Rethinking Neural vs. Matrix-Factorization Collaborative Filtering: the Theoretical Perspectives

Da Xu [1] Chuanwei Ruan [2] Evren Korpeoglu [1] Sushant Kumar [1] Kannan Achan [1]

Abstract

The recent work by Rendle et al. (2020), based on empirical observations, argues that matrix-factorization collaborative filtering (MCF) compares favorably to neural collaborative filtering (NCF), and conjectures the dot product's superiority over the feed-forward neural network as similarity function. In this paper, we address the comparison rigorously by answering the following questions: 1. what is the limiting expressivity of each model; 2. under the practical gradient descent, to which solution does each optimization path converge; 3. how would the models generalize under the inductive and transductive learning setting. Our results highlight the similar expressivity for the overparameterized NCF and MCF as kernelized predictors, and reveal the relation between their optimization paths. We further show their different generalization behaviors, where MCF and NCF experience specific tradeoff and comparison in the transductive and inductive collaborative filtering setting. Lastly, by showing a novel generalization result, we reveal the critical role of correcting exposure bias for model evaluation in the inductive setting. Our results explain some of the previously observed conflicts, and we provide synthetic and real-data experiments to shed further insights to this topic.

1. Introduction

Neural collaborative filtering (NCF) and matrix factorization collaborative filtering (MCF) are the two major approaches for learning the user-item embeddings and combining them into similarity scores. From the early stage of collaborative filtering (CF), the factorization methods are actively explored thanks to their interpretability and computation advantage. (Koren et al., 2009; Koren, 2008; Rendle, 2010). In recent years, the success of deep learning in various other domains has motivated the adaptation of universal model architectures, such as the feedforward neural network (FFN), to CF tasks. A prevalent practice is to use FFN to combine the user and item embeddings, which is referred to as the NCF (He et al., 2017). As a consequence, the comparison between MCF and NCF was led by whether the similarity measurement should be computed by the dot product, or learnt by the FFN in a data-adaptive fashion. He et al. (2017) claims that NCF is superior because it is capable of expressing MCF under a particular parameterization. Since then, a significant proportion of new models use FFN as the default similarity function, and they are often reported with more competitive performances than MCF (Zhang et al., 2019). Nevertheless, a recent paper argues based on empirical evidence that after training, FFN is unable to reconstruct the dot-product computation, and NCF can be outperformed by the MCF (Rendle et al., 2020).

It is not of surprise that the controversy arises since our understandings of both methods, specially NCF, are still somewhat limited from the theoretical standpoint. In particular, He et al. (2017) does not consider how training affects the parameters' dynamics, as well as how it alters the model expressivity in comparison with the dot-product formulation. Likewise, the conjectures in Rendle et al. (2020) regarding the trained FFN are based on empirical observations so the conclusions may not generalize to other settings. On the other hand, while MCF has been studied intensively under the matrix completion framework with many theoretical guarantees (Candès & Recht, 2009; Srebro et al., 2004; Srebro & Shraibman, 2005; Candès & Recht, 2009), it is nevertheless unclear how the inductive bias of using gradient descent training can affect the performance.

In addition to the choice of model, which is the main focus of the previous literature, we find that the testing performance is closely connected to how the tasks are designed, including the learning setting (inductive or transductive) as well as the relation between the empirical and exposure distribution. Towards that end, we summarize several major questions that prompt the ongoing debate:

[1] Walmart Labs, Sunnyvale, California, USA. [2] Instacart, San Francisco, California, USA. The work was done when the author was with Walmart Labs.. Correspondence to: Da Xu <daxu5180@gmail.com>.

Proceedings of the 38^{th} International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

1. what is the expressive power of (what functions can be efficiently expressed by) each model, both at initialization and during training;

2. among the potentially many global optimum (due to the overparameterization), which solutions will be located by the practical gradient descent optimization;

3. how would the training performances generalize to the unseen data which can be either obtained as random hold-outs of a fixed finite population (the transductive setting) or i.i.d samples according to some underlying distribution (the inductive setting).

We point out that the third question is particularly complicated for CF. When viewing CF as a matrix completion problem, the training and testing data are constructed via random splits of the finite matrix index set, and are thus fixed before training. On the other hand, when treating the testing data as new i.i.d samples, the exposure bias (Chen et al., 2020) causes the discrepancy between: how the training data are sampled in practice (often according to the empirical data distribution), and which testing distribution to generalize (users are more likely to interact with items that are exposed to them). As a consequence, the same model can show various generalization behaviors in different settings.

