Vietoris-Rips Complex: A New Direction for Cross-Domain Cold-Start Recommendation

Ajay Krishna Vajjala* Dipak Falgun Meher* Shrunal Pothagoni* Ziwei Zhu*
David S. Rosenblum*

Abstract

Cross-domain recommendation (CDR) has emerged as a promising solution to alleviating the cold-start problem by leveraging information from an auxiliary source domain to generate recommendations in a target domain. Most CDR techniques fall into a category known as *bridge-based methods*, but many of them fail to account for the structure and rating behavior of target users from the source domain into the recommendation process. Therefore, we present a novel framework called Vietoris-Rips Complex for Cross-Domain Recommendation (VRCDR), which utilizes the Vietoris-Rips Complex (a technique from computational geometry) to understand the underlying structure in user behavior from the source domain, and includes the learned information into recommendations in the target domain to make the recommendations more personalized to users' niche preferences. Extensive experiments on large, real-world datasets demonstrate that VRCDR consistently improves recommendations compared to state-of-the-art bridge-based CDR methods.

1 Introduction

Considering the exponential surge in online data, finding desired information for users can become nearly insurmountable[19, 2]. E-commerce websites, known for collecting vast amounts of data, rely on recommender systems to help users discover information that aligns with their individual preferences [21]. Researchers have spent significant effort in improving the state of recommender systems over the past few years [1]. However, most of the existing methods struggle to make relevant recommendations to new users—the so-called "cold-start problem" [10]—and fail to leverage available information outside the domain of interest to improve the overall accuracy of recommendations.

Recently, cross-domain recommender (CDR) systems have emerged as a promising solution to alleviate the cold-start problem [22, 25]. They transfer information from a dense source domain to a sparse target domain, with a goal of providing more relevant recommendations in the target domain. The state-of-the-art CDR methods fall into a category known as *bridge-based methods* [9, 23, 24, 27], and the Embedding and Mapping Approach for CDR (EMCDR) [11] is known to be one of the most effective bridge-based methods for cross-domain recommendation. EMCDR uses an embedding-based mapping function to create new representations for users in the target domain, using information from the source domain [11]. Given the success of this technique, many bridge-based methods have been proposed over the past few years [27, 9, 28, 24, 23].

While these methods are encouraging, there are three major drawbacks that can affect their performance. First, existing methods fail to utilize users' interaction history from the source domain effectively to make more personalized recommendations [26, 23, 11]. Existing methods can capture the major preferences of users, but they often overlook niche preferences. This is because they do not directly leverage information about users' interactions in the source domain during the mapping process. In addition, they learn one bridge function shared by all users, which lacks the ability to capture distinct personal preferences of users, especially for those niche users. Second, existing bridge-based methods cannot work for extreme cold-start users (i.e., users who have no interactions in the target domain), because current methods minimize the distance between the ground truth target user embedding and the mapped embedding. However, these representations are inaccurate for users without target domain interactions [9, 11, 23]. Finally, existing methods fail to use item information from the source domain when mapping source users to the target domain [22, 25]. The content of source domain items that are interacted by the target users can help improve personalized recommendations.

Addressing these challenges, we present a novel framework called **V**ietoris-**R**ips Complex for **C**ross-**D**omain **R**ecommendation, called **VRCDR**. The framework first learns user and item representations separately in the source and target domains, using a neural network model that incorporates item information. It then creates a characteristic vector for each user using the Vietoris-Rips complex (Rips Complex)[6], a technique from computational geometry, to understand the geometry behind each user's rating patterns, and

*George Mason University,
{akrish, dmeher, spothago, zzhu20, dsr}@gmu.edu

uses the vector as input to the bridge mapping function. Finally, it adopts a task-oriented loss, which updates the weights of the non-linear mapping function based on rating prediction, rather than minimizing the distance between embeddings. Our implementation is available at https://github.com/ajaykv1/VRCDR. In sum, the contributions in this paper are as follows:

- We present a novel framework named **VRCDR** that leverages unique user preferences from a source domain to provide personalized recommendations in a target domain.
- We present a novel area-based triangulation method called **ATE**, which uses the Rips Complex to create characteristic vectors for users, making our study the first to incorporate the Rips Complex for CDR.
- Through extensive experimentation on 4 CDR tasks, applied to real-world datasets, we demonstrate that VRCDR significantly outperforms existing state-of-the-art bridge-based methods. Specifically, our framework performs significantly and consistently better over a diverse range of cold-start situations, where cold-start user percentages are set to 20%, 50%, and 70%.

2 Preliminaries

In this section, we first introduce the CDR problem setting and notation, then go over the neural network architecture, which we use as the base model for VRCDR.

2.1 Notation In cross-domain recommendation, we have a source and a target domain. Each domain has a set of users $\mathbf{U} = \{u_1, u_2, ..., u_m\}$, a set of items $\mathbf{I} = \{i_1, i_2, ..., i_n\}$, and a rating matrix $\boldsymbol{R}$. Given that there exist interactions between users and items in each domain, each domain can be represented as a collaborative filtering (CF) problem [7]. Using this information, we can build a rating matrix for the source domain, $\mathbf{R}_s \in \{1, 2, 3, 4, 5\}^{|U_s| \times |I_s|}$, and a rating matrix for the target domain, $\mathbf{R}_t \in \{1, 2, 3, 4, 5\}^{|U_t| \times |I_t|}$.

