Deep Unified Representation for Heterogeneous Recommendation

Chengqiang Lu  
Alibaba Group  
China  
lulu.lcq@alibaba-inc.com

Mingyang Yin  
Alibaba Group  
China  
hengyang.ymy@alibaba-inc.com

Shuheng Shen  
Ant Financial Services Group  
China  
shuheng.ssh@antgroup.com

Luo Ji  
Alibaba Group  
China  
jiluo.lj@alibaba-inc.com

Qi Liu  
University of Science and Technology of China  
China  
qiliuql@ustc.edu.cn

Hongxia Yang  
Alibaba Group  
China  
yang.yhx@alibaba-inc.com

ABSTRACT
Recommendation system has been a widely studied task both in academia and industry. Previous works mainly focus on homogeneous recommendation and little progress has been made for heterogeneous recommender systems. However, heterogeneous recommendations, e.g., recommending different types of items including products, videos, celebrity shopping notes, among many others, are dominant nowadays. State-of-the-art methods are incapable of leveraging attributes from different types of items and thus suffer from data sparsity problems. And it is indeed quite challenging to represent items with different feature spaces jointly. To tackle this problem, we propose a kernel-based neural network, namely deep unified representation (or DURation) for heterogeneous recommendation, to jointly model unified representations of heterogeneous items while preserving their original feature space topology structures. Theoretically, we prove the representation ability of the proposed model. Besides, we conduct extensive experiments on the real-world datasets. Experimental results demonstrate that with the unified representation, our model achieves remarkable improvement (e.g., 4.1% ~34.9% lift by AUC score and 3.7% lift by online CTR) over existing state-of-the-art models.

1 INTRODUCTION
In the information explosion era, recommendation systems become more and more important to link users with their potential interesting items. Recently, along with the rising number and increasing diversity of available items, many online platforms provide different kinds of content to enrich users’ experiences. For example, e-commerce platforms recommend user-generated-contents (UGC) along with products, e.g., Amazon [31] and Alibaba[3]. Social networking platforms recommend potentially interesting books, movies, and music, e.g., Douban [26] and Facebook [38].

Traditional homogeneous recommendation models that deal with products alone are inapplicable in these scenarios [8, 11]. Therefore, heterogeneous recommendation [13], which remains a largely unexplored area, has attracted massive interest in both academia and industry.

Most recommendation models extract users’ and items’ latent representations implicitly or explicitly and then calculate the relevance between them via inner-product or other neural network methods, e.g., MLP. Traditional models, such as Collaborative Filtering [36] and Matrix Factorization [21, 56], exploit the interactions between users and items in-depth to learn representations. However, modeling the observed interactions is insufficient to make suitable recommendations as the informative attributes of users and items are ignored. In the past decade, deep learning has achieved great success in the areas of computer vision [16], natural language processing [50], and natural science [37] with its powerful representation learning ability. Therefore, it becomes increasingly ubiquitous in modern recommendation systems to exploit arbitrary continuous and categorical attribute features with deep neural networks [7]. These models usually compare favorably to the aforementioned traditional models because they could leverage items and users’ attribute features. Besides, these models could remedy the problem of inadequate labeled interaction data.
Although previous works have made much progress, recommendation on heterogeneous items is still challenging because different kinds of attributes do not share the same feature space.

Therefore, former models that were designed for homogeneous item recommendation could not be applied directly. A naive but common way to solve it is to merely use the interaction features or the same part of attribute features, where much useful information has to be dropped [18]. Another solution is to train one sub-model for each kind respectively while maintaining shared user embeddings [26]. However, in this setting, one kind of item’s recommendation could not benefit from other items’ information.

Cross-domain recommendation [2, 10, 48] is a similar problem but it deals with the samples from different distributions but with the same feature space. Heterogeneous Information Networks (HIN) [39] characterizes auxiliary data by constructing and mining a heterogeneous graph, e.g., a graph contains multiple types of entities such as movies, actors, and directors. Nevertheless, items to be recommended in HIN-based methods are still homogeneous, e.g., movies. Consequently, these models are not suitable for heterogeneous recommendation.

To address this heterogeneous recommendation problem, we propose the deep unified representation (DURation) for the heterogeneous item recommendation to form use of various item attribute features. We first introduce a simple and easy-to-extend model architecture. Compared with previous studies, we design the mapping module to transform different kinds of input features into a unified feature space. To learn the unified representation, we propose the alignment loss to reduce their dissimilarities inspired by the domain generation [32]. Besides, we prove that the proposed unified representation could capture the nuanced difference of items after the alignment. We also modify the CORAL loss [45] to preserve the topology of original feature spaces. At the end, we design an approximated implementation with both theoretical and empirical proofs of its effectiveness for large-scale deployment.

Results of relatively comprehensive experiments in the real-world datasets demonstrate the superiority of the proposed model.

To summarize, we make the following contributions:

- We formally define the problem of heterogeneous item recommendation, a general framework for various cross-domain recommendation problems, and discuss practical challenges of large-scale deployment.
- We design unified representations of heterogeneous items to leverage different attribute spaces simultaneously and alleviate the problem of data sparsity. In both theory and practice, we prove that such alignment can keep topologies of their original spaces.
- Extensive experiments on the real-world dataset demonstrate the effectiveness of our proposed model.

