EMCDR: A Classical Embedding and Mapping Approach for Cross-Domain Recommendation, Reproduction and Extension

Tong Man[12*] , Huawei Shen[12†], Xiaolong Jin[12†], and Xueqi Cheng[12†]

Abstract

To deal with data sparsity, one of the most challenging problems in recommender systems, an Embedding and Mapping framework for Cross-Domain Recommendation is proposed. It is one of the most classical modeling method based on embedding and mapping, moreover it has been used as the strongest baseline in many papers due to simplicity and conciseness. In this report, I mainly focus on reproduction of this work according to the available paper on the website https://www.ijcai.org/Proceedings/2017/0343.pdf. Lastly, as this paper proposes a framework paradigm, this model also has a lot of places worth exploring and adequate space for improvement.

Keywords： cross-domain recommendation, embedding and mapping, reproduction.

1 Introduction

The sparsity problem is the main problem faced by recommender system, which is also an important reason for the quality degradation of recommender system. In some large websites such as Amazon, the amount of items rated by users in only a tip of the iceberg relative to the total number of items on the website, which leads to the extremely daya sparsity of the user-item rating matrix and low accuracy when models taking these data as input. For example, collaborative filtering is widely adopted in recommender system.[1] One of the most classical collaborative filtering method is Matrix Factorization, i.e., MF, which achieved great achievement in the last decade.[2] In fact, MF method is actually borrowed from SVD, decomposing the rating matrix into user embedding matrix and item embedding matrix. However, in the practical scenario, the rating matrix is usually with high sparsity and even unevenly distributed. These kinds of problem become worse when they come to the new user and new item, which called the “cold-start problem”.

Cross-domain recommendation, being as one promising solution to the data sparsity problems, is getting more and more attention from researchers recently. Naturally, there is plenty of work focused on how to capture the correlation between the source domain and the target domain, which can be divided into several broad categories. EMCDR, namely, an Embedding and Mapping approach for Cross-Domain Recommendation, as

[1]CAS Key Laboratory of Network Data Science and Technology, Institute of Computing Technology, Chinese Academy of Sciences, China

[2]University of Chinese Academy of Sciences, China

[*]supermt@gmail.com

[†]{shenhuawei, jinxiaolong, cxq}@ict.ac.cn

the first of its kind, gives a different perspective for the cross-domain recommendation modeling approach and a large number of cross-domain recommendation methods based on mapping emerged after this paper.

Existing cross-domain recommendation methods, which deal with data sparsity, can mainly be divided into two categories.[3] The first type of work works in an asymmetrical manner, which aims to mitigate data sparsity in the target domain by using the data in the auxiliary domain.[4][5][6] In these kind of methods, knowledge or patterns learned from the auxiliary domain are directly transferred to the target domain. However, the capabilities of these approaches are limited if they don't make the use of the data from both domains. On the contrary, the second type of work works in a symmetrical manner, assuming that both the auxiliary domain and the target domain have the data sparsity problems, and expecting that these two domains can complement with each other.[7][8][9][10][11] In these kind of methods, there is no distinction between the auxiliary domain and the target domain, i.e., these two domains are treated equally. All the domains combat the data sparsity in a common way, typically learning the cross-domain mapping functions or explicitly differentiate between domain-specific factors and domain-sharing factors. The main concern with these approaches is that learning both domain-specific factors and domain-sharing factors may further exacerbate the problem of data sparsity.

EMCDR model studies the cross-domain recommendation problem from an embedding and mapping perspective, which is consistent with the second type of approach mentioned earlier, i.e., to learn the mapping function across the domains. The work focuses on two crucial issues: (1) How to represent the cross-domain mapping function in a linear or non-linear form? (2) Which part of data should be used to learn the mapping function? In the article, the authors give the answer that the model mainly contains two part: one part is for the latent factor modeling, the other part is for the latent space mapping procedure. And from the experiment comparing the method with the state-of-the-art method, shows the effectiveness of EMCDR in the cross-domain recommendation scenario.[3]

2 Related Works

Actually, there's a lot of work has the similar idea with the EMCDR framework. Borrowing the views stated in Zang et al., they attribute this kind of method to the user partial overlap & item partial non-overlap scenario, which is actually symmetrical with the user non-overlap & item partial overlap scenario.[12] In this scenario, the modeling approaches can be categorized into five classes, one of them that is embedding and mapping method. Others have *(i) collective matrix factorization, (ii) representation combination of overlapping users, (iii) graph neural network-based approaches, and (iv) capturing aspect correlations.*

