5.62	5.50
Librarything	SGP	0.3014*	0.3354*	0.3410*	0.3554*	0.3569*
	LightGCN	0.2631	0.2821	0.3101	0.3275	0.3310
	%Improv.	14.5	15.9	9.96	8.52	7.80
Yelp	SGP	0.1987*	0.2435*	0.2669*	0.2848*	0.2972*
	LightGCN	0.1671	0.2086	0.2381	0.2590	0.2735
	%Improv.	18.9	16.7	12.1	9.97	8.71

5.1. RQ1: Pre-trained Recommendation Performances

In order to answer **RQ1**, we use Table 3 to report the overall performance of our SGP model in comparison to 13 other baselines and the pre-training stage (the first stage only of the SGP model) in terms of 3 different metrics, namely NDCG, Recall and MAP. Comparing the performance of SGP (Pre-training) with other baselines, we can conclude that the pre-training stage itself cannot outperform all baselines. However, through the information distillation stage when we use randomly initialised embeddings concatenated with Multivariate Gaussian distributions extracted from the pre-trained embeddings, our SGP model achieves the best performance, constantly and significantly outperforming all other baselines in terms of all metrics on three used datasets. These results demonstrate that solely employing the GNN model with the available social relations is not sufficient to enhance the recommendation performance. This is likely because the social information should not be considered equally to the interaction information, since the interaction information makes the actual ground truth when inferring the users' main preferences and next items

Table 5
Performances of SGP (Pre-training) on the *extreme cold-start* users in comparison with the random and popularity-based baselines. In the table, SGP is the only approach where interactions are used. * and ↑ denote significant differences compared to the random and popularity baselines, respectively (paired t-test, p<0.01).

	Epinions			Librarything			Yelp		
	NDCG	Recall	MAP	NDCG	Recall	MAP	NDCG	Recall	MAP
Random	0.0676	0.1334	0.0403	0.0998	0.1938	0.0707	0.0887	0.1806	0.0615
Popularity	0.0712	0.1448	0.0423	0.1693	0.3011	0.1082	0.1937	0.3219	0.1287
SGP	$0.0870^{*\uparrow}$	$0.1691^{*\uparrow}$	$0.0589^{*\uparrow}$	$0.3324^{*\uparrow}$	$0.5134^{*\uparrow}$	$0.2958^{*\uparrow}$	$0.2972^{*\uparrow}$	$0.4631^{*\uparrow}$	$0.2347^{*\uparrow}$
SGP (Pre-training)	$0.0728^{*\uparrow}$	$0.1583^{*\uparrow}$	$0.0450^{*\uparrow}$	$0.2029^{*\uparrow}$	$0.3360^{*\uparrow}$	$0.1382^{*\uparrow}$	$0.2279^{*\uparrow}$	$0.3469^{*\uparrow}$	$0.1488^{*\uparrow}$

of interest. By reusing these pre-trained embeddings concatenated with randomly initialised embeddings, our SGP model can markedly and significantly enhance the recommendation performance. It is of note that the performances of all the evaluated models on the Epinions dataset are lower than on the Librarything and Yelp datasets. However, these performances are in line with those reported in the literature (e.g. NDCG@10 $\approx$ 0.3 on the Librarything dataset (Palumbo, Rizzo, Troncy, Baralis, Osella and Ferro, 2018; Valcarce, Landin, Parapar and Barreiro, 2019; Mauro, Ardissono and Hu, 2019); NDCG@10 < 0.1 on the Epinions dataset (Abdollahpouri, Burke and Mobasher, 2019)). These differences may be explained by the differing densities of user-item interactions in the used datasets (see Table 2). Additionally, by comparing other graph-based models, we observe that those more recent models such as SGL and UltraGCN do sometimes outperform the LightGCN model. However, LightGCN can still outperform SGL and UltraGCN for most of the times as shown in Table 3. This observation is likely related to our used data split, where we use 10 different testing sets to avoid the oracle testing set. In other words, our results suggest that the recent graph-based models have not achieved consistent and robust improvements over the LightGCN model. In addition to those graph-based baseline models, our SGP also significantly outperforms other social-aware recommender systems, including S^2-MHCN and SCDSR. This indicates that a social graph pre-training technique is more effective than the self-supervised learning technique on the recommendation task. Overall, in answer to **RQ1**, we can conclude that using the GNN model to leverage the social relations and generate pre-trained embeddings can improve the recommendation performance compared with SGP (Pre-training) and 16 competitive (neural and non-neural) baselines.

