

---

# An efficient manifold density estimator for all recommendation systems

---

**Jacek Dąbrowski**  
Synerise  
jack.dabrowski@synerise.com

**Barbara Rychalska**  
Synerise  
Warsaw University of Technology  
barbara.rychalska@synerise.com

**Michał Daniluk**  
Synerise  
michal.daniluk@synerise.com

**Dominika Basaj**  
Synerise  
Warsaw University of Technology  
dominika.basaj@synerise.com

**Konrad Gołuchowski**  
Synerise  
konrad.goluchowski@synerise.com

**Piotr Bąbel**  
Synerise  
piotr.babel@synerise.com

**Andrzej Michałowski**  
Synerise  
andrzej.michalowski@synerise.com

## Abstract

Many unsupervised representation learning methods belong to the class of similarity learning models. While various modality-specific approaches exist for different types of data, a core property of many methods is that representations of similar inputs are close under some similarity function. We propose EMDE (Efficient Manifold Density Estimator) – a framework utilizing arbitrary vector representations with the property of local similarity to succinctly represent smooth probability densities on Riemannian manifolds. Our approximate representation has the desirable properties of being fixed-size and having simple additive compositionality, thus being especially amenable to treatment with neural networks – both as input and output format, producing efficient conditional estimators. We generalize and reformulate the problem of multi-modal recommendations as conditional, weighted density estimation on manifolds. Our approach allows for trivial inclusion of multiple interaction types, modalities of data as well as interaction strengths for any recommendation setting. Applying EMDE to both top-k and session-based recommendation settings, we establish new state-of-the-art results on multiple open datasets in both uni-modal and multi-modal settings. We release the source code and our own real-world dataset of e-commerce product purchases, with special focus on modeling of the item cold-start problem.

## 1 Introduction

An increasingly common problem setting in many domains is that predictions must be made from multi-modal interaction data, with many interaction types, object metadata coming from multiple domains, and having different types of attributes. Interactions often have one or more weight values

attached, representing e.g. cardinality, strength, duration, or monetary value. Such a landscape is prevalent in applications from IoT sensor networks, through social networks to brick-and-mortar retail and e-commerce. A natural habitat for multimodal, multi-attribute interaction data is that of *recommender systems*. Their task is to suggest items which a user might find interesting, often in the setting of online stores or social media. There exist multiple recommendation-based sub-tasks such as session-based recommendation (where the input is an ordered collection of items based on a single user shopping session), or top-k recommendation (where the item collection is an unordered set of items which user found interesting, usually over a longer timespan). Usually, each of these sub-problems is solved by one specialized algorithm. Moreover, state-of-the-art collaborative filtering systems in both top-k and session-based recommendation usually offer no simple way to incorporate multi-modal item information i.e. images or text, and are limited to a single type of interaction. However, a joint multi-modal representation of a single object is a native construct of the human brain which enables us to form complex associations based on multi-level semantic similarity [36] [37], so performance gains can be expected from such representations.

In order to address these issues, we present EMDE (Efficient Manifold Density Estimator) – a one-pass algorithm for efficient, approximate representation of smooth probability densities on Riemannian manifolds. EMDE can utilize any locally-similar embedding to construct a structured representation of the vector embedding manifold (that we call *sketch*). It preserves local geometry, and allows estimation of arbitrary smooth probability densities out of samples scattered on the manifold.

In this paper, we generalize and reformulate recommendation tasks as conditional density estimation on manifolds allowing us to use EMDE. We then apply EMDE to both top-k and session-based recommendation tasks – considering *multiple types of interactions* as well as *multi-modal input*. Symmetrically we produce *multi-modal output*, which enables an elegant solution of the item cold-start problem (recommending items with no interactions, due to being freshly introduced into inventory). [9] and [26] observe the *phantom progress* problem in recommender systems: carefully tuned simple heuristics (such as nearest-neighbor methods) in practice often outperform complex deep learning models, while algorithm performance is heavily dependent on the dataset and chosen performance metrics. In response to this, we test our approach on a wide range of referential metrics and datasets. We show that our approach consistently outperforms both the most advanced models and simple heuristics alike.

Our contributions are the following:

**We introduce an efficient, approximate representation of smooth probability densities on Riemannian manifolds called EMDE.** It represents real-valued multisets of arbitrary high-dimensional vectors on a locally-similar embedding manifold in the form of *locality-sensitive sketches*. Representations produced by EMDE have constant size and are independent of the number of samples and original embeddings dimensions. Thus the size of downstream model, that utilizes this representations, does not depend on the number of samples and embedding dimensionality. Flexibility comes from the ability to combine various modalities of input data without the necessity to create a joint embedding model, the ability to aggregate multiple samples with attached weights into a fixed-size structure, and from an efficient readout method.

**We establish new state-of-the-art results on several datasets for session-based and top-k scenarios.** To this end, we introduce a generalization and reformulation of recommendation tasks as conditional density estimation on manifolds. Our formulation trivially allows the use of multi-modal data, multiple types of interactions, and arbitrary non-negative weights attached to interactions (e.g. counts, durations, monetary values), for all types of recommendation problems.

**We prove that EMDE naturally solves the item cold-start problem.** We also release a new dataset: *Online Sales*, aimed specifically at testing item cold-start scenario. The dataset is collected from real life user-item interactions in our system. This allows us to acquire naturally occurring cold start items, avoiding artificial manipulations of older datasets where cold start items are naturally few [13]. We also release the code for EMDE and our experiments.

## 2 Related Work

**Sketching-based density estimators.** [4] introduce methods for KDE based on locality-sensitive hashing (LSH). Subsequently [38] utilize a sketch-based structure for a compressed representation of

multiple LSH partitions for KDE estimation. Both methods require a computation-heavy sampling procedure to arrive at density estimates. [5] introduce RACE - a LSH sketch-based method for KDE, which does not require sampling to arrive at density estimates. [6] further explores the technique of LSH sketching for approximate nearest-neighbor search on streaming data. In these methods, the manifold considered is  $\mathbb{R}^n$ . We instead disentangle the notion of the manifold on which data lies from the densities to be estimated on the manifold. We do not use kernel functions, and our estimates are piecewise-constant. Our primary interest is in manifold probability densities as an input and output format for neural models.

**Session-based recommenders.** Following [26] we compare our method to both simple baselines like S-KNN, V-SKNN and recent deep-learning based models i.e. Gru4Rec [16], NARM [24], STAMP [24], NextFitNet [47], and SR-GNN [46].

**Top-k recommenders.** [9] find out that among 18 recently published algorithms, only 7 could be reproduced, and 6 of them can often be outperformed with simple heuristics. Thus, we compare our method to MultVAE [25], as it proves to be the only method among the 18 that could be both reproduced and is not outperformed by simple methods. We also implement two recent state-of-the-art methods MacridVAE [29] and EASE [39].