2.1 The Basic Paradigm

As one inter-domain recommendation approach where one domain is treated as the source domain and the other as the target domain. Figure 1 has shown a shematic of this class method, which contains three main steps, namely latent factor modeling, latent space mapping and cross-domain recommendation. In the latent factor modeling process, the goal is to generate the latent factors of users and items $\{U^s, V^s, U^t, V^t\}$ for each domain. In the process of latent space mapping, the aim is to train a mapping function f. The objective of function f is to build the relationships of latent space between the domains.

$$\min_{\theta} \sum_{u_i \in \mathcal{U}} L(f(U_i^s;\theta), U_i^t) \tag{1}$$

During the cross-domain recommendation process, for users only with latent factors in the source domain, it generates the latent factors of users in the target domain. Formally,

$$\hat{U}_i^t = f(U_i^s;\theta) \tag{2}$$

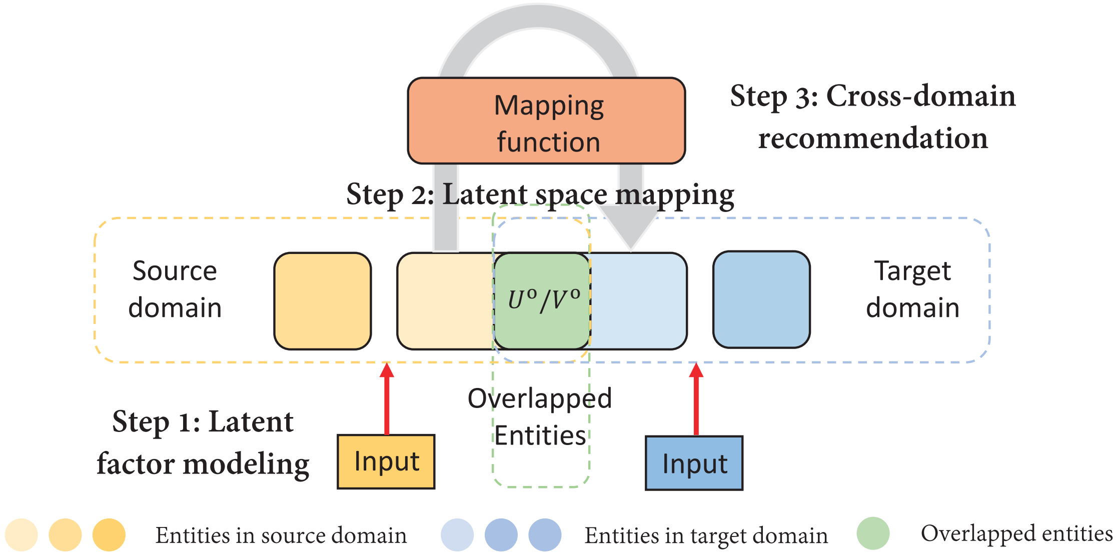

Figure 1: The schematic disgram of embedding and mapping

2.2 Different Paradigm Modeling

For EMCDR framework, it applied Matrix Factorization (MF) and Bayesian Personalized Ranking (BPR) to generate latent factors for users and items, respectively in the latent factor modeling process. For latent space mapping, it uses linear function and non-linear function based on Multi-Layer Perceptron (MLP) as the mapping function. The main objective is to approximate the latent factors of overlapping entities in the source domain mapped by f with the corresponding latent factors in the target domain[12].

Thanks to the proposed EMCDR framework, the idea has been extensively improved by many researchers, mainly in two aspects, one is the latent factor modeling process, the other is the latent space mapping process.