5.2. RQ2: GMM Information Distillation

To address **RQ2**, we show a box plot of our SGP model on the 3 used datasets across different number of pre-defined multivariate Gaussian distributions, k, in terms of NDCG@10, where for each k value, the model is trained and evaluated 50 times with different random seeds. In Figure 2, the max and min values for each set of experiments are shown as two bars at the top and bottom of each box, respectively. The mean value of each set of experiments is shown as an orange line lying in the middle of each box. We also report the best mean for each dataset in Table 3 (i.e. $k = 6$ for the Epinions dataset and $k = 8$ for the Librarything and Yelp datasets). Figure 2 shows that our SGP model only achieves better performances when k is larger than 4, whereas for all datasets when $k = 2$ or 4, the SGP model has a lower performance than several baselines. This can be explained by the fact that the users' preferences are hard to be estimated with simple distributions. Indeed, usually the users' preferences are formed by combinations of distributions, which cannot be easily factorised with 2 to 4 factors. Therefore, a small number of Gaussian distributions is not sufficient enough to represent the users' preferences. However, we also observe a performance degradation when k is too large. This is likely because the latent dimension has a limited size (usually up to a few hundreds), while each reconstructed embedding is a sample from the multiple extracted Gaussian distributions. Therefore, when k becomes larger, elements sampled from each distribution become fewer, thereby leading to a loss in the accuracy of the representation of its intended original factor. For example, when the latent dimension is 100, if $k = 10$ is applied, only 10 elements are sampled from each Gaussian distribution. Moreover, when k is larger, the performance of our SGP model is relatively stable. This demonstrates that our SGP model is effective in distilling information from the pre-trained embeddings given that enough Gaussian distributions are employed i.e. when k is sufficiently large, the model stabilises and shows less variance. In addition, we observe that the performance of SGP in terms of NDCG@10 varies across different datasets. For example, when $k = 4$, SGP is relatively less effective than the other configurations on the Librarything dataset while SGP is dramatically improved when $k \geq 6$ on the Epinions dataset.

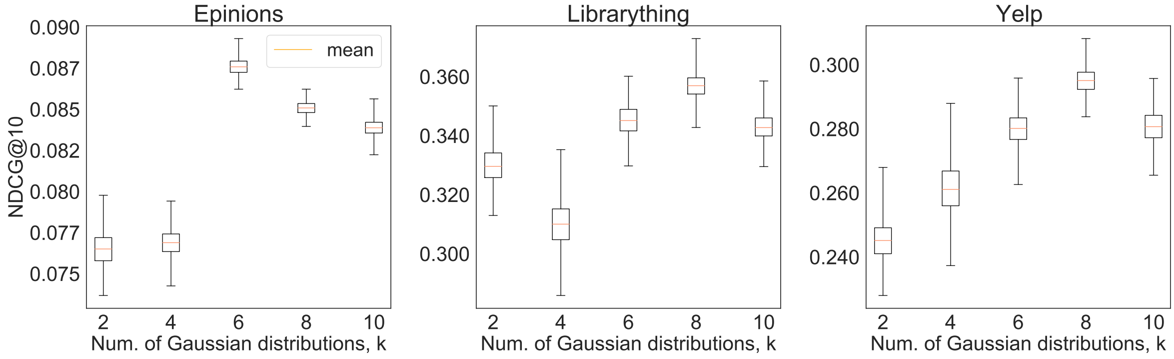

Figure 2: Max/Min/Mean values of NDCG@10 for SGP on 3 datasets with different number of Gaussian distributions.

As shown in Table 2, different datasets have different social densities leading to different structures of social networks. Therefore, these structures of the social networks can affect the usefulness of the social relations thereby also affecting the performance of social-aware recommenders such as SGP. Overall, in answer to **RQ2**, we can conclude that the GMM can be used to effectively distill information from the pre-trained embeddings. We also suggest preferable k values, which can be used to enhance the recommendation performance.

5.3. RQ3: *Cold-start* Performances

To address **RQ3**, in Table 4, we examine the performance of our SGP model for different groups of users who have less than {5, 10, 15, 20} interactions, respectively, in comparison to the best baseline, LightGCN, in terms of NDCG@10. From the table, we note that our SGP model overall significantly outperforms LighGCN, while users with less than 10 interactions particularly benefit from our model compared with the other groups of users. Overall, it is reasonable to observe that *cold-start* users benefit more from our model because when their interaction information is too sparse, incorporating more social information will likely enable the SGP model to predict their possible unknown preferences. However, users with sufficient interactions tend to have their preferences accurately captured by the recommender systems, therefore adding more social relations may not be beneficial for them. Indeed, from Table 4, we observe that there is a clear decrease in the reported percentage improvement when we consider the group of users who have less than 10 interactions in comparison to those users who have more than 15 interactions.

