Geometric Collaborative Filtering with Convergence

Hisham Husain
Amazon International Machine Learning

Julien Monteil
Amazon International Machine Learning

Abstract

Latent variable collaborative filtering methods have been a standard approach to modelling user-click interactions due to their simplicity and effectiveness. However, there is limited work on analyzing the mathematical properties of these methods in particular on preventing the overfitting towards the identity, and such methods typically utilize loss functions that overlook the geometry between items. In this work, we introduce a notion of generalization gap in collaborative filtering and analyze this with respect to latent collaborative filtering models. We present a geometric upper bound that gives rise to loss functions, and a way to meaningfully utilize the geometry of the items to improve recommendations. We show how these losses can be minimized and gives the recipe to a new latent collaborative filtering algorithm, which we refer to as GeoCF, due to the geometric nature of our results. We then show experimentally that our proposed GeoCF algorithm can outperform other all existing methods on the Movielens20M and Netflix datasets, as well as two large-scale internal datasets. In summary, our work proposes a theoretically sound method which paves a way to better understand generalization of collaborative filtering at large.

1 Introduction

The importance of recommender systems cannot be overstated and play a pivotal role in many e-commerce and entertainment applications. The general problem of recommendation is to consume user-item interactions, and learn meaningful relationships to better recommend items that the user has not seen, but is likely to enjoy. A number of different algorithmic solutions, from neighbourhood methods, e.g. [19], linear regressions, e.g. [23], factorization models, e.g. [31], to sequential models, e.g. [42], and latent generative models, e.g. [32], have been proposed throughout the years to solve this practical problem. Significant effort has been dedicated to surpass the performance of state-of-the-art methods on public datasets, focusing on learning the best representation of the sparse binarized user-item interaction matrix. Less interest has been vested on leveraging item and user metadata information to boost the learning capabilities of the algorithms, with most existing approaches leveraging regularizers to ground the learning of the latent variables, e.g. [24], in particular in the context of cold-start recommendations, e.g. [40].

One particular top-performing method involves latent collaborative filtering, which seeks to reconstruct the user-item interaction with the use of a generative process. The assumption behind the success that generative models have in reconstructing the click history may be that the stochastic process helps in preventing the overfitting towards the identity, which is considered to be the principal cause behind the loss of predictive capability [34]. One popular approach in this family is the MultiVAE [22], and its subsequent variations [37, 18] which dominate the MovieLens20M and Netflix leaderboards, allow for flexible architectures, and are efficient. Despite the success, there are two main caveats. Firstly, the majority of these approaches are agnostic to user and item metadata and second, there is little theoretical understanding of these approaches from a generalization perspective. While we discuss these as two separate problems, they form the underpinning of a more compelling question we seek to answer:

> *Can we derive a latent collaborative filtering approach that achieves theoretical guarantees on generalization via item metadata?*

In order to answer this question, we first need to define a notion of generalization for latent collaborative

Proceedings of the 28[th] International Conference on Artificial Intelligence and Statistics (AISTATS) 2025, Mai Khao, Thailand. PMLR: Volume 258.

filtering. The closest concept to generalization has been commonly recollected as cold-start [39], and while the literature on the topic has become denser, the endeavours have been largely empirical, see e.g. [26, 43]. In Statistical Learning Theory (SLT) [4], the motivation behind generalization is to counter overfitting. Despite the prevalence of overfitting in collaborative filtering [34], there is surprisingly little work in the formalization of generalization. In this work, we borrow machinery from SLT and define, to the best of our knowledge, the first formalization of a generalization gap in latent collaborative filtering. We show that our definition aligns well and parallels existing intuition in latent model CF such as the trepidation of overfitting to the identity model.

We then present a geometrically dependent upper bound on the generalization gap for an arbitrary latent collaborative filtering method, that gives rise to loss functions. We show how these losses can be minimized and gives the recipe to a new latent collaborative filtering algorithm, which we call GeoCF, due to the geometric nature of our results.

In our geometrical upper bound, we find a convergence rate of $n^{-\frac{1}{d^*}}$ (n being the number of observed data points) where d^* is the *intrinsic* dimensionality of our user-item distribution induced by the item metadata. The quantity d^* depends directly on how well the user-item space is metrized by embeddings, thus emphasizing the importance representation learning plays for performance from a theoretical perspective. Finally, we show that GeoCF empirically outperforms existing baselines on popular baselines such as Movielens20M and Netflix datasets.

In summary, our contributions are three-fold:

- A formalization of generalization gap in latent collaborative filtering methods, which parallels and subsumes our intuition about existing methods.
- (GeoCF) A new latent collaborative filtering algorithm motivated directly from the geometric generalization bound that utilizes item metadata for recommendation.
- Convergence result for GeoCF, establishing the importance of metadata in improving generalization from a formal perspective.

2 Preliminaries on Collaborative Filtering

In this section, we will first introduce preliminaries for latent collaborative filtering models followed by a brief summary of the optimal transport tools to be used in building the proposed algorithm.

Collaborative Filtering We use Ω to denote a compact Polish space and denote Σ as the standard Borel σ-algebra on Ω, $\mathbb{R}$ denotes the real numbers and $\mathbb{N}$ natural numbers. Let $I \in \mathbb{N}$ denote the number of all items and we will consider $\Omega = \mathbb{R}^I$ to represent users where each $X \in \Omega$ corresponds to the click-vector of a specific user X. For simplicity, we will binarize this vector: for item i, user X clicked on it if $X^i = 1$ and $X^i = 0$ otherwise. We use $\Delta(\Omega)$ to denote the set of all probability measures over Ω.

We observe a set of item-user click history consisting of n users, which we denote by $\mathsf{X} = \{X_i\}_{i=1}^n$ where each $X_i \in \Omega$. The goal of collaborative filtering is to recommend items to users by first identifying similar users based on their history. For example, if the history of two users X_i and X_j differ only by one click, then it is likely that item is of interest to both users. Thus, we would like to produce a function F that takes in a user $X \in \Omega$ and outputs a item-click vector $\tilde{X} \in \Omega$ that the user would most likely be interested in. Thus, we can formulate this as a least regression problem:

$$\min_{F:F\neq \mathrm{Id}} \left\| \mathsf{X} - \mathsf{X}F^\top \right\|, \tag{1}$$

where we treat $\mathsf{X} \in \mathbb{R}^{n\times I}, F \in \mathbb{R}^{I\times I}$. We want to avoid the identity solution, and this is often enforced in the form of a regularizer, e.g. [33]. Various developments in this line of work study how to decompose F to incorporate such intuition. A straightforward way to perform collaborative filtering is via neighbourhood methods, which consist of computing similarities across items or users, where the similarity is evaluated as a dot product on click vectors, and possibly item and user metadata [14]. The Netflix competition and subsequent research works on available public datasets have shown that SLIM [23] and matrix factorization techniques tend to outperform neighbourhood methods [20, 35]. A push for neural approximations of matrix factorization took place [13, 7], with the conclusion that the proposed MLP architectures fail to learn a better nonlinear variant of the dot product and are outperformed by careful implementations of matrix factorization [9, 30, 31]. The success of Deep Cross Networks on learning to rank tasks [28] also showed the importance of explicitly encoding the dot product in the network architecture. The framework of Collaborative Metric Learning (CML) [1, 2, 3] consists on learning two linear projection layers to project the user and item onto the same Euclidean metric space, and then to minimize a pairwise empirical risk to preserve the consistency of user preferences written as a triplet loss. The authors then derive a generalization error upper bound which borrows and adapts machinery from the Rademacher complexity [4], and which relies on this specific set-up with the projection layers and

