Holographic Factorization Machines for Recommendation

Yi Tay,[1*] Shuai Zhang,[2*] Anh Tuan Luu,[3]
Siu Cheung Hui,[1] Lina Yao,[2] Tran Dang Quang Vinh[1]
[1]Nanyang Technological University, Singapore
[2]University of New South Wales, Australia
[3]Institute for Infocomm Research, Singapore
ytay017@e.ntu.edu.sg, shuai.zhang@unsw.edu.au

Abstract

Factorization Machines (FMs) are a class of popular algorithms that have been widely adopted for collaborative filtering and recommendation tasks. FMs are characterized by its usage of the inner product of factorized parameters to model pairwise feature interactions, making it highly expressive and powerful. This paper proposes Holographic Factorization Machines (HFM), a new novel method of enhancing the representation capability of FMs without increasing its parameter size. Our approach replaces the inner product in FMs with holographic reduced representations (HRRs), which are theoretically motivated by associative retrieval and compressed outer products. Empirically, we found that this leads to consistent improvements over vanilla FMs by up to 4% improvement in terms of mean squared error, with improvements larger at smaller parameterization. Additionally, we propose a neural adaptation of HFM which enhances its capability to handle nonlinear structures. We conduct extensive experiments on **nine** publicly available datasets for collaborative filtering with explicit feedback. HFM achieves state-of-the-art performance on all **nine**, outperforming strong competitors such as Attentional Factorization Machines (AFM) and Neural Matrix Factorization (NeuMF).

Introduction

In an era of information overload and content overdrive, consumers naturally suffer from overchoice. After all, there are easily a million songs, a thousand videos and hundreds of restaurants to choose from at a given time. This is the exact problem that recommender systems are designed for - making lives easier, by automatically providing and recommending the best choices to users. At the intersection of information retrieval and user profiling, collaborative filtering (CF) algorithms (Goldberg et al. 1992) are highly popular and effective recommendation algorithms. The core intuition behind CF is that it tries to predict the preference of a given user by gathering preferences from other users.

Across the rich history of CF research, techniques based on matrix factorization (MF) (Mnih and Salakhutdinov 2008) were highly dominant. The key idea behind MF is to factorize a user-item interaction matrix, learning latent patterns of user behavior and approximating and completing the missing values. As such, a rating score can be approximated for an item a user has never seen or used.

Factorization Machines (FM) (Rendle 2010) were later proposed, combining the key ideas of factorization models (e.g., MF, SVD) with general purpose machine learning techniques such as Support Vector Machines (SVMs) (Steinwart and Christmann 2008). The key idea between FMs is to model pairwise feature interaction using the inner product of two vectors (factorized parameters). FMs have demonstrated widespread success both in general machine learning tasks, collaborative filtering and recommendation tasks (Rendle 2012; 2010; Zheng, Noroozi, and Yu 2017; Tay, Luu, and Hui 2018; Pasricha and McAuley 2018).

At its core, an FM model comprises a collection of embedding vectors $\{v_1, v_2 \dots v_n\} \in \mathbb{R}^k$ in which the inner product between v_i and v_j is used to approximate feature interaction (x_i, x_j). Unfortunately, the usage of solely inner products may be sub-optimal for FMs as inner products only consider element-wise interactions between (v_i, v_j). Moreover, FMs sum over a series of inner products, which may result in further information loss.

This paper presents an improved memory-enhanced adaptation of factorization machines. More specifically, we enhance FMs with holographic reduced representations (HRRs) (Plate 1995), replacing the inner products with HRR operators such as circular convolution and circular correlation. There are several key benefits to investigating such an architecture:

- Operations such as circular convolution behave as *compressed* outer products. Hence, our proposed model acts as an outer product adaptation of FMs without actually incurring the parameter cost of the standard outer product. Unlike inner products which are element-wise operations, outer products model pairwise interactions between parameters. Hence our proposed FM can be regarded as a *tensorized* factorization machine, albeit compressed.
- Aside from enriched representation capability, the internal memory of the FM additionally acts as an associative memory array. Each parameter pair (v_i, v_j) now acts as a key-value pair in this addressible distributed memory. As such, each input feature vector is modeled with a *memory trace* and individual feature interactions can be efficiently retrieved by an associative retrieval mech-

*Equal contribution.

anism which takes place during gradient-based optimization. This is unlike the standard FM that sums over factorized interactions. We provide more details in subsequent sections.

Overall, we hypothesize that augmenting FMs with HRR can lead to improvement in performance. Notably, our work is inspired by the successful application of HRRs to standard connectionist methods, giving rise to architectures such as the Associative Long Short-Term Memory (Danihelka et al. 2016), Holographic Recurrent Networks (Plate 1992) and Holographic Embeddings (Nickel, Rosasco, and Poggio 2016).