Table 5 shows a comparison of our SGP model with a random recommender and a popularity-based recommender for the *extreme cold-start* users case. The random and popularity-based recommenders are two commonly used baselines when no interaction data is available. Here, we aim to simulate the situation when users register to an App or a Web service following the suggestions of their friends. In this case, the model only knows about the users' friends while it does not have access to the historical interactions. Instead of making random recommendations or only recommending popular items, our SGP model generates embeddings by constructing multivariate Gaussian distributions by evenly sampling elements from their friends' embeddings, which is also produced by the pre-training stage of our SGP model. By comparing our proposed SGP model with its first pre-training stage only (denoted by SGP (Pre-training)) with both baselines and the full SGP model for the *extreme cold-start users*, we find that SGP (Pre-training) significantly outperforms both the random and the popularity-based recommenders. On the other hand, it is reasonable and natural that when no interactions are observed and no training is conducted, SGP (Pre-training) is far worse than the full SGP model. However, SGP (Pre-training) significantly outperforms both the random and the popularity-based recommenders on the Librarything and Yelp datasets and is comparable to the results of SGP on the Epinions dataset. Overall, in answer to **RQ3**, we can conclude that our proposed method SGP is effective at tackling the *cold-start* problem and is particularly useful in alleviating the practical extreme cold-start issue.

5.4. RQ4: Impact of Social Relations

In order to determine the effect of social relations and to answer **RQ4**, we conduct an ablation study where we randomly dropout different proportions of social relations. Figure 3 shows how the performance of our SGP model and Diffnet++ model are affected when different proportions of social relations are randomly masked out. From this figure, we can clearly observe a trend that when more social relations are masked during the pre-training, the more the recommendation performances of SGP and Diffnet++ are degraded across three datasets. This trend reveals that the social relations do indeed help the SGP model to achieve a better pre-training thereby enhancing the final recommendation performance. However, we also observe some variance in the performance on the Epinions dataset, compared with the consistent decline of performance on the Librarything and Yelp datasets. This is because the raw data of the Epinions dataset provides bidirectional social relations i.e. both the 'trust' and 'trustedby' relations are given. Since our current SGP model cannot distinguish between these bidirectional relations, for the sake of simplicity, we unify these two types of relations as one unidirectional social network to fit our implementation. Although unifying the bidirectional relations does bring an overall performance improvement to SGP over other baselines, this unifying method itself is not optimal and can possibly induce noise, because the social influences are not bidirectionally equal. By comparing the Diffnet++ model with SGP across three datasets over different dropout ratios, we observe consistent performances improvement from our proposed model, which further justifies our previous results. Therefore, from our conducted ablation study, in answer to **RQ4**, we can conclude that using the social relations on the pre-training stage can help enhance the recommendation performance of our SGP model. Furthermore, we postulate that the performance can be further enhanced by enabling our current SGP model to distinguish among bidirectional social relations. We leave the adequate integration of bidirectional social relations into our SGP model to future work.