The remainder of this paper is as follows. After introducing some related works in Section 2, we elaborate on our proposed model architecture and the learning method for the unified representations in Section 3 and 4 respectively. We then perform experimental evaluations in Section 5 and conclude the paper in Section 6.

2 RELATED WORK

In this section, we briefly review recent progress in the cross-domain recommendation and transfer learning, especially their insufficiencies to be directly applied to the heterogeneous recommendation.

2.1 Cross-domain Recommendation

A considerable amount of literature has been published on the problem of cross-domain recommendation systems [2, 9, 28]. In the different literature, researchers have considered distinct definitions of domain. For the sake of clarity, we follow the definition of [32]. Given input space $X$ and output space $Y$, the domain is a joint distribution $P_{XY}$ on $X \times Y$. This definition covers the attribute-level and type-level domain defined in [9] but does not include the item-level domain.

Cross-domain recommendation systems have achieved great success. For instance, multi-view neural networks [15, 25, 26], that construct separated sub-networks for each kind of item (so-called view) and let them share the same user embedding layer, were proposed and widely used. However, the items of one kind could barely benefit from the information of the others in these models. Besides, probabilistic topic models are often used to learn topic distributions from different kinds of items then build relevance of these topic distributions [47, 48]. Conversely, topic models require establishing linkages across items by text data such as item descriptions and user-generated text and could not utilize raw attributes such as numerical or categorical features. In addition, neighbor-based models are also popular in building relations between different users. One category of these models leverages users’ similarity based on the users’ social network [55] that are not suitable for the platform lacking social information such as e-commerce platform. Furthermore, they do not perform well for the users with a weak social bond. The second category of neighbor-based models builds user similarity based on user latent features extracted from user attributes and history behaviors [52]. One drawback of them is their poor scalability for large-scale online systems. Another drawback is all these models suffer from data sparsity.

Along with the aforementioned definition of the domain, these models could extract latent features of items sampled from different distributions $P_X$. Nonetheless, they require the original features of items in the same input space $X$, which is not consistent with the setting of the heterogeneous item recommendation. Therefore, most cross-domain models could not be applied directly to the problem. Actually, we demonstrate that cross-domain recommendation could be considered as a particular case of the heterogeneous item recommendation in Section 3.1.

2.2 Transfer Learning

To learn the representation of items in different spaces, transfer learning is a conventional powerful tool in multiple areas including natural language processing [35], computer vision [53], natural science [23] and recommendation system [5, 29]. EMCDR (Embedding and Mapping framework for Cross-Domain Recommendation) uses a multi-layer perceptron to capture the nonlinear mapping function to transfer knowledge from the source domain to the target domain [29]. The main disadvantage of EMCDR is that it only takes the
interaction matrices of different items as input and cannot exploit
the attribute features. Along this line, Li and Tuzhilin [25] propose
a novel model based on the mechanism of dual learning that trans-
fers information between two kinds of items so that this approach
could make the use of different attribute features. It learns the la-
tent orthogonal mapping function to preserve similarities of users' preferences. MMT-Net (Multi-Linear Module Transfer Network) leverages contextual invariances to develop shared modules across
different items [22].

Domain adaptation (also known as covariate shift) is another form of transfer learning [34] that could be used in both supervised and unsupervised problems. Domain adaptation was proposed primarily to deal with the inconsistency between training and test distributions. Wang and Mahadevan [51] construct linear mapping functions to link different feature spaces using manifold alignment with the aid of label. However, the linear mapping function is not suitable for a web-scale recommendation. Therefore, we need
attributes of different kinds of items are in different feature spaces
In the setting of heterogeneous recommendation, the representations of users

\[ u \]

Label Matrix

\[ Y \]

Mapping Function

\[ f \]

Input Space

\[ X \]

Unified Feature Space

\[ R \]

Kernel Function

\[ \mu \]

Mean Map

\[ \hat{Y} \]

Prediction Matrix

\[ S_p \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]

\[ p_{th} \text{ Item Representations} \]

\[ P \]

\[ R \]

\[ Y \]

\[ f_p \]

\[ \hat{Y} \]

\[ p_{th} \text{ Item Representations} \]

\[ Y \]
the dimension of attribute feature space of the \( p_{th} \) kind of items
and \( d \) is the dimension of the unified feature space. Here, we define
the mapping function \( f_p \) as an MLP too. After the transformation,
we concatenate the transformed attribute feature vector with interaction feature vector then feed them into another MLP
\( x_i = MLP(Y_{attri} \oplus f_p(R_{attri})). \)

4 UNIFIED REPRESENTATION LEARNING

The main impediment of applying existing models to the heterogeneous item recommendation is that the attributes features of different kinds of items are in different feature spaces. Simply dropping them leads to the loss of important side information of items, so that the learning of unified representations is the overarching task in the heterogeneous recommendation. There are three challenges here: firstly, the distribution \( P_{X_p} \) of unified representations of different kinds of items should be aligned; secondly, the new representations \( X_p \) should preserve the topology of the original feature space; thirdly, we aim to keep the functional relationship \( P(Y|R_p) \) in the transformed feature space fixed so that the final deep features could be discriminative enough to train a strong recommendation system.