Our Contributions

The overall contributions of our paper are summarized as follows:

- We propose a memory-enhanced factorization machine - the Holographic Factorization Machine (HFM) for collaborative filtering. We show that a relative performance improvement of $1\% - 4\%$ in terms of mean squared error over the vanilla FM model on nine benchmark datasets. Performance improvement is also robust when varying hyperparameter settings.
- We propose a further neural extension of HFM - HFM+, which imbues the HFM with the ability to handle nonlinearity using fully-connected layers.
- HFM and HFM+ achieve extremely competitive performance on nine benchmark datasets. HFM+ achieves state-of-the-art, outperforming recently proposed models such as NeuMF (Neural Matrix Factorization) and AFM (Attentional Factorization Machines).

Background

In this section, we discuss the relevant background that forms the basis of our work.

Factorization Machines (FM)

Factorization machines (Rendle 2010) are general machine learning methods which are commonly applied for recommendation tasks due to its strength at modeling sparse categorical data. The FM operates based on the following equation:

$$F(x) = w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \langle v_i, v_j \rangle \, x_i \, x_j \qquad (1)$$

where $x \in \mathbb{R}^k$ is a real-valued input feature vector. $\langle ., . \rangle$ is the dot product. The parameters $\{v_1 \dots v_n\}$ are factorized parameters (vectors of $v \in \mathbb{R}^k$) used to model pairwise interactions (x_i, x_j). w_0 is the global bias and $\sum_{i=1}^{n} w_i x_i$ represents a linear regression component. The output of $F(x)$ is a scalar, representing the strength of the user-item interaction.

Holographic Reduced Representation (HRR)

HRR was originally proposed by (Plate 1995) as a distributed form of associative memory. The key idea is a series of encoding and decoding operations that are used to emulate storage and retrieval in holography. First, we introduce the key operators in HRRs shown as follows:

$$[a \circledast b]_k = \sum_{i=0}^{d-1} a_i \, b_{(k-i) \bmod d} \qquad (2)$$

$$[a \star b]_k = \sum_{i=0}^{d-1} a_i \, b_{(k+i) \bmod d} \qquad (3)$$

where $\star : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the circular correlation operator and $* : \mathbb{R}^d \times \mathbb{R}^d \rightarrow \mathbb{R}^d$ denotes the circular convolution operator. For notational convenience, we use zero-indexed vectors. We refer to circular convolution as CCOV and circular correlation as CCOR for the remainder of the paper.

Associative Memory CCOV and CCOR are inverse operators and act as encoding-decoding pairs in associative memory. Let a and b be real-valued vectors. In HRR, CCOV $(\circledast)$ is used to associate two vectors to form a memory trace m. Subsequently, a nice property is that we are able to retrieve a noisy version of b by decoding a from m using CCOR, the decoding operator.

$$m = a \circledast b \qquad (4)$$

$$a \star m \approx b + n \qquad (5)$$

where n is a noise term and $\star$ is the CCOR operator. Generally, this is known as associative retrieval. Next, we formally introduce the associative memory operators. A memory trace can be extended in form of a *trace composition*:

$$m = a_1 \circledast b_1 + a_2 \circledast b_2 + a_3 \circledast b_3 \qquad (6)$$

where addition is the trace composition operator. Similarly, decoding $a_1 \star m$ returns a noisy version of b_1. As such, individual elements can be retrieved even under the additive composition (summation) of several constituent memory traces (Plate 1995; Danihelka et al. 2016).

Computation of HRR There are several ways to compute HRR operations. The most naive way is to literally compress the outer product. However, this incurs an undesirable complexity of n^2. Fortunately, we are able to exploit computation in the frequency domain, exploiting Fast Fourier Transforms (FFT) for computation with a log-linear runtime.

$$a \circledast b = \mathfrak{F}^{-1}(\mathfrak{F}(a) \odot \mathfrak{F}(b)) \qquad (7)$$

$$a \star b = \mathfrak{F}^{-1}(\overline{\mathfrak{F}(a)} \odot \mathfrak{F}(b)) \qquad (8)$$

where $\mathfrak{F}$ and $\mathfrak{F}^{-1}$ are the FFT and inverse FFT operations respectively. $\odot$ is the Hadamard product. Alternatively, complex-valued representations can be used to achieve a similar effect. (Danihelka et al. 2016) uses complex-valued vectors (complex inner product, i.e., $\bar{a}^\top b$ where $a, b \in \mathbb{C}^n$) as their encoding operation. Following (Nickel, Rosasco, and Poggio 2016), we use the real-valued FFT and extract the real components from its output.

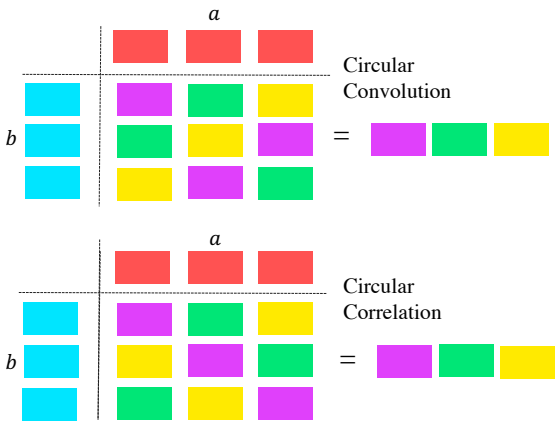


Figure 1: HRRs as Compressed Outer Products (*Best viewed in Color*). The outer product of vectors a and b of d-dimensions are compressed into a d dimensional vector. Colours denote compression by summation patterns.