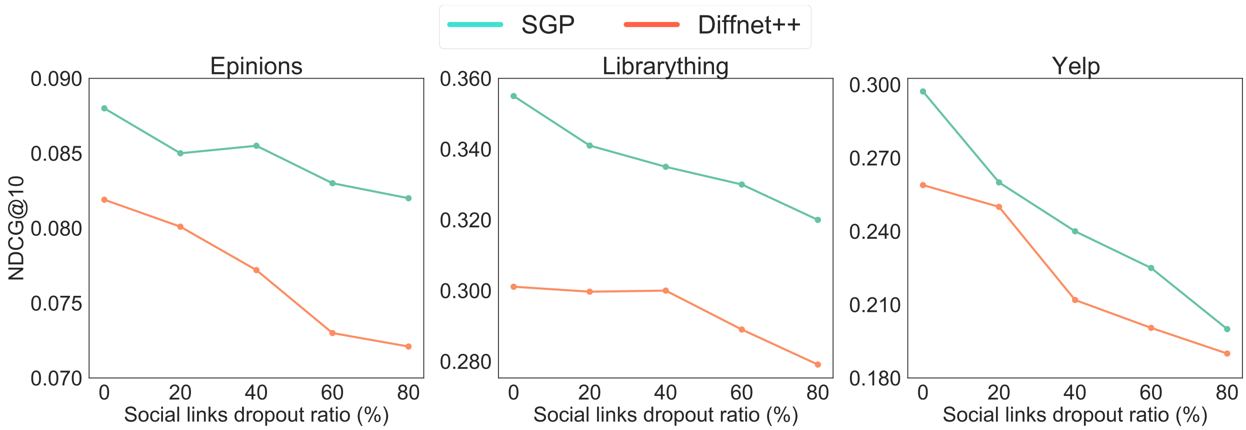

Figure 3: An ablation study of performances of SGP and Diffnet++ (different proportions of social relations are masked out).

5.5. RQ5: Hyperparameter Analysis

In this section, we aim to answer **RQ5** by examining how the performance of SGP and that of the second-best baseline LightGCN are affected by different recommendation cut-offs and embedding dimensions. First, we plot when the recommendation lists are generated with different cut-offs in Figure 4a. From this figure, we can observe that our SGP consistently outperforms LightGCN across different cut-offs. Specifically, SGP mainly surpasses LightGCN for larger cut-offs (i.e. when cut-offs $\geq$ 10). This is due to the fact that we consider social relations as side information, and they are only leveraged during the pre-training stage. As a result, those items that are easy to predict will be preserved as top-ranked items, while social relations play an important role for our SGP model in obtaining a higher effectiveness at lower rank cutoffs. For example, in the venue recommendation scenario, users may visit venues suggested by their friends when travelling to different countries. When these venues are in the test set, a general recommender relying only on the interaction information will unlikely rank these venues at the top of the ranking list for such users. Our proposed SGP model is likely to benefit this case if the users' friends have visited/liked these venues before. Specifically, SGP gives higher scores to those venues visited/recommended by each user's friends. As a result, those venues lowly ranked by other recommenders will have a higher chance to appear in the top-10/top-20/top50 ranking lists as shown in Figure 4a. Here, we must emphasise the difference between the results shown in Figure 4a and Table 4 to avoid a possible confusion. In Table 4, we have reported performances across different user groups defined by the number of historical interactions of each user. This is different from what we plot in this section, which is based on different cut-offs. Figure 4b shows how the NDCG@10 measure is affected when different embedding dimensions

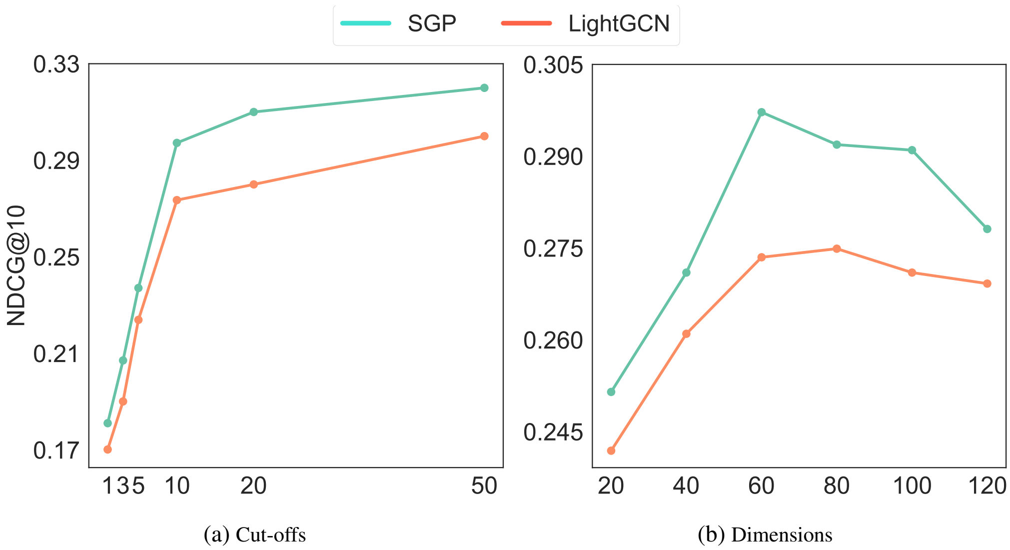

Figure 4: A performance comparison between SGP and LightGCN over different dimensions and different cut-offs on the Yelp dataset.

are applied to SGP and LightGCN. This figure demonstrates that our SGP can bring consistent improvements over the baseline for different embedding dimensions. To conclude on **RQ5**, our SGP model can constantly outperform the strong LightGCN baseline when different hyperparameters are applied.