4.1 Alignment

Intuitively, the performance of back-end model \( h \) which works after learning unified representation depends in part on the dissimilarity of the distributions \( P_{X_1}, P_{X_2}, \ldots, P_{X_p} \). Besides, lack of feature distribution alignments can lead to overfitting when data is sparse [30, 51]. Moreover, closely aligned feature distributions will make it tractable for downstream tasks such as item clustering. Therefore, alignment is one of the most essential parts in the learning of unified representations.

To solve the problem of alignment, we propose to reduce the dissimilarity between feature spaces by minimizing the distance between empirical distributions of samples and those of their corresponding transformed ones by the mapping functions. Thus, the first step is to define the representation of distributions. Embedding probability distributions into reproducing kernel Hilbert spaces (RKHS) was introduced to resolve this issue [1]. This embedding represents any distribution as a mean element in an RKHS [40].

Given a kernel \( \kappa \), we define the mean map of the distribution \( P_{X_p} \) as

\[
\mu[P_{X_p}] := \int_X \kappa(x, \cdot) dP_{X_p}(x). \tag{1}
\]

Here \( P_{X_p} \) is the distribution of \( p_{th} \) kind of items after transformation and \( X \) is the unified feature space. In our model we choose the Gaussian Radial Basis Function (RBF) kernel \( \kappa(x, x') = \exp(-\lambda \|x - x'\|^2) \) due to its universality, which means for every function \( f \) induced by \( \kappa \) and every \( \epsilon > 0 \) there exists a function \( g \) induced by \( k \) with \( \|f - g\|_{\infty} \leq \epsilon \). This property ensures that all continuous functions can be approximated by certain kernel expressions [44] and benefits the embedding of distributions which will be discussed later.

By utilizing the mean map and RBF kernel, the distance between distributions is measured as the inner product of two distributions

\[
\mu[P_{X_p}] \cdot \mu[P_{X_q}] = \int_X \kappa(x, \cdot) dP_{X_p}(x) \cdot \int_X \kappa(x, \cdot) dP_{X_q}(x).
\]

To prove the suitability of this measure, let us introduce the characteristic kernel first [42]:

**Definition 1 (Characteristic kernel).** A bounded measurable kernel \( \kappa \) is said to be characteristic if \( \mu \rightarrow \int_X \kappa(x, \cdot) d\mu(x) \) is injective, where \( \mu \) is a Borel probability measure on \( X \).

Let \( \mu \) be the mean map using RBF as the kernel function. The following theorem shows that the inner product between RBF kernel mean embedding is suitable as a measure of the distance between distributions.

**Theorem 1A.** For any two distributions \( P \) and \( Q \) in \( X \), \( \mu[P] = \mu[Q] \) if and only if \( P = Q \).

**Theorem 1B.** There exist two distinct distributions \( P \) and \( Q \) defined on \( X \) satisfying \( \mu[P], \mu[Q])_{H^2} < \epsilon \) for any arbitrarily small \( \epsilon > 0 \).

The Theorem 1 demonstrates that distributions are close in embedding space when their differences occur at higher frequencies.
and this embedding could capture the nuanced difference. We provide the proof of this theorem in Supplementary Material A.1.1. After embedding the distributions, a popular method to measure the dissimilarity between distributions is using the distributional variance [32]:

**Definition 2 (Distributional variance).** Given a set of distributions \( \mathcal{P} = \{ \mathcal{P}_X, \mathcal{P}_{X_2}, \ldots, \mathcal{P}_{X_P} \} \), embed them to RKHS \( \mathcal{H} \) and then define the \( P \times P \) Gram matrix \( G \) with entries \( G_{ij} = \langle p_{X_i}, p_{X_j} \rangle_\mathcal{H} \). Let \( \Sigma = G - 1_P G^\top - 1_P G \). The distributional variance is

\[
\mathbb{V}_G(\mathcal{P}) = \frac{1}{P} \text{tr}(\Sigma) = \frac{1}{P} \text{tr}(G) - \frac{1}{P^2} \sum_{i=1}^{P} \sum_{j=1}^{P} G_{ij}.
\] (2)

Here \( 1_P \) denotes an \( P \times P \) all-ones matrix. Given \( P \) sample sets \( \mathcal{S} = \{ S_i \}_{i=1}^{P} \) drawn from \( \mathcal{P}_{X_1}, \mathcal{P}_{X_2}, \ldots, \mathcal{P}_{X_P} \), previous study [32] proved that the empirical estimator \( \mathbb{V}_G(\mathcal{S}) = \frac{1}{P} \text{tr}(\hat{\Sigma}) \) is a consistent estimation of \( \mathbb{V}_G(\mathcal{P}) \) where \( \hat{\Sigma} \) is obtained from the Gram matrix

\[
\hat{G}_{ij} := \frac{1}{N_i \cdot N_j} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} \kappa(x_k^i, x_l^j).
\] (3)

Here \( i, j \) is the index of distributions, and \( k, l \) is the index of the samples in their distribution. \( N_i \) denotes the number of samples drawn from \( S_i \). To investigate this further, aiming to minimize \( \mathbb{V}_G(\mathcal{S}) \), we define the alignment loss

\[
A = \sum_{i=1}^{P} \sum_{j=1}^{P} \sum_{k=1}^{N_i} \sum_{l=1}^{N_j} A_{ijkl} = \begin{cases} \frac{P-1}{P \cdot N_i \cdot N_j} \kappa(x_k^i, x_l^j) & i = j \\ \frac{-1}{P \cdot N_i \cdot N_j} \kappa(x_k^i, x_l^j) & i \neq j \end{cases} \]

(4)

The derivation process from \( \mathbb{V}_G(\mathcal{S}) \) to Equation 4 is investigated in Supplementary Material A.1.2. Because the complexity of Equation 4 is high, we make an approximation which is discussed in Section 4.3.