Compressed Outer Products We show that CCOV and CCOR are equivalent to compressing an outer product. Unlike inner products, outer product forms a similarity matrix between two vectors. However, outer products can be undesirable as they can be cumbersome in the sense that they convert vectors to matrices. CCOV and CCOR do not have this restriction, i.e., the compressed outer product of two vectors is still a vector. This is an extremely attractive property that we exploit. By inspecting Equation (3), it is clear that CCOV and CCOR are compressing the outer product into a vector representation. Figure 1 depicts this phenomena.

Holographic Factorization Machines (HFM)

In this section, we describe HFM, our proposed architecture.

Modeling Pairwise Interactions with HRR

The main contribution in our HFM model is to replace the inner products in FMs with HRRs (Holographic Reduced Representations). More specifically, we replace the inner product in FMs with $(v_i \circledast v_j)$ as follows:

$$F(x) = L(x) + h^\top (\sum_{i=1}^{n} \sum_{j=i+1}^{n} (v_i \circledast v_j)\, x_i\, x_j) \qquad (9)$$

where $L(x) = w_0 + \sum_{i=1}^{n} w_i x_i$ represents the linear regression part of the FM formulation. $\circledast$ is the circular convolution operation. $h \in \mathbb{R}^n$ and $\{v_1, \dots v_n\}$ are parameters of the HFM layer (refered to as a parameter store in Figure 2). Alternatively, we may also use the inverse operator, $\star$ - the circular correlation operator:

$$F(x) = L(x) + h^\top (\sum_{i=1}^{n} \sum_{j=i+1}^{n} (v_i \star v_j)\, x_i\, x_j) \qquad (10)$$

which uses the conjugate transpose of $\mathfrak{F}(v_i)$ instead. Finally, the entire summation over memory traces is then be reduced to a scalar by h.

Associative Key-Value Memory In this section, we draw connections between HFM and associative memory models. The goal is to provide some intuition regarding how the associative retrieval mechanism works under the hood of our proposed model. Previously, we noted that FMs sum across pairwise interactions which inevitably results in information loss. In HFMs, pairwise interactions are actually retrievable. Recall the concepts of a trace composition mentioned earlier corresponds nicely to the summation operation in HFMs.

$$m = \alpha_{12}(v_1 \circledast v_2) + \alpha_{13}(v_1 \circledast v_3) + \cdots \qquad (11)$$

where α_{ij} is the interaction between features (x_i, x_j). This has two interpretations. Firstly, when calling $v_1 \star m$, we retrieve a combination of $v_2, v_3 \cdots v_n$ weighted by the feature interaction strength. Conversely, there is no way to retrieve individual feature interactions with standard FMs as information is lost during summation. In HFM, each parameter vector acts as a key-value pair to all other parameter vectors, forming a distributed key-value store.

Notably, the encoding-decoding operation of associative memories take effect during gradient-based optimization. This stems from the fact that the gradient of CCOV is its inverse operation - CCOR (Nickel, Rosasco, and Poggio 2016) (and vice versa). During the backward pass (gradient updates), the model learns relationships between parameter vectors, i.e., modeling the relationship of each user-item pair using $h^\top m$ (omitting $L(x)$ for simplicity). During the forward pass (inference), specific contributions of each interaction are retrieved by decoding m with $v_1, v_2 \cdots v_n$. As such, HFMs implement an associative retrieval mechanism within the FM parameters.

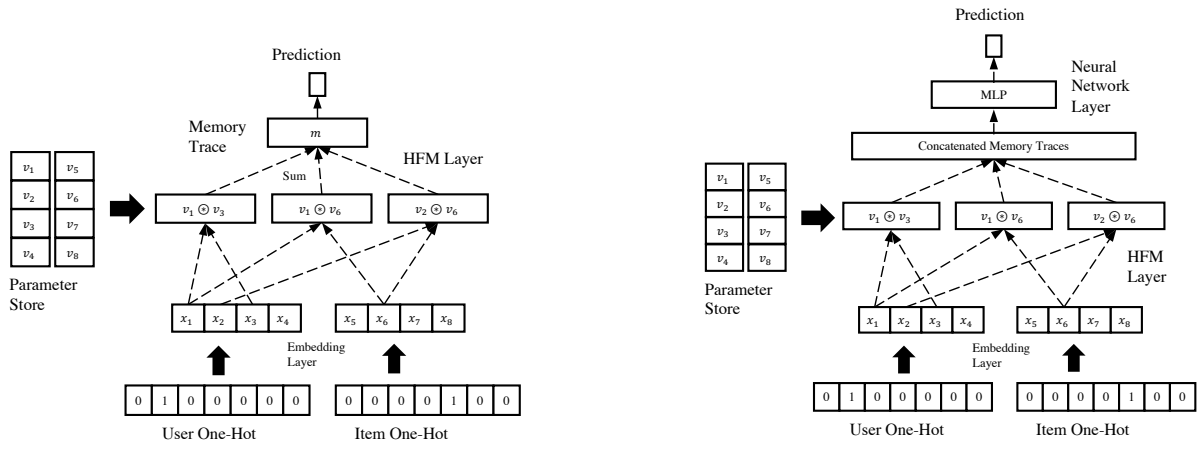
End-to-End Learning for Recommendation

In this section, we describe the overall model architecture of HFM for CF tasks. Each training instance of HFM accepts an interaction tuple (p, q, r). Each of the user p and item q are passed into the network as a sparse one-hot-encoded vector. r is a real-valued number from $[0, 1]$ which serves as the supervision signal for the model.