5.6. The Embedding Visualisation

In this section, we aim to analyse how our SGP model affects the users' embeddings in the latent space, compared to the embeddings obtained from a classic MF model that does not encapsulate social relations. We visualise all users' embeddings of the Librarything dataset[6], trained by the SGP model in comparison with embeddings trained by the MF model[7] using the t-distributed stochastic neighbour embedding (t-SNE) approach. Figure 5(a) shows the t-SNE for MF, while Figure 5(b) provides the t-SNE for SGP. In both plots, we highlight three anchor users (represented as yellow/green/red dots), along with their corresponding friends (triangles) and their target items (stars). Both the green and orange anchor users are fortunate to have their friends close to their target items, hence, these two anchor users are pulled closer to their target items, as shown in Figure 5(b). In contrast, in Figure 5 (a), these two anchor users are clustered far apart from their target items by MF, due to the fact that social relations are not considered by MF. For the red user's case, he/she has a dissimilar friend, who is located relatively far away from the target item and his/her friends. Our SGP model can still handle this case by relocating the red user to the space between this dissimilar friend and two other similar friends, thereby bringing this user closer to the target item. Through the provided three examples of users, we illustrated different situations where users might possibly benefit from our SGP model, thereby improving the recommendation performances as observed in the reported results across three datasets.

6. Conclusions

In this paper, we explored how to leverage a GNN model to generate pre-trained embeddings using the existing social relations among users. Next, we used the Gaussian Mixture Model to carefully extract prior knowledge contained in those pre-trained embeddings for the subsequent fine-tuning and recommendations. Our proposed *Social-aware Gaussian Pre-trained* (SGP) model can significantly outperform competitive baselines, as demonstrated by the extensive experiments conducted on three public datasets. Furthermore, a detailed user analysis showed that

[6]The Librarything dataset is a less sparse dataset with a higher or equal social density compared to the Epinions and Yelp datasets therefore, for illustration purposes, we have more users to choose from. However, note that we do nevertheless observe similar trends on the Epinions and Yelp datasets

[7]The MF model is chosen because it is also an embedding-based method and is not socially aware, and therefore can offer us a clear comparison between a social-aware model and a non-social model.

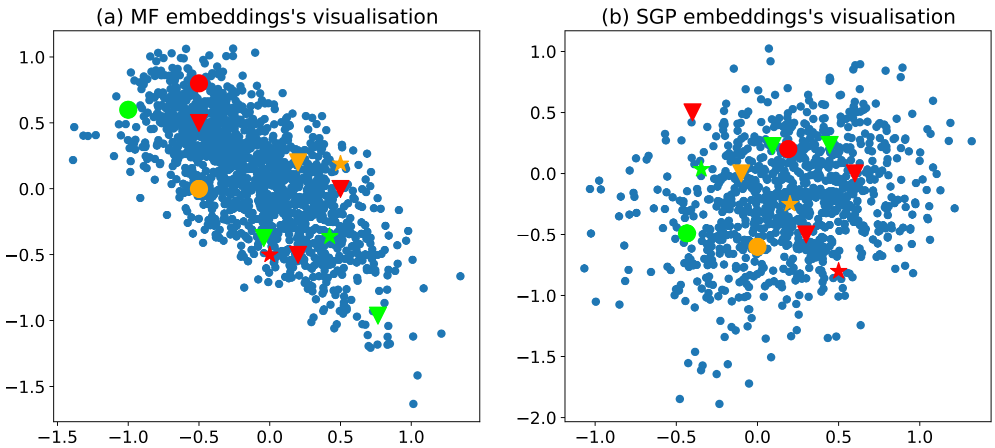

Figure 5: The t-SNE plot of all users' embeddings of the Librarything dataset obtained from the MF model (a) and our SGP model (b), where the red dot represents an anchor user, the red triangles are this user's friends and the red start is the target item. The similar configuration is also applied to another two users with their corresponding friends and target items, which are plotted with the colour of green and orange, respectively.

by incorporating the social relations, users who have less than 10 interactions particularly benefit from our SGP model. Moreover, we showed that our SGP model can practically serve *extreme cold-start* users with reasonable recommendations when it only knows about the friend's preferences of these newly registered users. Finally, we used an ablation study to examine the effect of social relations on our proposed model and a hyperparameter analysis to study the effects of different cut-offs and embedding dimensions, followed by the visualisation of the generated embeddings to further illustrate how our proposed model could benefit recommendations. As future work, we aim to investigate how to leverage the bidirectional nature of social relations so that we can alleviate the issue of 'trust'/'trustedby' relations, as mentioned in Section 5.4. In addition, we aim to leverage more effective fine-tuning methods for SGP, such as the contrastive graph learning method (Yang et al., 2023), instead of the plain training method.