**Content-based recommenders.** Methods with explicit side-information are often tuned to specific datasets or tasks (e.g. news recommendation) [10] [17]. To the best of our knowledge, there has been no attempt at a generalized multi-modal system in session-based setting. The more general top-k recommender systems ingest image [15], audio [42] or knowledge base [48] [19] side information, or a selected combination of modalities [2]. [45] learn a joint representation of content information and collaborative filtering ratings. [13] extend SVD to enable incorporation of categorical side data. Although some of them are very recent, none of these methods has been shown to outperform the top performing uni-modal models such as EASE or MacridVAE, their evaluation is often done against weaker baselines.

Our work differs from cited papers in several aspects: 1) we outperform the best currently known recommendation systems, 2) we impose no restrictions on the task (top-k or session-based), or dataset, 3) we make no assumptions as to the type and number of utilized data modalities 4) we support partial modality coverage and do not discard items with missing modalities, and 5) we support a cold-user scenario, where test set can contain unseen users.

### 3 Algorithm

Intuitively, EMDE is a form of a piecewise-constant probability density estimator which can work on arbitrary Riemannian manifolds locally embedded in  $\mathbb{R}^n$ , while efficiently scaling to extremely large dimensionalities. Our solution is inspired by two well known algorithms: locality sensitive hashing (LSH) [20] and count-sketch / count-min sketch (CMS) [8]. We utilize vector representations coming from upstream metric representation learning methods on any modality of data to perform multiple partitionings of the embedding manifold, using data-dependent LSH methods. The partitionings define regions of the manifold analogous to CMS hash function buckets. When multiple such partitionings are combined, intersection of regions from independent partitionings allows to describe sub-regions much smaller than the resolution of each individual partition. This way of representation constitutes a compressed map of the manifold, allowing to store and accumulate values assigned to local regions. Pointwise estimates can then be retrieved efficiently by aggregating stored values over regions overlapping the query point.

#### 3.1 Unsupervised similarity learning and Riemannian manifolds

The goal of (deep) metric representation learning is to learn a function  $h_\theta(x) : \mathbb{X} \rightarrow \mathbb{R}^n$  mapping inputs from the data manifold in  $\mathbb{X}$  onto points in  $\mathbb{R}^n$  which are metrically close if and only if they are semantically similar. In practice,  $h_\theta(\mathbb{X}) \subset \mathbb{R}^n$  forms a Riemannian manifold locally embedded in euclidean space. Our method requires that the aforementioned property of *local similarity* under a locally-euclidean metric holds at least approximately. Not all methods of deep representation learning follow the metric learning paradigm, even though they are very popular in practice, optimizing e.g. skip-gram, masked-language-model, next sentence prediction, CPC, InfoNCE or DeepInfoMax objectives [31][11][43][18]. Nonetheless, [23] show that all the aforementioned self-supervised ob-

jectives correspond to InfoNCE, while [40] observe that InfoNCE has a direct formulation in terms of metric learning. Thus, most existing representation learning methods produce embeddings indirectly optimized for local similarity, and can be utilized by our method.

The starting point of our algorithm is a manifold  $\mathbb{M} := h_\theta(\mathbb{X})$  locally embedded in  $\mathbb{R}^n$ . Our goal is to create a compressed, approximate, piecewise-constant representation for any smooth probability density on the manifold, from point samples.

### 3.2 Efficient Manifold Density Estimator

Given samples from a data manifold  $\mathbb{M} \subset \mathbb{R}^n$  and parameters  $N, K \in \mathbb{N}^+$  we first construct a density-dependent mapping function  $V(x) : \mathbb{M} \rightarrow (e_1, \dots, e_N)$  where  $e_i \in \{1, \dots, 2^K\}$  assigning inputs  $x \in \mathbb{M}$  to multiple local regions of the manifold. To this end, we utilize a modified version of LSH algorithm we call Density-dependent LSH (DLSH). We start with choosing  $K$  random vectors  $r_i$ , then for  $v \in \mathbb{M}$  we let  $hash_i(v) = sgn(v \cdot r_i - b_i)$ , drawing the bias value  $b_i$  from  $Q_i(U \sim \text{Unif}[0, 1])$ , where  $Q_i$  is the quantile function of  $\{v \cdot r_i : v \in \mathbb{M}\}$ . In contrast to LSH, this scheme is density-dependent, cutting the the manifold into non-empty parts, thus avoiding unutilized regions. We combine  $K$  independent binary hash values into bit strings, which we interpret as short integers, giving a partitioning of the input manifold into  $2^K$  regions. We perform the procedure  $N$  times, resulting in a *sketch structure* of width  $2^K$  and *depth*  $N$ . Local similarity of the data manifold allows  $V(x)$  to assign semantically similar inputs to the same sketch regions frequently. This effect captures the local metric prior induced by the underlying representation learning method which produced the vectors  $\mathbb{M}$ . In practice  $\mathbb{M}$  can represent a single view or modality of information, such as text, image, audio or network interaction embeddings.

An empty *sketch* is instantiated as 2-dimensional array  $N \times 2^K$  and can be indexed by the outputs of  $V$ . Filled with zeros, it represents a degenerate zero density. We add samples  $x \in H$  weighted by  $w \in R^+$  from a smooth probability measure on the manifold  $\mathbb{M}$ , where  $f$  is the probability density function, by performing  $sketch_{samples}[V(x)] := sketch_{samples}[V(x)] + w_x$  for all  $x \in H$  and their corresponding weights  $w \in R^+$ . Our final representation is  $sketch := sketch_{samples} / \sum w_x$ . From the definitions of  $sketch_{samples}$  and  $\sum w_x$  it is clear that both are additive, can be merged with simple point-wise summation and can be constructed incrementally in a streaming setting. We call values stored in the final array *sketch content*.

To get an un-normalized point estimate  $\hat{f}$  of the probability density function  $f$ , for any  $z \in \mathbb{M}$ , we let  $\hat{f}(z) = GeometricMean(sketch[V(z)])$ . We verify the choice of geometric mean empirically, while strong theoretical arguments behind its suitability can be found in [12] and [21], when we notice that every sketch contains  $N$  probability mass function estimates over all  $2^K$  regions of the manifold  $\mathbb{M}$  each, effectively forming an ensemble of probability densities.

Due to their additive compositionality and fixed-size representation, our sketch structures are a natural fit for representing real-valued multisets of vectors (i.e. sets of vectors with attached weights). Sketch *structure*, defined by the mapping function  $V(x)$ , captures local semantic similarity of data – inputs metrically close on the underlying manifold have a high probability to share buckets in the sketch array. Sketch *content* refers to the accumulated density estimates. For example user’s purchased, viewed, and searched items can all be represented in sketches with visual item similarity *structure*, but represent different *content*. Multiple such *structure-content* combinations are possible, enabling numerous usage scenarios. Sketches with different *structure* and *content* can be concatenated, while retaining their favorable properties.

### 3.3 Conditional and multi-modal estimation

A simple feed-forward neural network can be used as a conditional density estimator, where sketches are both the input and output format. This allows for both multi-modal input and output. We set the loss function for conditional estimation to be *Softmax + CrossEntropy*, where both are applied only width-wise – independently for each level of depth for every sketch, and the resulting values averaged. This approach is consistent with sketches representing an ensemble of PMFs over regions of the manifold.