Embedding Layer This layer transforms the one-hot encoded representation into a dense real-valued representation (a.k.a embeddings). As such, this layer is parameterized by $W_p \in \mathbb{R}^{|P| \times k}$ and $W_q \in \mathbb{R}^{|Q| \times k}$ respectively. P is the set of all users and Q is the set of all items.

HFM Layer This layer is mainly used to model the interaction between user and item embeddings (or latent features). We concatenate the user and item embedding to form a feature vector of $2k$ dimensions. The HFM models pairwise interactions between each **latent** feature which have been described in earlier sections. The output of this layer is a scalar value, which represents the score of the user-item pair.

Optimization and Learning Our model is then trained end-to-end using stochastic gradient descent (in particular, Adam optimizer (Kingma and Ba 2014)), minimizing the binary cross-entropy loss. We scale the output between $[0, 1]$



Holographic Factorization Machines (HFM)

Nonlinear Holographic Factorization Machines (HFM+)

Figure 2: Proposed Model Architectures for Recommendation. HFM extends FM with Holographic reduced representations (HRR), exploiting rich compressed tensor products instead of inner products. Nonlinear HFMs (HFM) pass the concatenated memory trace into a MLP layer, enhancing its representation capability with nonlinear transformations.

with the sigmoid function σ.

$$J(\theta) = -y\,(\log(p)) + (1-y)\log(1-p) + \lambda||\theta||_{L2} \quad (12)$$

where y is the ground truth label, $p = \sigma(F(x))$ is the output of the model and $||\theta||_{L2}$ is the regularization term weighted by λ.

On the Formulation of the FM Function Finally, a notable difference from (Rendle 2010; 2012; Xiao et al. 2017) is that we use the concatenation of user and item embedding as input to the FM function, i.e., the FM model operates on latent dimensions. This follows DeepCoNN (Zheng, Noroozi, and Yu 2017) which was proposed for the review rating problem. This difference in choice is largely due to the difference in problem formulation. For FM and AFM (Xiao et al. 2017), the authors considered a sparse categorical regression problem using a myriad of categorical features. Naturally, the input to these models are sparse feature vectors in which the learned feature embeddings actually correspond to the FM parameters. In their case, the FM model actually reverts to the MF model when only considering user-item interaction (Rendle 2012). Hence, it is more appropriate for our work to follow (Zheng, Noroozi, and Yu 2017).

Nonlinear HFMs (HFM+)

One of the biggest weaknesses of FMs is that they are inherently linear models and may have problems on complex datasets. As such, we propose to enhance HFMs with nonlinear transformations which gives rise to HFM+, our second architecture. One of the biggest differences with NFM is that our approach applies multi-layered perceptron to the interaction matrix instead of a pooling layer. Recall that each pairwise interaction returns a vector of n dimensions. We flatten this interaction matrix and pass them into a FC layer. Let $S \in \mathbb{R}^{n \times \frac{(n-1)}{2}}$ be the interaction matrix and $s \in \mathbb{R}^{\frac{n(n-1)}{2}}$ be the flattened version of S. The output of the nonlinear (and second order) part of the HFM is now:

$$L(x) = W_3(\sigma_r(W_2(\sigma_r(W_1(s + b_1)) + b_2))) + b_3 \quad (13)$$

where W_*, b_* with $* = \{1, 2, 3\}$ are the parameters of this layer. The final output of HFM+ is now:

$$F(x) = w_0 + \sum_{i=1}^{n} w_i\, x_i + L(x) \quad (14)$$

where $L(x)$ is the nonlinear component of the HFM+ model. To the best of our knowledge, this formulation is novel in the sense that we unflatten and concatenate the entire memory trace.

Empirical Evaluation

In this section, we describe our experiments and report empirical results. First, we define the research questions (RQ) that our experiments are designed to answer.

- **RQ1** - Does HFM+ and HFM achieve state-of-the-art performance on CF benchmarks?
- **RQ2** - Does HRR improve the performance on FMs? What is the relative improvement on different settings?
- **RQ3** - What are the impacts of some key hyperparameters (e.g., number of latent dimensions) on performance.

Datasets

In this spirit of experimental rigor, we compare our method against others using **nine** publicly available benchmark datasets.