Answering each question requires in-depth understandings for the underlying modelling, optimization and generalization mechanisms. This paper also aims to explain how the different factors contribute to the controversies in the previous literature. The recent developments in the neural tangent kernel (NTK) provide powerful tools to study the limiting behavior of overparameterized models (Jacot et al., 2018). We leverage them to first study the expressivity of NCF and MCF and reveal their relations via the lens of kernelized predictors. By studying the local updates in the space of parameters, we then provide a unified analysis on the inductive bias of gradient descent (Soudry et al., 2018) for both NCF and MCF. We find that both optimization paths converge in direction to some norm-constraint max-margin solutions. While our result is a restatement for the FFN (Gunasekar et al., 2018a), it is novel for MCF and connects the two cases. By further deriving the (provably tight) norm-constraint capacities of NCF and MCF, we show their different generalization behaviors under the transductive and inductive learning setting. We also reveal a novel generalization result that accounts for the discrepancy between the empirical data and exposure distribution. We conclude our contributions as follow.

1. We reveal the limiting expressivity of NCF and MCF as kernelized predictors under a specific type of kernel that conforms to the collaborative filtering principle.

2. We provide a unified analysis that shows for NCF and MCF the gradient descent converge to the corresponding max-margin solutions.

3. We study and compare the transductive and inductive generalization behaviors for NCF and MCF, and provide a novel generalization result for the inductive setting that accounts for the impact of exposure.

4. We provide experiments on synthetic and real-data to examine and illustrate our results in practical applications.

Compared with Rendle et al. (2020), we reveal the complex dynamics of model expressivity, optimization and generalization that underlies the comparison between MCF and NCF. Our results rigorously justify the conclusions drawn under each perspective and problem instance.

2. Related Work

The kernel induced by deep learning models, specially the neural tangent kernel (NTK), has been identified as the key component to study the limiting behavior of overparameterized neural networks (Daniely et al., 2016; Jacot et al., 2018; Yang, 2019; Chizat & Bach, 2019) and connect the gradient flow training to the reproducing kernel Hilbert space (RHKS) (Arora et al., 2019c; Mei et al., 2019).

Nevertheless, the limiting-width (scale) results may not fully explain the success of training practical models to zero loss and still being able to generalize (Zhang et al., 2016). Towards that end, considerable efforts are spent studying the optimization geometry and the inductive bias of gradient descent (Soudry et al., 2018), and the convergence results have been shown for various FFN models (Lyu & Li, 2019; Gunasekar et al., 2018a; Chizat & Bach, 2020). However, the analogous results for matrix factorization are mostly established for the the matrix sensing problem under the squared loss (Gunasekar et al., 2018a; Arora et al., 2019a; Li et al., 2018), and are therefore not application to the implicit recommendation setting[1], which is the focus of this paper.

Finally, while the generalization results of FFN and matrix factorization have been studied previously (Wei et al., 2019; Srebro et al., 2004; Srebro & Shraibman, 2005), their settings and assumptions are different from the collaborative filtering tasks (see Section 6 for detail), and are therefore less informative for analyzing NCF and MCF. On the other hand, despite the increasing recognition of adjusting for the exposure bias when training and evaluating recommenders (Xu et al., 2020; Schnabel et al., 2016; Liang et al., 2016; Yang et al., 2018; Gilotte et al., 2018), it is unclear how the common re-weighting approach affects generalization. This line of research also relates to the data missing-not-at-random (MNAR) domain, where the inverse propensity weighting method is heavily investigated in the context of

[1] The users express their interests implicitly, e.g. via click or not click, instead of providing explicit ratings.

recommender systems (Yang et al., 2018; Saito et al., 2020).

3. Notations and Background

To be consistent with the reference work of Rendle et al. (2020), we also adopt the implicit feedback setting which is more practical for the modern recommender systems.

We use $u \in \mathcal{U}$ and $i \in \mathcal{I}$ to denote the user and item, and use $\mathcal{D} = \{(u,i) \,|\, u \in \mathcal{U}, i \in \mathcal{I}\}$ to denote the collected user-item interactions. The vectors, random variables, and matrices and denoted by the lower-case bold-font, upper-case, and upper-case bold-fond letters. The vector ℓ_2 norm, matrix Frobenius norm and nuclear norm are given by $\|\cdot\|_2$, $\|\cdot\|_F$, and $\|\cdot\|_*$. The sigmoid function is given by $\sigma(\cdot)$.