triplet loss. CML exhibits strong performance on small and medium scale datasets for the first few ranking positions, however it comes with a high computational cost, and cannot readily incorporate side information. The variational autoencoder approach [22] (MultiVAE) is amongst the most competitive neural approaches but the least squares approximation of SLIM, i.e., a linear autoencoder with projections to ensure the zero diagonal constraint (EASE) [33, 35], still beat it on 2 out 3 public datasets. It was shown in [18] that augmenting the MultiVAE approach with flexible priors and gating mechanisms led to state of the art performance. The MultiVAE approach was also refined and ensembled with a neural EASE to beat baselines on MovieLens20M and Netflix [37]. Finally, a recent approach related to the proposed framework in this paper is SinkhornCF [21], which replaces the reconstruction error by the Sinkhorn distance, a loss that will help guide the model to recommend items based on their metadata. We present the preliminaries of the Sinkhorn loss and optimal transport in the next section.

Optimal Transport and Divergences When it comes to comparing two probability distributions $P, Q \in \Delta(\Omega)$, a very well regarded and canonical choice is the Wasserstein distance [38], which is defined using a ground cost $c : \mathcal{X} \times \mathcal{X} \to \mathbb{R}_+$

$$\mathscr{W}_c(P,Q) := \inf_{\pi \in \Pi(P,Q)} \left\{ \int_{\mathcal{X}\times\mathcal{X}} c(x,y) d\pi(x,y) \right\}, \quad (2)$$

where $\Pi(P,Q)$ is the *set of couplings* between P and Q

$$\Pi(P,Q) = \left\{ \pi \in \mathscr{P}(\mathcal{X}\times\mathcal{X}) : \int_{\mathcal{X}} \pi(x,y)dx = P, \int_{\mathcal{X}} \pi(x,y)dy = Q \right\}. \quad (3)$$

Finding the exact value of the Wasserstein distance is expensive and one often resorts to regularizing the objective with the entropy of π, yielding the following objective:

$$S^\varepsilon(P,Q) = \inf_{\pi\in\Pi(P,Q)} \left\{ \inf_{\mathcal{X}\times\mathcal{X}} c(x,y)d\pi(x,y) + \varepsilon \int_{\mathcal{X}\times\mathcal{X}} \log \pi(x,y) d\pi(x,y) \right\}. \quad (4)$$

This objective can be solved in a more stable and tractable manner, referred to as the Sinkhorn algorithm [8]. Since S^ε serves as a viable approximation to $\mathscr{W}_c$, it has been popularly used in various machine learning applications. Another choice of divergence we will utilize is the Maximum Mean Discrepancy (MMD) with respect to a kernel k $d_{\mathrm{MMD}}(P,Q)$ which is typically used for its computational convenience.

3 Generalization for Collaborative Filtering

In this section, we present to the best of our knowledge, the first formal pursuit of a generalization study for latent collaborative filtering. We begin by introducing a notion of generalization error, akin to the generalization gap existing in supervised learning [4]. Letting 2^Ω refer to the power set of Ω, we represent a collaborative filtering algorithm as a mapping $C : \Omega \times 2^\Omega \to \Delta(\Omega)$ so that $C(\mathsf{X}, \mathcal{D})$ represents the distribution of possible recommendations for user X upon training with data $\mathcal{D}$.

Definition 3.1 (Collaborative Filtering Generalization Error)**.** Let $P \in \Delta(\Omega)$, let $C : \Omega \times 2^\Omega \to \Delta(\Omega)$ denote a collaborative filtering algorithm, then the error of C under n samples is defined as

$$\mathcal{E}_{n,P}(C) := \mathbb{E}_{X^n \sim P^n} \left[d_{\mathrm{TV}} \left(\int_{\mathcal{X}} C(\mathsf{X}, X^n) dP(\mathsf{X}), P \right) \right], \quad (5)$$

where $d_{\mathrm{TV}}(\mu,\nu) = \sup_{A\in\Sigma} (\mu(A) - \nu(A))$, is the total-variation distance between distributions $\mu, \nu \in \Delta(\Omega)$.

We are interested in analyzing how well the collaborative filtering algorithm C consumes a dataset X^n of size n from P to reconstruct the user distribution (and their preferences). Thus, we analyze the discrepancy using the total variation metric, and take the expectation across all datasets of size n. Note that the quantity $\mathcal{E}_{n,P} \in [0,1]$ and thus we want to make sure it decreases to 0 at a fast rate as $n \to \infty$. We now show that the identity solution characterized as $C_{\mathrm{ID}}(\mathsf{X}, \mathcal{D}) = \frac{1}{|\mathcal{D}|} \sum_{\mathsf{X}'\in\mathcal{D}} \delta_{\mathsf{X}'}(\mathsf{X})$ that overfits to the data achieves 1, which is the maximal possible value.

Proposition 3.2. *Let $P \in \Delta(\Omega)$ then $\mathcal{E}_{n,P}(C_{\mathrm{ID}}) = 1$ for any $n < \infty$.*

Proof. Note that the distribution C_{ID} is not absolutely continuous with respect to P and thus the Total Variation Distance will be 1 with probability 1. □

While this is not a surprise and anticipated, as discussed in [34], we would like to emphasize this is the first negative result connecting to a formal notion of generalization. In the proof, we can see that the negative result stems from the fact that the model is unable to recommend for unseen users and so we want the resulting distribution $C(\mathsf{X}, \mathcal{D})$ to exemplify some smoothness. We now analyze the behaviour of $\mathcal{E}_{n,P}$ for latent collaborative filtering algorithms.

In such models, we have a latent space $\mathcal{Z}$ with a prior distribution $\gamma(z) \in \Delta(\mathcal{Z})$. We use ϑ and ξ to denote

the parameters of a decoder and encoder distribution $p_\vartheta(\mathsf{X} \mid z)$ and $q_\xi(z \mid \mathsf{X})$ respectively. We will also assume that we have a deterministic decoder so that $p_\vartheta(X \mid z) = \delta_{D_\vartheta(z)}(X)$ where $D_\vartheta : \mathcal{Z} \to \mathcal{X}$. During training time, we minimize a loss $\mathsf{L}(\vartheta, \xi)$ and the resulting collaborative filtering algorithm for recommendation on a user X is

$$C_{\vartheta,\xi}(\mathsf{X}, \mathcal{D}) = \int_{\mathcal{Z}} p_\vartheta(\cdot \mid z) q_\xi(z \mid \mathsf{X}) dz. \tag{6}$$

The above describes the reconstruction process given by the encoder and decoder, which we expect will be much smoother than in the case of the dirac delta, as seen in the identity case. Our goal is to provide an analysis of $\mathcal{E}_{n,P}(C_{\vartheta,\xi})$ and we do so by exploiting the geometry of the space in Ω. To this end, assume we have a cost $c : \Omega \times \Omega \to \mathbb{R}$ that metrizes Ω, which allows us to define the following machinery from [41].