For the rating prediction task, the ratings that users give to items can be derived from:

$$\hat{r}_{ui} = f(u, i|\Theta), \tag{2.1}$$

where f is the interaction function [10], $\hat{r}_{ui}$ is the predicted rating, and Θ are model parameters. For neural network based collaborative filtering approaches, the function f is parameterized by a neural network [7],

$$f(\mathbf{x}_{ui}|\mathbf{P}, \mathbf{Q}, \theta) = \phi_o(\phi_L(\dots(\phi_1(\boldsymbol{x}_{ui}))\dots)), \tag{2.2}$$

where $\mathbf{P} \in \mathbb{R}^{m \times d}$ and $\mathbf{Q} \in \mathbb{R}^{n \times d}$ are embedding matrices for users and items, respectively, $\mathbf{x}_{ui}$ is the concatenated representation of user u's and item i's embeddings, and θ represents the weights and biases..

2.2 Base Model In this paper, we adopt a Multi-Layer Perceptron (MLP) network as the base model to learn user and item embeddings from the source and target domains. In addition to $\mathbf{P}$ and $\mathbf{Q}$, which are user and item embedding matrices, we also use $\mathbf{C}$ and $\mathbf{D}$, which are category and description embedding matrices that are associated with each item. Each item is associated with a category and a description, and we believe that leveraging content information can improve the overall learned representations for users and items in the source and target domains. In this paper, we adopt a pre-trained transformer known as DistilBERT [16] to retrieve embeddings for the text information. DistilBERT is a distilled version of BERT, which is 60% faster and 40% lighter than the original BERT model, and can achieve similar performances as BERT [16].

The base MLP network is separately trained on the source and target domain data. For each domain, the user, item, category, and description embeddings are concatenated together and passed onto multiple layers,

$$f(\mathbf{x}_{uicd}|\mathbf{P}, \mathbf{Q}, \mathbf{C}, \mathbf{D}, \theta) = \phi_o(\phi_L(\dots(\phi_1(\mathbf{x}_{uicd}))\dots)), \tag{2.3}$$

where $\mathbf{C} \in \mathbb{R}^{n \times d}$ and $\mathbf{D} \in \mathbb{R}^{n \times d}$ are the embedding matrices for content and descriptions (retrieved from DistilBERT), and $\mathbf{x}_{uicd}$ is the concatenated representation of user u's embedding, item i's embeddings, and the content and description embeddings for item i. The source and target domain are pre-trained individually using this base model, and the user and item embeddings from both domains are retrieved from the respective pre-trained models.

3 Architecture Overview for VRCDR

In this section, we present a high-level overview of the architecture for VRCDR. There are three main components: (i) The pre-training phase; (ii) The employment of a non-linear mapping function; (iii) The use of a task-oriented loss function. Figure 1 shows the overall architecture of the presented framework.

3.1 Pre-training Phase In Section 2.2, we introduced an MLP as our base model. Leveraging this base model, we pre-trained two distinct CF models, one for the source domain data ($\mathbf{M}_s$) and one for the target domain data ($\mathbf{M}_t$). The main objective of this pre-training phase is to derive robust embeddings for users and items across these domains. From $\mathbf{M}_s$, we extract $\mathbf{U}_s$ and

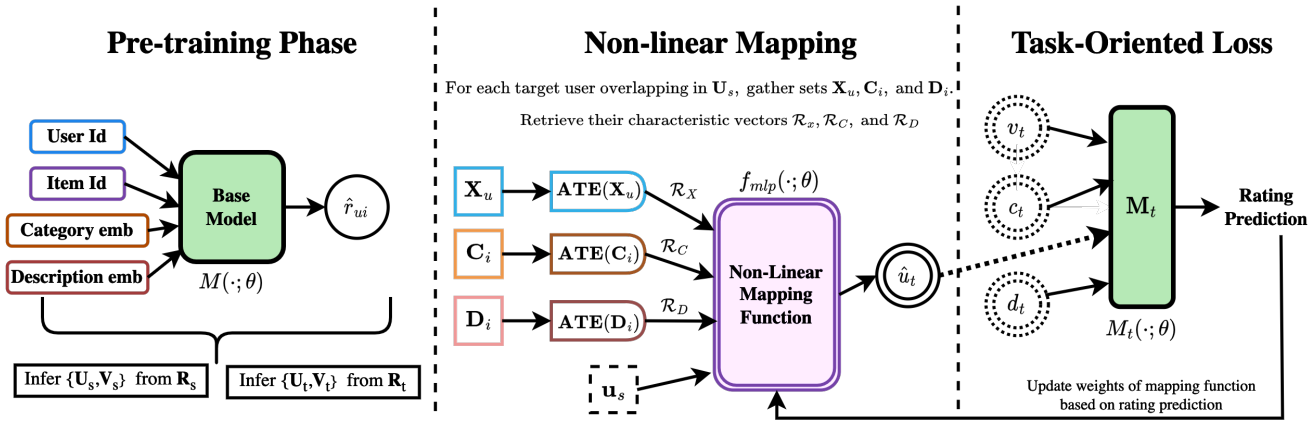


Figure 1: **VRCDR** has three components. First, user and item embeddings are inferred from the base source and target models. Next, the target user's interacted items and content information from the source domain are passed into **ATE** (Algorithm 1) to retrieve the characteristic vectors. Then, the outputs, along with the target user's corresponding source domain embedding (u_s), are fed into the mapping function. Finally, the transformed user embedding is passed to $\mathbf{M}_t$. Along with this, the original target item embedding (v_t) and its content information (d_t and c_t) are provided for rating prediction, which is used to update weights of the mapping function.

$\mathbf{V}_s$ representing the user and item embeddings of the source domain. Similarly, from $\mathbf{M}_t$, we obtain $\mathbf{U}_t$ and $\mathbf{V}_t$. These embeddings can be simply extracted from the embedding layers of both base models ($\mathbf{M}_s$ and $\mathbf{M}_t$).