We use $y_{u,i} \in \{-1,+1\}$ to denote the implicit user feedback, $o_{u,i} \in \{0,1\}$ to indicate if the item has been exposed to the user, and P_O to denote the exposure distribution: $p(O_{u,i} = 1)$. The empirical data distribution is denoted by P_n. The classification model (predictor) is selected from $\mathcal{F} = \{f(\boldsymbol{\theta};\cdot) \,|\, \boldsymbol{\theta} \in \boldsymbol{\Theta}\}$. We use $F(\boldsymbol{\theta})$ to denote the *decision boundary* associated with $f(\boldsymbol{\theta};\cdot)$, and use $\ell(\cdot)$ to denote the loss function. Both MCF and NCF use embeddings to represent user and items[2], which we denote by $\mathbf{z}_u, \mathbf{z}_i \in \mathbb{R}^d$. Throughout this paper, we assume the models are overparameterized, so analytically there exists parameterizations $\boldsymbol{\theta}^*$ that perfectly separates the data: $\forall (u,i) \in \mathcal{D} : y_{u,i} f(\boldsymbol{\theta}^*; (u,i)) \geq 0$.

3.1. Dot product and learnt similarity

The general formulation of MCF is given by[3]:

$$f^{\text{MCF}}(u,i) := f\big((\mathbf{Z}_U, \mathbf{Z}_I); (u,i)\big) = \langle \mathbf{z}_u, \mathbf{z}_i \rangle,$$

where $\mathbf{Z}_U$ and $\mathbf{Z}_I$ are matrix collections of the user and item embeddings. When viewed as the matrix completion problem, MCF relies only on the dot product to compute the similarity and recover the observed user-feedback matrix $\mathbf{Y} \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ by minimizing: $\mathcal{L}\big(\mathbf{Z}_u, \mathbf{Z}_i\big) = \frac{1}{|\mathcal{D}|} \sum_{(u,i)\in\mathcal{D}} \ell\big(\big\langle \mathbf{Y}_{u,i}, \big[\mathbf{Z}_U \mathbf{Z}_I^{\intercal}\big]_{u,i} \big\rangle\big)$.

The NCF, on the other hand, employs a feed-forward neural network $\phi(\boldsymbol{\theta}^{\text{NN}};\cdot)$ to learn the similarity of the user and item by first combining their embeddings via addition or concatenation[4]: $f^{\text{NCF-a}}(u,i) := \phi\big(\boldsymbol{\theta}^{\text{NN}}; \mathbf{z}_u + \mathbf{z}_i\big)$ and $f^{\text{NCF-c}}(u,i) := \phi\big(\boldsymbol{\theta}^{\text{NN}}; [\mathbf{z}_u, \mathbf{z}_i]\big)$. Without loss of generality, we assume the following multi-layer percepton (using the addition $\mathbf{x} := \mathbf{z}_u + \mathbf{z}_i$ as an example):

$$\phi(\boldsymbol{\theta}^{\text{NN}}; \mathbf{x}) = \mathbf{W}_1 \tilde{\sigma}(\mathbf{W}_2 \mathbf{x}),\ \mathbf{W}_1 \in \mathbb{R}^{1\times d_1},\ \mathbf{W}_2 \in \mathbb{R}^{d_1 \times d},$$

where $\tilde{\sigma}(\cdot)$ is the ReLU activation. The risk is then given by: $\mathcal{L}\big(\theta^{\text{NN}}, \mathbf{Z}_u, \mathbf{Z}_i\big) = \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}} \ell\big(y_{u,i}, f^{\text{NCF}}(u,i)\big)$.

3.2. Transductive and inductive learning for collaborative filtering

Transductive and inductive learning differ primarily in whether the learnt patterns shall generalize to a specific testing data dependent on the training data, or the samples with respect to some distribution of the population from which the training data is also drawn (Vapnik, 1982; Olivier et al., 2006). For CF in practice, the two learning settings may appear in different scenarios, and each setting can be found suitable by particular tasks.

Specifically for collaborative filtering, transductive learning treats the training and testing (u,i) pairs as obtained in advance via a random split of the finite $\mathcal{D}$, which is equivalent to *sampling without replacement* and thus the dependency between the training and testing data. Compared with the inductive learning whose testing samples are drawn from some unknown distribution over the (u,i) indices, transductive learning aims at a specific subset of (u,i) pairs. The traditional CF tasks, such as movie rating and item-to-item recommendation, are better characterized by transductive learning because their goal is to predict well on a specific subset of entries whose indices are also known in advance.