References

Abdollahpouri, H., Burke, R., Mobasher, B., 2019. Managing popularity bias in recommender systems with personalized re-ranking, in: Proceedings of AAAI.

van den Berg, R., Kipf, T.N., Welling, M., 2018. Graph convolutional matrix completion, in: Proceedings of SIGKDD.

Brown, T.B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D.M., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., Amodei, D., 2020. Language models are few-shot learners, in: Proceedings of NeurIPS.

Camacho, L.A.G., Alves-Souza, S.N., 2018. Social network data to alleviate cold-start in recommender system: A systematic review. Information Processing & Management 54, 529–544.

Chen, L., Wu, L., Hong, R., Zhang, K., Wang, M., 2020. Revisiting graph based collaborative filtering: A linear residual graph convolutional network approach, in: Proceedings of the AAAI conference on artificial intelligence, pp. 27–34.

Dacrema, M.F., Cremonesi, P., Jannach, D., 2019. Are we really making much progress? a worrying analysis of recent neural recommendation approaches, in: Proceedings of RecSys.

Devlin, J., Chang, M., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for language understanding, in: Proceedings of NAACL-HLT.

Ebesu, T., Shen, B., Fang, Y., 2018. Collaborative memory network for recommendation systems, in: The 41st international ACM SIGIR conference on research & development in information retrieval, pp. 515–524.

Elahi, M., Kholgh, D.K., Kiarostami, M.S., Saghari, S., Rad, S.P., Tkalčič, M., 2021. Investigating the impact of recommender systems on user-based and item-based popularity bias. Information Processing & Management 58, 102655.

Erhan, D., Courville, A.C., Bengio, Y., Vincent, P., 2010. Why does unsupervised pre-training help deep learning?, in: Proceedings of AISTATS.

Fan, W., Ma, Y., Li, Q., He, Y., Zhao, E., Tang, J., Yin, D., 2019. Graph neural networks for social recommendation, in: Proceedings of WWW, pp. 417–426.

González, J.Á., Hurtado, L.F., Pla, F., 2020. Transformer based contextualization of pre-trained word embeddings for irony detection in twitter. Information Processing & Management 57, 102262.

Gou, J., Yu, B., Maybank, S.J., Tao, D., 2021. Knowledge distillation: A survey. International Journal of Computer Vision 129, 1789–1819.

Hamilton, W., Ying, Z., Leskovec, J., 2017. Inductive representation learning on large graphs, in: Proceedings of NeurIPS.

Hao, B., Zhang, J., Yin, H., Li, C., Chen, H., 2021. Pre-training graph neural networks for cold-start users and items representation, in: Proceedings of WSDM.

He, X., Deng, K., Wang, X., Li, Y., Zhang, Y., Wang, M., 2020. LightGCN: Simplifying and powering graph convolution network for recommendation, in: Proceedings of SIGIR.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., Chua, T.S., 2017. Neural collaborative filtering, in: Proceedings of WWW.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: Proceedings of ICLR.

Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: Proceedings of ICLR.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer 42.

Krichene, W., Rendle, S., 2020. On sampled metrics for item recommendation, in: Proceedings of SIGKDD.

Li, J., Wang, Y., McAuley, J., 2020. Time interval aware self-attention for sequential recommendation, in: Proceedings of WSDM.

Li, X., Liu, Z., Guo, S., Liu, Z., Peng, H., Philip, S.Y., Achan, K., 2021. Pre-training recommender systems via reinforced attentive multi-relational graph neural network, in: Proceedings of IEEE BigData.

Liang, D., Krishnan, R.G., Hoffman, M.D., Jebara, T., 2018. Variational autoencoders for collaborative filtering, in: Proceedings of WWW.

Litjens, G., Kooi, T., Bejnordi, B.E., Setio, A.A.A., Ciompi, F., Ghafoorian, M., Van Der Laak, J.A., Van Ginneken, B., Sánchez, C.I., 2017. A survey on deep learning in medical image analysis. Medical Image Analysis 42.