Arguing that the mapping function in EMCDR is learned based on shared entities, Kang et al. proposed a semi-supervised method, called **SSCDR**, which uses the data of non-overlapping entities to learn the mapping function and enhance the robustness of the mapping function.[13]

Generally, the users in two domains overlap only a small number of users, thus the mapping function learned usually becomes overfitted on overlapping users, reducing the generalization ability of EMCDR model. Moreover, using Mean Square Error (MSE) metric requires high quality of the target vector. However, in the cold-start entities scenario, the quality of the target vector is unsatisfactory, which will bring noise in the entities representation learning. Zhu et al. proposed a Transfer Meta framework, called **TMCDR**, which divides cross-domain recommendation into two stage: Transfer stage and Meta stage.[14]

Lastly, on the one hand, **CDLFM**[15] modifies the matrix factorization and mapping process by using user neighbors. **RC-DFM** extended stacked denoising autoencoders (SDAE) to effectively fuse the review text and item contents to train the entity factors, which achieves the SOTA performance in recommending the cold-start users.[16] These work focus on the improvement of the latent factor modeling. On the other hand, **DCDCSR** further expands EMCDR by generating benchmark factor matrices as the benchmark of the mapping function to solve cross-domain and cross-system recommendation problem.[17] Both DCDCSR and SSCDR aim at the improvement of the latent space mapping process.

3 Method

3.1 Notation

Table 1: Some Notations

D^s, D^t	source domain and target domain
$\mathcal{U} \in \{u_1, u_2, ...\}$	user sets
$\mathcal{I} \in \{i_1, i_2, ...\}$	item sets
$\mathcal{U}^s, \mathcal{I}^s$	the user set and item set in D^s
$\mathcal{U}^t, \mathcal{I}^t$	the user set and item set in D^t
$\mathcal{R}^s, \mathcal{R}^t$	user-item ratings in D^s and D^t, respectively
$f(\cdot)$	mapping function
$\hat{U}/\hat{V}$	affine factors of entities

3.2 Overview

The Embedding and Mapping framework for Cross-Domain Recommendation (EMCDR) is illustrated in Figure 2, which contains three major steps, i.e., latent factor modeling, latent space mapping and cross-domain recommendation.