However, for the negative-sampling-based training and evaluation approach, which often applies to the modern large-scale problems, the data is characterized by an incoming stream of i.i.d samples. As such, inductive learning appears to be the more appropriate setting, yet there still exists a critical gap caused by the exposure bias: the underlying distribution is often shifted by how the items were exposed to users. Inversely weighting each sample by its exposure probability gives the unbiased estimator as if the exposure was *uniformly as random* (Schnabel et al., 2016), i.e.

$$\begin{aligned}
&\mathbb{E}_{P_O}\Big[\frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}} \frac{\ell(y_{u,i}, \hat{y}_{u,i})}{p(O_{u,i}=1)}\Big] \\
&= \frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}} \frac{\ell(y_{u,i}, \hat{y}_{u,i})}{p(O_{u,i}=1)} p(O_{u,i}=1) \\
&= \mathbb{E}_{\hat{P}_{\text{unif}}}[\ell(Y, \hat{Y})].
\end{aligned}$$

The uniform exposure scenario is ideal because the training and testing will not be affected by the previous exposure, thus leading to fair evaluation and comparison. Clearly, the inverse weighting approach can effectively correct for the distribution shift caused by the exposure bias if $p(O_{u,i} = 1)$

[2] The user and item embeddings are allowed to have different dimensions in NCF. Here, we assume they have the same dimensions to be comparable to MCF.

[3] Some previous work reformulate MCF as: $\beta_u + \beta_i + \langle \mathbf{z}_u, \mathbf{z}_i \rangle$, which is a special case of the general formulation.

[4] The original paper of He et al. (2017) suggests using the concatenation, but in our experiments we find adding the embeddings sometimes provide superior performances so we study them both.

is known in advance. We will discuss in Section 7 the limitation of this presumption. Another view of the reweighting method is to think of the testing distribution as P_{unif}, which is uniform on $\mathcal{D}$, and the training distribution as $P(O_{u,i}=1)$, which corresponds to the previous exposure.

3.3. Experiment Setup

We use the *MovieLens-100K*[5] dataset for the real-data and semi-synthetic experiments. Following Rendle et al. (2020) and (He et al., 2017), we first covert the data to implicit feedback by treating the positive ratings as clicks. The primary reasons for using this relatively smaller dataset (which consists of the 100,000 ratings from 1,000 users on 1,700 movies) is the concerning computation feasibility:

1. to show the limiting expressivity and convergence results, we often need to compute the ground truth by solving convex optimization problems exactly and run the gradient descent optimization for several thousands of epochs;

2. a recent paper points out that ranking metrics computed via subsampling can be misleading in practice (Rendle, 2019), so we conduct full scans for metric computation.

The MCF and NCF are implemented with *Tensorflow* using the stochastic gradient descent optimizer and log loss: $\ell(u)=\log(1+\exp(-u))$. All the ranking metrics, i.e. ranking AUC, top-k hitting rate (HR@k) and normalized discounted cumulative gain (NDCG@K), are computed after ranking all the movies for the user. Following (Rendle et al., 2020; He et al., 2017), for each user, we use the last interaction for testing and the rest for training and validation, and equip each positive sample with four negative samples unless specified otherwise. The detailed settings for each section are relegated to the appendix. All the reported results are computed from ten repetitions.

4. The Limiting Expressivity and Collaborative Filtering Kernel

The key challenge for directly comparing the expressivity of dot product and learnt similarity is that the parameters are constantly changing during the gradient descent update: $\boldsymbol{\theta}^{(t)}=\boldsymbol{\theta}^{(t-1)}-\eta\nabla\mathcal{L}(\boldsymbol{\theta}^{(t-1)})$. The different training dynamics and the randomness in initialization prohibit the shoulder-to-shoulder comparison, unless we can find some key factors invariant to training and decisive for model expressivity. A feasible direction is toward the infinite-width domain[6], where the overparameterized models are observed to converge to zero training loss while their parameters hardly vary (Allen-Zhu et al., 2019; Du et al., 2019). The invariant factor that is recognized to fully describe the training dynamics is the neural tangent kernel (NTK) (Jacot et al., 2018; Arora et al., 2019b):

$$K\big((u,i),(u',i')\big)=\big\langle\nabla f\big(\boldsymbol{\theta};(u,i)\big),\nabla f\big(\boldsymbol{\theta};(u',i')\big)\big\rangle,$$

which is a fixed quantity that depends only on the model architecture as $d\to\infty$. We refer to the model expressivity under the infinite-width NTK as its limiting expressivity.

To see the critical role of a fixed NTK for model expressivity, note that by Taylor expansion, infinite-width MF and FFN behave like a linear function around their *scaled initializations* (i.i.d with $N(0,\alpha/d)$ for some constant α):

$$f\big(\boldsymbol{\theta};(u,i)\big)=f\big(\boldsymbol{\theta}^{(0)};(u,i)\big)+\\ \Big\langle\boldsymbol{\theta}-\boldsymbol{\theta}^{(0)},\nabla f\big(\boldsymbol{\theta}^{(0)};(u,i)\big)\Big\rangle+\mathcal{O}\big(\sqrt{1/d}\big).$$