- **Netflix** is a popular dataset for explicit CF, popularized by the Netflix Prize competition. Netflix[1] is a video streaming website and is concerned with recommending movies/videos to user. Due to the large dataset size, we extract ratings from the year 2005. Nevertheless, the number of ratings is still 40 million.
- **MovieLens** is another popular benchmark for recommendation. Once again, this dataset[2] is concerned with movie ratings. We utilize three different sizes of this dataset which have 20M (MovieLens20M), 1M (MovieLens1M) and 100K (MovieLens100K) respectively where the suffix denotes the number of interations in the dataset.
- **IMDb** is another movie-based CF dataset constructed by (Diao et al. 2014). IMDb[3] is also a movie rating website and comprises user preference scores for movies. However, different from Netflix and MovieLens, the rating scale of this dataset is from $1 - 10$.
- **Amazon Product Reviews** is a review rating dataset[4] based on customer reviews on Amazon (McAuley et al. 2015; He and McAuley 2016). We use multiple versions of this dataset which have been split into product categories. More specifically, we use four categories - Gourmet Food, Kindle Store, CDs and Vinyl, and finally Beauty. Note that subsets were selected largely based on domain diversity and also dataset size.

All datasets were setup and filtered to a 20-core setting. The only exception is the Netflix dataset which we had to use a 100-core setting due to hardware limitations. Notably, the resulting Netflix dataset still contains over 40 million interactions which still poses a computational challenge for high end graphic cards. Table 1 reports the dataset statistics of all datasets (after filtering). For all datasets, we use a time-based split, i.e., we sort all of a user's items by timestamps and withhold the last two as the development and testing sets respectively.

Dataset	Ratings	Users	Items	Density
Netflix	44M	75K	13K	4.5
MovieLens20M	16M	53K	27K	1.1
MovieLens1M	1M	6K	4K	4.2
MovieLens100K	100K	1.6K	0.9K	6.0
IMDb	117K	0.8K	114K	0.1
Grocery Food	120K	3K	39K	0.1
Kindle Store	800K	15K	185K	0.03
CDs and Vinyl	933K	16K	290K	0.02
Beauty	92K	3K	42K	0.07

Table 1: Statistics of nine datasets adopted in our experimental evaluation. Density reports the number of interactions with respect to the total size of user/item matrix.

[1]https://www.netflix.com/browse.

[2]https://grouplens.org/datasets/movielens/

[3]https://www.imdb.com/.

[4]http://jmcauley.ucsd.edu/data/amazon/

Competitive Baselines

We compare with two standard baselines and two state-of-the-art models.

- **Matrix Factorization (MF)** is a popular standard baseline for CF. MF models each user and item pair using the inner product $p \odot q$.
- **Factorization Machines (FM)** is a strong CF baseline proposed in (Rendle 2010). It learns pairwise feature interactions using factorized parameters. Following (Zheng, Noroozi, and Yu 2017), we concatenate the user and item embedding as input into a standard FM model.
- **Attentional Factorization Machines (AFM)** is a state-of-the-art model proposed by Xiao et al. (Xiao et al. 2017). This model follows the implementation of our FM model. However, an attention mechanism is applied on top of FM enabling it to select the best and most informative pairwise features to be used for prediction.
- **Neural Matrix Factorization (NeuMF)** is a state-of-the-art model proposed in (He et al. 2017). It proposes a joint matrix factorization and neural network approach, achieving highly competitive performance on multiple recommendation benchmarks. We use an identical structure, i.e., combining the generalized MF and a three-layered pyramidal multi-layered perceptron (MLP). Note that there are dual embedding spaces for the MF/MLP model. Notably, under our problem formulation, the recent Neural Factorization Machines (He and Chua 2017) is subsumed by NeuMF.

Experimental Setup

The evaluation metric employed is the standard mean squared error (MSE). We implement all models in Tensorflow[5].The latent dimensions (embedding size) of all baselines are tuned[6] in the range of $\{4, 8, 16, 32\}$ since we found that for most datasets, performance does not increase beyond $k = 32$. The batch size is set to 1024 in all our experiments. All methods are optimized with Adam (Kingma and Ba 2014) with a learning rate of 0.0003 (varying the learning rate in the range of $[0.001, 0.0001]$ did not yield any improvements). All models are optimized with sigmoid cross entropy loss since we found that it was significantly more stable compared to minimizing the raw mean squared error. Therefore, rating values are scaled to $[0, 1]$ and then renormalized upon inference. A dropout of 0.2 is applied to all feed-forward layers. For all FM based models, the number of latent factors is tuned amongst $\{4, 8\}$. We train each model for a maximum of 50 epochs and compute the score on the held-out set at every epoch. We apply early stopping, i.e., we stop training if performance on the held-out set does not improve after 5 epochs. We report the test scores on the model with the best score on the held-out set. We additionally tune between using circular correlation and circular convolution for our HFM model.

[5]https://www.tensorflow.org/.

[6]Due to hardware limitations we could not tune the latent dimensions on MovieLens20M and Netflix and set it to a standard $k = 16$ for all datasets.

RQ1 - Does HFM achieve state-of-the-art performance?