Definition 3.3 (c-Covering Numbers)**.** For a set $S \subseteq \Omega$, we denote $N_\eta(S)$ to be the *η-covering number* of S, which is the smallest $m \in \mathbb{N}_*$ such that there exists closed balls $B_1, \ldots, B_m$ of radius η with $S \subseteq \bigcup_{i=1}^m B_i$. For any $P \in \Delta(\Omega)$, the (η, τ)-*dimension* is $d_\eta(P, \tau) := \frac{\log N_\eta(P,\tau)}{-\log \eta}$, where $N_\eta(P, \tau) := \inf\{N_\eta(S) : P(S) \geq 1 - \tau\}$.

Definition 3.4 (1-Upper c-Wasserstein Dimension)**.** The 1-*Upper Wasserstein dimension* of any $P \in \Delta(\Omega)$ is $d_c^*(P) := \inf\left\{s \in (2, \infty) : \limsup_{\eta \to 0} d_\eta(P, \eta^{\frac{s}{s-2}}) \leq s\right\}$.

The quantity $d_c^*(P)$ represents the *intrinsic* dimension of where P is largely supported. For example, if P lies on a low dimensional manifold in some high dimensional ambient space, then $d_c^*(P)$ would correspond to the dimension of said manifold. We now present an upper bound on $\mathcal{E}_{n,P}(C_{\vartheta,\xi})$. For that we introduce the following terms $q_{\xi^*}\#\hat{P}^n(z) = \int_{\mathcal{X}} q_{\xi^*}(z \mid \mathsf{X}) d\hat{P}^n(\mathsf{X})$, $p_{\vartheta^*}\#\gamma = \int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z)\gamma(z)$, with $\hat{P}^n$ denoting n samples from P, and the GeoCF loss, $\mathsf{L}(\vartheta, \xi) := \int_{\mathcal{Z}} d_I(X, D_\vartheta(z)) dq_\xi(z \mid X) d\hat{P}^n(X) + \lambda \cdot d_{\text{MMD}}\left(q_\xi \# \hat{P}^n, \gamma\right)$, which we will show how to derive in the remainder of the paper, see Section 4 leading up to equation (10).

Theorem 3.5. *Let $\mathcal{F}_c$ denote the set of 1-Lipschitz functions with respect to c and $\mathcal{H}_k$ denote the unit ball of Reproducing Kernel Hilbert Space functions with kernel k. For any $P \in \Delta(\Omega)$ and $\lambda > 0$, consider the minimizers $\vartheta^*, \xi^* = \arg\inf_{\vartheta,\xi} \mathsf{L}(\vartheta, \xi)$. It then holds that with high probability*

$$\begin{aligned}\mathcal{E}_{n,P}(C_{\vartheta^*,\xi^*}) \leq & \delta(\vartheta^*, \xi^*; P) + \frac{1}{4}\mathbb{E}\Bigg[\mathscr{W}_c\left(\hat{P}^n, p_{\vartheta^*}\#\gamma\right) \\ & + d_{\text{MMD}}\left(q_{\xi^*}\#\hat{P}^n, \gamma\right)\Bigg] + A n^{-\frac{1}{d_c^*(P)}},\end{aligned} \tag{7}$$

for some $A > 0$ where $q_{\xi^*}\#\hat{P}^n(z) = \int_{\mathcal{X}} q_{\xi^*}(z \mid \mathsf{X}) d\hat{P}^n(\mathsf{X})$, $p_{\vartheta^*}\#\gamma = \int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z)\gamma(z)$ and $\delta(\vartheta^*, \xi^*; P) = 2\inf_{h \in \mathcal{F}_c} \left\|\log \frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty + 2\inf_{h \in \mathcal{H}_k} \left\|\log \frac{\gamma}{q_{\xi^*}\#P} - h\right\|_\infty$.

Proof. (Sketch, full proof in Appendix) We apply a triangle inequality to the Total Variation (TV) distance in $\mathcal{E}_{n,P}$ over the latent space. We then utilize two applications of Pinskers inequality to relate the TV distances to the Kullback-Leibler divergence and Integral Probabiltiy Metrics (IPMs), whose dual forms correspond to the Wasserstein distance and the MMD. Applying concentration inequalities on these IPMs allows us to get fast rates of convergence for MMD and a dimensionality dependence when applied to the Wasserstein distance, resulting in the final inequality. □

The proof of this Theorem can be found in the Appendix. The above result tells us that the generalization gap is bounded by three terms. The first term is independent of n and depends on how well P and γ can be approximated by the encoder and decoder distributions. In particular, noting that $\mathcal{H}_k$ are universal function approximators, these terms can be kept arbitrarily small. The second term is the sum of two expected gaps: the model p_{ϑ^*} and $\hat{P}^n$, along with the gap between the prior distribution γ and the encoder q_{ξ^*}.

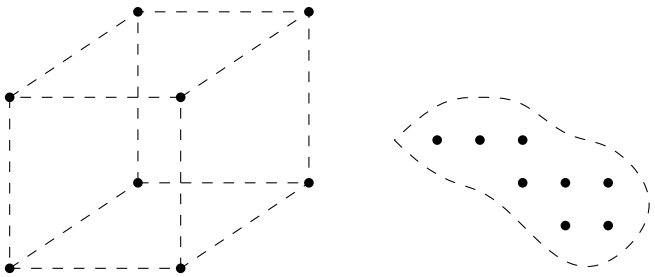

Figure 1: An illustration of the distribution P embedded in the space Ω for two different choices of c_I. The left corresponds to when users are more spread out whereas the right corresponds to a more meaningful representation so that the distribution P is supported on a lower dimensional manifold.

The third term is the rate of convergence, which depends on the complexity of our distribution P and choice of metric c_I. To better understand this, observe that $d_c^*(P)$ is a measure of how well the support of P can be covered by η-radius balls, where this radius is measured by $d(u, u') = \mathscr{W}_{c_I}(p_u, p_{u'})$. Although it is practically hard to exactly compute this intrinsic

dimension $d_c^*(P)$, see e.g. the recent work of [5], we can work with the knowledge that this intrinsic dimension is reduced if we find a way to separate users so that the natural distribution P is compactly preserved, and the semantic embeddings of item metadata present a reasonable way of achieving so. We illustrate in Figure 1 how the user space Ω can change drastically depending on the metric c_I used.

We will now derive an algorithm in the next section inspired by the upper bound.

4 GeoCF: utilizing geometry to close the generalization gap

We saw in the previous section the generalization gap of collaborative filtering can be lowered by controlling parameters ϑ, ξ and metric c of the upper bound:

$$\underbrace{\mathbb{E}\left[\mathscr{W}_c\left(\hat{P}^n, p_\vartheta \# \gamma\right)\right]}_{\text{Decoder}} + \underbrace{\mathbb{E}\left[d_{\text{MMD}}\left(q_\xi \# \hat{P}^n, \gamma\right)\right]}_{\text{Encoder}} + \underbrace{A n^{-\frac{1}{d_c^*(P)}}}_{\text{Metric}}. \tag{8}$$

Schematically, we can derive an algorithm by first picking a choice of c that helps decompose the user distribution P into a more compact representation and secondly, minimizing the two terms with respect to ϑ and ξ.