Analytically, we can always use reparameterization to discard the intercept term $f\big(\boldsymbol{\theta}^{(0)};(u,i)\big)$ by finding $f(\boldsymbol{\theta};\,\cdot)=g(\boldsymbol{\theta}_1;\,\cdot)-g(\boldsymbol{\theta}_1;\,\cdot)$ and letting $\boldsymbol{\theta}_1^{(0)}=\boldsymbol{\theta}_2^{(0)}$. Therefore, the limiting expressivity of $f\big(\boldsymbol{\theta};(u,i)\big)$ is well-approximated by $\Big\langle\boldsymbol{\theta}-\boldsymbol{\theta}^{(0)},\nabla f\big(\boldsymbol{\theta}^{(0)};(u,i)\big)\Big\rangle$ as $d\to\infty$. It is now a straightforward analogy that $\nabla f\big(\boldsymbol{\theta}^{(0)};(u,i)\big)$ plays the role of the feature lift that maps all the (u,i) pair to their representations in the reproducing kernel Hilbert space (RHKS) induced by the NTK, i.e. $K:(\mathcal{U},\mathcal{I})\times(\mathcal{U},\mathcal{I})\to\mathbb{R}$ (Shawe-Taylor et al., 2004). Consequently, under the infinite-width limit, applying gradient descent in the original function space is equivalent to finding the optimal RHKS predictor.

We point out that the RHKS view of MCF and NCF during optimization is a natural continuation of their expressive power at initialization. At initialization, the dot-product formulation of MCF directly characterizes the Gram matrix for the user-item similarity kernel (Paterek, 2007), and standard FFN is known to concentrate to a Gaussian process characterized by its covariance kernel (Neal, 1996; Yang, 2019). In the sequel, we wish to understand the relation between the NTK of MCF and NCF, and find out which solutions are optimal in the corresponding RHKS. Interestingly, both MCF and NCF lead to a specific type of kernel that we believe interprets collaborative filtering in principle. We summarize the results as below.

Theorem 1. *Let $\mathcal{D}_{train}\subseteq\mathcal{D}$ be the training data. Under the exponential loss $\ell(u)=\exp(-u)$ or log loss $\ell(u)=\log(1+\exp(-u))$, by applying gradient descent with small learning rate and scaled initializations, the decision boundary for infinite-width MCF and NCF connects to the minimum-norm RHKS predictor via:*

$$\lim_{t\to\infty}\lim_{d\to\infty}F\Big(\frac{\boldsymbol{\theta}^{(t)}}{\|\boldsymbol{\theta}^{(t)}\|_2}\Big)\xrightarrow{\textit{stationary points of}}\\ \Big\{\mathop{\arg\min}_{f:(\mathcal{U},\mathcal{I})\to\mathbb{R}}\big\|f\big\|_{K_{CF}}\,s.t.\,y_{u,i}f(u,i)\geq 1,\,\forall(u,i)\in\mathcal{D}_{train}\Big\},$$

[5] https://grouplens.org/datasets/movielens/100k/

[6] The width of each layer tends to infinity, i.e. $d,d_1\to\infty$ for the models in Section 3.1. Here, we assume $d_1=d$, w.l.o.g.

where $\|\cdot\|_K$ is the induced RKHS norm and K_{CF} is the ***collaborative filtering kernel*** *parameterized via:*

$$K_{CF}\big((u,i),(u',i')\big) = a + b\cdot \mathbb{1}[i=i'] + c\cdot 1[u=u'],$$

for some $a,b,c \geq 0$ when $(u,i) \neq (u',i')$.

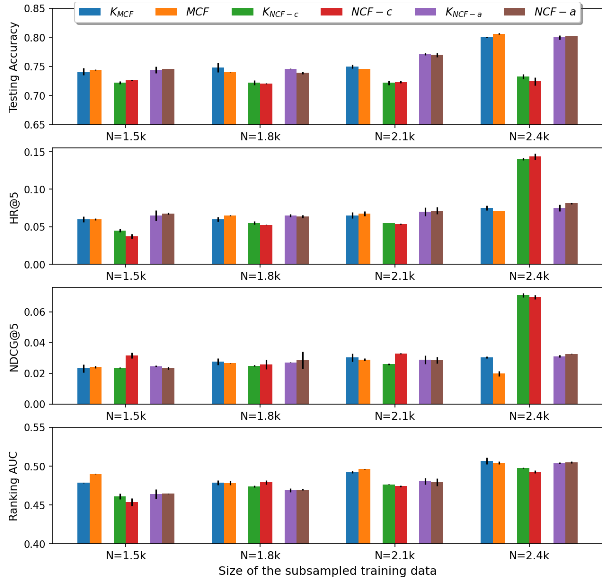

Figure 1: The comparison between the RKHS predictors in Theorem 1 and their corresponding MCF or NCF methods on the testing data. We use *LIBSVM* (Chang & Lin, 2011) for the kernelized support vector machine (SVM). The subsampling is performed by sampling 200 items according to the popularity, and sampling 200 users uniformly at random. For each (u,i) pair with a positive feedback, we sample different number of negative instances without replacement to obtain the desired sample size of N. We set $d=128$ and $d_1=64$ for MCF and NCF to approximate the large-width setting as the dataset is very small. To be consistent with the theory, we use the standard scaled normal initialization and train 3,000 epochs under the constant learning rate 0.01.