In order to understand the unique preferences of target domain users, we believe it is important to leverage their rating information from the source domain. Therefore, for each target user $u \in \mathbf{U}_t$, we gather the list of their interacted items from the source domain. This list is denoted as $\mathbf{X}_u = \{{I_s}^1, {I_s}^2, \ldots, {I_s}^n\}$, where ${I_s}^n$ represents the n^{th} item interacted with by user u in the source domain. Along with the interacted items, we retrieve the category and description embeddings that are associated with each item from $\mathbf{X}_u$. The category list is denoted as $\mathbf{C}_i = \{{C_s}^1, {C_s}^2, \ldots, {C_s}^n\}$. Similarly, the description list is represented as $\mathbf{D}_i = \{{D_s}^1, {D_s}^2, \ldots, {D_s}^n\}$. Here, ${C_s}^n$ and ${D_s}^n$ correspond to the category and description embeddings of the n^{th} item interacted with by user u from the source domain.

3.2 Nonlinear MLP mapping function For each list gathered during the pre-training phase ($\boldsymbol{X}_u$, $\boldsymbol{C}_i$, $\boldsymbol{D}_i$), we construct a characteristic vector by employing the Rips Complex. The characteristic vectors preserve geometrical and structural information from the embeddings within the lists, which can provide additional information about the target user's rating behavior in the source domain. Therefore, by employing the Rips Complex, we obtain three distinct characteristic vectors that correspond to each list: $\boldsymbol{\mathcal{R}}_{\mathcal{X}}$, $\boldsymbol{\mathcal{R}}_{\mathcal{C}}$, and $\boldsymbol{\mathcal{R}}_{\mathcal{D}}$. The details of how this is done are provided later in Section 4, where we introduce a novel algorithm called **ATE** (Algorithm 1). Once the characteristic vectors for each list are computed, we employ a non-linear mapping function to output new embeddings for the user in the target domain. We adopt the MLP mapping function from EMCDR, due to its effectiveness in modeling the non-linear relationships between the input data [11]. In EMCDR, the user embedding from the source domain is used as the only input to the mapping function. We believe that solely using the user embedding as the input is not effective. We also believe that by providing additional information as input to the MLP network, the outputs can be more accurate representations for users in the target domain. Therefore, the modified MLP mapping function can be defined as:

$$\hat{\boldsymbol{u}}_t = f_{mlp}([u, \boldsymbol{\mathcal{R}}_X, \boldsymbol{\mathcal{R}}_C, \boldsymbol{\mathcal{R}}_D]; \theta), \tag{3.4}$$

where $u \in \boldsymbol{U}_s$ represents the user embedding from the source domain for the corresponding target user, $\boldsymbol{\mathcal{R}}_X$, $\boldsymbol{\mathcal{R}}_C$ and $\boldsymbol{\mathcal{R}}_D$ represents the characteristic vectors obtained from using the Rips Complex on $\boldsymbol{X}_u$, $\boldsymbol{C}_i$, and $\boldsymbol{D}_i$, $\hat{\boldsymbol{u}}_t$ represents the transformed target user embedding, and θ represents the model parameters. We can use $\hat{\boldsymbol{u}}_t$ for the rating prediction task.

3.3 Task oriented Loss Function For our framework, we adopt a task oriented loss function for optimizing the weights of the the mapping function. Many existing works utilize the MSE loss to minimize distance between transformed user embeddings and ground truth target user embeddings. The MSE loss function is de-

fined as follows [11]:

$$L = \sum_{u_t \in \boldsymbol{OU}} ||\hat{\boldsymbol{u}}_t - u_t||^2, \tag{3.5}$$

where $\hat{\boldsymbol{u}}_t$ is the transformed target user embedding, u_t is the ground truth target user embedding, and $\boldsymbol{OU}$ represents the overlapping users between the source and target domains. The ground truth embeddings for extreme cold-start users in the target domain will not be accurate, since users will have no interactions. To address this issue, we adopt a task oriented loss function, which has been adopted recently in CDR [28]:

$$L = \frac{1}{N} \sum_{r_{ij} \in \boldsymbol{R}_t^{OU}} [r_{ij} - (\hat{\boldsymbol{u}}_i \cdot v_j)]^2 \tag{3.6}$$

where r_{ij} represents the ground truth rating for which user i gave item j, $\hat{\boldsymbol{u}}_i$ represents the transformed target user embedding from the mapping function, v_j represents the ground truth target item from $\boldsymbol{M}_t$, where $v_j \in \boldsymbol{V}_t$, and $\boldsymbol{R}_t^{OU}$ represents the rating matrix for user and item interactions in the target domain, where the users overlap between both domains. This loss function relies only on rating information as ground truth information, which alleviates the shortcomings faced by unstable user embeddings.

4 The Rips Complex for VRCDR

In order to understand more about a target user's preferences, we analyze their rating behavior from the source domain, and include the learned information into the recommendation process as a characteristic vector. To do this, we gather the target user's interacted items in the source domain, and analyze the geometrical relationship between them using the Rips Complex. We argue that the geometrical relationship between items can provide additional information about how items relate to each other in higher dimensional spaces, which can help uncover the unique preferences for each target user. Therefore, we present a novel algorithm called **ATE** (Algorithm 1) that uses the Rips Complex to compute the characteristic vectors. **ATE** takes in a list of embeddings as input, and outputs a single characteristic vector that captures the geometrical and structural information among them. In this section, we first introduce the concept and notations for the Rips Complex. Then, we describe our novel approach, **ATE**.