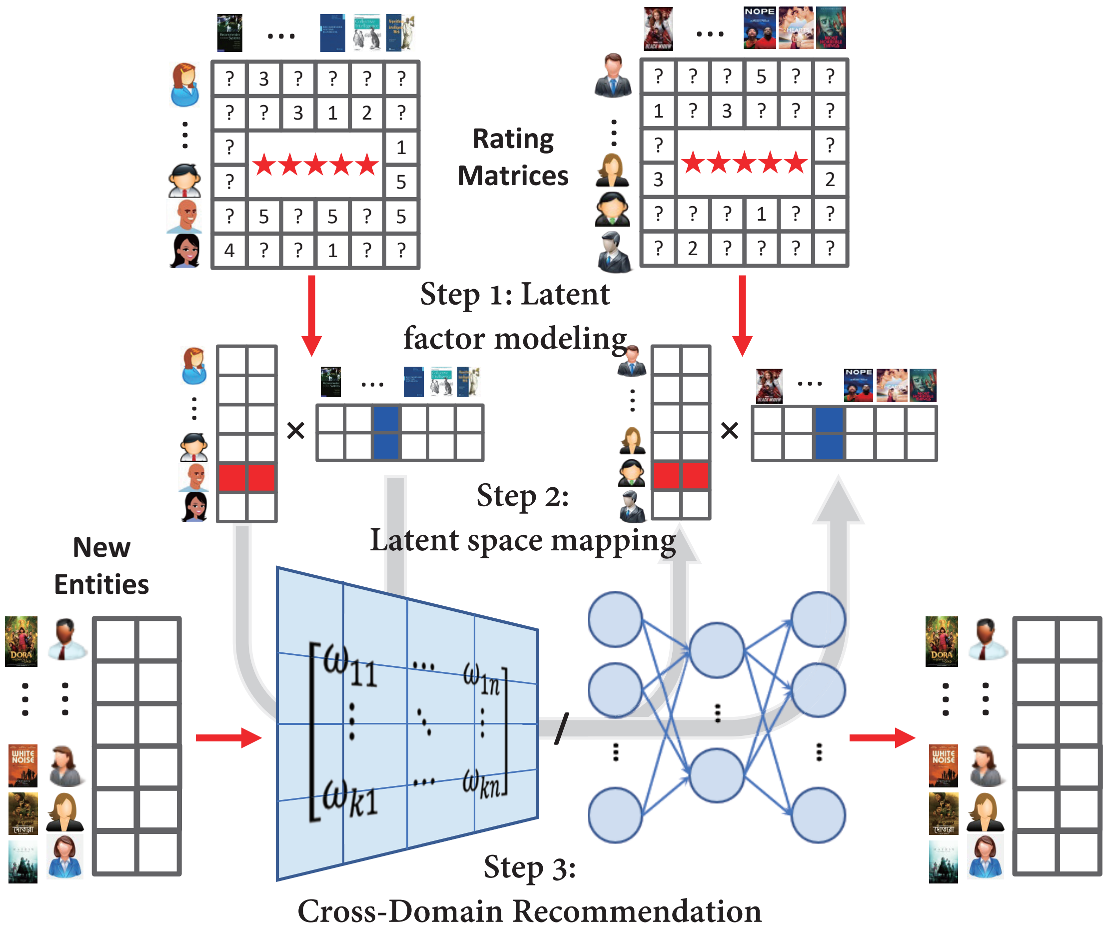

Figure 2: Overview of the method

3.3 Problem Definition

Without loss of generality, one domain is called the source domain, D^s and the other as the target domain, D^t. Suppose that $\mathcal{U} \in \{u_1, u_2, ...\}$ and $\mathcal{I} \in \{i_1, i_2, ...\}$ be the user sets and item sets. $\mathcal{R}^s$ and $\mathcal{R}^t$ be the two rating matrices from the source domain and the target domain, respectively, where R^s_{ui} is the rating that user u gives to item i in the source domain D^s and R^t_{ui} represents in a similar way.

Consider the two observed matrices in two domains, R^s and R^t, and cross-domain user and item sets, $\mathcal{U}$ and $\mathcal{I}$, we aim to make recommendation for entities with less information in the target domain by leveraging the information across domains.

3.4 Model

In this section, I will briefly introduce two different implementations of latent factor modeling part and latent space mapping part in the EMCDR framework.

3.4.1 Latent Factor Modeling

The first step in the EMCDR framework is to learn the latent factors of entities (i.e., users or items) in the source domain and target domain, respectively. Two different kinds of models, **MF**[18] and **BPR**[19], are used in the EMCDR framework providing us with a rating-oriented recommendation algorithm and a ranking-oriented one. Typically, MF decomposes a rating matrix into two low-rank matrices. Assuming that $\mathcal{R}$ is a rating matrix of $|\mathcal{U}| \times |\mathcal{I}|$ size, with Gaussian noise, given a user-item pair, the probability of the rating r_{ui} being obversed is modeled as follows:[2]

$$p(r_{ui}|U_u, V_i; \sigma^2) = \mathcal{N}(r_{ui}|U_u^T V_i, \sigma^2) \tag{3}$$

where K is the dimension of the latent factor, U is the user latent factor matrix of size $K \times |\mathcal{U}|$, in which U_u represents the u-th column of the matrix, that is the latent factor of user u, and V is the item latent factor matrix of size $K \times |\mathcal{I}|$, in which the definition is quite similar as fore, V_i represents the latent factor of item i. In addition, $\mathcal{N}(x|\mu, \sigma^2)$ represents the probability density function of Gaussian distribution with mean μ and variance σ^2.

The optimization problem of MF is to maximize the probability of obversed conditional distribution on the rating matrix, $\mathcal{R}$, which can be described as below:

$$\min_{U,V} \left(\sum_u \sum_i \|y_{ui} \cdot (r_{ui} - U_u^T V_i)\|_F^2 + \lambda_U \sum_u \|U_u\|_F^2 + \lambda_V \sum_i \|V_i\|_F^2 \right) \tag{4}$$

where y_{ui} is the indicator variable, $y_{ui} = 1$ if user u has rated on item i, otherwise, $y_{ui} = 0$, $\| \cdot \|_F^2$ denotes the Frobenius norm, λ_U and λ_V are both the coefficients for the regularization terms.

Unlike the assumption of MF, which bases on pointwise preference on an item, where 1 and 0 are used to denote “like” and “dislike” for an oobserved user-item pair and an unobversed user-item pair, respectively. BPR has the assumption of pairwise preferences over two items, which relaxes the assumption of pointwise preferences, that can be represented as follows,

$$\hat{r}_{ui} > \hat{r}_{uj}, i \in \mathcal{I}_u, j \in \mathcal{I} \backslash \mathcal{I}_u \tag{5}$$

where the relationship $\hat{r}_{ui} > \hat{r}_{uj}$ means that a user u is likely to prefer an item $i \in \mathcal{I}_u$ to an item $j \in \mathcal{I} \backslash \mathcal{I}_u$.