For instance, under the standard scaled initialization $N(0,1/d)$, we have $a=0$, $b=c=1$ for K_{MCF}; $a=1/\pi$, $b=c=\frac{1}{2}-\frac{1}{2\pi}$ for $K_{\text{NCF-c}}$, and $a=1/\pi$, $b=c=\frac{1}{2}-\frac{(2-\sqrt{3})}{2\pi}$ for $K_{\text{NCF-a}}$. The details and proofs are relegated to the appendix.

The significance of Theorem 1 is that we now have a dual charactierzation of model expressivity for the *trained* MCF and NCF via the lens of RKHS predictors. Furthermore, K_{CF} provide a natural kernel interpretation for the CF principle by emphasizing how predictions are made from the collaborative user or item signal. Recall from the standard SVM arguments that the limiting classifier in Theorem 1 is given by:

$$f(u',i') = \sum_{(u,i)\in\mathcal{D}_{\text{train}}} \alpha_{u.i} y_{u,i} K_{\text{CF}}\big((u,i),(u',i')\big),$$

where $\alpha_{u,i} \geq 0$ are the corresponding dual variables (Drucker et al., 1997). Specifically for K_{CF}, the value of a, which gives the kernel value when $u \neq u'$ and $i \neq i'$, captures the global (population) bias; b and c capture the relative importance[7] of neighboring data points who share the same user or item. In Figure 1, we compare the testing results of large-width MCF, NCF and their K_{CF}-induced SVM on the sampled subsets of the MovieLens data. The subsampling is necessary because the kernel method has a space complexity of $\mathcal{O}\big(|\mathcal{D}|\times|\mathcal{D}|\big)$. We observe highly comparable results from the trained CF and their corresponding RHKS predictor on testing data, which suggests their similar limiting performances [8].

Therefore, regardless of the distinguished formulations of dot product and learnt similarity, both MCF and NCF are capable of expressing the same type of RKHS predictor in their limit. The only difference lies in their specific kernel parameterization, which immediately implies that each model's limiting performance can depend on the specific data distribution. In particular, Bartlett & Mendelson (2002) show that the inductive generalization performance of the RKHS predictor in Theorem 1 satisfies:

$$\text{Test error} \leq \text{Train error} + \mathcal{O}\left(\sqrt{\frac{\boldsymbol{\alpha}^{\intercal} K_{\text{CF}}\boldsymbol{\alpha}\cdot \text{trace}(K_{\text{CF}})}{|\mathcal{D}_{\text{train}}|}}\right),$$

where $\boldsymbol{\alpha}$ is the vector of dual variables which are data-dependent. Therefore, it is impossible to draw an exclusively comparison between the two methods, because each dataset can cause a particular $\boldsymbol{\alpha}$ that lead to different comparisons of $\boldsymbol{\alpha}^{\intercal} K_{\text{CF}}\boldsymbol{\alpha}$. Consequently, our results suggest that the comparison between MCF and NCF should not be concluded sightly just based on their expressivity, at least not in the limiting sense. We still need to understand how the parameterization interacts with gradient descent under the practical model size and initialization, which are the topics of the next section.

5. Optimization Paths of MCF and NCF under Gradient Descent

The key observation that motivates this section is that when trained with gradient descent on the MovieLens data without using any explicit regularization (Figure 3), the classification and ranking metric on testing data keep increasing

[7] We illustrate in the appendix on how the relative magnitude of a,b,c can be changed by the initializations.

[8] The results does not imply that MCF and NCF can be effectively replaced in practice because the kernel methods are impractical even for moderate recommendation dataset.

while the training loss decreases to zero. Our observation is consistent with Zhang et al. (2016) and the follow-up analysis of Soudry et al. (2018); Gunasekar et al. (2018b), that gradient descent carries the inductive bias of selecting the global optimum that tends to generalize well. Since this phenomenon may hold for any model size and initialization as pointed out by Soudry et al. (2018), the underlying mechanism may provide a more complete understanding of the trained MCF and NCF in addition to the previous section.