Table 2 reports an overall performance comparison of all compared models and baselines. Firstly, we note that HFM+ (and HFM) consistently achieves state-of-the-art performance, outperforming strong baselines on all nine benchmark datasets. Amongst the baselines, we find that AFM generally outperforms FM (6 out of 9 datasets). FM strongly outperforms MF on 5 out of 9 datasets, with a 200% improvement on the IMDB dataset. On the other 4 datasets, FM performs marginally worse. Amongst all competitors, NeuMF performs consistently well because of its ability to model nonlinearity. Notably, it also uses dual embedding spaces, which easily doubles its parameter size. On that note, while FM is strongly outperformed by NeuMF, HFM comes close in performance to NeuMF. Finally, HFM+ consistently outperforms NeuMF on all datasets.

RQ2 - Does HRR improve FMs?

Table 3 reports the relative performance improvement of our proposed HFM model and FM. We observe that, across **all** latent dimensions, our proposed HFMs are superior to FM. Given that HFMs are just HRR-enhanced FMs, this ascertains the effectiveness of using HRR in FMs. Aside from the robust improvement, the improvements over selected best models are also notable, ranging from $1\% - 4\%$ across seven datasets. One interesting observation is that performance improvements are relatively higher with smaller parameterization (e.g., IMDB, food and CDs). The least improvement often comes from $k = 16$ and higher performance gains are obtained at $k = 4$ or $k = 8$. We believe that this is because HRRs augment the representation capability of the FM model. When both models have large latent dimensions, the parameters of the FMs may be sufficient on certain datasets, reducing the benefits of using HRR. Lastly, we note that HFM+ outperforms FM on almost all dimensions and datasets - with a relative improvement of 12.5% on the netflix dataset. All in all, we are able to reasonably improve the performance of the base FM across all latent dimensions **without** incurring any additional parameter costs.

RQ3 - What are the impacts of Hyperparameters on model performance?

Figure 3 and Figure 4 report the effect of varying the latent dimensions k for all models on two datasets. The optimal dimensions are $k = 32$ (IMDB) and $k = 24$ (Kindle). On IMDB, we observe the performance of HFM is clearly superior to FM. On Kindle, the performance is significantly better at smaller k but converges to marginal improvement at higher k. Notably, the best performance of HFM ($k = 8$) on Kindle is much better than AFM. On the Amazon Kindle dataset, we also observe that HFM+ outperforms NeuMF on all latent dimensions.

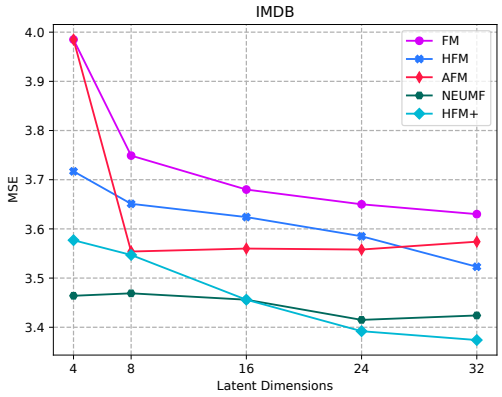


Figure 3: Effect of Latent Dimensions on Performance (IMDb dataset). HFM outperforms FM on all dimensions and HFM+ achieves the overall best performance.

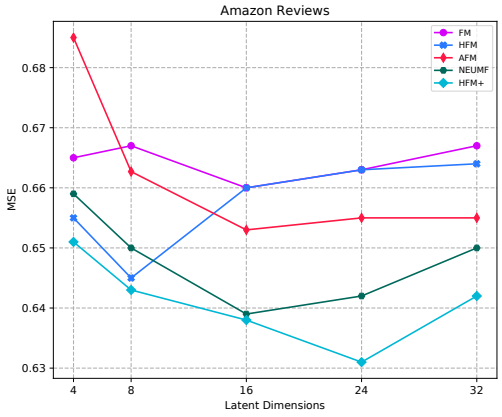


Figure 4: Effect of Latent Dimensions on Performance (Amazon Reviews Kindle). Across all dimensions, HFM is always better than FM and HFM+ is always better than NeuMF and all other baselines.

Related Work

The term Collaborative Filtering was first coined by (Goldberg et al. 1992), who proposed the first recommender system *Tapestry*. Many prior research in this field utilised a factorization-based approach to learn user-item associations. In particular, probabilistic matrix factorization (Mnih and Salakhutdinov 2008) and SVD (Koren 2008) were amongst the more popular CF algorithms. Factorization machines (Rendle 2010; 2012) were later proposed as a general-purpose machine learning model, i.e., regression and classification. Nevertheless, FMs also see wide adoption across a potpourri of recommendation tasks (Juan et al. 2016; Rendle et al. 2011; Zheng, Noroozi, and Yu 2017; Tay, Luu, and Hui 2018; Pasricha and McAuley 2018).