### 3.4 Recommendation as conditional density estimation on manifolds

An example research field where multiple modalities of data are present and multisets of events must be aggregated into compact representations, is that of *recommender systems*. Items can be described by their interactions with users (sometimes interactions of different types, e.g. click, purchase, favorite), by their names, attributes, images – all of which constitute different modalities of input. Users are represented by their interactions with items. Assuming each item is described in several modalities  $\mathbb{X}_1, \dots, \mathbb{X}_p$ , and via upstream unsupervised or self-supervised representation learning methods, and those modalities are encoded on manifolds  $\mathbb{M}_1, \dots, \mathbb{M}_p$ , we can construct sketches structurally representing the manifolds and concatenate them.

Recommendation problems have a natural interpretation as density estimation – user-item interactions can be considered to be samples from a user’s interest distribution over the space of all items. For top-k recommendations, we treat both input and output as simple sets, which can be represented as a uniform probability distribution over all inputs in the set. So in the density representation, we set all weights  $w = 1$ . For session-based recommendations, when items can occur multiple times, with timestamps or sequential ordering, we can utilize weights  $w$  to reflect their relative importance, e.g. setting  $w = \text{occurrences}(item)$  or  $w = \lambda^{\text{position}(item)}$  to reflect decaying interest in items seen long ago. It is worth noting that other non-negative scalars accompanying interactions, e.g. amount of money spent on an item, duration of time an item was viewed, etc. can be trivially incorporated with our framework if available – via the weights  $W$  either in place of, or in addition to the proposed sketches via concatenation.

We apply this form of multi-modal encoding to both input and target of the learning problem, using a simple feed-forward neural network to learn conditional estimation in between. The input may have more sketches, e.g. differentiating between most recent and historic interactions, while the output needs to have at least one modality to perform item score retrieval. Retrieving item scores is performed by applying the point estimate procedure  $\hat{f}(z)$  to item vectors  $z \in \mathbb{M}$ . We pick items with the highest scores as output recommendations. Contrary to conventional recommendation systems, provided that item embeddings are given beforehand, computational complexity of the training phase is independent of the number of items, and depends exclusively on the number and dimensions of input and output sketches – item scores are retrieved only during inference or validation with recommendation-specific metrics.

### 3.5 Simple baseline for locally similar network embedding

In order to disentangle performance of our framework from the quality of common upstream representation learning methods, we utilize an extremely simple network embedding method with the desired property of local similarity for both item attributes and interaction data.

Given an interaction network (e.g. between users and items) with edges  $E$ , we define the random walk transition matrix  $\mathbf{M}$ , where where  $e_{ij}$  is the number of edges running from  $i$  to  $j$ , as  $\mathbf{M}_{ij} = \frac{e_{ij}}{d_i}$  for  $ij \in E$ ; 0 for  $ij \notin E$ .

We initialize the embedding matrix  $\mathbf{T}_0 \in \mathbb{R}^{N \times d} \sim U(-1, 1)$ , where  $d$  is the dimensionality. Then for  $q$  iterations, we perform:  $\mathbf{T}_i = \mathbf{M} \cdot \mathbf{T}_{i-1}$ ;  $L_2$  normalize rows of  $\mathbf{T}_i$ .  $\mathbf{T}_q$  is our final embedding matrix. The method can be interpreted as iterated  $L_2$  normalized weighted averaging of neighboring nodes’ representations. After just 1 iteration, nodes with similar 1-hop neighborhoods in the network will have similar representations. Further iterations extend the same principle to q-hop neighborhoods. As our experiments show, despite its extreme simplicity, the method works well enough for EMDE.

## 4 Experiments

We report results for unimodal EMDE (no multimodal data, only basic user-item interactions), and EMDE MM – configurations where selected multimodal channels are present. For each experiment, we carefully fine-tune the baselines or use the best configurations reported by the authors. We implement all algorithms in the same frameworks of [9] [27], keeping to their selected performance measures and datasets. We simplify our experimental setting by using modality embeddings obtained with our embedding algorithm described in 3.5, which is the same for representing both text

and interaction networks (for text we create a graph of item-word edges). We leave experiments with elaborate embeddings such as BERT [11] for future research. We explore only a subset of all possible dataset modalities. In Tables 1 and 2 we show the results while detailed descriptions of the datasets can be found in the Appendix.

#### 4.1 EMDE as a Session-Based Recommender System

We compare against benchmark methods from [27], adding one recent graph neural model [46].

**Datasets.** We conduct experiments on six datasets, three from the music domain (NOWP [35], 30MUS [41], AOTM) and three from the e-commerce domain (RETAIL, DIGI, RSC15). We use evaluation framework from [27], with already optimized hyper-parameters for other methods and a number of metrics evaluated on these datasets. Each dataset was split into five time-contiguous subsets to be able to make multiple measurements in order to minimize the risk of random effects.

**EMDE Configuration** On top of EMDE, we use a simple 3-layer feed forward network with leaky ReLU activations, batch normalization and residual connections between layers. Whenever possible, we incorporate multimodal item features such as additional metadata or data from different sources such as search queries, playlists, purchases etc. More details about training configuration for each dataset are available in Appendix 1.2.

**EMDE Performance.** As presentend in Table 1, EMDE significantly outperforms competing approaches when multi-modal data is available (EMDE MM), and it is a very strong baseline in the uni-modal case (EMDE). [27] find that neural methods generally underperform compared to simple approaches (nearest-neighbors). In contrast, EMDE MM – a neural model – outperforms all competitors on RETAIL, DIGI and 30MU datasets by a large margin. We were unable to locate additional modalities of item data for 2 music datasets (AOTM and NOWP). This problem is rather artificial as in a practical setting each item is nearly always characterized with some multimodal data (i.e. product name and attributes at the very least). Still, EMDE achieves state-of-the-art results on NOWP and has the highest MRR on AOTM. Consistently with [27], the RSC15 dataset proves to be an outlier where EMDE is either the top performer or a close follower of NARM depending on metric.

#### 4.2 EMDE as a top-k Recommender System

**Datasets.** We conduct experiments on two real-world, large-scale datasets: Netflix Prize [3] and MovieLens20M. The datasets contain movie ratings from viewers. Dataset characteristics and pre-processing are consistent with [9].

**Baselines.** We compare against recent state-of-the-art VAE-based neural models: MultVAE [25] and MacridVAE [29], and a non-neural algorithm EASE [39], as well as simpler baselines used by [9].

**EMDE Configuration** On top of EMDE, we use a simple one-hidden-layer feed-forward neural network with 12,000 neurons and leaky ReLU activations. As additional modalities we choose the interactions of users with disliked items (items which received a rating lower than 4). For MovieLens20M, we also experiment with textual descriptions of movies and lists of movie actors, treated as a network with our embedding method (edges between items and tokens, or items and cast members). We do not observe improvements with incorporation of these modalities, which we hypothesize to be implicitly modeled by liked and disliked item interactions. For Netflix, there is no multi-modal data accessible apart from movie titles and movie production year (both have low relevance for user preferences). We put 80% of randomly shuffled user items into the input sketch, and the remaining 20% into the output sketch to reflect train/test split ratio of items for a single user.