Liu, H., Zheng, C., Li, D., Zhang, Z., Lin, K., Shen, X., Xiong, N.N., Wang, J., 2022. Multi-perspective social recommendation method with graph representation learning. Neurocomputing 468, 469–481.

Liu, S., Ounis, I., Macdonald, C., Meng, Z., 2020. A heterogeneous graph neural model for cold-start recommendation, in: Proceedings of SIGIR.

Luo, S., Yang, Y., Zhang, K., Sun, P., Wu, L., Hong, R., 2022. Self-supervised cross domain social recommendation, in: Proceedings of the 8th International Conference on Computing and Artificial Intelligence, pp. 286–292.

Ma, H., Zhou, D., Liu, C., Lyu, M.R., King, I., 2011. Recommender systems with social regularization, in: Proceedings of WSDM.

Ma, X., Guo, J., Zhang, R., Fan, Y., Ji, X., Cheng, X., 2021. Prop: Pre-training with representative words prediction for ad-hoc retrieval, in: Proceedings of WSDM.

Van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. Journal of Machine Learning Research 9.

Mao, K., Zhu, J., Xiao, X., Lu, B., Wang, Z., He, X., 2021. Ultragcn: Ultra simplification of graph convolutional networks for recommendation, in: Proceedings of CIKM.

Mauro, N., Ardissono, L., Hu, Z.F., 2019. Multi-faceted trust-based collaborative filtering, in: Proceedings of UMAP.

McLachlan, G.J., 1999. Mahalanobis distance. Resonance 4, 20–26.

Meng, Z., McCreadie, R., Macdonald, C., Ounis, I., Liu, S., Wu, Y., Wang, X., Liang, S., Liang, Y., Zeng, G., et al., 2020. BETA-Rec: Build, Evaluate and Tune Automated Recommender Systems, in: Proceedings of RecSys.

Ning, X., Karypis, G., 2011. Slim: Sparse linear methods for top-n recommender systems, in: Proceedings of ICDM.

Palumbo, E., Rizzo, G., Troncy, R., Baralis, E., Osella, M., Ferro, E., 2018. Translational models for item recommendation, in: Proceedings of ESWC.

Park, W., Kim, D., Lu, Y., Cho, M., 2019. Relational knowledge distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 3967–3976.

Ramchoun, H., Idrissi, M.A.J., Ghanou, Y., Ettaouil, M., 2016. Multilayer perceptron: Architecture optimization and training. Journal of Interactive Multimedia and Artificial Intelligence 4.

Rendle, S., Freudenthaler, C., Gantner, Z., Schmidt-Thieme, L., 2009. BPR: Bayesian personalized ranking from implicit feedback, in: Proceedings of UAI.

Rendle, S., Krichene, W., Zhang, L., Anderson, J.R., 2020. Neural collaborative filtering vs. matrix factorization revisited, in: Proceedings of RecSys.

Reynolds, D.A., 2009. Gaussian mixture models. Encyclopedia of biometrics 741.

Sarwar, B., Karypis, G., Konstan, J., Riedl, J., 2001. Item-based collaborative filtering recommendation algorithms, in: Proceedings of WWW.

Seyedhoseinzadeh, K., Rahmani, H.A., Afsharchi, M., Aliannejadi, M., 2022. Leveraging social influence based on users activity centers for point-of-interest recommendation. Information Processing & Management 59, 102858.

Shen, Y., Wu, Y., Zhang, Y., Shan, C., Zhang, J., Letaief, B.K., Li, D., 2021. How powerful is graph convolution for recommendation?, in: Proceedings of CIKM.

Turc, I., Chang, M.W., Lee, K., Toutanova, K., 2019. Well-read students learn better: On the importance of pre-training compact models. arXiv preprint arXiv:1908.08962 .

Valcarce, D., Landin, A., Parapar, J., Barreiro, Á., 2019. Collaborative filtering embeddings for memory-based recommender systems. Engineering Applications of Artificial Intelligence 85.

Van Engelen, J.E., Hoos, H.H., 2020. A survey on semi-supervised learning. Machine Learning 109.

Wang, C., Samari, B., Siddiqi, K., 2018. Local spectral graph convolution for point set feature learning, in: Proceedings of ECCV.

Wang, X., He, X., Wang, M., Feng, F., Chua, T.S., 2019. Neural graph collaborative filtering, in: Proceedings of SIGIR.