### 4.2 Topology Preservation

During the construction of the mapping functions, we aim to preserve the topology of the original space for the raw feature of each kind of item. Given two items that belong to the same category \( i \), their raw and mapped embeddings are \( r_k^i \) and \( x_k^i \) respectively. Basically, if \( r_k^i \) and \( r_l^j \) are similar in their original spaces, the \( x_k^i \) and \( x_l^j \) should also close in the transformed feature space and vice versa. To solve the problem, several attempts have been made to minimize the distances between pairs of items in the new space with their similarity in raw space as an adjusted rate [51]. However, this approach suffers from the choice of similarity metric: Gaussian kernel similarity could not ensure the dissimilar items are well-separated from each other in the new space; Euclidean distance varies greatly which makes it intractable in practice; adjusted cosine similarity performs unstable in experiments.

As a remedy, we propose a modification of the CORAL loss [45] which is defined as the distance between the second-order statistics. Suppose we are given \( i \) kind training examples \( D_{Ri} = \{ r_k^i \}, r \in \mathbb{R}^d \) where \( d \) is embedding dimension. \( D_{Xi} = \{ x_k^j \}, x \in \mathbb{R}^d \) denotes the samples after the transformation. Therefore, the covariance matrices of the raw and transformed data are given by:

\[
C_{Ri} = \frac{1}{d-1} \left( D_{Ri} D_{Ri}^\top - \frac{1}{d} (1^\top D_{Ri}^\top) (1^\top D_{Ri}) \right),
\]

\[
C_{Xi} = \frac{1}{d-1} \left( D_{Xi} D_{Xi}^\top - \frac{1}{d} (1^\top D_{Xi}^\top) (1^\top D_{Xi}) \right),
\]

where \( 1 \) is a column vector with all elements equal to 1. Then we define the topology loss as:

\[
T = \sum_{i=1}^{P} \frac{1}{4N_i} \| C_{Ri} - C_{Xi} \|_F^2.
\]

(5)

(6)

where \( \| \cdot \|_F^2 \) denotes the squared matrix Frobenius norm.

### 4.3 Objective Function

Because we model the recommendation as a classification problem, we utilize the Binary Cross Entropy Loss (BCE Loss) as the classification loss:

\[
C = -\frac{1}{M} \sum_{i=1}^{M} y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i).
\]

(7)

where \( M \) is the number of all labeled interactions between users and items and \( y_i \) is the user’s feedback to the item in one interaction. \( \hat{y}_i = g(x_i) \) indicates the predictive value where \( h \) is the readout function mentioned in Section 3.2.

We desire our model to simultaneously achieve the aforementioned three goals in the new space and obtain a unified representation. Consequently, combining Equation 4, 6 and 7, the overall objective function to be minimized is

\[
L = C + \alpha A + \beta T,
\]

(8)

where \( \alpha \) and \( \beta \) are hyperparameters. We optimize the Equation 8 with Adam optimizer. For each distribution, we randomly sample \( B \) samples to calculate the mini-batch stochastic gradient \( \nabla L_{PB} = \frac{1}{B} \sum_{b=1}^{B} \left( \nabla C_{PB} + \alpha \nabla A_{PB} + \beta \nabla T_{PB} \right) \). As sample indexes to estimate the loss terms in Equation 4, 6 and 7 are all independent, \( \nabla L_{PB} \) is an unbiased estimation of \( \nabla L \), which guarantees a \( \frac{1}{P} \) convergence rate. Therefore, this approximation ensures that our model could be applied to the web-scale recommendation.

**Summary.** With latent representations of users and items, a readout function \( h \) is applied for the final prediction, which is denoted as \( \hat{y}_ij = h(u_i, x_j) \). The choice of \( h \) can be diverse, such as bidirectional encoder representations [20, 46] which is capable of handling sequential recommendation, or deep ranking models [24] to better capture high order interaction. In the consideration of online efficiency for large-scale system, we use inner product of two vectors as \( h(u_i, x_j) = \langle u_i, x_j \rangle \).

### 5 EXPERIMENTS

This section, we conduct extensive experiments on a real-world public dataset and on a real-world large-scale online recommendation system. Besides, the code of our proposed model and baselines has been published to GitHub \(^1\) so that everyone could validate the experimental results using a one-line code with proper configuration. For more details about reproducibility, such as the dataset,
code usage, set-up, and hyperparameters, please refer to the Supplementary Material A.2.