Also, in Figure 3, we show another phenomenon that the norms of parameter matrices diverge as the loss converges to zero. It is not a surprising result since the gradient-descent training is conducted under the following conditions[9]:

C1. The loss function has the exponential-tail behavior such as the exponential loss and log loss;

C2. Both the MCF and NCF in Section 3.1 are L-homogeneous, i.e. $f(\boldsymbol{\theta};\,\cdot) = \|\boldsymbol{\theta}\|_2^L \cdot f\big(\boldsymbol{\theta}/\|\boldsymbol{\theta}\|_2;\,\cdot\big)$ for some $L > 0$, and have some smoothness property;

C3. The data is separable with respect to the overparameterized MCF and NCF (introduced in Section 3).

For clarity purpose, we consider the exponential loss $\ell(u) = \exp(-u)$. Setting aside the technical details, the heuristic explanation is that with a proper learning rate, gradient descent on smooth objectives converges to the stationary points (gradient vanishes). Since the gradient is given by:

$$\frac{1}{|\mathcal{D}|}\sum_{(u,i)\in\mathcal{D}} -y_{u,i}\exp\big(-y_{u,i}f\big(\boldsymbol{\theta};(u,i)\big)\big)\nabla f\big(\boldsymbol{\theta};(u,i)\big),$$

disregarding the corner cases where the gradient terms are linear dependent, the necessary condition for a zero gradient is: $\forall (u,i)\in\mathcal{D} : y_{u,i}f\big(\boldsymbol{\theta};(u,i)\big) \to \infty$, which implies that $\|\boldsymbol{\theta}\|_2 \to \infty$ since f is homogeneous. The norm divergence plays an important role here because the loss function is now dominated by: $\min_{(u,i)\in\mathcal{D}} y_{u,i}f\big(\boldsymbol{\theta};(u,i)\big)$, which corresponds to the worst margin, due to the fact that the loss function has an exponential behavior. Therefore, as the parameter norm diverges, the direction of the optimization path resembles that of the hard-margin SVM (shown as below) since $\mathcal{D}$ is separable.

$$\max_{\boldsymbol{\theta}:\|\boldsymbol{\theta}\|_2\le 1}\ \min_{(u,i)\in\mathcal{D}} y_{u,i}f\big(\boldsymbol{\theta};(u,i)\big),$$

The same reasoning holds for MCF under the nuclear norm, since it relates to the ℓ_2 norm via: $\|\mathbf{X}\|_* = \frac{1}{2}\min_{\mathbf{U}\mathbf{V}^\intercal=\mathbf{X}}(\|\mathbf{U}\|_F^2 + \|\mathbf{V}\|_F^2)$.

Providing the rigorous arguments for the above heuristic is nontrivial since $f(\boldsymbol{\theta};\,\cdot)$ can be non-convex for NCF, so the connection between stationarity and optimality can be tricky. We refer to a similar idea in Lyu & Li (2019), which bridges the discrepancy by showing a specific type of constraint qualification that leads to the KKT points. We summarize our results as below.

Theorem 2. *Under condition C1, C2 and C3, with a small constant learning rate,* $\boldsymbol{\theta}^{NCF} := [\boldsymbol{\theta}^{NN}, \mathbf{Z}_U, \mathbf{Z}_I]$ *converges in direction, i.e.* $\lim_{t\to\infty}\boldsymbol{\theta}^{(t)}/\|\boldsymbol{\theta}^{(t)}\|_2$*, to the KKT point of:*

$$\min\ \|\boldsymbol{\theta}\|_2 \quad s.t. \quad y_{u,i}f^{NCF}\big(\boldsymbol{\theta};(u,i)\big) \ge 1,\ \forall (u,i)\in\mathcal{D}_{train}.$$

The predictor of MCF: $\mathbf{X} = \mathbf{Z}_U\mathbf{Z}_I^\intercal$ *converges in direction, i.e.* $\lim_{t\to\infty}\mathbf{X}^{(t)}/\|\mathbf{X}^{(t)}\|_*$*, to the stationary point of:*

$$\min\ \|\mathbf{X}\|_* \quad s.t. \quad y_{u,i}\mathbf{X}_{u,i} \ge 1,\ \forall (u,i)\in\mathcal{D}_{train}. \qquad (1)$$

Theorem 2 extends the results in Soudry et al. (2018); Lyu & Li (2019); Gunasekar et al. (2018b) to matrix factorization setting, and complements Gunasekar et al. (2018a); Arora et al. (2019a) who study matrix sensing under the squared loss. We also provide a unified proof in the appendix.