**EMDE Performance.** We observe that our approach consistently and significantly outperforms the baselines especially for lower  $k$  values in the top- $k$  recommended item rankings, which is consistent with CMS being a heavy-hitters estimator. The density of output sketches is determined by data distribution, e.g. the median user liked item count for Netflix is 60, so the median sketch density is 12 items (20% out of 60) - the sketch is not expected to decode items for much higher  $k$ . In practice, the very top recommended items are key for user satisfaction as they are given the most attention by users, considering the limitation of item display capabilities and user’s attention in the real world.

Table 1: Session-based recommendation results

| Model          | MRR@20   | P@20          | R@20          | HR@20         | MAP@20        | Model          | MRR@20   | P@20          | R@20          | HR@20         | MAP@20        |
|----------------|--|---------------|---------------|---------------|---------------|----------------|--|---------------|---------------|---------------|---------------|
| RETAIL         |  |               |               |               |               | RSC15          |  |               |               |               |               |
| EMDE MM        | <b>0.3649</b>  | <b>0.0556</b> | <b>0.4974</b> | <b>0.6202</b> | <b>0.0302</b> | EMDE MM        | <b>0.3116</b>  | <b>0.0743</b> | 0.5000        | 0.6680        | 0.0352        |
| EMDE           | 0.3522   | 0.0512        | 0.4643        | 0.5774        | 0.0277        | EMDE           | 0.3104   | 0.0730        | 0.4936        | 0.6619        | 0.0346        |
| VS-KNN [28]    | 0.3395   | 0.0531        | 0.4632        | 0.5745        | 0.0278        | CT [30]        | 0.3072   | 0.0654        | 0.4717        | 0.6359        | 0.0316        |
| S-KNN [26]     | 0.337  | 0.0532        | 0.4707        | 0.5788        | 0.0283        | NARM [24]      | 0.3047   | 0.0735        | <b>0.5109</b> | <b>0.6751</b> | <b>0.0357</b> |
| Gru4Rec [16]   | 0.3237   | 0.0502        | 0.4559        | 0.5669        | 0.0272        | STAMP [24]     | 0.3033   | 0.0713        | 0.4979        | 0.6654        | 0.0344        |
| NARM [24]      | 0.3196   | 0.044         | 0.4072        | 0.5549        | 0.0239        | SR [22]        | 0.301  | 0.0684        | 0.4853        | 0.6506        | 0.0332        |
| SR-GNN [46]    | 0.2957   | 0.0459        | 0.4273        | 0.5277        | 0.0249        | AR [1]         | 0.2894   | 0.0673        | 0.476         | 0.6361        | 0.0325        |
| STAMP [24]     | 0.2527   | 0.0428        | 0.3922        | 0.462         | 0.0229        | VS-KNN [28]    | 0.2872   | 0.0707        | 0.4937        | 0.6512        | 0.0341        |
| SR [22]        | 0.2453   | 0.0362        | 0.3359        | 0.4174        | 0.0194        | Gru4Rec [16]   | 0.2826   | 0.0682        | 0.4837        | 0.648         | 0.0334        |
| AR [1]         | 0.2407   | 0.0387        | 0.3533        | 0.4367        | 0.0205        | S-KNN [26]     | 0.262  | 0.0657        | 0.4658        | 0.5996        | 0.0318        |
| CT [30]        | 0.2305   | 0.0308        | 0.2902        | 0.3632        | 0.0162        | NEXTITNET [47] | <i>could not complete hyperparameter tuning [27]</i> |               |               |               |               |
| NEXTITNET [47] | 0.2038   | 0.032         | 0.3051        | 0.3779        | 0.0173        | SR-GNN [46]    | <i>could not complete hyperparameter tuning</i>      |               |               |               |               |
| DIGI           |  |               |               |               |               | NOWP           |  |               |               |               |               |
| EMDE MM        | 0.1731   | <b>0.0620</b> | <b>0.3849</b> | <b>0.4908</b> | <b>0.0268</b> | EMDE MM        | <i>no multimodal data available</i>                  |               |               |               |               |
| EMDE           | 0.1714   | 0.0595        | 0.3697        | 0.4691        | 0.0254        | EMDE           | <b>0.1108</b>  | <b>0.0617</b> | <b>0.1847</b> | <b>0.2665</b> | 0.0179        |
| VS-KNN [28]    | <b>0.1784</b>  | 0.0584        | 0.3668        | 0.4729        | 0.0249        | CT [30]        | 0.1094   | 0.0287        | 0.0893        | 0.1679        | 0.0065        |
| S-KNN [26]     | 0.1714   | 0.0596        | 0.3715        | 0.4748        | 0.0255        | Gru4Rec [16]   | 0.1076   | 0.0449        | 0.1361        | 0.2261        | 0.0116        |
| Gru4Rec [16]   | 0.1644   | 0.0577        | 0.3617        | 0.4639        | 0.0247        | SR [22]        | 0.1052   | 0.0466        | 0.1366        | 0.2002        | 0.0133        |
| SR-GNN [46]    | 0.1437   | 0.0534        | 0.331         | 0.4277        | 0.0221        | SR-GNN [46]    | 0.0981   | 0.0414        | 0.1194        | 0.1968        | 0.0101        |
| NEXTITNET [47] | 0.1424   | 0.038         | 0.2416        | 0.2922        | 0.0149        | STAMP [24]     | 0.0897   | 0.0455        | 0.1245        | 0.1919        | 0.0111        |
| NARM [24]      | 0.1392   | 0.0528        | 0.3254        | 0.4188        | 0.0218        | NARM [24]      | 0.0894   | 0.0463        | 0.1274        | 0.1849        | 0.0118        |
| STAMP [24]     | 0.1314   | 0.0489        | 0.304         | 0.3917        | 0.0201        | VS-KNN [28]    | 0.081  | 0.0664        | 0.1828        | 0.2534        | 0.0193        |
| AR [1]         | 0.128  | 0.0463        | 0.2872        | 0.372         | 0.0189        | AR [1]         | 0.071  | 0.0564        | 0.1544        | 0.2076        | 0.0166        |
| SR [22]        | 0.1216   | 0.0406        | 0.2517        | 0.3277        | 0.0164        | S-KNN [26]     | 0.0687   | 0.0655        | 0.1809        | 0.245         | <b>0.0186</b> |
| CT [30]        | 0.1075   | 0.0294        | 0.186         | 0.2494        | 0.0115        | NEXTITNET [47] | <i>could not complete hyperparameter tuning [27]</i> |               |               |               |               |
| 30MU           |  |               |               |               |               | AOTM           |  |               |               |               |               |
| EMDE MM        | <b>0.2703</b>  | <b>0.1106</b> | <b>0.2503</b> | <b>0.4105</b> | <b>0.0331</b> | EMDE MM        | <i>no multimodal data available</i>                  |               |               |               |               |
| EMDE           | <b>0.2673</b>  | <b>0.1102</b> | <b>0.2502</b> | <b>0.4104</b> | <b>0.0330</b> | EMDE           | <b>0.0123</b>  | 0.0083        | 0.0227        | 0.0292        | 0.0020        |
| CT [30]        | 0.2502   | 0.0308        | 0.0885        | 0.2882        | 0.0058        | CT [30]        | 0.0111   | 0.0043        | 0.0126        | 0.0191        | 0.0006        |
| SR [22]        | 0.241  | 0.0816        | 0.1937        | 0.3327        | 0.024         | NARM [24]      | 0.0088   | 0.005         | 0.0146        | 0.0202        | 0.0009        |
| Gru4Rec [16]   | 0.2369   | 0.0617        | 0.1529        | 0.3273        | 0.015         | STAMP [24]     | 0.0088   | 0.002         | 0.0063        | 0.0128        | 0.0003        |
| NARM [24]      | 0.1945   | 0.0675        | 0.1486        | 0.2956        | 0.0155        | SR [22]        | 0.0074   | 0.0047        | 0.0134        | 0.0186        | 0.001         |
| VS-KNN [28]    | 0.1162   | 0.109         | 0.2347        | 0.383         | 0.0309        | Gru4Rec [16]   | 0.0074   | 0.002         | 0.0063        | 0.013         | 0.0003        |
| AR [1]         | 0.096  | 0.0886        | 0.193         | 0.3088        | 0.0254        | SR-GNN [46]    | 0.0068   | 0.0053        | 0.0153        | 0.0197        | 0.0009        |
| S-KNN [26]     | 0.0898   | 0.1073        | 0.2217        | 0.3443        | 0.029         | NEXTITNET [47] | 0.0065   | 0.0024        | 0.0071        | 0.0139        | 0.0004        |
| STAMP [24]     | 0.0819   | 0.0411        | 0.0875        | 0.1539        | 0.0093        | AR [1]         | 0.0059   | 0.0076        | 0.02          | 0.0233        | 0.0018        |
| NEXTITNET [47] | <i>could not complete hyperparameter tuning [27]</i> |               |               |               |               | VS-KNN [28]    | 0.0057   | 0.0116        | 0.0312        | 0.0352        | 0.0032        |
| SR-GNN [46]    | <i>could not complete hyperparameter tuning</i>      |               |               |               |               | S-KNN [26]     | 0.0054   | <b>0.0139</b> | <b>0.039</b>  | <b>0.0417</b> | <b>0.0037</b> |