5.1 Dataset
In previous work, synthetic datasets (e.g. sample data from the homogeneous database such as Google Books, IMDB, and Spotify respectively and then mixed them up) are widely used [25, 29]. However, these datasets are simply combinations of some well-known homogeneous datasets and do not correspond with reality. Therefore, we evaluated our model on the real-world Douban dataset. Douban\footnote{www.douban.com} is a popular Chinese social networking platform that allows users to create content related to movies, books, and music. Here we convert the recommendation problem in the Douban dataset to a binary classification problem (like or not like). The detailed statistics of this dataset are shown in Table 2. Besides, we also deploy our model into production and conduct online A/B test experiments which will be explained in Section 5.3.2.

| # Item | # Rate  | # User  | # Attribute |
|-------|---------|---------|-------------|
| Book  | 150955  | 356001  | 25496 11    |
| Music | 102571  | 269760  | 25426 8     |
| Movie | 50299   | 470479  | 31664 10    |
| All   | 303825  | 1096240 | 36585 /     |

5.2 Baselines
5.2.1 General Models. The first group consists of three general models. They merely use the interaction information between user and items and drop the attributes of items.
- **DeepMF** [56] is a neural network enhanced matrix factorization model.
- **FISM** [19] is an item-based model that learns the item-item similarity for the recommendation.
- **NAIS** [17] is a deep attention model that is capable of distinguishing which historical items in a user profile are more important.

5.2.2 Contextual Models. The second group contains six contextual models that could exploit the attributes of homogeneous items. To apply them in the heterogeneous recommendation, we feed them with the overlapping features of different kinds of items such as the average rate of items. Besides, we also run these models in each kind of homogeneous item for comparison.
- **DeepFM** [14] combines the power of factorization machines for recommendation and deep learning for feature learning.
- **xDeepFM** [27] uses the compressed interaction network to generate combinatorial features and merges them into a classic deep neural network.
- **AFM** [54] is the attentional factorization machine that could learn the importance of different feature interactions.
- **DSSM** [18] is one of the first proposed two-tower models used in the recommendation system.
- **Wide&Deep** [6] is an approach that uses wide linear models and deep neural networks to combine the benefits of memorization and generalization.
- **AutoInt** [41] is a deep model that could learn the high-order feature interactions of input features via self-attentive neural networks.

5.2.3 Heterogeneous Models. The third group contains two cross-domain models that could handle heterogeneous items mentioned in related work.
- **CCCFNet** [48] combines the collaborative filtering feature and content-based feature in a multi-view neural network.
- **DDTCDR** [25] develops a novel latent orthogonal mapping to extract user preferences over multiple sources while preserving the relations between users across different latent spaces. We extend the origin DDTCDR model to a multi-source version so that it could be applied to the heterogeneous item recommendation.

5.3 Effectiveness
5.3.1 Experiments on Public Dataset. We conduct effectiveness experiments in all of the baselines and our proposed model. For the contextual models, we run them in both heterogeneous and homogeneous settings. In the former setting (marked with He), these models use the overlapping attributes of different kinds of items as the contextual features; in the latter setting (marked with Ho), these models use full attributes of homogeneous items.

Table 3 shows the recommendation performance of baselines and our model. We have the following main observations.

Our model achieves the highest AUC scores in all kinds of items. In the heterogeneous setting, our model could exploit more attribute features of items. Besides, our model compares favorably to homogeneous contextual models due to its benefit from the data of all kinds of items. Basically, contextual models perform better than general models, and contextual models in the homogeneous setting perform better than them in the heterogeneous setting. Besides, the heterogeneous models (CCCFNet and DDTCDR) outperform the contextual models. To pursue this further, our model achieves significant improvements over other models. We could clearly see the positive effect of utilization of attribute features: general models drop all attributes of items; heterogeneous contextual models merely use the common attributes of items; homogeneous models could manipulate all attribute of one kind items; our model enables the usage of all attributes of different kinds of items. As the interaction matrix of items gets more sparse, the superiority of our model becomes more salient. Take the book item as an example, its density of the interaction matrix (i.e., # Rate/(# Item × # User)) is much lower than other two kinds of items (see Table 2). Table 3 illustrates that, when recommending books, the advantage of DURation is more significant, which demonstrates that our model profits from interaction data of the heterogeneous items.

Next, we conduct experiments to demonstrate that unified representations could benefit from attributes of all kinds of items. The representation of one kind of item also benefits from the information of other items. Aiming to show the connection between representations of different kinds of items, we conduct a case study. Firstly, we select some famous books and their movie adaptation.
Table 3: Comparative results with baselines and DURation. Different categories of models are separated by vertical lines. He indicates the heterogeneous setting and Ho indicates the homogeneous setting. The highest scores are marked in bold.

| Category | NAIS | DeepMF | FISM | DeepFM | xDeepFM | AFM | DSSM | Wide&Deep | AutoInt | CCCFNet | DDTCDR | DURation |
|---------|------|--------|------|--------|---------|-----|------|-----------|---------|---------|--------|---------|
| Book    | 0.554 | 0.588 | 0.571 | 0.721  | 0.737   | 0.721| 0.732| 0.732     | 0.738   | 0.722   | 0.739  | 0.722   | 0.738   | 0.741   | 0.734   | 0.766   |
| Music   | 0.563 | 0.631 | 0.576 | 0.756  | 0.782   | 0.759| 0.769| 0.769     | 0.769   | 0.769   | 0.759  | 0.784   | 0.768   | 0.784   | 0.786   | 0.818   |
| Movie   | 0.651 | 0.656 | 0.656 | 0.76   | 0.777   | 0.761| 0.78 | 0.771     | 0.582   | 0.707   | 0.763  | 0.763   | 0.784   | 0.782   | 0.774   | 0.792   |
| All     | 0.592 | 0.647 | 0.596 | 0.77   | /       | 0.77 | /   | 0.708     | /       | 0.569   | /      | 0.771   | /       | 0.771   | /       | 0.804   |

Figure 2: Book-Movie Similarity Heatmap. The horizontal axis represents ten books while the vertical axis represents their movie adaptations. The color of the block indicates the similarity between the corresponding book and movie. The bluer the higher the similarity and vice versa.