Metrizing Ω with c Based on the theoretical result, we need to metrize Ω so that d_c^* is reduced. Intuitively, this means we must find a way to separate users so that the natural distribution P is compactly preserved, and for this task we rely on item metadata. By item metadata, here we mean any movie metadata, such as movie title, movie description, list of actors, directors, etc. Suppose we have an embedding function $E : \{1, \dots, I\} \to \mathbb{R}^k$ that embeds each of the I items into a representation into $\mathbb{R}^k$. In the presented experiments of Section 5, we will simply use the Sentence Transformer embeddings [29] of movie titles. By considering a user $u \in \Omega$ as point-cloud distribution consisting of items they clicked on: $p_u = \frac{1}{|I_u|}\sum_{i \in I_u} \delta_i$ where I_u is the set of items clicked on by user u, we can define a distance between users: $d_I(u, u') = \mathscr{W}_{c_I}(p_u, p_{u'})$ where $c_I(i, i') = \|E(i) - E(i')\|_2$ is used as the ground cost between items. The distance d_I compares users by considering the semantic similarity of their respectively engaged items, and since our hypothesis is that P naturally clusters users with shared interests together, it is natural that $d_{d_I}^*(P)$ results in a smaller quantity. In the extreme opposite case where there are no embeddings or 01-loss is used, for example in MultiVAE (see [21][Section 2.3]), then the dimensionality of the support of P will be higher which in turn leads to slow convergence. Note that we just presented a convenient way to metrize Ω, and more sophisticated metric learning approaches may be considered as well.

Learning ξ For the encoder parameters ξ, the most straight forward way is to minimize the term $d_{\text{MMD}}(q_\xi \# \hat{P}^n, \gamma)$ which can indeed be computed in closed form with $O(m^2)$ sample complexity if we take m samples from γ and $q_\xi \# \hat{P}^n$.

Learning ϑ Similar to the encoder, we can minimize decoder parameters ϑ with respect to $\mathscr{W}_{d_I}\left(\hat{P}^n, p_\vartheta \# \gamma\right)$. In order to compute this quantity, one may break it down using certain advances in optimal transport.

Theorem 4.1 ([36])**.** *Let $P \in \Delta(\Omega)$ and for some $n \in \mathbb{N}$ and $\hat{P}^n$ correspond to the empirical distribution of P from n i.i.d samples. Let $\mathcal{Q} = \left\{\pi : \int_\Omega \pi(z \mid X) d\hat{P}^n(X) = \gamma(z)\right\}$, we then have that*

$$\mathscr{W}_{d_I}\left(\hat{P}^n, p_\vartheta \# \gamma\right) = \inf_{\pi \in \mathcal{Q}} \int_\Omega \int_{\mathcal{Z}} d_I(X, D_\vartheta(z)) d\pi(z \mid X) d\hat{P}^n(X). \tag{9}$$

The main contribution of this Theorem is that instead of finding an optimal coupling, as is standard in Wasserstein distances, we can search for this conditional distribution π instead by searching within $\mathcal{Q}$. Notice however that π is precisely an encoder distribution mapping from $\mathcal{X}$ to a distribution over $\mathcal{Z}$ and the constraint of $\mathcal{Q}$ amounts to ensuring $\pi \# \hat{P}^n = \gamma$, which can be enforced by minimizing $d_{\text{MMD}}(\pi \# \hat{P}^n, \gamma)$, which is precisely the same objective we derived for the encoder q_ξ. Hence we can replace the minimization over π with q_ξ, leading to the overall loss function

$$\mathsf{L}(\vartheta, \xi) := \underbrace{\int_{\mathcal{Z}} d_I(X, D_\vartheta(z)) dq_\xi(z \mid X) d\hat{P}^n(X)}_{\text{Reconstruction error}} + \lambda \cdot \underbrace{d_{\text{MMD}}\left(q_\xi \# \hat{P}^n, \gamma\right)}_{\text{Prior regularization}}, \tag{10}$$

where λ can be interpreted as a Lagrangian multiplier for the constraint $q_\xi \# \hat{P}^n = \gamma$. The first term can be approximated with Monte Carlo sampling and resembles a reconstruction error. In order to approximate d_I, we utilize the Sinkhorn distance with cost matrix c_I which has demonstrated to be an efficient differentiable proxy to Wasserstein distances. The MMD term can also easily be computed, thus leading to a computationally viable algorithm.

5 Experiments

In this section, we compare the proposed methods to validate our theoretical findings related to the benefits provided by the geometric formulation. We will focus our experiments on four different data sets: Movielens20M, Netflix and two internal datasets from an internal large-scale streaming service.

5.1 Experimental setup

Metrics and statistical significance In order to evaluate the methods, we will utilize Recall@K and nDCG@K defined as follows. Recall@K = $\sum_{k=1}^{K} \frac{[\![r(k)\in I_1]\!]}{\min(K,|I_1|)}$ and DCG@K = $\sum_{k=1}^{K} \frac{2^{[\![r(k)\in I_1]\!]}-1}{\log(k+1)}$ where $[\![A]\!]$ is the Iverson bracket for an event A (denoting 1 if A is true and 0 otherwise), $r(k)$ indicates the item at rank k and I_1 denotes the positive items set for the user. We then average across all users for both recall and nDCG. Note that the used definition of Recall@K, which is used in numerous works [22, 34, 9, 21], is sometimes called capped Recall@K, as it interpolates between Precision@K and Recall@K, to deal with the case when the number of clicks is higher than the value of K (in which case Recall@K upper bound is not 1). Recall indicates if the first predicted elements match with ground truth whereas DCG measures how well the predicted ranking matches the ground truth ranking. nDCG@K is obtained by normalizing DCG by dividing by the maximum possible value. We also perform bootstrapping analysis to assess the significance of our results by sampling 20% of the users in each test dataset and repeating the process 1000 times, in order to compute 95% confidence intervals, and we indicate when our results achieve statistical significance.

Baselines We will be considering the following competitive baselines.

- **ItemKNNCF** [9] A classic item-based nearest-neighbor method that will utilize Cosine Similarity to find the closest items. We will pick neighborhood size from $[5, 1000]$.
- **P3alpha** [6]. P3alpha is a graph-based method that performs random walks between users and items based on the observed interaction matrix.
- **RP3beta** [27] A variant of P3alpha where the similarity distance between two items is divided by the popularity and raised to the power of an additional parameter β.
- **SLIM** [23] SLIM is a linear latent factor model which solves a constrained linear optimization problem and finds an asymmetric item-similarity matrix via sparsity and ℓ_1, ℓ_2 regularization.
- **WMF** [15] Weighted Matrix Factorization (WMF) is a generalized matrix factorization by adopting a Gaussian observation sampling distribution with a item-dependent diagonal variance matrix, where unobserved items are set to 1.
- **NeuMF** [10, 13] A generalization of matrix factorization method that learns a non-linear interaction between the item and user interactions. As suggested in [13], we adopt a 3-layer MLP along with the number of predictive factors selected from $\{8, 16, 32, 64\}$.
- **MultiVAE** [22] MultiVAE is a successful deep generative model inspired by Variational Inference, which derives a latent model for collaborative filtering. The autoencoder architecture for MultiVAE uses 1 hidden layer of dimensionality 600 and has the following end-to-end structure: $\mathbb{R}^{|I|} \to \mathbb{R}^{600} \to \mathbb{R}^{200} \to \mathbb{R}^{600} \to \mathbb{R}^{|I|}$.
- **CML** [1] We consider the scalable Sampling-Free Collaborative Metric learning approach (SFCML), which proposes an efficient alternative to CML without the needs for negative sampling, while achieving state-of-the-art performance on a number of relatively small datasets, e.g. CityULike, Steam 200k, MovieLens100k.
- **LightGCN** [12] LightGCN is the best-in-class graph convolutional network for recommendation. It learns user and item embeddings by linearly propagating them on the user-item interaction graph, and uses the dot product of the weighted sum of the embeddings learned at all layers to produce final scores.
- **SinkhornCF** [21] This is a method that proposes using a Sinkhorn loss in addition to MultiVAE with the same motivation as our approach. This work uses the same architecture as MultiVAE with $\varepsilon = 1.0$ set as the regularization parameter for the Sinkhorn algorithm, batch sizes of 500 are used with 100 epochs. This method also relies on a cost matrix to be prespecified similar to our approach and estimates this via the cosine value.