4.1 The Rips Complex A key tenet of computational geometry is that data has shape and shape has meaning, and the Rips Complex is a tool used to generate and study the shape of data [6]. Now, we introduce the notations and concepts for the Rips Complex. Geometrically, a k-simplex is interpreted as a generalization of triangles, shown in Figure 2. For any k-simplex, its *boundary* is the set of $(k-1)$-simplices that encompass it. For instance, the boundary of a 1-simplex are its two 0-simplices, the boundary of a 2-simplex are the bordering three 1-simplices, etc. We often denote the boundary of a simplex $[\sigma]$ by $\partial[\sigma]$. A k-simplex is *open*, denoted (s), if it does not contain its boundary (i.e. $[\sigma] - \partial[\sigma]$) and *closed* if the boundary is contained.

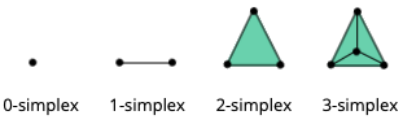


Figure 2: Visualization of simplices up to 3 dimensions.

Definition 4.1. A *simplicial complex* K is a finite set of open simplices in some $\mathbb{R}^n$ such that:

(1) if $(s) \in K$ *then all open faces of* $[s] \in K$*;*

(2) if $(s_1) \cap (s_2) \neq \emptyset$ *then* $(s_1) = (s_2)$.

A simplicial complex defines a set of rules that allow us to combine simplices of various dimensions in a way that is geometrically intuitive. To reference the set of n dimensional simplicies of a complex K we define the *n-skeleton*, denoted by $\text{skel}_n(K)$, to be the set of all of the simplices in K of dimension n. In practice we tend to only compute up to the 2-skeleton.

Definition 4.2. Given a set of points $\mathcal{X} = \{x_1, x_2, \ldots, x_k\} \subset \mathbb{R}^n$ and an $\epsilon > 0$, an k-simplex $\sigma = [x_{i_1}, x_{i_2}, \ldots x_{i_k}]$ is in the Vietoris-Rips Complex $\mathit{Rips}_\epsilon(\mathcal{X})$ if and only if:

$$\mathbb{B}_\epsilon(x_{i_j}) \cap \mathbb{B}_\epsilon(x_{i_{j'}}) \neq \emptyset$$

where $\mathbb{B}_\epsilon(x_{i_j})$ is an open ball at x_{i_j}.

Although there are a plethora of methods that can be used to create a simplicial complex from a point cloud, we use the Rips Complex due to the existence of fast implementations relative to other methods [30].

Given that the Rips Complex can find underlying shapes from embedded data, we adopt this method to understand the niche preferences of target domain users. To give an example, lets say a target user interacted with a few items from the source domain. Figure 3 shows the Rips Complex for the target user's interacted source domain items. We can see that there are two 2-simplices created. The first 2-simplex is formed around the items Twilight, Avatar, and Inception, while the

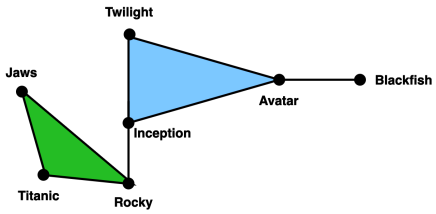


Figure 3: Example Rips Complex for user's interactions

second is formed around Jaws, Rocky, and Titanic. Using this triangulation, we can derive the preferences for the user based on the items that form each 2-simplex. The first 2-simplex contains items from the fantasy genre, while the second contains items from the realistic fiction genre. This structure reveals that the two genres are closely linked with the target user's preferences. In general, we are computing the Rips Complex to possibly find these implicit (niche) preferences within the data.

4.2 Characteristic Vectors using the Rips Complex Existing CDR methods fail to leverage the niche preferences of users effectively, primarily because they don't analyze the relationship between items from the source domain that target users have interacted with [27, 9, 28, 23]. We aim to fill that gap by applying the Rips Complex [6] to learn the shape and structure of item embeddings that target users have interacted with from the source domain.

To begin, let ϵ be some fixed edge-length. We then compute the Rips Complex over a list of embeddings $\boldsymbol{\mathcal{L}}$ which we denote:

$$\boldsymbol{\mathcal{K}} = Rips_{\epsilon}(\boldsymbol{\mathcal{L}}) \tag{4.7}$$

As stated before, our implementation of the Rips Complex will compute the structure of the simplicial complex up to its 2-skeleton. Once the 2-simplices are formed, we need a way to accurately capture the structural information to effectively understand the user preferences. Therefore, we develop a method that creates a weight coefficient for each two-simplex, which emphasizes the niche preferences for a particular user. We call this novel algorithmic approach as **A**rea-based **T**riangulated **E**mbedding **(ATE)** (see Algorithm 1). The algorithm works by first computing the area of a given 2-simplex from $\boldsymbol{\mathcal{K}}$ using Heron's formula [20]:

$$\text{Area} = \sqrt{S(S-A)(S-B)(S-C)} \tag{4.8}$$

where S is the semi-perimeter of the 2-simplex and A, B, and C are the length of the three edges from $\partial\sigma$. The area is then used to scale the averaged embeddings, which is expressed as:

$$\text{AE} = \frac{1}{|\partial\sigma|} \sum_{v \in \partial\sigma} v \tag{4.9}$$

We iterate this method with every 2-simplex from the 2-skeleton until we are left with a set of new embeddings denoted $\boldsymbol{\mathcal{E}}$. The last step is to then average the embeddings of $\boldsymbol{\mathcal{E}}$ with all the other unused embeddings from $\boldsymbol{\mathcal{K}}$ to generate the characteristic vector $\boldsymbol{\mathcal{R}}_{\mathcal{L}}$. For each list gathered during the pre-training