So for the BPR model, it should first create a training set $D := \{(u, i, j)|r_{ui} > r_{uj}\}$. We use $\sigma(\hat{r}_{uij})$ to approximate the probability of a preference pair (r_{ui}, r_{uj}), given the latent factor $\{U_u, V_i, V_j\}$, is modeled as

$$\begin{aligned} p(r_{ui} > r_{uj}) = \sigma(\hat{r}_{uij}) &= \sigma(\hat{r}_{ui} - \hat{r}_{uj}) \\ &= \sigma(U_u^T V_i - U_u^T V_j) \end{aligned} \tag{6}$$

where $\hat{r}_{uij} = \hat{r}_{ui} - \hat{r}_{uj}$, $\sigma(\cdot)$ is the sigmoid function. BPR optimizes the following objective function from the posterior probability of the latent factors.

$$\min_{U,V} \left(\sum_{(u,i,j) \in D} - \ln \sigma(\hat{r}_{uij}) + \lambda_U \sum_u \|U_u\|_F^2 + \lambda_V \sum_i \|V_i\|_F^2 \right) \tag{7}$$

According to the paper of EMCDR, the parameters of both MF and BPR models are optimized via stochastic gradient descent optimizer. In practice, the whole gradient update process is automated through the deep learning frameworks like tensorflow and pytorch. Formally, for the MF model, given a rating r_{ui} for one user-item pair (u, i), the updating rules are

$$
\begin{aligned}
U_u &\leftarrow U_u - \eta \cdot (r_{ui} - U_u^T V_i) V_i + \lambda_U U_u \\
V_i &\leftarrow V_i - \eta \cdot (r_{ui} - U_u^T V_i) U_u + \lambda_V V_i
\end{aligned} \tag{8}
$$

For a randomly sampled triple (u, i, j) in the BPR model, the updating rule is:

$$
\begin{aligned}
U_u &\leftarrow U_u - \eta \cdot -\sigma(-\hat{r}_{uij})(V_i - V_j) + \lambda_U U_u) \\
V_i &\leftarrow V_i - \eta \cdot -\sigma(-\hat{r}_{uij}) U_u + \lambda_V V_i \\
V_j &\leftarrow V_j - \eta \cdot -\sigma(-\hat{r}_{uij})(-U_u) + \lambda_V V_j
\end{aligned} \tag{9}
$$

3.4.2 Latent Space Mapping

With the latent factor models, so we can get the latent factors of users and items in the source and target domain $\{U^s, V^s, U^t, V^t\}$. The EMCDR framework assumes that cross-domain relationships can be captured through mapping functions. They have adopted two different mapping functions: one for the linear function, and the other is the function based on Multi-Layer Perceptron (MLP), which is non-linear.[20] In general, the mapping function f is optimized by the following formula.

$$
\min_{\theta} \sum L(f(E^s; \theta), E^t) \tag{10}
$$

where $L(\cdot, \cdot)$ is the loss function and E refers to the latent factor matrices of entities like U and V.

In **Linear Mapping** process, a mapping function is defined as a transfer matrix M and a MLP with the structure of one-hidden layer in **MLP-based Nonlinear Mapping** process. The corresponding formulas are shown in Equation 11.

$$
\begin{aligned}
&\min_{\theta} \sum L(M \times E^s, E^t) + \Omega(M) \\
&\min_{\theta} \sum L(f_{mlp}(E^s; \theta), E^t)
\end{aligned} \tag{11}
$$

where θ refers to the parameter set and $\Omega(M)$ is the regularization term on the transfer matrix.

3.4.3 Cross-domain Recommendation

Given the user/item in the target domain, we may estimate a credible latent factor by the latent factor learned in the source domain and the mapping function. By the following mapping process, we can obtain its affine latent factor in the target domain:

$$
\hat{U}_u^t = f(\hat{U}_u^s; \theta) \tag{12}
$$

4 Implementation Details

4.1 Algorithm

The complete EMCDR framework is presented in Algorithm 1.

Algorithm 1 The EMCDR framework.