GeoCF architecture For GeoCF, we adopt the same architecture as MultiVAE and SinkhornCF. We use $\mathbb{R}^{|I|} \to \mathbb{R}^{600} \to \mathbb{R}^{200} \to \mathbb{R}^{600} \to \mathbb{R}^{|I|}$. We use a batch size of 500 and 100 epochs, as for SinkhornCF. For the Maximum Mean Discrepancy (MMD) term, we use Gaussian RBF kernel and select the kernel bandwidth parameter within the range $\{0.05, 0.1, 0.5, 1.0\}$. We use an exponential scheduler over the epochs, decreasing the Lagrangian multiplier λ and use $\varepsilon = 1.0$ as the regularization parameter for the Sinkhorn algorithm.

Table 1: Recall & nDCG for MovieLens20M. Best result is made bold, and the runner-up is underlined. * indicates statistical significance.

Model	Recall				nDCG			
	@20	@50	@75	@100	@20	@50	@75	@100
ItemKNNCF	0.311	0.435	0.504	0.555	0.266	0.306	0.329	0.346
P3alpha	0.291	0.397	0.446	0.475	0.244	0.280	0.298	0.308
RP3beta	0.340	0.463	0.528	0.571	0.290	0.331	0.353	0.368
SLIM	0.375	0.498	0.566	0.615	0.328	0.367	0.391	0.407
WMF	0.361	0.493	0.561	0.611	0.307	0.351	0.376	0.392
NeuMF	0.161	0.321	0.419	0.494	0.110	0.168	0.195	0.221
SFCML	0.363	0.451	0.520	0.549	0.323	0.354	0.372	0.382
LightGCN	0.259	0.412	0.488	0.544	0.231	0.274	0.307	0.326
MultiVAE	0.400	0.538	0.610	0.661	0.339	0.387	0.412	0.429
SinkhornCF	0.405	0.542	0.613	0.665	0.349	0.395	0.420	0.438
GeoCF	**0.411***	**0.550***	**0.622***	**0.673***	**0.353***	**0.400***	**0.423***	**0.443***
Improvement (%)	1.48	1.47	1.46	1.20	1.14	1.26	0.71	1.14

Table 2: Recall & nDCG for Netflix. Best result is made bold, and the runner-up is underlined. * indicates statistical significance.

Model	Recall				nDCG			
	@20	@50	@75	@100	@20	@50	@75	@100
ItemKNNCF	0.271	0.348	0.406	0.454	0.251	0.269	0.288	0.305
P3alpha	0.259	0.298	0.321	0.337	0.242	0.250	0.259	0.266
RP3beta	0.307	0.378	0.426	0.463	0.288	0.302	0.319	0.334
SLIM	0.347	0.426	0.486	0.534	0.325	0.341	0.361	0.379
WMF	0.314	0.393	0.450	0.496	0.293	0.310	0.330	0.347
NeuMF	0.107	0.213	0.299	0.373	0.089	0.130	0.162	0.190
SFCML	0.320	0.399	0.452	0.486	0.315	0.334	0.349	0.359
LightGCN	0.223	0.368	0.446	0.505	0.218	0.262	0.291	0.312
MultiVAE	0.350	0.441	0.504	0.555	0.321	0.344	0.366	0.385
SinkhornCF	0.356	0.445	0.507	0.558	0.327	0.349	0.371	0.389
GeoCF	**0.360***	**0.450***	**0.511***	**0.561**	**0.330***	**0.351***	**0.374***	**0.391***
Improvement (%)	1.12	1.12	0.79	0.54	0.53	0.57	0.81	0.53

5.2 Public Datasets

Table 1 and Table 2 present the performance of different models in terms of nDCG and Recall on Movielens20M[1] and Netflix[2], respectively. Movielens20M consists of 136,677 users and 20,108 movies, while Netflix contains 480,189 different users and 17,770 items, where each record contains a user, item pair followed by a score rating the user has for the given item. We follow the same process as in [21], keeping users that have watched at least five movies and threshold items with at least four ratings as a positive interaction. We split the user data into a training, validation and test set. For every user in the training set, we utilize all interaction history however for users in the validation or test set, a fraction of the history (80%) is used to predict the remaining interaction.

We highlight the best results in bold and underline the runner-ups. We can see that GeoCF achieves the best performance across the datasets with SinkhornCF consistently following as the runner-up. We believe these results testify to the importance of using geometric losses given such consistency. In particular, we note that the benefits gained by GeoCF are slightly higher on MovieLens20M compared to Netflix.

[1] https://grouplens.org/datasets/movielens/20m/
[2] https://www.kaggle.com/netflix-inc/netflix-prize-data

Table 3: nDCG results on two internal datasets. Best result is made bold, and the runner-up is underlined. * indicates statistical significance. Left: *Storefront* service. Right: *NextUp* service.

Model	nDCG			
	@10	@20	@50	@100
ItemKNNCF	0.521	0.573	0.599	0.611
P3alpha	0.497	0.517	0.538	0.576
RP3beta	0.500	0.522	0.535	0.541
SLIM	0.543	0.563	0.581	0.588
WMF	0.501	0.517	0.525	0.530
NeuMF	0.221	0.256	0.277	0.190
SFCML	0.481	0.506	0.521	0.528
LightGCN	0.473	0.502	0.523	0.531
MultiVAE	0.522	0.562	0.580	0.590
SinkhornCF	<u>0.602</u>	<u>0.613</u>	<u>0.626</u>	<u>0.628</u>
GeoCF	**0.613***	**0.631***	**0.632***	**0.632***
+ (%)	1.83	2.94	0.96	0.64

Model	nDCG			
	@10	@20	@50	@100
ItemKNNCF	0.502	0.538	0.576	0.605
P3alpha	0.484	0.500	0.518	0.532
RP3beta	0.488	0.502	0.519	0.534
SLIM	0.525	0.541	0.561	0.579
WMF	0.493	0.510	0.530	0.547
NeuMF	0.178	0.260	0.324	0.380
SFCML	0.445	0.462	0.477	0.488
LightGCN	0.471	0.502	0.529	0.548
MultiVAE	0.417	0.422	0.443	0.447
SinkhornCF	<u>0.558</u>	<u>0.587</u>	<u>0.602</u>	<u>0.611</u>
GeoCF	**0.608***	**0.628***	**0.646***	**0.649***
+ (%)	8.87	6.86	7.21	6.17

5.3 Internal Datasets

We now apply GeoCF to two internal datasets, which come from a streaming service exposed to a number of customers that go beyond the scale of the public datasets considered above. The first internal dataset involves all user-item interactions on the *Storefront* of the service. The second is related to the *NextUp* auto-play service, which triggers next title in auto-play mode upon completion of the current title. For both datasets, we collect 9 weeks of streaming data, and use the following week for testing. The last week of the 9 weeks training period is used for validation. We filter customers with at least 1 click over the train dataset.