Algorithm 1 Area-based Triangulated Embedding

Require: $\boldsymbol{\mathcal{L}}$
Ensure: $\boldsymbol{\mathcal{L}} \neq \emptyset$
1: $\boldsymbol{\mathcal{K}} = Rips_{\epsilon}(\boldsymbol{\mathcal{L}})$
2: $\boldsymbol{\mathcal{E}} = [\,]$
3: **for** $\sigma \in \text{skel}_2(\boldsymbol{\mathcal{K}})$ **do**
4: $\quad w_\sigma = \text{Area}(\sigma)$
5: $\quad \boldsymbol{\mu} = w_\sigma \cdot \text{AE}(\partial\sigma)$
6: $\quad \boldsymbol{\mathcal{E}}.\text{append}(\boldsymbol{\mu})$
7: **end for**
8: $\boldsymbol{\mathcal{R}}_{\mathcal{L}} = \text{AE}(\boldsymbol{\mathcal{E}}, \boldsymbol{\mathcal{K}} - \text{skel}_2(\boldsymbol{\mathcal{K}}))$

phase ($\boldsymbol{X}_u$, $\boldsymbol{C}_i$, $\boldsymbol{D}_i$), we compute their characteristic vectors ($\boldsymbol{\mathcal{R}}_{\mathcal{X}}$, $\boldsymbol{\mathcal{R}}_{\mathcal{C}}$, $\boldsymbol{\mathcal{R}}_{\mathcal{D}}$) using **ATE**. This allow us to incorporate the user's individual preferences directly into the recommendation process. In the case of cold-start and warm-start users, the characteristic vectors provide additional information about their preferences from other domains. This can potentially lead to more personalized recommendations when they don't have much rating information in the target domain.

5 Experiments

We conducted extensive experiments to evaluate the VRCDR framework, and aim to answer the following research questions: **RQ1** How does VRCDR perform compared to state-of-the-art baselines? **RQ2** How does VRCDR perform in extreme cold-start situations? **RQ3** What impact do hyperparameters from the Rips Complex have on recommendation performance? **RQ4** How effective is the Rips Complex in VRCDR?

5.1 Experimental Setup

5.1.1 Dataset For experimentation, we adopt the widely used Amazon Review 5-scores dataset [12], where each user has at least 5 ratings. We chose five categories to represent domains in our experimental study. The domains we chose are movies_and_tv (Movie), cds_and_vinyl (Music), Electronics, Video_Games (Games), and Grocery_and_Gourmet_Food (Food).

CDR Tasks	Domain Combination		Item #		User #			Rating #	
	Source	Target	Source	Target	Source	Target	Overlap	Source	Target
Task 1	Movie	Electronics	50052	63001	123960	192403	15500	1860253	1785863
Task 2	Movie	Music	50052	64443	123960	75258	18031	1860253	1283356
Task 3	Movie	Food	50052	8713	123960	14681	3330	1860253	152369
Task 4	Electronics	Games	63001	10672	192403	24303	7419	1785863	279769

Table 1: Statistics for domains from the Amazon Review Dataset, along with cross-domain recommendation tasks.

From the selected domains, we define four CDR tasks: **Task 1:** Movie→Electronics, **Task 2:** Movie→Music, **Task 3:** Movie→Food, and **Task 4:** Electronics→Games. Each task has more rating information in the source domain compared to the target domain, which mimics real life applications for CDR. Table 1 shows details for the dataset and tasks.

5.1.2 Baseline Models Given that VRCDR falls into the category of bridge-based methods for CDR, we choose mainly bridge-based CDR methods for the baseline approaches. We employ Matrix Factorization [10] (MF) and Generalized MF (GMF) [17] as the base embedding models for the CDR baselines to capture the linear and non-linear performances. As a result, the baselines chosen for experimentation are as follows:

- **TGT** [10]: A naive Matrix Factorization model, which is trained solely on the target domain data.
- **CMF** [13]: A CDR model which extends matrix factorization by factorizing rating matrices for multiple domains simultaneously while maintaining a shared global user embedding.
- **EMCDR** [11]: A mapping based approach for cross-domain recommendation that uses a MLP network as the general bridge function.
- **PTUPCDR** [28]: A personalized bridge-based approach for cross-domain recommendation that uses a meta-network and a linear mapping function to create personalized bridges for users.

5.1.3 Evaluation Metrics Given that the Amazon Review dataset [12] contains numerical ratings, we use Root Mean Squared error and Mean Absolute Error as the evaluation metrics for the rating prediction task, which has been used widely for CDR evaluation [11, 28].

(5.10)
$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}, \quad \text{MAE} = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

5.1.4 Implementation Details We implemented our framework using Tensorflow, and used the open source code provided by the authors of PTUPCDR [28] for the baselines. We tuned the learning rate for the Adam optimizer using 5-fold cross validation within {0.00001, 0.00002, 0.00005, 0.0001, 0.0002, 0.0005} for all methods, and set the dropout rate to 0.2. The batch size is set to 256 and the embedding dimension is set to 16 for all methods. Each model is trained for 50 epochs or until the validation performance does not improve for 5 epochs. For PTUPCDR, the meta-network is a two-layer neural network with $2\times K$ hidden units, where K is the embedding dimension. For EMCDR and VRCDR, the mapping function is a one-layer MLP with $2 \times K$ hidden units, where the hidden layer uses tanh, and the output layer uses sigmoid as the activation functions.