Then we obtain their unified representations and calculate their pair-wise similarity to construct the correlation matrix. Figure 2 is the heatmap of the correlation matrix. The vertical axis represents the movies and the horizontal axis indicates their original books. Each block shows the similarity between the book and the movie. The bluer the block is, the more similar the book and the movie are. When the color becomes redder, the book and the movie are more distinct. We can see that the block in the diagonal is bluer than the other parts of the figure dramatically. It shows that the unified representations could capture the adaptation relationship of books and movies. Besides, it is reasonable to see that "Love in the Time of Cholera" is more distinct from other books/movies because this book and its film adaptation are relatively unpopular.

As a summary, we could draw the conclusion that the unified representations of different kinds of items are integrated well in the new feature space.

5.3.2 Online A/B Testing.

In addition to the offline experiments in the public dataset, we also apply DURation model in a real-world large-scale online recommendation system. In this scenario, there are four types of items. They are merchandise, videos, QAs (Question-Answer), and articles. In this recommendation system, the Daily Active Users (DAU) are about six million. We train our model with the data set that sampled from user behaviors in one day and deploy this model in the next day’s recommendation. The amount of data in our training set here is about 50 million.

The original algorithm in this scenario is a CCCFNet-based [48] model. Except for the basic components of the naive CCCFNet, this model also utilizes self-attention to extract the user representation from the sequence of user behavior.

In the offline experiment, the AUC of our model is 0.675 while the AUC of the baseline is 0.667. Compared with this baseline, our proposed method contributes to 3.67% CTR gain, 4.26% PV gain, and 1.75% UV gain in the online A/B test. Especially, our model performs relatively better in the setting of cold-start, which leads to a 27.22% CTR increase in the recommendation of new items and a 2.59% CTR increase in new users. The definition of a new user/item is the user/item that had no activity in the past 30 days. Table 4 illustrates the detailed results of online experiments including the performance of our model in different types of items.

Table 4: Online experimental results.

| Category | PV gain | UV gain | CTR gain |
|----------|---------|---------|----------|
| All      | +4.26%  | +1.75%  | +3.67%   |
| Merchandise | -1.05%  | +1.21%  | +1.25%   |
| Article  | +10.67% | +4.05%  | +2.62%   |
| QA       | +6.78%  | +1.01%  | +3.04%   |
| Video    | +1.5%   | +1.31%  | +5.70%   |

5.4 Ablation Study

In this subsection, we conduct experiments to demonstrate how each part of our model affects the learning of the unified representation. To validate the necessity of the alignment loss and topology loss, we remove them and then test our model. The results show that the AUC score drops from 0.804 to 0.798 in the cold-start setting. As the cold-start problem is one of the most significant challenges in the recommendation system, we also test the performance of our model in the cold-start setting. In this setting, the user in the test set is inactive and we only reserve up to three samples of these users each in the training set. Removing A and T decreases the AUC score remarkably (from 0.675 to 0.649) in the cold-start setting. For detailed results, please refer to Table 5.
To get a better understanding of alignment loss $A$, we use t-SNE [4, 49] to do dimensionality reduction and then plot the distribution of our unified representations using kernel density estimate (KDE). Figure 3 illustrates the results of visualization. The upper plot is the representations learned by the model without alignment loss and the underplot is the output of our model with alignment loss. Different color indicates a different kind of items. We can clearly see that the cliques in the upper sub-figure are more distinct and the distributions in the under sub-figure are well-mixed. It proves that the alignment loss helps minimize of dissimilarity of different kinds of items in the unified feature space so that the unified representation becomes more tractable.

| Setting     | Model     | All  | Book | Music | Movie |
|-------------|-----------|------|------|-------|-------|
| Cold-start  | Duration  | 0.576| 0.707| 0.667 | 0.675 |
|             | Remove $A$ and $T$ | 0.565| 0.69  | 0.619 | 0.649 |
| Full data   | Duration  | 0.804| 0.766| 0.819 | 0.792 |
|             | Remove $A$ and $T$ | 0.798| 0.759| 0.817 | 0.79  |

Table 5: Performance result in ablation study.

We also design an experiment to validate the competency of the topology loss $T$. Firstly, for each item kind, we cluster the raw item representations using the K-Means algorithm, then assign each item a pseudo label. Afterward, we cluster the items with their unified representations learned in two settings (with/without topology loss). Then we calculate the macro pairwise F1 score of the two cluster results generated by raw representations and unified representations. For each pair of items, if their raw representations belong to the same cluster, this pair is regarded as a positive sample. In that situation, if their deep unified representations also belong to the same cluster, it leads to a true-positive prediction. Analogously, we could count the number of true negatives, false positives, and false negatives. Accordingly, the pairwise precision and pairwise recall are easy to be induced by the definition in the simple binary classification context. Figure 4 presents the pairwise F1 score of models in different settings. We can clearly see that if removed the topology loss $T$, the F1 score goes down. This indicates that the topology loss indeed boosts the ability to preserve the topology information of origin feature space.