We present the results left and right of Table 3 respectively. Similarly to MovieLens-20M and Netflix datasets, we find that GeoCF and SinkhornCF appear dominant quantitatively. In particular, we see larger boosts on the *NextUp* service, which goes with the intuition that there is a stronger semantic bias in auto-play recommendations. The lifts obtained with GeoCF are also significantly larger with respect to the Movielens20M and Netflix data splits, which we attribute to the filtering logic in preparing the data splits: in the public datasets, users with at least 4 clicks were selected, whereas in the internal datasets, we selected all users with at least 1 click. This underlines the generalization capability of our GeoCF algorithm which can be applied to any type of user click distribution, making it more robust to the filter on the number of clicks used in the dataset.

Infrastructure and training times Our approach and all the baselines were evaluated on a p3.2xlarge AWS EC2 instance. In terms of computational cost, we observed the training times to be approximately x2 for SinkhornCF with respect to MultiVAE, and x1.2 for GeoCF with respect to SinkhornCF: for example, for MovieLens20M, it took 450 seconds for MultiVAE, 984 seconds for Sinkhorn CF, and 1146 seconds for GeoCF.

6 Conclusion

We propose a notion of generalization gap in collaborative filtering and analyze this with respect to latent collaborative filtering models. We present a geometric upper bound that gives rise to a loss function, and a way to meaningfully utilize the geometry of item-metadata to improve recommendations. The resulting algorithm has deep connections to similiar approaches in generative modelling such as the Wasserstein Autoencoder and SinkhornCF, both of which advocate for the use of geometry. The proposed approach achieves an improvement on both public and internal datasets, establishing the practical significance of our approach and validating the theoretical findings. The benefits of our method can be further extended with a more deliberate approach to learning an item-similarity matrix in the form of a bi-level optimization. We leave such pursuits as the subject of future work.

Acknowledgments

The authors would like to thank Conor Hassan and Vu Nguyen for proofreading the manuscript. The authors would also like to thank Xurong Liang for his help during the rebuttal.

References

[1] Shilong Bao, Qianqian Xu, Zhiyong Yang, Xiaochun Cao, and Qingming Huang. Rethinking collaborative metric learning: Toward an efficient alternative without negative sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(1):1017–1035, 2022.

[2] Shilong Bao, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. The minority matters: A diversity-promoting collaborative metric learning algorithm. *Advances in Neural Information Processing Systems*, 35:2451–2464, 2022.

[3] Shilong Bao, Qianqian Xu, Zhiyong Yang, Yuan He, Xiaochun Cao, and Qingming Huang. Improved diversity-promoting collaborative metric learning for recommendation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2024.

[4] Peter L Bartlett and Shahar Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research*, 3(Nov):463–482, 2002.

[5] Adam Block, Zeyu Jia, Yury Polyanskiy, and Alexander Rakhlin. Intrinsic dimension estimation using wasserstein distance. *Journal of Machine Learning Research*, 23(313):1–37, 2022.

[6] Colin Cooper, Sang Hyuk Lee, Tomasz Radzik, and Yiannis Siantos. Random walks in recommender systems: exact computation and simulations. In *Proceedings of the 23rd international conference on world wide web*, pages 811–816, 2014.

[7] Paul Covington, Jay Adams, and Emre Sargin. Deep neural networks for youtube recommendations. In *Proceedings of the 10th ACM conference on recommender systems*, pages 191–198, 2016.

[8] Marco Cuturi. Sinkhorn distances: Lightspeed computation of optimal transport. *Advances in neural information processing systems*, 26, 2013.

[9] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. Are we really making much progress? a worrying analysis of recent neural recommendation approaches. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 101–109, 2019.

[10] Gintare Karolina Dziugaite and Daniel M Roy. Neural network matrix factorization. *arXiv preprint arXiv:1511.06443*, 2015.

[11] Arthur Gretton, Karsten M Borgwardt, Malte J Rasch, Bernhard Schölkopf, and Alexander Smola. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

[12] Xiangnan He, Kuan Deng, Xiang Wang, Yan Li, Yongdong Zhang, and Meng Wang. Lightgcn: Simplifying and powering graph convolution network for recommendation. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 639–648, 2020.

[13] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182, 2017.

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE International Conference on Data Mining*, pages 263–272, 2008.

[15] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *2008 Eighth IEEE international conference on data mining*, pages 263–272. Ieee, 2008.

[16] Hisham Husain, Richard Nock, and Robert C Williamson. A primal-dual link between gans and autoencoders. *Advances in Neural Information Processing Systems*, 32, 2019.

[17] G Kerkyacharian and D Picard. Density estimation in besov spaces zyxwvutsrqponmlkjihgfedcbazyxwvut. *Statistics & probability letters*, 13:15–24, 1992.

[18] Daeryong Kim and Bongwon Suh. Enhancing vaes for collaborative filtering: flexible priors & gating mechanisms. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 403–407, 2019.

[19] Yehuda Koren. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 426–434, 2008.

[20] Yehuda Koren, Robert Bell, and Chris Volinsky. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37, 2009.

[21] Xiucheng Li, Jin Yao Chin, Yile Chen, and Gao Cong. Sinkhorn collaborative filtering. In *Proceedings of the web conference 2021*, pages 582–592, 2021.

[22] Dawen Liang, Rahul G Krishnan, Matthew D Hoffman, and Tony Jebara. Variational autoencoders for collaborative filtering. In *Proceedings of the 2018 world wide web conference*, pages 689–698, 2018.

[23] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *2011 IEEE 11th international conference on data mining*, pages 497–506. IEEE, 2011.

[24] Xia Ning and George Karypis. Sparse linear methods with side information for top-n recommendations. In *Proceedings of the sixth ACM conference on Recommender systems*, pages 155–162, 2012.

[25] Kazusato Oko, Shunta Akiyama, and Taiji Suzuki. Diffusion models are minimax optimal distribution estimators. In *International Conference on Machine Learning*, pages 26517–26582. PMLR, 2023.

[26] Feiyang Pan, Shuokai Li, Xiang Ao, Pingzhong Tang, and Qing He. Warm up cold-start advertisements: Improving ctr predictions via learning to learn id embeddings. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 695–704, 2019.

[27] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. Updatable, accurate, diverse, and scalable recommendations for interactive applications. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 7(1):1–34, 2016.

[28] Zhen Qin, Le Yan, Honglei Zhuang, Yi Tay, Rama Kumar Pasumarthi, Xuanhui Wang, Mike Bendersky, and Marc Najork. Are neural rankers still outperformed by gradient boosted decision trees? 2021.

[29] Nils Reimers, I Sentence-BERT Gurevych, et al. Sentence embeddings using siamese bert-networks. arxiv 2019. *arXiv preprint arXiv:1908.10084*, 10, 1908.

[30] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *Fourteenth ACM Conference on Recommender Systems*, pages 240–248, 2020.

[31] Steffen Rendle, Walid Krichene, Li Zhang, and Yehuda Koren. Revisiting the performance of ials on item recommendation benchmarks. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 427–435, 2022.

[32] Ilya Shenbin, Anton Alekseev, Elena Tutubalina, Valentin Malykh, and Sergey I Nikolenko. Recvae: A new variational autoencoder for top-n recommendations with implicit feedback. In *Proceedings of the 13th international conference on web search and data mining*, pages 528–536, 2020.

[33] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference*, pages 3251–3257, 2019.