Table 2: Top-k recommendation results

| Model                                     | R@1                       | NDCG@3        | R@3           | NDCG@5        | R@5           | NDCG@10       | R@10          | NDCG@20       | R@20          | NDCG@50       | R@50          | NDCG@100      | R@100         |
|---|---------------------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|---------------|
| MovieLens 20M                             |                           |               |               |               |               |               |               |               |               |               |               |               |               |
| EMDE MM                                   | <b>0.0670</b>             | <b>0.1992</b> | <b>0.1408</b> | <b>0.2378</b> | <b>0.1963</b> | <b>0.2890</b> | <b>0.2780</b> | <b>0.3358</b> | <b>0.3710</b> | <b>0.3901</b> | 0.5033        | <b>0.4274</b> | 0.6054        |
| EMDE                                      | 0.0496                    | 0.1585        | 0.1152        | 0.1925        | 0.1587        | 0.2417        | 0.2385        | 0.2907        | 0.3369        | 0.3517        | 0.4859        | 0.3912        | 0.5974        |
| EASE [39]                                 | 0.0507                    | 0.1618        | 0.1150        | 0.1990        | 0.1616        | 0.2530        | 0.2465        | 0.3078        | 0.3542        | 0.3756        | 0.5146        | 0.4194        | 0.6347        |
| MultiVAE [25]                             | 0.0514                    | 0.1589        | 0.1164        | 0.1955        | 0.1627        | 0.2493        | 0.2488        | 0.3052        | 0.3589        | 0.3758        | <b>0.5293</b> | 0.4216        | <b>0.6551</b> |
| SLIMElasticNet [32]                       | 0.0474                    | 0.1528        | 0.1085        | 0.1885        | 0.1533        | 0.2389        | 0.2318        | 0.2916        | 0.3350        | 0.3572        | 0.4893        | 0.4012        | 0.6103        |
| MacridVAE [29]                            | 0.0435                    | 0.1413        | 0.1024        | 0.1767        | 0.1475        | 0.2271        | 0.2268        | 0.2803        | 0.3331        | 0.3488        | 0.4971        | 0.3953        | 0.6256        |
| RP3beta [34]                              | 0.0380                    | 0.1243        | 0.0884        | 0.1550        | 0.1279        | 0.2004        | 0.2007        | 0.2501        | 0.3018        | 0.3146        | 0.4592        | 0.3594        | 0.5842        |
| ItemKNN CF [9]                            | 0.0398                    | 0.1268        | 0.0912        | 0.1558        | 0.1281        | 0.1989        | 0.1973        | 0.2431        | 0.2857        | 0.3037        | 0.4342        | 0.3467        | 0.5563        |
| P3alpha [7]                               | 0.0329                    | 0.1073        | 0.0770        | 0.1342        | 0.1116        | 0.1745        | 0.1765        | 0.2174        | 0.2616        | 0.2702        | 0.3851        | 0.2965        | 0.4486        |
| PureSVD                                   | 0.0257                    | 0.0834        | 0.0601        | 0.1029        | 0.0847        | 0.1299        | 0.1269        | 0.1595        | 0.1845        | 0.2028        | 0.2892        | 0.2402        | 0.3988        |
| TopPop [9]                                | 0.0180                    | 0.0566        | 0.0389        | 0.0717        | 0.0580        | 0.0934        | 0.0912        | 0.1201        | 0.1441        | 0.1569        | 0.2320        | 0.1901        | 0.3296        |
| Netflix Prize                             |                           |               |               |               |               |               |               |               |               |               |               |               |               |
| EMDE MM                                   | <b>0.0388</b>             | <b>0.1374</b> | <b>0.0891</b> | <b>0.1699</b> | <b>0.1253</b> | <b>0.2155</b> | <b>0.1875</b> | <b>0.2619</b> | 0.2645        | 0.3223        | 0.3870        | 0.3668        | 0.4924        |
| EMDE                                      | 0.0310                    | 0.1160        | 0.0741        | 0.1448        | 0.1059        | 0.1876        | 0.1652        | 0.2332        | 0.2432        | 0.2952        | 0.3708        | 0.3419        | 0.4823        |
| EASE [39]                                 | 0.0323                    | 0.1236        | 0.0772        | 0.1558        | 0.1120        | 0.2050        | 0.1782        | 0.2589        | 0.2677        | <b>0.3329</b> | <b>0.4157</b> | <b>0.3879</b> | 0.5443        |
| MultiVAE [25]                             | 0.0313                    | 0.1174        | 0.0765        | 0.1485        | 0.1109        | 0.1957        | 0.1756        | 0.2483        | 0.2645        | 0.3224        | 0.4155        | 0.3785        | <b>0.5484</b> |
| SLIMElasticNet [32]                       | 0.0307                    | 0.1178        | 0.0734        | 0.1484        | 0.1062        | 0.1952        | 0.1688        | 0.2474        | 0.2552        | 0.3196        | 0.3996        | 0.3746        | 0.5291        |
| MacridVAE [29]                            | 0.0291                    | 0.1098        | 0.0700        | 0.1389        | 0.1020        | 0.1834        | 0.1629        | 0.2336        | 0.2476        | 0.3054        | 0.3935        | 0.3600        | 0.5228        |
| RP3beta [34]                              | 0.0243                    | 0.0946        | 0.0595        | 0.1191        | 0.0863        | 0.1578        | 0.1390        | 0.2012        | 0.2116        | 0.2629        | 0.3343        | 0.3075        | 0.4304        |
| ItemKNN CF [9]                            | 0.0241                    | 0.0930        | 0.0586        | 0.1171        | 0.0850        | 0.1548        | 0.1370        | 0.1969        | 0.2092        | 0.2595        | 0.3394        | 0.3093        | 0.4601        |
| P3alpha [7]                               | 0.0244                    | 0.0869        | 0.0574        | 0.1093        | 0.0826        | 0.1452        | 0.1327        | 0.1851        | 0.1997        | 0.2396        | 0.3056        | 0.2782        | 0.3866        |