To test the effectiveness of VRCDR, we remove all ratings for a fraction of the overlapping users in the target domain randomly, and use them as test users for extreme cold-start scenarios. The data points that were not removed are used as the training set for the mapping function. We followed the same dataset partitioning as PTUPCDR in our experimental setup [28]. We set the proportion of cold-start users ($\boldsymbol{\beta}$) as 20%, 50%, 70%, and report the average results over 10 runs.

5.2 Results

5.2.1 RQ1: Performance against baselines In this section, we demonstrate the effectiveness of VRCDR on 4 CDR scenarios. The results are shown in Table 2. Several findings can be derived from the experimental results. (i) Given that TGT is a naive single domain recommendation model, the poor performance is expected. All other CDR approaches outperformed TGT, highlighting the effectiveness of transferring information from other domains to improve recommendation performance. (ii) We employed MF and GMF as the base models for CMF, EMCDR, and PTUPCDR. CMF is a linear-based approach, while both EMCDR and PTUPCDR are deep learning-based approaches. This allows us to capture the performance when using em-

Task	Metric	TGT	MF-CMF	MF-EMCDR	MF-PTUPCDR	GMF-CMF	GMF-EMCDR	GMF-PTUPCDR	VRCDR
1	MAE	4.405	2.2463	1.3633	1.2592	2.0597	1.4051	1.1211	**0.9631***
	RMSE	4.9675	2.8422	1.6905	1.6275	2.4825	1.6889	1.4771	**1.1641***
2	MAE	4.5434	1.8199	2.0693	1.5133	1.6614	2.1542	1.4195	**0.8788***
	RMSE	5.3371	2.4213	2.4075	2.0242	2.0852	2.4471	1.9132	**1.1281***
3	MAE	4.5287	2.5415	3.1997	2.5105	2.2363	3.0202	2.1399	**0.9137***
	RMSE	5.3501	3.2475	3.5654	3.2813	2.8045	3.3525	2.9305	**1.1355***
4	MAE	4.5488	2.4314	3.1371	2.1838	2.5162	2.8393	1.8659	**0.9025***
	RMSE	5.4177	3.0427	3.4133	2.8388	3.0384	3.1207	2.4465	**1.1429***

Table 2: Results for 4 CDR tasks against state-of-the-art baseline methods. Best results are in **bold**, and * indicates a p-value less than 0.01 from two-tailed paired t-test between VRCDR and best baseline.

beddings from both linear and non-linear models. We can see that the MF based CDR approaches performed poorly compared to the GMF based approaches. This is expected, as GMF is able to capture nonlinear interaction patterns between users and items, which can lead to more effective knowledge transfer. The results show that leveraging deep learning and non-linearity can help improve performance for CDR tasks. (iii) VRCDR performed significantly better on both evaluation metrics across all tasks. This demonstrates the effectiveness of leveraging the structure of users' interaction patterns along with item content to improve CDR performance.

5.2.2 RQ2: Cold-start Performance In this section, we evaluate the performance of VRCDR in extreme cold-start scenarios. Figures 4 and 5 show results for RMSE and MAE for VRCDR, respectively, compared to the best baseline methods. We choose 20%, 50%, and 70% to be the values for $\boldsymbol{\beta}$, and analyze the effect of recommendation performance with respect to the percent of extreme cold-start users. From the results, we can see that VRCDR significantly outperforms the baselines across all tasks and $\boldsymbol{\beta}$ values, except for task 2 when $\boldsymbol{\beta}$ is 20%. When $\boldsymbol{\beta}$ is 20% for task 2, we find that the RMSE between the best baseline and VRCDR is not significant, which implies similar performance between the two methods. However, the MAE of the best baseline is significantly better than the performance of VRCDR. This can be considered an outlier, given that all other tasks performed significantly better with VRCDR for every value of $\boldsymbol{\beta}$. We show that VRCDR can outperform the best baselines in most situations, demonstrating that the it is effective for cold-start recommendation.

5.2.3 RQ3: Hyperparameter Analysis In this section, we perform additional experiments to understand how the hyperparameter values can impact performance of VRCDR. The results are shown in Table 3. The Rips Complex is tuned using the edge-length ($\boldsymbol{\alpha}$) hyperparameter, which refers to the minimum length

CDR Tasks	Hyperparameters for Edge-Length				
	$\boldsymbol{\alpha} = 0.95$	$\boldsymbol{\alpha} = 1.0$	$\boldsymbol{\alpha} = 1.5$	$\boldsymbol{\alpha} = 2.0$	$\boldsymbol{\alpha} = 2.5$
Task 1	1.16701	**1.15907**	1.18561	1.16798	1.17047
Task 2	1.13025	**1.12145**	1.12541	1.13539	1.12725
Task 3	1.12216	**1.11583**	1.13364	1.12890	1.12299
Task 4	1.14267	**1.14155**	1.14822	1.14285	1.14670