Input:

Source domain R^s, target domain R_t;

User set $\mathcal{U}$, item set $\mathcal{I}$;

Object:

Make recommendation for entities in the target domain: $\{u \in R^s \& u \notin R^t\}/\{i \in R^s \& i \notin R^t\}$;

Latent Factor Model

1: Learn $\{U^s, V^s\}$ from $\mathcal{R}^s$;

2: Learn $\{U^t, V^t\}$ from $\mathcal{R}^t$;

Latent Factor Model

1: Learn the mapping function $f(\cdot)$ by entities across domain

Cross-domain Recommendation

1: Get affine factors $\hat{U}^t(\hat{V}^t)$ of target entities

2: Make recommendation for target entities

4.2 Experimental Environment Setup

The entire reproduction process is based on python 3.6.13 and tensorflow 1.12.0. The whole experimental procedure is carried out in strict accordance with what is said in the paper, as the learning rate and regularization coefficients are optimized via 5-fold cross validation on the training set. However, for the final latent factor used for mapping I have used best performance it can reach according to the 5-fold cross validation experiment. Specifically, with the best number of iterations and the best parameter setting, I used the whold data for the training phase to avoid the circumstance that latent factors can not contain all the information of data. In the end, for the authors did not publish the dataset they used during the experiment, I myself have processed the cross-domain dataset on **MovieLens** alignment with **Netflix** on movies. The satistics of MovieLens-Netflix dataset is quite different from what is reported on the paper, through I have tried to process data as closely as possible to the data statistics, which may affect the reproduction results. The statistics of MovieLens-Netflix dataset I have processed is showed in Table 2, which compared with the data used in the origin paper, the matrix size is flatter. The statistics of the data used in original paper is also showed in Table 3.

Table 2: Statistics of the MovieLens-Netflix dataset.

Statistics	MovieLens	Netflix
Num. of users	98,907	99,683
Num. of movies	4,100	4,100
Num. of ratings	7,465,524	11,305,262

Table 3: Statistics of the MovieLens-Netflix dataset.

Statistics	MovieLens	Netflix
Num. of users	69,258	70,132
Num. of movies	5,871	5,871
Num. of ratings	7,891,832	11,658,783

In order to evaluate the effectiveness and efficiency of the EMCDR framework for cross-domain recommendation tasks, the authors randomly remove all rating information of some entities in the target domain and set the different fractions for cold-start entities, namely, 10%, 20%, 30%, 40% and 50%, for the sake of stringency. Due to the time issue, I didn't follow the original paper that sample entities for $L = 10$ times repeatly to generate different sets in every experiment, in some cases, I set L as 5 to get experimental results faster.

Here, the dimension K of the latent factor is only set as 100, for evaluating all of them will cost plently of time! Moreover, there is a phenomenon of anti-common sense in my experimental results and the exact cause is still under investigation. The learning rate and regularization coefficients are optimized via 5-fold cross validation on the training dataset, followed by the original paper, so as the structure of MLP mapping function. In order to get better experimental results, I only use the *tanh* activation function instead of tan-sigmoid function, which employed as the activation function in the original paper and the weight and bias parameters of the MLP is initialized in Gaussian Distribution, which differs from the rule in[20].

4.3 Main Contributions

In the reproduction of this paper, I reconstructed the data which is similar to the data size in the original paper and evaluate according to the experiemental settings in the original paper, which expects to verify some potential rules and model performance. The whole results are shown in Table 4 and 5.

The improvement idea of EMCDR framework is very clear, which can be divided according to the parts of its framework, i.e., latent factor model and latent space mapping. For the part of latent factor modeling, we have learned the embedding of entities through MF and BPR models previously. If we can learn the higher-quality embedding, it will undoubtedly help improve the performance of the whole framework. Additionally, for the part of latent space mapping, the process of learning the mapping function is essentially to find the transfer relationship between source domain and target domain. In the original paperr, MLP is used to complete the transformation process of the transfer mapping, and also the best results are obtained in the experiment. Variational AutoEncoder (**VAE**) learns and obtains a set of mean values and standard deviations, which is used to generate Gaussian distribution and sample from it to obtain z_u, for the reconstruction process of entity embeddings. Benefitted from this kind of modeling method, in theory, VAE should be more able to capture the variants and invariants between domains. KL divergence can not only greatly enhance the model generation ability, but also help the framework become more robust. So if we use the latent factors learned from source domain to reconstruct the latent factors of target domain by VAE model, the effectiveness of the combined model is very desirable, which is illstrated in Figure 3.