| PureSVD                                   | 0.0121                    | 0.0478        | 0.0279        | 0.0608        | 0.0416        | 0.0810        | 0.0682        | 0.1052        | 0.1079        | 0.1542        | 0.2134        | 0.1954        | 0.3170        |
| TopPop [9]                                | 0.0050                    | 0.0292        | 0.0168        | 0.0406        | 0.0289        | 0.0568        | 0.0486        | 0.0761        | 0.0782        | 0.1159        | 0.1643        | 0.1570        | 0.2718        |
| Online Sales Dataset (Cold Item Scenario) |                           |               |               |               |               |               |               |               |               |               |               |               |               |
| EMDE MM mout                              | 0.0839                    | 0.1136        | 0.1287        | <b>0.1236</b> | <b>0.1513</b> | <b>0.1338</b> | <b>0.1801</b> | <b>0.1442</b> | <b>0.2201</b> | <b>0.1559</b> | <b>0.2752</b> | <b>0.1648</b> | <b>0.3256</b> |
| EMDE                                      | 0.0774                    | 0.1021        | 0.1134        | 0.1107        | 0.1329        | 0.1197        | 0.1589        | 0.1259        | 0.1809        | 0.1309        | 0.2039        | 0.1351        | 0.2273        |
| RP3beta [34]                              | 0.0852                    | 0.1159        | <b>0.1295</b> | 0.1208        | 0.1400        | 0.1299        | 0.1657        | 0.1353        | 0.1850        | 0.1385        | 0.1988        | 0.1404        | 0.2082        |
| EASE [39]                                 | 0.0833                    | <b>0.1141</b> | 0.1278        | 0.1219        | 0.1447        | 0.1279        | 0.1618        | 0.1351        | 0.1887        | 0.1401        | 0.2107        | 0.1432        | 0.2277        |
| ItemKNN CF [9]                            | <b>0.0866</b>             | <b>0.1141</b> | 0.1266        | 0.1208        | 0.1415        | 0.1286        | 0.1631        | 0.1349        | 0.1854        | 0.1389        | 0.2021        | 0.1403        | 0.2093        |
| P3alpha [7]                               | 0.0858                    | 0.1129        | 0.1251        | 0.1209        | 0.1425        | 0.1289        | 0.1648        | 0.1343        | 0.1841        | 0.1380        | 0.1998        | 0.1396        | 0.2072        |
| MacridVAE [29]                            | 0.0739                    | 0.1026        | 0.1171        | 0.1101        | 0.1335        | 0.1214        | 0.1659        | 0.1278        | 0.1891        | 0.1365        | 0.2295        | 0.1414        | 0.2568        |
| SLIMElasticNet [32]                       | 0.0811                    | 0.1088        | 0.1203        | 0.1162        | 0.1366        | 0.1212        | 0.1506        | 0.1266        | 0.1696        | 0.1296        | 0.1826        | 0.1313        | 0.1913        |
| PureSVD                                   | 0.0670                    | 0.0925        | 0.1042        | 0.0992        | 0.1195        | 0.1062        | 0.1383        | 0.1137        | 0.1652        | 0.1184        | 0.1873        | 0.1232        | 0.2144        |
| TopPop [9]                                | 0.0130                    | 0.0183        | 0.0209        | 0.0233        | 0.0330        | 0.0254        | 0.0395        | 0.0285        | 0.0509        | 0.0366        | 0.0905        | 0.0435        | 0.1302        |
| MultiVAE [25]                             | <i>could not converge</i> |               |               |               |               |               |               |               |               |               |               |               |               |

### 4.3 EMDE as a Multimodal Output Recommender System for cold item prediction

In the above experiments recommender output consisted of only one sketch with both structure and density based on user-item interactions. Nonetheless, we can utilize a concatenation of multi-modal sketches on the output. This approach permits additional functionalities, such as solving the problem of cold items. Such items can be few in mature datasets collected over a longer timespan (MovieLens20M only has 612 of them for 20,720 total items), so dataset manipulations to artificially sample cold items are necessary [13]. In practice however cold items can be numerous, especially in newly opened stores or quickly changing item stock. There are works which explicitly focus on the cold item scenario [2] [44], however we treat this functionality as a nice added benefit to other advantages of EMDE.

**Online Sales Dataset.** To truthfully reflect the scale of the problem in the real world top-k setting, we collect user-item interactions from a single store in our production system for one month. After filtering out users with less than 2 interactions, we obtain 12491 purchased items and 12958 users from which we sample 1500 users for both valid and test set. For the 3000 non-trainable users, we obtain 1675 cold items, which means that on average, 56% of users have bought a cold item. Additionally, as the second input/output modality, for each item we collect product titles, which are represented as a bag-of-words composed of hashed tokens. Whenever an item cannot be retrieved from the regular interactions sketch output, given no interactions in the training set, its score is retrieved from just the item title sketch. Non-cold items have their scores averaged from both interaction and title sketches. Our results show that this way many relevant cold-start items can be retrieved by multioutput EMDE (EMDE MM mout) (Table 2). Contrary to previous tendencies, here we obtain significant gains especially for large  $k$ . This is because densities in the titles sketch tend to be more uniform (i.e. lower for top results). We boost the scores of cold items by multiplying them by a constant of 1.3 estimated empirically on the validation set. Depending on practical need, the scores can be artificially boosted even further to suggest more recently added items.

Simple baselines generally outperform other neural methods on this dataset, probably because of the small size of this dataset which makes complex models hard to optimize. Especially MultVAE proved hard to converge, as the obtained results were close to 0 for all metrics irrespective of the parameter configuration.