6 CONCLUSION

In this work, we propose a novel deep unified representation for the heterogeneous recommendation. We leverage the strengths of attributes of different kinds of items to learn the unified exquisite representations. We design an easy-to-extend framework and introduce the alignment loss, topology loss, and classification loss in the learning of unified representation. In theory and in practice, we prove that the representation could align different transformed distributions while preserving the topology of raw feature spaces. Furthermore, our model achieves much better performance compared with baselines on real-world offline and online datasets.

REFERENCES

[1] Alain Berlinet and Christine Thomas-Agnan. 2011. Reproducing kernel Hilbert spaces in probability and statistics. Springer Science & Business Media.
you need. arXiv preprint arXiv:1706.03762 (2017).

[51] Chang Wang and Sridhar Mahadevan. 2011. Heterogeneous domain adaptation using manifold alignment. In IJCAI proceedings-international joint conference on artificial intelligence, Vol. 22. 1541.

[52] X. Wang, Z. Peng, Senzheng Wang, Philip S. Yu, Wenzing Fu, and Xiaoguang Hong. 2018. Cross-Domain Recommendation for Cold-Start Users via Neighborhood Based Feature Mapping. arXiv abs/1803.01617 (2018).

[53] Karl Weiss, Taghi M Khoshgoftaar, and DingDing Wang. 2016. A survey of transfer learning. Journal of big data 3, 1 (2016), 1–40.

[54] Jun Xiao, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. arXiv preprint arXiv:1708.04617 (2017).

[55] Z. Xu, Huizhen Jiang, Xiangjie Kong, Jialiang Kang, W. Wang, and Feng Xia. 2016. Cross-domain item recommendation based on user similarity. Comput. Sci. Inf. Syst. 13 (2016), 359–373.

[56] Hong-Jian Xue, Xinyu Dai, Jianbing Zhang, Shujian Huang, and Jiajun Chen. 2017. Deep Matrix Factorization Models for Recommender Systems. In IJCAI. Vol. 17. Melbourn, Australia, 3203–3209.

[57] Xinyang Yi, Ji Yang, Lichan Hong, Derek Zhiyuan Cheng, Lukasz Heldt, Aditee Kumthekar, Zhe Zhao, Li Wei, and Ed Chia. 2019. Sampling-bias-corrected neural modeling for large corpus item recommendations. In Proceedings of the 13th ACM Conference on Recommender Systems. 269–277.

[58] Wayne Xin Zhao, Shanlei Mu, Yu Peng Hou, Zihan Lin, Kaiyuan Li, Yuduo Chen, Yujie Lu, Hui Wang, Changxin Tian, Xingyu Pan, Yingqian Min, Zhichao Feng, Xinyan Fan, Xu Chen, Pengfei Wang, Wendi Ji, Yaliang Li, Xiaoling Wang, and Ji-Rong Wen. 2020. RecBole: Towards a Unified, Comprehensive and Efficient Framework for Recommendation Algorithms. arXiv preprint arXiv:2011.01731 (2020).
**A SUPPLEMENTARY MATERIAL**

**A.1 Proofs**

**A.1.1 Proof of Theorem 1.**

Proof. We prove the Theorem 1A first. It is clear that $\mu[\mathbb{F}] = \mu[\mathbb{Q}]$ if $\mathbb{F} = \mathbb{Q}$. If $\mathbb{F} \neq \mathbb{Q}$, then MMD$(C(X), \mathbb{P}, \mathbb{Q}) > 0$ based on the Lemma 1 in [12] where MMD is maximum mean discrepancy and $C(X)$ is the space of bounded continuous functions on $X$. Therefore, we could prove the converse by showing that MMD$(C(X), \mathbb{P}, \mathbb{Q}) = D$ for some $D > 0$ implies $\mu[\mathbb{F}] \neq \mu[\mathbb{Q}]$.

Because the RBF kernel $k$ is universal, its associated RKHS $\mathcal{H}$ is also universal. Let $\mathcal{F}$ be the unit ball in $\mathcal{H}$. If MMD$(C(X), \mathbb{P}, \mathbb{Q}) = D$, then there exists some $f \in C(X)$ for which $E_{\mathbb{P}}[f] - E_{\mathbb{Q}}[f] \geq D/2$. Therefore, for $\epsilon = D/8$, there exists $f^* \in \mathcal{F}$ so that $\|f^* - f\|_{\infty} < \epsilon$ since $\mathcal{H}$ is dense in $C(X)$ with respect to the $L_{\infty}$ norm. Then, we have

\[
\|E_{\mathbb{P}}[f^*] - E_{\mathbb{Q}}[f^*]\|_{\mathcal{H}} > \epsilon
\]

and consequently

\[
\left|E_{\mathbb{P}}[f^*] - E_{\mathbb{Q}}[f^*]\right| > \left|E_{\mathbb{P}}[f] - E_{\mathbb{Q}}[f]\right| - 2\epsilon
\]

Then we obtain

\[
\frac{E_{\mathbb{P}}[f^*] - E_{\mathbb{Q}}[f^*]}{\|f^*\|_{\mathcal{H}}} \geq \frac{D}{4} > 0.
\]

Therefore, the definition of MMD establishes that MMD$(\mathcal{F}, \mathbb{P}, \mathbb{Q}) > 0$. Since MMD$(\mathcal{F}, \mathbb{P}, \mathbb{Q}) = \|\mu[\mathbb{P}] - \mu[\mathbb{Q}]\|_{\mathcal{H}}$, we have $\mu[\mathbb{F}] \neq \mu[\mathbb{Q}]$. Finally, we can conclude that now $\mu[\mathbb{F}] = \mu[\mathbb{Q}]$ if and only if $\mathbb{P} = \mathbb{Q}$.