[34] Harald Steck. Autoencoders that don't overfit towards the identity. *Advances in Neural Information Processing Systems*, 33:19598–19608, 2020.

[35] Harald Steck, Maria Dimakopoulou, Nickolai Riabov, and Tony Jebara. Admm slim: Sparse recommendations for many users. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pages 555–563, 2020.

[36] Ilya Tolstikhin, Olivier Bousquet, Sylvain Gelly, and Bernhard Schoelkopf. Wasserstein autoencoders. *arXiv preprint arXiv:1711.01558*, 2017.

[37] Vojtěch Vančura and Pavel Kordík. Deep variational autoencoder with shallow parallel path for top-n recommendation (vasp). In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part V 30*, pages 138–149. Springer, 2021.

[38] Cédric Villani et al. *Optimal transport: old and new*, volume 338. Springer, 2009.

[39] Maksims Volkovs, Guangwei Yu, and Tomi Poutanen. Dropoutnet: Addressing cold start in recommender systems. *Advances in neural information processing systems*, 30, 2017.

[40] Wenjie Wang, Xinyu Lin, Liuhui Wang, Fuli Feng, Yinwei Wei, and Tat-Seng Chua. Equivariant learning for out-of-distribution cold-start recommendation. In *Proceedings of the 31st ACM International Conference on Multimedia*, pages 903–914, 2023.

[41] Jonathan Weed and Francis Bach. Sharp asymptotic and finite-sample rates of convergence of empirical measures in wasserstein distance. 2019.

[42] Liwei Wu, Shuqing Li, Cho-Jui Hsieh, and James Sharpnack. Sse-pt: Sequential recommendation via personalized transformer. In *Proceedings of the 14th ACM conference on recommender systems*, pages 328–337, 2020.

[43] Xiaoxiao Xu, Chen Yang, Qian Yu, Zhiwei Fang, Jiaxing Wang, Chaosheng Fan, Yang He, Changping Peng, Zhangang Lin, and Jingping Shao. Alleviating cold-start problem in ctr prediction with a variational embedding learning framework. In *Proceedings of the ACM Web Conference 2022*, pages 27–35, 2022.

A Appendix

A.1 Proof of Generalization

Proof of Theorem 3.5. Let $\mathcal{F}_c$ denote the set of 1-Lipschitz functions with respect to c and $\mathcal{H}_k$ denote the unit ball of Reproducing Kernel Hilbert Space functions with kernel k. For any $P \in \Delta(\Omega)$ and $\lambda > 0$, consider the minimizers $\vartheta^*, \xi^* = \arg\inf_{\vartheta,\xi} \mathsf{L}(\vartheta, \xi)$. It then holds that with high probability

$$\begin{aligned}\mathcal{E}_{n,P}(C_{\vartheta^*,\xi^*}) \leq & \delta(\vartheta^*, \xi^*; P) \\ & + \frac{1}{4}\mathbb{E}\left[\mathscr{W}_c\left(\hat{P}^n, p_{\vartheta^*}\#\gamma\right) + d_{\mathrm{MMD}}\left(q_{\xi^*}\#\hat{P}^n, \gamma\right)\right] \\ & + An^{-\frac{1}{d_c^*(P)}},\end{aligned} \tag{11}$$

for some $A > 0$ where $q_{\xi^*}\#\hat{P}^n(z) = \int_{\mathcal{X}} q_{\xi^*}(z \mid \mathsf{X}) d\hat{P}^n(\mathsf{X})$, $p_{\vartheta^*}\#\gamma = \int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z)\gamma(z)$ and $\delta(\vartheta^*, \xi^*; P) = 2\inf_{h\in\mathcal{F}_c}\left\|\log\frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty + 2\inf_{h\in\mathcal{H}_k}\left\|\log\frac{\gamma}{q_{\xi^*}\#P} - h\right\|_\infty$.

Proof. Throughout, we will assume that $\mathcal{F}_c$ are the set of 1-Lipschitz functions with respect to c. Additionally, we will use $d_{\mathcal{F}}$ to denote the Integral Probability Metric (IPM) with function class $\mathcal{F}$: $d_{\mathcal{F}}(P, Q) = \sup_{f\in\mathcal{F}}\{\mathbb{E}_P[f] - \mathbb{E}_Q[f]\}$. First note that

$$\mathcal{E}_{n,P}(C_{\vartheta^*,\xi^*}) = \mathbb{E}\left[d_{\mathrm{TV}}\left(P, \int_\Omega\int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z) q_{\xi^*}(z \mid X) dP(\mathsf{X})\right)\right] \tag{12}$$

$$\leq \mathbb{E}\left[d_{\mathrm{TV}}\left(P, \int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z) d\gamma(z)\right)\right] + \tag{13}$$

$$\mathbb{E}\left[d_{\mathrm{TV}}\left(\int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z) d\gamma(z), \int_\Omega\int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z) q_{\xi^*}(z \mid X) dP(\mathsf{X})\right)\right] \tag{14}$$

We now focus on bounding the first term. Using Pinskers inequality, we have for any $h \in \mathcal{F}_c$:

$$4d_{\mathrm{TV}}^2(P, p_{\vartheta^*}\#\gamma) \tag{15}$$

$$\leq d_{\mathrm{KL}}(P \parallel p_{\vartheta^*}\#\gamma) + d_{\mathrm{KL}}(p_{\vartheta^*}\#\gamma \parallel P) \tag{16}$$

$$\leq \int_\Omega \log\frac{P(X)}{p_{\vartheta^*}\#\gamma(X)} \cdot (P(X) - p_{\vartheta^*}\#\gamma(X)) dX \tag{17}$$

$$= \int_\Omega \left(\log\frac{P(X)}{p_{\vartheta^*}\#\gamma(X)} - h(X)\right) \cdot (P(X) - p_{\vartheta^*}\#\gamma(X)) dX + \int_\Omega h(X) \cdot (P(X) - p_{\vartheta^*}\#\gamma(X)) dX \tag{18}$$

$$\leq \int_\Omega \left(\log\frac{P(X)}{p_{\vartheta^*}\#\gamma(X)} - h(X)\right) \cdot (P(X) - p_{\vartheta^*}\#\gamma(X)) dX + d_{\mathcal{F}_c}(P, p_{\vartheta^*}\#\gamma) \tag{19}$$

$$\leq \left\|\log\frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty \|P - p_{\vartheta^*}\#\gamma\|_1 + d_{\mathcal{F}_c}(P, p_{\vartheta^*}\#\gamma) \tag{20}$$

$$\leq 2\left\|\log\frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty + d_{\mathcal{F}_c}(P, p_{\vartheta^*}\#\gamma), \tag{21}$$

where h is then taken to be the minimizer from $\mathcal{F}_c$. Next note that

$$d_{\mathcal{F}_c}(P, p_{\vartheta^*}\#\gamma) \leq d_{\mathcal{F}_c}(P, \hat{P}^n) + d_{\mathcal{F}_c}(\hat{P}^n, p_{\vartheta^*}\#\gamma). \tag{22}$$