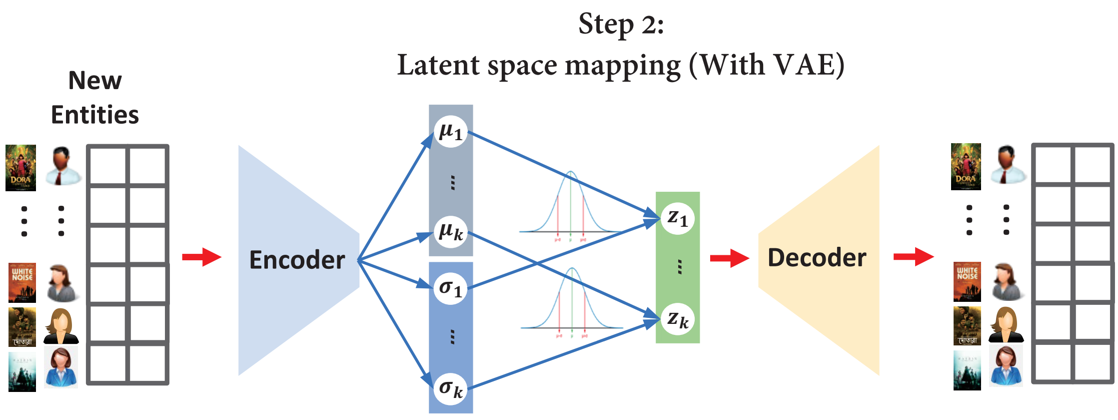

Figure 3: Extension of Latent Space Mapping

5 Results and Analysis

For the reproduction, experimental results of RMSE and AUC on the MovieLens-Netflix dataset are demonstrated on Table 4 and 5, respectively. The dataset I have processed here is quite different from the original paper, so the relevant metrics cannot be directly compared. Relatively, in the same experiment, comparison between different experimental groups is a more reliable choice. From the original paper, we can determine some intuitive relationships: for example, the higher the fraction of the cold-start entity, the lower the experimental metrics obtained; Moreover, if MLP is used as the function of latent space mapping, the experimental results should outperform those using linear mapping.

Table 4: Recommendation performance in terms of RMSE on the MovieLens-Netflix dataset.

	Model \ CSF[1]	10%	20%	30%	40%	50%
K=100	MF+LM	1.0521 ± 0.0004	1.0564 ± 0.0005	1.0625 ± 0.0005	1.0617 ± 0.0003	1.0625 ± 0.0003
	MF+MLP	1.0486 ± 0.0005	1.0536 ± 0.0005	1.0605 ± 0.0008	1.0600 ± 0.0003	1.0603 ± 0.0003

[1] CSF: Cold-Start entity Fraction.

Table 5: Recommendation performance in terms of AUC on the MovieLens-Netflix dataset.

	Model \ CSF[1]	10%	20%	30%	40%	50%
K=100	BPR+LM	0.5610 ± 0.0112	0.5559 ± 0.0100	0.5533 ± 0.0062	0.5618 ± 0.0130	0.5553 ± 0.0071
	BPR+MLP	0.5681 ± 0.0129	0.5590 ± 0.0081	0.5580 ± 0.0109	0.5627 ± 0.0092	0.5602 ± 0.0060

[1] CSF: Cold-Start entity Fraction.

Due to the time issue, I did not test at the circumstances of $K = 20, 50$ as the original paper did, for I consider that the embedding parameters need to be re-adjusted when changing the dimension of embeddings. Further, for this cross-domain recommendation method, the quality of learned embedding in the target domain is particularly critical. which will directly affect the performance of the entire experiment. From the Table 4 and 5, on the one hand, we can see that the experimental results of the experimental groups using MLP are better than those using linear mapping function (LM), which is intuitive. On the other hand, in general, with the increase of the fraction of cold start entities, the results of the experimental group tend to decrease. Except for the slight abnormality when $CSF = 40\%$ and 50%, we think that this may be caused by the random seed setting or maybe the learned embeddings are not good enough, although the learning process strictly follows the five-fold cross validation to select the best parameters of learning rate and regularization coefficient.