Wen, Y., Guo, L., Chen, Z., Ma, J., 2018. Network embedding based recommendation method in social networks, in: Proceedings of WWW.

Wong, C.M., Feng, F., Zhang, W., Vong, C.M., Chen, H., Zhang, Y., He, P., Chen, H., Zhao, K., Chen, H., 2021. Improving conversational recommender system by pretraining billion-scale knowledge graph, in: Proceedings of ICDE.

Wu, F., Souza, A., Zhang, T., Fifty, C., Yu, T., Weinberger, K., 2019. Simplifying graph convolutional networks, in: Proceedings of ICML.

Wu, J., Wang, X., Feng, F., He, X., Chen, L., Lian, J., Xie, X., 2021. Self-supervised graph learning for recommendation, in: Proceedings of SIGIR.

Wu, L., Li, J., Sun, P., Hong, R., Ge, Y., Wang, M., 2020a. Diffnet++: A neural influence and interest diffusion network for social recommendation. IEEE Transactions on Knowledge and Data Engineering .

Wu, L., Sun, P., Hong, R., Fu, Y., Wang, X., Wang, M., 2018. Socialgcn: An efficient graph convolutional network based model for social recommendation. CoRR abs/1811.02815. URL: http://arxiv.org/abs/1811.02815, arXiv:1811.02815.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., Philip, S.Y., 2020b. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 32.

Xiao, C., Xie, R., Yao, Y., Liu, Z., Sun, M., Zhang, X., Lin, L., 2021. Uprec: User-aware pre-training for recommender systems. CoRR .

Xie, X., Sun, F., Liu, Z., Gao, J., Ding, B., Cui, B., 2020. Contrastive pre-training for sequential recommendation. arXiv .

Xin, X., Pimentel, T., Karatzoglou, A., Ren, P., Christakopoulou, K., Ren, Z., 2022. Rethinking reinforcement learning for recommendation: A prompt perspective, in: Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1347–1357.

Xue, H.J., Dai, X., Zhang, J., Huang, S., Chen, J., 2017. Deep matrix factorization models for recommender systems., in: Proceedings of IJCAI.

Yang, X., Guo, Y., Liu, Y., Steck, H., 2014. A survey of cf based social recommender systems. Comput Commun 41.

Yang, Y., Wu, Z., Wu, L., Zhang, K., Hong, R., Zhang, Z., Zhou, J., Wang, M., 2023. Generative-contrastive graph learning for recommendation .

Yi, Z., Ounis, I., Macdonald, C., 2023. Contrastive graph prompt-tuning for cross-domain recommendation. ACM Transactions on Information Systems .

Yu, J., Yin, H., Gao, M., Xia, X., Zhang, X., Viet Hung, N.Q., 2021a. Socially-aware self-supervised tri-training for recommendation, in: Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining, pp. 2084–2092.

Yu, J., Yin, H., Li, J., Wang, Q., Hung, N.Q.V., Zhang, X., 2021b. Self-supervised multi-channel hypergraph convolutional network for social recommendation, in: Proceedings of the Web Conference 2021, pp. 413–424.

Yu, J., Yin, H., Xia, X., Chen, T., Li, J., Huang, Z., 2022a. Self-supervised learning for recommender systems: A survey. arXiv preprint arXiv:2203.15876 .

Yu, J., Yin, H., Xia, X., Cui, L., Nguyen, Q.V.H., 2022b. Graph augmentation-free contrastive learning for recommendation, in: Proceedings of SIGIR.

Zhang, B., Ling, H., Shen, J., Wang, Q., Lei, J., Shi, Y., Wu, L., Li, P., 2021. Mixture distribution graph network for few shot learning. IEEE Transactions on Cognitive and Developmental Systems .

Zhang, Q., Yang, L.T., Chen, Z., Li, P., 2018. A survey on deep learning for big data. Information Fusion 42.

Zhao, T., McAuley, J., King, I., 2014. Leveraging social connections to improve personalized ranking for collaborative filtering, in: Proceedings of CIKM.

Zheng, Z., Hui, K., He, B., Han, X., Sun, L., Yates, A., 2021. Contextualized query expansion via unsupervised chunk selection for text retrieval. Information Processing & Management 58, 102672.

Zhou, J., Cui, G., Hu, S., Zhang, Z., Yang, C., Liu, Z., Wang, L., Li, C., Sun, M., 2020. Graph neural networks: A review of methods and applications. AI Open 1.