Table 3: Hyperparameter analysis using RMSE

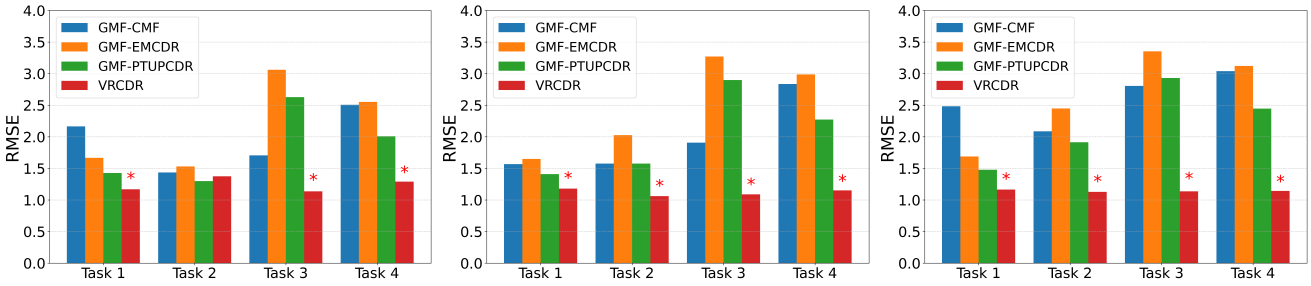
required between a pair of points in our point cloud to generate an edge [4]. Following the definition of the Rips Complex, lower values of the edge-length will isolate points as edges cannot be formed between them. This results in a lack of structure of the embedded data. Conversely, larger edge length values will trivially connect any two points with an edge. The best choice for edge-length is usually found somewhere between these two extremes, where noise is not a factor, and the algorithm can precisely understand the structure of the embedded data [4]. We choose 5 different values for $\boldsymbol{\alpha}$, which are 0.95, 1.0, 1.5, 2.0, and 2.5. We can see that VRCDR performs best when using 1.0 for $\boldsymbol{\alpha}$. This result aligns with the intuition of Rips Complex [6], where the edge length should not be too low or too high [4].

CDR Tasks	Variations of VRCDR			
	UA-CA	UR-CA	UA-CR	VRCDR
Task 1	1.16931	1.20075	1.17241	**1.16411**
Task 2	1.16103	1.15152	1.12123	**1.11407**
Task 3	1.13804	1.14055	1.12955	**1.12322**
Task 4	1.18105	1.16623	1.14723	**1.14448**

Table 4: Ablation study for VRCDR using RMSE

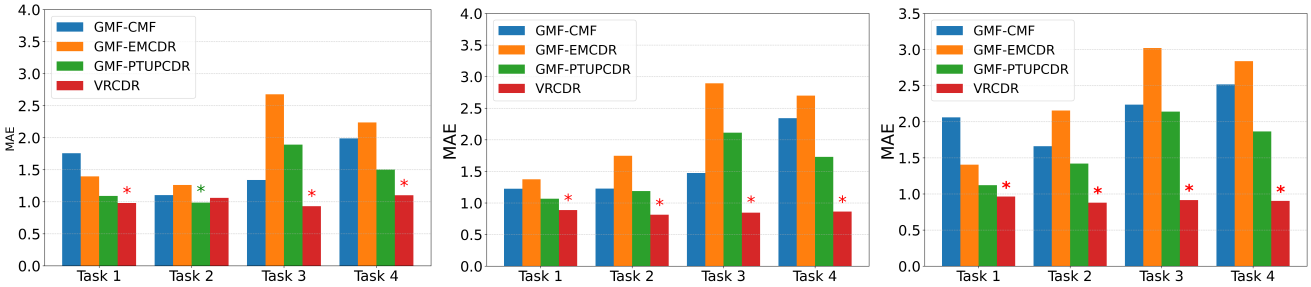
5.2.4 RQ4: Ablation Study To examine the effectiveness of the Rips Complex in VRCDR, we conduct an ablation study to compare VRCDR with three variants and report the results in table 4. The variations are:

- **UA-CA:** In this model, we remove the Rips Com-



(a) $\beta = 20\%$ (b) $\beta = 50\%$ (c) $\beta = 70\%$

Figure 4: RMSE comparison over 4 CDR tasks for extreme cold-start users. The * above the bar indicates a p-value less than 0.01 from the two-tailed paired t-test between VRCDR and the best baseline.



(a) $\beta = 20\%$ (b) $\beta = 50\%$ (c) $\beta = 70\%$

Figure 5: MAE comparison over 4 CDR tasks for extreme cold-start users. The * above the bar indicates a p-value less than 0.01 from the two-tailed paired t-test between VRCDR and the best baseline.

plex for computations for the user and item characteristic vectors. Instead, we replace the Rips Complex with a simple average.

- **UR-CA:** In this model, we keep the Rips Complex for the user characteristic vector, and use a simple average for the content characteristic vectors.
- **UA-CR:** In this model, an average is used for the user characteristic vector, and the Rips Complex is used for the content characteristic vectors.

The results in Table 4 show the three variants to perform worse compared to VRCDR. In UA-CA, adopting an average for both users and content information might oversimplify the underlying structure and information in the data. Although an average may be computationally more efficient, the user and item characteristics might not be adequately captured, which can explain the poor performance. UR-CA is able to capture the structure of the user's interaction patterns using the Rips Complex, but fails to capture the structure of the content information. It can be the case that the content information is complex to understand, and by using an average, the information is not accurately captured, leading to poor performance. In UA-CR, the content's structure is captured effectively, but an average on the user side might lead to information loss, resulting in inferior performance to VRCDR. Based on the results of the ablation study, we conclude that the use of Rips Complex is essential to achieve competitive results.

6 Related Work

Recently, cross-domain recommender systems have emerged as promising methods to alleviate the cold-start issue. By adopting transfer learning techniques, CDR approaches are able to leverage information from auxiliary domains to improve recommendations in a given target domain [15, 22, 14, 29]. The majority of the proposed CDR models fall into the bridge-based category, where they use mapping functions to map embeddings for users and items across domains [9, 11, 23, 24].

One of the pioneering efforts for CDR is known as CMF [18]. This method factorizes matrices for multiple domains simultaneously while maintaining a shared global user embedding matrix. Recent studies have built off of CMF, with CST [13] being one of the more effective methods, where it initializes target domain user embeddings using the learned user embeddings in the source domain. This framework introduced the idea of a bridge-based method, and more algorithms have been developed over the past few years to try and build off of the success that those methods have shown [25, 3, 8, 5].