Today, deep learning based approaches have claimed state-of-the-art in many recommendation and CF tasks (Zhang et al. 2017; He et al. 2017; Tay, Luu, and Hui 2018; Zhang et al. 2018; Tay, Anh Tuan, and Hui 2018). (He et al. 2017) showed that the standard matrix factorization can be formulated as

Dataset	MF	FM	AFM	NEUMF	HFM	HFM+
Netflix	0.687	0.746	0.742	0.677	0.719	**0.663**
MovieLens20M	0.711	0.742	0.766	0.711	0.729	**0.708**
MovieLens1M	1.149	1.144	1.155	1.142	**1.134**	1.142
MovieLens100K	1.037	1.067	1.046	1.042	1.031	**1.030**
IMDb	7.131	3.545	3.544	3.424	3.429	**3.374**
Beauty	3.360	1.353	1.311	1.293	1.300	**1.274**
Cds and Vinyl	1.923	1.013	1.012	0.991	1.000	**0.979**
Grocery Food	2.828	1.222	1.198	1.194	1.208	**1.164**
Kindle Stoe	1.528	0.663	0.653	0.642	0.645	**0.631**

Table 2: Performance comparison (mean squared error) of all models on 9 benchmark datasets. Best result is in boldface. HFM outperforms FM and HFM+ outperforms all baselines.

Dataset	HFM vs FM				HFM+ vs FM			
	4	8	16	Best	4	8	16	Best
Netflix	-	-	+3.9%	+3.9%	-	-	+12.5%	+12.5%
MovieLens20M	-	-	+1.8%	+1.8%	-	-	+4.9%	+4.9%
MovieLens1M	+1.5%	+1.1%	+0.9%	+1.8%	+0.3%	+0.9%	+0.2%	+0.2%
MovieLens100K	+0.8%	+3.7%	+3.5%	+3.5%	+5.0%	+3.1%	+1.1%	+3.6%
IMDb	+7.2%	+2.7%	+1.5%	+3.4%	+11%	+8.1%	+6.5%	+5.1%
Beauty	+5.2%	+3.9%	+4.1%	+4.1%	+3.9%	+6.3%	+3.3%	+6.2%
CDs and Vinyl	+1.4%	+0.4%	+0.1%	+1.0%	+1.5%	+1.0%	+2.2%	+3.5%
Grocery Food	+1.2%	+1.0%	+0.7%	+1.2%	+3.2%	+2.5%	+2.2%	+5.0%
Kindle Store	+1.5%	+3.4%	+0.1%	+2.8%	+2.2%	+3.7%	+1.1%	+5.1%

Table 3: Relative Performance Improvement ($+\%$) against FM model across various latent dimension size k. 'Best' settings refer to the performance improvement over the best values of k for both models in Table 2. HFM provides a modest to large boost to the FM model on all benchmark datasets. Notably, HFM does not increase the parameter size of the FM model.

a network. The authors go on to propose Neural Matrix Factorization (NeuMF), a combined model that takes advantages of factorization models and nonlinear multi-layered perceptrons. FMs have also received neural makeovers. (He and Chua 2017) proposed enhancing standard FMs with nonlinear layers while (Xiao et al. 2017) equipped FMs with attention layers. This enables a more selective modeling of feature interactions with attractive properties such as avoiding overfitting. DeepFM (Guo et al. 2017), proposed for CTR prediction, combines the prediction scores of a deep neural network and FM model.

Owing to its effectiveness in capturing feature interactions, FMs have also been recently adopted in other variations of recommender tasks such as review-based or sequential recommender systems. DeepCoNN (Zheng, Noroozi, and Yu 2017) used a FM on top of user and item convolutional neural network (CNN) for review rating prediction. The recent Multi-Pointer Co-Attention Networks (Tay, Luu, and Hui 2018) similarly adopts a FM prediction layer. (Pasricha and McAuley 2018) proposed a translational sequential recommender model based on factorization machines.

Our work is strongly inspired by applications of HRR (Plate 1995) and associative memory models (Gabor 1969). Holographic recurrent networks (HRN) (Plate 1992) proposed using HRRs for recursive computation within the recurrent network cell. Associative LSTMs (Danihelka et al. 2016) proposed enhancing the memory of the LSTM cell using HRRs, albeit using complex-valued parameters. (Tay et al. 2017) proposed using HRRs for matching question answer pairs. (Nickel, Rosasco, and Poggio 2016) proposed HOLE, a knowledge graph embedding that exploits circular correlation for learning entity relationships. (Hayashi and Shimbo 2017) showed the equivalence of HRRs with complex-valued inner product, drawing parallels with HOLE and ComplEx, a complex-valued embedding model for link prediction (Trouillon et al. 2016).

Conclusion

We proposed Holographic Factorization Machines (HFM), a novel application of Holographic Reduced Representations (HRRs) to FMs. HFM achieves superior performance relative to standard FMs, owing to enhanced representation capacity due to compression of outer products and associative retrieval mechanisms. Additionally, we equip HFMs with the ability to handle nonlinear complexities (nonlinear HFM, i.e., HFM+), achieving state-of-the-art performance. Experimental results on nine diverse benchmark datasets demonstrate the effectiveness of HFM. Moreover, our ablative studies show that HFM also provides robust improvements to the base FM model across all latent dimensions and datasets without actually increasing the parameter size of the model.