Since the Theorem 1A is proved, namely, the mean map $\mu \rightarrow \int_X k(x, y) \, d\mu(x)$ is injective and hence the RBF kernel $k$ is characteristic by the Definition 1. Along this line, we can construct $\mathbb{F} \neq \mathbb{Q}$ such that $\mu[\mathbb{F}] = \mu[\mathbb{Q}]$ for any arbitrary small $\epsilon > 0$ which is proved in previous work ([43], Theorem 9).

**A.1.2 Proof of Equation 4.**

Proof. From $\mathcal{G}_H(S)$ to Equation 4, we have

\[
\mathcal{G}_H(S) = \frac{1}{P} \text{tr}(\Sigma) = \frac{1}{P} \text{tr}(\hat{G}) = \frac{1}{P^2} \sum_{i,j} \sum_{l} \hat{G}_{ij} = \frac{1}{P^2} \sum_{i,j} \sum_{l} \sum_{k=1}^{N_k} \kappa(x_i^k, x_j^l) - \frac{1}{P^2} \sum_{i,j} \sum_{l} \sum_{k=1}^{N_k} \nu_{ij} \kappa(x_i^k, x_j^l) = \frac{P}{P^2} \sum_{i,j} \sum_{l} \sum_{k=1}^{N_k} A_{ijkl}
\]

where

\[
A_{ijkl} = \begin{cases} \frac{P-1}{P^2 N_k} \kappa(x_i^k, x_j^l) & i = j \\ \frac{-1}{P^2 N_k} \kappa(x_i^k, x_j^l) & i \neq j \end{cases}
\]

\[\square\]

**A.2 Reproducibility**

**A.2.1 Dataset.** To facilitate the reproduction of the results in our paper, we provide the processed Douban dataset in the supplementary material. Download and uncompress it to \texttt{/HRec/data/douban} folder. The dataset have been desensitized so that it contains no personally identifiable information or offensive content. In this dataset, there are 1,096,240 interaction records between 303,825 items and 36,585 users. The interaction data is a rate from one-star to five-stars. For the convenience of comparison with some classic models, we convert the recommendation problem in the Douban dataset to a binary classification problem. The rates more than three-stars are treated as positive labels and others are treated as negative labels.

**A.2.2 Code.** The code of our proposed model and baselines is available in Github\(^1\). We provide the easy-to-use code so that anyone could run it with one simple line of code.

The code is written in Python3.7. We use the experience of ReCBole [58] for the implementation of part of the baselines.

**Usage**

The usage of our code is quite simple. Firstly, prepare a configuration file in JSON format. You can refer to the existing configuration files in the \texttt{configs} folder for examples.

For instance, given a configuration file named `duration.json` in \texttt{configs}, just run the following command:

```bash
$ python train_hete.py duration
```

For the homogeneous models, the usage is the same except for replacing `train_hete.py` with `train_homo.py`. Take the deepmf model as an example:

```bash
$ python train_homo.py deepmf
```

Once the program starts successfully, it creates a subfolder in `output`. The subfolder [Model name]–[Dataset name]–[time] contains the config file, the log file, and the best and the last checkpoints of the training model.

**A.2.3 Setup.** In our offline experiments, the Douban dataset is divided into the train set (70%), the validate set (20%), and the test set (10%). All of the models are trained with mini-batch Adam optimizer using early stopping criteria: if the performance of the model in the validation set does not improve within a patience threshold (five epochs here) then the training stops. Otherwise, the maximum number of training epochs is 100. The batch size is set to be 1024 and the initial learning rate is 0.001.

After training, we select the checkpoint which performs best in the validate set and apply it in the test set to obtain the final result. As the evaluation metric, we use the Area Under Curve (AUC) to measure the probability that a model will assign a higher score to a randomly chosen positive item than a randomly chosen negative item. A higher AUC score indicates higher performance. We calculate the AUC for each individual kind of item along with the overall AUC of all items.

For a recommendation system, an important hyperparameter is the embedding size of users/items. We vary them in the set of \{32,
64, 128) and observe that the optimal performance is obtained with 64. With regard to $\alpha$ and $\beta$ in Equation 8, the selection of these two hyperparameters is related to the batch size due to our optimization method mentioned in Section 4.3. Therefore, the guideline for tuning $\alpha$ and $\beta$ is to make the three loss terms in the same order of magnitude. When batch size is 1024, experimental results show that the setting of this model $\alpha = 5 \times 10^8, \beta = 0.001$ leads to the best performance. Overall, differences are minor indicating the insensitivity of hyperparameters. We list the remaining hyperparameters of baselines in Table 6.

Table 6: Hyperparameters of baselines.