EMCDR [11] has shown the most promise in terms

of improving recommendations for cold-start users. Numerous new algorithms have been released in the last five years, which build off of the same idea [9, 23, 24, 27]. Most of these methods develop a shared bridge function for all users, which can generalize well over large amounts of data. However, Zhu et al recentley proposed PTUPCDR [28], which builds off of EMCDR by providing personalized bridges to each user, making it the first studies to deviate from the shared bridge methods.

7 Conclusion

In this paper, we presented a novel CDR framework, called VRCDR, that can leverage geometry of users' interacted items from the source domain in the recommendation process to improve recommendations in the target domain. Extensive experiments showed the effectiveness of VRCDR compared to state-of-the-art bridge-based CDR methods in real-world situations. For future work, we plan to extend VRCDR to include context about users in the recommendation process, and utilize more techniques from computational geometry to better understand embedded data in the domain of CDR.

References

[1] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *KBS*, 2013.

[2] J. Cao, X. Lin, X. Cong, J. Ya, T. Liu, and B. Wang. Disencdr: Learning disentangled representations for cross-domain recommendation. In *SIGIR*, 2022.

[3] C. Gao, X. Chen, F. Feng, K. Zhao, X. He, Y. Li, and D. Jin. Cross-domain recommendation without sharing user-relevant data. In *WWW*, 2019.

[4] R. Ghrist. Barcodes: the persistent topology of data. *Am. Math. Soc*, 2008.

[5] L. Guo, L. Tang, T. Chen, L. Zhu, Q. Nguyen, and H. Yin. Da-gcn: A domain-aware attentive graph convolution network for shared-account cross-domain sequential recommendation. *arXiv*, 2021.

[6] J. Hausmann. On the vietoris-rips complexes and a cohomology theory for metric spaces. *Ann Math*, 1995.

[7] G. Hu, Y. Zhang, and Q. Yang. Conet: Collaborative cross networks for cross-domain recommendation. In *CIKM*, 2018.

[8] M. Jiang, P. Cui, X. Chen, F. Wang, W. Zhu, and S. Yang. Social recommendation with cross-domain transferable knowledge. *TKDE*, 2015.

[9] S. Kang, J. Hwang, D. Lee, and H. Yu. Semi-supervised learning for cross-domain recommendation to cold-start users. In *CIKM*, 2019.

[10] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 2009.

[11] T. Man, H. Shen, X. Jin, and X. Cheng. Cross-domain recommendation: An embedding and mapping approach. In *IJCAI*, 2017.

[12] J. Ni, J. Li, and J. McAuley. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *EMNLP-IJCNLP*, 2019.

[13] W. Pan, E. Xiang, N. Liu, and Q. Yang. Transfer learning in collaborative filtering for sparsity reduction. In *AAAI*, 2010.

[14] G. Pinto, Z. Wang, A. Roy, T. Hong, and A. Capozzoli. Transfer learning for smart buildings: A critical review of algorithms, applications, and future perspectives. *Adv. App*, 2022.

[15] A. Sahu. User and item spaces transfer from additional domains for cross-domain recommender systems. *Appl. Intell.*, 2023.

[16] V. Sanh, L. Debut, J. Chaumond, and T. Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv*, 2019.

[17] H. Shan and A. Banerjee. Generalized probabilistic matrix factorizations for collaborative filtering. In *ICDM*, 2010.

[18] A. Singh and G. Gordon. Relational learning via collective matrix factorization. In *KDD*, 2008.

[19] H. Su, Y. Zhang, X. Yang, H. Hua, S. Wang, and J. Li. Cross-domain recommendation via adversarial adaptation. In *CIKM*, 2022.

[20] E. Weisstein. Heron's formula. *MathWorld*, 2003.

[21] J. Yu, H. Yin, X. Xia, T. Chen, J. Li, and Z. Huang. Self-supervised learning for recommender systems: A survey. *TKDE*, 2023.

[22] T. Zang, Y. Zhu, H. Liu, R. Zhang, and J. Yu. A survey on cross-domain recommendation: taxonomies, methods, and future directions. *ACM TOIS*, 2022.

[23] F. Zhu, C. Chen, Y. Wang, G. Liu, and X. Zheng. Dtcdr: A framework for dual-target cross-domain recommendation. In *CIKM*, 2019.

[24] F. Zhu, Y. Wang, C. Chen, G. Liu, M. Orgun, and J. Wu. A deep framework for cross-domain and cross-system recommendations. *arXiv*, 2020.

[25] F. Zhu, Y. Wang, C. Chen, J. Zhou, L. Li, and G. Liu. Cross-domain recommendation: challenges, progress, and prospects. *arXiv*, 2021.

[26] F. Zhu, Y. Wang, J. Zhou, C. Chen, L. Li, and G. Liu. A unified framework for cross-domain and cross-system recommendations. *TKDE*, 2021.

[27] Y. Zhu, K. Ge, F. Zhuang, R. Xie, D. Xi, X. Zhang, L. Lin, and Q. He. Transfer-meta framework for cross-domain recommendation to cold-start users. In *SIGIR*, 2021.

[28] Y. Zhu, Z. Tang, Y. Liu, F. Zhuang, R. Xie, X. Zhang, L. Lin, and Q. He. Personalized transfer of user preferences for cross-domain recommendation. In *WSDM*, 2022.

[29] F. Zhuang, Z. Qi, K. Duan, D. Xi, Y.n Zhu, H. Zhu, H. Xiong, and Q. He. A comprehensive survey on transfer learning. *IEEE*, 2020.

[30] A. Zomorodian. Fast construction of the vietoris-rips complex. *Comput.Gr.*, 2010.