References

Danihelka, I.; Wayne, G.; Uria, B.; Kalchbrenner, N.; and Graves, A. 2016. Associative long short-term memory. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, 1986–1994.

Diao, Q.; Qiu, M.; Wu, C.; Smola, A. J.; Jiang, J.; and Wang, C. 2014. Jointly modeling aspects, ratings and sentiments for movie recommendation (JMARS). In *The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014*, 193–202.

Gabor, D. 1969. Associative holographic memories. *IBM J. Res. Dev.* 13(2):156–159.

Goldberg, D.; Nichols, D.; Oki, B. M.; and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35(12):61–70.

Guo, H.; Tang, R.; Ye, Y.; Li, Z.; and He, X. 2017. Deepfm: A factorization-machine based neural network for ctr prediction. *arXiv preprint arXiv:1703.04247*.

Hayashi, K., and Shimbo, M. 2017. On the equivalence of holographic and complex embeddings for link prediction. *arXiv preprint arXiv:1702.05563*.

He, X., and Chua, T.-S. 2017. Neural factorization machines for sparse predictive analytics.

He, R., and McAuley, J. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *WWW*, 507–517. International World Wide Web Conferences Steering Committee.

He, X.; Liao, L.; Zhang, H.; Nie, L.; Hu, X.; and Chua, T.-S. 2017. Neural collaborative filtering. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, 173–182. Republic and Canton of Geneva, Switzerland: International World Wide Web Conferences Steering Committee.

Juan, Y.; Zhuang, Y.; Chin, W.-S.; and Lin, C.-J. 2016. Field-aware factorization machines for ctr prediction. In *Recsys*, 43–50. ACM.

Kingma, D. P., and Ba, J. 2014. Adam: A method for stochastic optimization. *CoRR* abs/1412.6980.

Koren, Y. 2008. Factorization meets the neighborhood: a multifaceted collaborative filtering model. In *SIGKDD*, 426–434. ACM.

McAuley, J.; Targett, C.; Shi, Q.; and Van Den Hengel, A. 2015. Image-based recommendations on styles and substitutes. In *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 43–52. ACM.

Mnih, A., and Salakhutdinov, R. R. 2008. Probabilistic matrix factorization. In *NIPS*, 1257–1264.

Nickel, M.; Rosasco, L.; and Poggio, T. A. 2016. Holographic embeddings of knowledge graphs. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA.*, 1955–1961.

Pasricha, R., and McAuley, J. 2018. Translation-based factorization machines for sequential recommendation. In *RecSys*.

Plate, T. 1992. Holographic recurrent networks. In *Advances in Neural Information Processing Systems 5, [NIPS Conference, Denver, Colorado, USA, November 30 - December 3, 1992]*, 34–41.

Plate, T. A. 1995. Holographic reduced representations. *IEEE Trans. Neural Networks* 6(3):623–641.

Rendle, S.; Gantner, Z.; Freudenthaler, C.; and Schmidt-Thieme, L. 2011. Fast context-aware recommendations with factorization machines. In *SIGIR*, 635–644. ACM.

Rendle, S. 2010. Factorization machines. In *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, 995–1000.

Rendle, S. 2012. Factorization machines with libfm. *ACM Trans. Intell. Syst. Technol.* 3(3):57:1–57:22.

Steinwart, I., and Christmann, A. 2008. *Support vector machines*. Springer Science & Business Media.

Tay, Y.; Anh Tuan, L.; and Hui, S. C. 2018. Latent relational metric learning via memory-based attention for collaborative ranking. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, 729–739. International World Wide Web Conferences Steering Committee.

Tay, Y.; Phan, M. C.; Tuan, L. A.; and Hui, S. C. 2017. Learning to rank question answer pairs with holographic dual lstm architecture. In *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '17, 695–704. New York, NY, USA: ACM.

Tay, Y.; Luu, A. T.; and Hui, S. C. 2018. Multi-pointer co-attention networks for recommendation. In *SIGKDD*, KDD '18, 2309–2318. New York, NY, USA: ACM.

Trouillon, T.; Welbl, J.; Riedel, S.; Gaussier, É.; and Bouchard, G. 2016. Complex embeddings for simple link prediction. In *ICML*, 2071–2080.

Xiao, J.; Ye, H.; He, X.; Zhang, H.; Wu, F.; and Chua, T. 2017. Attentional factorization machines: Learning the weight of feature interactions via attention networks. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017, Melbourne, Australia, August 19-25, 2017*, 3119–3125.

Zhang, S.; Yao, L.; Sun, A.; and Tay, Y. 2017. Deep learning based recommender system: A survey and new perspectives. *arXiv preprint arXiv:1707.07435*.

Zhang, S.; Yao, L.; Sun, A.; Wang, S.; Long, G.; and Dong, M. 2018. Neurec: On nonlinear transformation for personalized ranking. In *IJCAI*, 3669–3675.

Zheng, L.; Noroozi, V.; and Yu, P. S. 2017. Joint deep modeling of users and items using reviews for recommendation. In *WSDM*, 425–434.