### 4.4 Ablation studies

In order to understand the effects of crucial parameters for training and decoding of EMDE, we conduct additional experiments in session-based scenario. We run experiments on RETAIL dataset, treating the best configuration for this dataset as our baseline. We report the results for MRR@20 and P@20 in Table 3. Results for other metrics are available in the Appendix.

**Manifold partitioning.** In addition to our density-dependent LSH variant (DLSH) we verify the impact of partitioning the manifold with product quantization methods (PQ) which decompose high-dimensional space into the Cartesian product of low-dimensional subspaces which are quantized separately. We also analyze its enhanced and optimized version (OPQ) [14]. We can see that DLSH is a strong baseline, leading at MRR@20 in comparison to other coders. However, we note that OPQ achieves competitive results, which indicates potential for improvement in density-based manifold partitioning methods.

**Score ensembling.** We perform point estimates from the output sketch by ensembling independent probability mass functions across sketch depth using geometric mean. While arguments behind the choice of geometric mean for ensembling probability measures can be found in [12] and [21], we empirically confirm the choice, comparing with: minimum, arithmetic mean and harmonic mean.

**Sketch width and depth.** Considering the fact that we ensemble densities from all sketch regions depth-wise, the higher the *depth*, the lower is the variance of final values. Sketch *width* is responsible for the amount of regions the manifold is split into. Holding  $depth \times width$  constant, we investigate the trade-off between them. Obtained results show that *width* of 128 achieves best results. In our experiments with other datasets, *width* of 128 seems to be a universally good value.

**Choice of manifold** We investigate how the choice of input embedding manifold influences performance of downstream recommendation tasks. We compare our multi-modal item sketches with uni-modal sketches built on item-user interaction embeddings, item attribute metadata embeddings,



Table 3: Ablation study results. Bolded headers indicate parameters used in experiments.

| Metrics | Partitioning method |         |         | Ensembling method |         |         |         | depth x width |         |         | Input embedding    |              |          |         |
|---------|---------------------|---------|---------|-------------------|---------|---------|---------|---------------|---------|---------|--------------------|--------------|----------|---------|
|         | <b>DLSH</b>         | OPQ     | PQ      | <b>gmean</b>      | min     | mean    | hmean   | <b>10x128</b> | 5x256   | 20x64   | <b>multi-modal</b> | interactions | metadata | random  |
| P@20    | 0.05564             | 0.05708 | 0.0519  | 0.05564           | 0.04634 | 0.05254 | 0.05172 | 0.05564       | 0.05522 | 0.05565 | 0.0556             | 0.05067      | 0.05144  | 0.03044 |
| MRR@20  | 0.36493             | 0.35937 | 0.35138 | 0.36493           | 0.31195 | 0.35922 | 0.34112 | 0.36493       | 0.35517 | 0.35892 | 0.3649             | 0.35019      | 0.32813  | 0.27865 |

and random item embeddings without a metric prior. All sketches have the same dimensions for a fair comparison. Not only random item vectors have the lowest performance due to the lack of a metric prior, but the manifolds for item interaction similarity and item attribute similarity confer different benefits, outperforming each other in MRR@20 and P@20 respectively. The multi-modal sketch is a clear winner in both metrics.

## 5 Conclusions

We present an efficient multimodal density estimator on Riemannian manifolds, which achieves state-of-the-art results in recommendation system setting. EMDE is computationally efficient, scalable and allows to efficiently encode multisets. In contrast to other recommendation systems, we can use one algorithm for session-based and top-k recommendations. Although we focused on recommendations here, efficiency and flexibility of EMDE indicate that it could be applied to other machine learning tasks, which we leave for future research. In the next steps we plan to investigate EMDE interpretability, a reformulation for scalar field estimation and applications to cross-modality recommendation.

## 6 Broader Impact

We believe that our work has a generally positive overall impact. More accurate recommendations will allow consumers to receive suggestions more to their liking, reducing their time browsing shop inventories. It is known that exposition to too many irrelevant options leads to the *paradox of choice* – an action paralysis and poor final choice, followed by dissatisfaction [33]. With growing sizes of shop inventories, it is important to shield customers from this negative psychological effect. Greater customer exhibition to relevant items will also mean higher turnover in retail.

A negative effect might be that more shops will be forced to invest in appropriate hardware for running the more elaborate recommender systems. However, we believe that shops will be willing to invest as the overall effort of creating valid recommendations will be lowered by the use of an accurate digital recommender.

## References

- [1] Rakesh Agrawal, Tomasz Imielinski, and Arun Swami. Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, page 207–216, New York, NY, USA, 1993. Association for Computing Machinery.
- [2] Oren Barkan, Noam Koenigstein, Eylon Yogev, and Ori Katz. Cb2cf: A neural multiview content-to-collaborative filtering model for completely cold item recommendations. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 228–236, New York, NY, USA, 2019. Association for Computing Machinery.
- [3] James Bennett, Stan Lanning, and Netflix Netflix. The netflix prize. In *In KDD Cup and Workshop in conjunction with KDD*, 2007.
- [4] M. Charikar and P. Siminelakis. Hashing-based-estimators for kernel density in high dimensions. In *2017 IEEE 58th Annual Symposium on Foundations of Computer Science (FOCS)*, pages 1032–1043, 2017.
- [5] Benjamin Coleman and Anshumali Shrivastava. Sub-linear race sketches for approximate kernel density estimation on streaming data. pages 1739–1749, 04 2020.
- [6] Benjamin Coleman, Anshumali Shrivastava, and Richard G. Baraniuk. Race: Sub-linear memory sketches for approximate near-neighbor search on streaming data, 2019.