6 Conclusion and Future Work

In this reproduction work, I reproduced a framework for cross-domain recommendation according to the settings of the original paper. In the **Latent Factor Modeling** part, first, I learned the embeddings of users and items through some simple latent factor models; In the process of **Latent Space Mapping**, learn the mapping function to capture the corresponding relationship of embeddings between cross-domain entities. In the third part, make recommendations by mapping the embeddings of the source domain to the target domain. The EMCDR framework is often used as the baseline in the recently published paper, so the work of reproducing the paper is likely to be used in the future. At the same time, the model itself is relatively simple, so it has greater scalability.

References

[1] SARWAR B, KARYPIS G, KONSTAN J, et al. Item-based collaborative filtering recommendation algorithms[C]//Proceedings of the 10th international conference on World Wide Web. 2001: 285-295.

[2] MNIH A, SALAKHUTDINOV R R. Probabilistic matrix factorization[J]. Advances in neural information processing systems, 2007, 20.

[3] MAN T, SHEN H, JIN X, et al. Cross-domain recommendation: An embedding and mapping approach.[C] //IJCAI: vol. 17. 2017: 2464-2470.

[4] LI B, YANG Q, XUE X. Can movies and books collaborate? cross-domain collaborative filtering for sparsity reduction[C]//Twenty-First international joint conference on artificial intelligence. 2009.

[5] LI B, YANG Q, XUE X. Transfer learning for collaborative filtering via a rating-matrix generative model[C]//Proceedings of the 26th annual international conference on machine learning. 2009: 617-624.

[6] PAN W, XIANG E, LIU N, et al. Transfer learning in collaborative filtering for sparsity reduction[C] //Proceedings of the AAAI conference on artificial intelligence: vol. 24: 1. 2010: 230-235.

[7] SINGH A P, GORDON G J. Relational learning via collective matrix factorization[C]//Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining. 2008: 650-658.

[8] AGARWAL D, CHEN B C, LONG B. Localized factor models for multi-context recommendation[C] //Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining. 2011: 609-617.

[9] JAMALI M, LAKSHMANAN L. Heteromf: recommendation in heterogeneous information networks using context dependent factor models[C]//Proceedings of the 22nd international conference on World Wide Web. 2013: 643-654.

[10] LI C Y, LIN S D. Matching users and items across domains to improve the recommendation quality[C]//Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining. 2014: 801-810.

[11] LIU Y F, HSU C Y, WU S H. Non-linear cross-domain collaborative filtering via hyper-structure transfer[C]//International Conference on Machine Learning. 2015: 1190-1198.

[12] ZANG T, ZHU Y, LIU H, et al. A survey on cross-domain recommendation: taxonomies, methods, and future directions[J]. ArXiv preprint arXiv:2108.03357, 2021.

[13] KANG S, HWANG J, LEE D, et al. Semi-supervised learning for cross-domain recommendation to cold-start users[C]//Proceedings of the 28th ACM International Conference on Information and Knowledge Management. 2019: 1563-1572.

[14] ZHU Y, GE K, ZHUANG F, et al. Transfer-meta framework for cross-domain recommendation to cold-start users[C]//Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2021: 1813-1817.

[15] WANG X, PENG Z, WANG S, et al. Cross-domain recommendation for cold-start users via neighborhood based feature mapping[C]//International conference on database systems for advanced applications. 2018: 158-165.

[16] FU W, PENG Z, WANG S, et al. Deeply fusing reviews and contents for cold start users in cross-domain recommendation systems[C]//Proceedings of the AAAI Conference on Artificial Intelligence: vol. 33: 01. 2019: 94-101.

[17] ZHU F, WANG Y, CHEN C, et al. A deep framework for cross-domain and cross-system recommendations[J]. ArXiv preprint arXiv:2009.06215, 2020.

[18] KOREN Y, BELL R, VOLINSKY C. Matrix factorization techniques for recommender systems[J]. Computer, 2009, 42(8): 30-37.

[19] RENDLE S, FREUDENTHALER C, GANTNER Z, et al. BPR: Bayesian personalized ranking from implicit feedback[J]. ArXiv preprint arXiv:1205.2618, 2012.

[20] GLOROT X, BENGIO Y. Understanding the difficulty of training deep feedforward neural networks[C]//Proceedings of the thirteenth international conference on artificial intelligence and statistics. 2010: 249-256.