Thus putting this together, we have

$$\mathbb{E}[d_{\mathrm{TV}}(P, p_{\vartheta^*}\#\gamma)] \leq \frac{1}{2}\inf_{h\in\mathcal{F}_c}\left\|\log\frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty + \frac{1}{4}\mathbb{E}[d_{\mathcal{F}_c}(P, \hat{P}^n)] + \frac{1}{4}\mathbb{E}[d_{\mathcal{F}_c}(\hat{P}^n, p_{\vartheta^*}\#\gamma)] \tag{23}$$

$$\leq \frac{1}{2}\inf_{h\in\mathcal{F}_c}\left\|\log\frac{P}{p_{\vartheta^*}\#\gamma} - h\right\|_\infty + O\left(n^{-\frac{1}{d^*(P)}}\right) + \frac{1}{4}\mathbb{E}\left[d_{\mathcal{F}_c}(\hat{P}^n, p_{\vartheta^*}\#\gamma)\right], \tag{24}$$

where we utilize [16, Lemma 21] for the last step and the infinum is taken by definition of being a minimizer. Note that we have $d_{\mathcal{F}_c} = \mathscr{W}_c$ by the Kantorovich-Rubinstein duality. For the second term in Eq (14), we can decompose it by taking the dual formulation of total variation:

$$\mathbb{E}\left[d_{\mathrm{TV}}\left(\int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z)d\gamma(z), \int_{\Omega}\int_{\mathcal{Z}} p_{\vartheta^*}(\cdot \mid z)q_{\xi^*}(z \mid X)dP(\mathsf{X})\right)\right] \tag{25}$$

$$= \mathbb{E}\left[\sup_{g:\|g\|_\infty \leq 1}\left\{\int_{\Omega}\int_{\mathcal{Z}} g(\mathsf{X})dp_{\vartheta^*}(\mathsf{X} \mid z)d\gamma(z) - \int_{\Omega}\int_{\Omega}\int_{\mathcal{Z}} g(\mathsf{X})dp_{\vartheta^*}(\mathsf{X} \mid z)dq_{\xi^*}(z \mid \mathsf{X})dP(\mathsf{X})\right\}\right] \tag{26}$$

$$\leq \mathbb{E}\left[d_{\mathrm{TV}}(\gamma, q_{\xi^*}\#P)\right]. \tag{27}$$

Let now $\mathcal{H}_k$ be the Reproducing Kernel Hilbert Space ball of unit 1 [11] then we have by the same analysis above except for $h \in \mathcal{H}_k$:

$$4d^2_{\mathrm{TV}}(\gamma, q_{\xi^*}\#P) \tag{28}$$

$$\leq d_{\mathrm{KL}}(\gamma \parallel q_{\xi^*}\#P) + d_{\mathrm{KL}}(q_{\xi^*}\#P \parallel \gamma) \tag{29}$$

$$\leq \int_{\Omega} \log\frac{\gamma(z)}{q_{\xi^*}\#P(z)} \cdot (\gamma(z) - q_{\xi^*}\#P(z))dz \tag{30}$$

$$= \int_{\Omega}\left(\log\frac{\gamma(z)}{q_{\xi^*}\#P(z)} - h(z)\right)\cdot(\gamma(X) - q_{\xi^*}\#P(z))dz + \int_{\Omega} h(z)\cdot(\gamma(z) - q_{\xi^*}\#P(z))dz \tag{31}$$

$$\leq \int_{\Omega}\left(\log\frac{\gamma(z)}{q_{\xi^*}\#P(z)} - h(z)\right)\cdot(\gamma(z) - q_{\xi^*}\#P(z))dz + d_{\mathcal{H}_k}(\gamma, q_{\xi^*}\#P) \tag{32}$$

$$\leq \left\|\log\frac{\gamma}{q_{\xi^*}\#P} - h\right\|_\infty \|\gamma - q_{\xi^*}\#P\|_1 + d_{\mathcal{H}_k}(\gamma, q_{\xi^*}\#P) \tag{33}$$

$$\leq 2\left\|\log\frac{\gamma}{q_{\xi^*}\#P} - h\right\|_\infty + d_{\mathcal{H}_k}(\gamma, q_{\xi^*}\#P). \tag{34}$$

Utilizing again the triangle inequality we have

$$d_{\mathcal{H}_k}(\gamma, q_{\xi^*}\#P) \leq d_{\mathcal{H}_k}(\gamma, q_{\xi^*}\#\hat{P}^n) + d_{\mathcal{H}_k}(q_{\xi^*}\#\hat{P}^n, q_{\xi^*}\#P). \tag{35}$$

Note that the second term converges as a rate of $O(n^{-\frac{1}{2}})$ due to [11] and is thus dominated by the rate of convergence $O(n^{-\frac{1}{d^*_c(P)}})$. The first term here is the dual form of the MMD and thus combining this with the first bound concludes the proof. □

A.2 Collaborative Filtering lower bound

We begin this section by defining Besov smoothness.

Definition A.1. For a function $f \in L^p(\Omega)$ for some $p \in (0, \infty]$, the r-th modulus of smoothness of f is defined by

$$w_{r,p}(f,t) = \sup_{\|h\|_2 \leq t} \|\Delta^r_h(f)\|_p, \quad \text{where} \quad \Delta^r_h(f)(x) \tag{36}$$

$$= \begin{cases} \sum_{j=0}^r \binom{r}{j}(-1)^{r-j} f(x+jh) & (\text{if } x + jh \in \Omega \text{ for all } j) \\ 0 & (\text{otherwise}). \end{cases} \tag{37}$$

Definition A.2 (Besov space $B^s_{p,q}(\Omega)$)**.** For $0 < p, q \leq \infty, s > 0, r := \lfloor s \rfloor + 1$, let the seminorm $|\cdot|_{B^s_{p,q}}$ be

$$|f|_{B^s_{p,q}} = \begin{cases} \left(\int_0^\infty (t^{-s} w_{r,p}(f,t))^q \frac{dt}{t}\right)^{\frac{1}{q}} & (q < \infty), \\ \sup_{t>0} t^{-s} w_{r,p}(f,t) & (q = \infty). \end{cases} \tag{38}$$

The norm of the Besov space $B^s_{p,q}$ is defined by $\|f\|_{B^s_{p,q}} = \|f\|_p + |f|_{B^s_{p,q}}$, and we have $B^s_{p,q} = \{f \in L^p(\Omega) | \, \|f\|_{B^s_{p,q}} < \infty\}$.

The Besov spaces constitute a rather large space of densities and is a standard assumption for the majority of density estimations works [17]. These lines of work study minimax estimation rates for densities lying in Besov spaces, to which we prove a similar result for the case of collaborative filtering.

We now show, under an assumption that the target density satisfies Besov continuinity conditions.

Proposition A.3. *Let $P \in \Delta(\Omega)$. Let $0 < p, q \leq \infty, s > 0$ such that*

$$s > \max\left(d\left(\frac{1}{p} - \frac{1}{2}\right), d\left(\frac{1}{p} - 1\right), 0\right). \tag{39}$$

Then, we have that

$$\inf_{C:\Omega\to\Delta(\Omega)} \sup_{P\in B^s_{p,q}} \mathcal{E}_{n,P}(C) \gtrsim n^{-\frac{s}{d+2s}}, \tag{40}$$

for some constant $c > 0$.

Proof. This result is a consequence of Proposition 5.2 from [25]. □

The main takeaway from this result is that under smoothness assumptions (characterized by the parameter s), the best rate we can achieve depends on the dimensionality d, which arises precisely from the embedding functions we use to compare similarity between items. In particular, if we do not use any item similarity embedding, then the dimensionality d will be the number of items in our collaborative filtering framework which can be quite large, leading to a larger error rate. Thus, this lower bound advocates for the use of encoders that reduce dimensionality.