- [7] Colin Cooper, Sang Lee, Tomasz Radzik, and Yiannis Siantos. Random walks in recommender systems: exact computation and simulations. pages 811–816, 04 2014.
- [8] Graham Cormode and S. Muthukrishnan. An improved data stream summary: The count-min sketch and its applications. In Martín Farach-Colton, editor, *LATIN 2004: Theoretical Informatics*, pages 29–38, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- [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*, RecSys '19, page 101–109, New York, NY, USA, 2019. Association for Computing Machinery.
- [10] Gabriel de Souza Pereira Moreira, Felipe Ferreira, and Adilson Marques da Cunha. News session-based recommendations using deep neural networks. In *Proceedings of the 3rd Workshop on Deep Learning for Recommender Systems*, DLRS 2018, page 15–23, New York, NY, USA, 2018. Association for Computing Machinery.
- [11] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [12] Pierre Dognin, Igor Melnyk, Youssef Mroueh, Jerret Ross, Cicero Dos Santos, and Tom Sercu. Wasserstein barycenter model ensembling, 2019.
- [13] Evgeny Frolov and Ivan Oseledets. Hybridsvd: When collaborative information is not enough. In *Proceedings of the 13th ACM Conference on Recommender Systems*, RecSys '19, page 331–339, New York, NY, USA, 2019. Association for Computing Machinery.
- [14] T. Ge, K. He, Q. Ke, and J. Sun. Optimized product quantization for approximate nearest neighbor search. In *2013 IEEE Conference on Computer Vision and Pattern Recognition*, pages 2946–2953, 2013.
- [15] Ruining He and Julian McAuley. Vbpr: Visual bayesian personalized ranking from implicit feedback. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 144–150. AAAI Press, 2016.
- [16] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks. In *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, 2016.
- [17] Balázs Hidasi, Massimo Quadrana, Alexandros Karatzoglou, and Domonkos Tikk. Parallel recurrent neural network architectures for feature-rich session-based recommendations. In *Proceedings of the 10th ACM Conference on Recommender Systems*, RecSys '16, page 241–248, New York, NY, USA, 2016. Association for Computing Machinery.
- [18] Devon Hjelm, Alex Fedorov, Samuel Lavoie-Marchildon, Karan Grewal, Philip Bachman, Adam Trischler, and Yoshua Bengio. Learning deep representations by mutual information estimation and maximization. In *ICLR 2019*. ICLR, April 2019.
- [19] Jin Huang, Wayne Xin Zhao, Hongjian Dou, Ji-Rong Wen, and Edward Y. Chang. Improving sequential recommendation with knowledge-enhanced memory networks. In *The 41st International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '18, page 505–514, New York, NY, USA, 2018. Association for Computing Machinery.
- [20] Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. *Conference Proceedings of the Annual ACM Symposium on Theory of Computing*, 604-613, 10 2000.
- [21] Mitsuhiro Itoh and Hiroyasu Satoh. Geometric mean of probability measures and geodesics of fisher information metric, 2017.
- [22] Iman Kamehkhosh, Dietmar Jannach, and Malte Ludewig. A comparison of frequent pattern techniques and a deep learning method for session-based recommendation. In *RecTemp@RecSys*, 2017.

- [23] Lingpeng Kong, Cyprien de Masson d’Autume, Lei Yu, Wang Ling, Zihang Dai, and Dani Yogatama. A mutual information maximization perspective of language representation learning. In *International Conference on Learning Representations*, 2020.
- [24] Jing Li, Pengjie Ren, Zhumin Chen, Zhaochun Ren, Tao Lian, and Jun Ma. Neural attentive session-based recommendation. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management, CIKM ’17*, page 1419–1428, New York, NY, USA, 2017. Association for Computing Machinery.
- [25] 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, WWW ’18*, page 689–698, Republic and Canton of Geneva, CHE, 2018. International World Wide Web Conferences Steering Committee.
- [26] Malte Ludewig and Dietmar Jannach. Evaluation of session-based recommendation algorithms. *User Modeling and User-Adapted Interaction*, 28(4–5):331–390, December 2018.
- [27] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems*, pages 462–466, 2019.
- [28] Malte Ludewig, Noemi Mauro, Sara Latifi, and Dietmar Jannach. Performance comparison of neural and non-neural approaches to session-based recommendation. In *Proceedings of the 13th ACM Conference on Recommender Systems, RecSys ’19*, page 462–466, New York, NY, USA, 2019. Association for Computing Machinery.
- [29] Jianxin Ma, Chang Zhou, Peng Cui, Hongxia Yang, and Wenwu Zhu. Learning disentangled representations for recommendation. In *NeurIPS*, 2019.
- [30] Fei Mi and Boi Faltings. Context tree for adaptive session-based recommendation. *CoRR*, abs/1806.03733, 2018.
- [31] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.
- [32] Xia Ning and George Karypis. Slim: Sparse linear methods for top-n recommender systems. In *Proceedings of the 2011 IEEE 11th International Conference on Data Mining, ICDM ’11*, page 497–506, USA, 2011. IEEE Computer Society.
- [33] Antti Oulasvirta, Janne P. Hukkinen, and Barry Schwartz. When more is less: The paradox of choice in search engine use. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR ’09*, page 516–523, New York, NY, USA, 2009. Association for Computing Machinery.
- [34] Bibek Paudel, Fabian Christoffel, Chris Newell, and Abraham Bernstein. Updatable, accurate, diverse, and scalable recommendations for interactive applications. *ACM Trans. Interact. Intell. Syst.*, 7(1), December 2016.
- [35] Asmita Poddar, Eva Zangerle, and yi-hsuan Yang. nowplaying-rs: A new benchmark dataset for building context-aware music recommender systems. 07 2018.
- [36] Rodrigo Quian, Alexander Kraskov, Christof Koch, and Itzhak Fried. Explicit encoding of multimodal percepts by single neurons in the human brain. *Current biology : CB*, 19:1308–13, 08 2009.
- [37] Rodrigo Quian, Leila Reddy, Gabriel Kreiman, Christof Koch, and Itzhak Fried. Invariant visual representation by single neurons in the human brain. *Nature*, 435:1102–7, 07 2005.
- [38] Paris Siminelakis, Kexin Rong, Peter Bailis, Moses Charikar, and Philip Levis. Rehashing kernel evaluation in high dimensions. In *International Conference on Machine Learning*, pages 5789–5798, 2019.
- [39] Harald Steck. Embarrassingly shallow autoencoders for sparse data. In *The World Wide Web Conference, WWW ’19*, page 3251–3257, New York, NY, USA, 2019. Association for Computing Machinery.
- [40] Michael Tschannen, Josip Djolonga, Paul K. Rubenstein, Sylvain Gelly, and Mario Lucic. On mutual information maximization for representation learning. In *International Conference on Learning Representations*, 2020.

- [41] Roberto Turrin, Massimo Quadrana, Andrea Condorelli, Roberto Pagano, and Paolo Cremonesi. 30music listening and playlists dataset. In *RecSys Posters*, 2015.
- [42] Aaron van den Oord, Sander Dieleman, and Benjamin Schrauwen. Deep content-based music recommendation. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2643–2651. Curran Associates, Inc., 2013.
- [43] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.
- [44] Manasi Vartak, Arvind Thiagarajan, Conrado Miranda, Jeshua Bratman, and Hugo Larochelle. A meta-learning perspective on cold-start recommendations for items. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 6904–6914. Curran Associates, Inc., 2017.
- [45] Hao Wang, Naiyan Wang, and Dit-Yan Yeung. Collaborative deep learning for recommender systems. In *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '15, page 1235–1244, New York, NY, USA, 2015. Association for Computing Machinery.
- [46] Shu Wu, Yuyuan Tang, Yanqiao Zhu, Liang Wang, Xing Xie, and Tieniu Tan. Session-based recommendation with graph neural networks. In *AAAI*, 2018.
- [47] Fajie Yuan, Alexandros Karatzoglou, Ioannis Arapakis, Joemon M. Jose, and Xiangnan He. A simple but hard-to-beat baseline for session-based recommendations. *CoRR*, abs/1808.05163, 2018.
- [48] Fuzheng Zhang, Nicholas Jing Yuan, Defu Lian, Xing Xie, and Wei-Ying Ma. Collaborative knowledge base embedding for recommender systems. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, page 353–362, New York, NY, USA, 2016. Association for Computing Machinery.