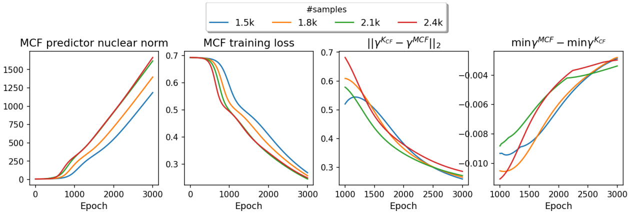

Figure 2: We adopt the sub-sampled dataset described in Figure 1. We first use the *CVX solver* (Grant & Boyd, 2014) to obtain the exact solutions of the convex programming of e.q. (1). In contrast to the optimization setup of Figure 1, here we consider the more practical scenario by using the unscaled $N(0, 0.1)$ initialization, the moderate width of $d = 32$ and the learning rate of 0.1. We do not show the error bar in the line charts for visualization clarity.

The optimization paths described in Theorem 2 reveal an intrinsic difference between MCF and NCF in terms of which minimum-norm solutions are tended by the gradient descent in a practical setting. Since the corresponding hard-margin SVM problem for MCF (state in (1)) is convex, we are able to to obtain its exact global minimum, and use it as an oracle to compare with the actual behavior of MCF during training. We particularly focus on the margins of the normalized (by nuclear norm) predictor, which we denote by $\boldsymbol{\gamma}$. In Figure 2, we first verify that the nuclear norm divergence indeed accompanies the convergence of training loss. Crucially, both $\|\boldsymbol{\gamma}^{K_{\mathrm{CF}}} - \boldsymbol{\gamma}^{\mathrm{MCF}}\|_2$ and $\min\boldsymbol{\gamma}^{\mathrm{MCF}} - \min\boldsymbol{\gamma}^{K_{\mathrm{CF}}}$ converge (by decreasing and increasing) to zero, as anticipated by the conclusions of Theorem 2.

Compared with Section 4, the results here reveal optimization path as a more likely factor that distinguishes MCF and NCF in practice. The nuclear-norm and ℓ_2-norm constraint

[9] The more refined discussions for each condition, as well as their implications, are provided in the appendix.

optimization paths imply the different capacities of the predictor, whose impact on the generalization performance is among the core investigation in the learning theory. We provide the detailed characterization and their comparisons in the next section.

6. Generalization for Transdutive and Inductive Collaborative Filtering

Before we state the main results, we point out that our setting covers a broad range of settings for MCF and NCF. Firstly, we consider the neural network in NCF to be any $q-$layer FFN, i.e. parameterized by $[\mathbf{W}_1, \ldots, \mathbf{W}_q, \mathbf{Z}_U, \mathbf{Z}_I]$, with ReLU activation. Secondly, we allow the explicit ℓ_2-norm regularizations, which are sometimes found useful to improve model performance under finite-step training. Together with the results from Section 5, we expect that both MCF and NCF will tend to the norm-constraint solutions, either by training toward convergence without the explicit norm regularization or by training under the explicit norm regularizations.

As we mentioned in Section 3.2, the transductive and inductive CF requires studying different generalization mechanisms, so MCF and NCF may compare differently in each setting. For inductive CF, the exposure distribution often decides which (u, i) pairs are more likely to appear in the training data, so its discrepancy with the idea uniform-exposure scenario under P_{unif} is also a critical factor for a fair comparison of MCF and NCF. For transductive CF, the goal is to bound the classification error on the given $\mathcal{D}_{\text{test}}$, i.e. $\text{Err}_{\mathcal{D}_{\text{test}}}(f) := \sum_{(u,i)\in\mathcal{D}_{\text{test}}} 1\big[y_{u,i} f(u,i) \leq 0\big]$, while for the inductive CF we wish to bound $\mathbb{E}_{(u,i)\sim P_{\text{unif}}} 1\big[y_{u,i} f(u,i) \leq 0\big]$ under the ideal uniform-exposure scenario that we explained in Section 3. For many real-world problems, and in the MovieLens dataset that we are experimenting with, the number of items far exceeds the number of users. So we assume $|\mathcal{U}| < |\mathcal{I}|$ without loss of generality.

For the transductive CF, the testing data is often made proportional to the training data so here we assume $|\mathcal{D}_{\text{test}}| = \beta|\mathcal{D}_{\text{train}}|$ for some $0 < \beta < 1$. Notice that $\mathcal{D}_{\text{test}} \cap \mathcal{D}_{\text{train}} = \emptyset$ due to the sampling without replacement. Also, $\mathcal{D} = \mathcal{D}_{\text{test}} \cup \mathcal{D}_{\text{train}}$. We first state the generalization result for transductive CF in terms of the classification error.

Theorem 3 (Transductive CF). *Let f^{MCF}, $f^{NCF} : \mathcal{U} \times \mathcal{I} \rightarrow \mathbb{R}$ be the predictor for MCF and NCF. Suppose that $\max_{(u,i)\in\mathcal{D}} \|\mathbf{z}_u\|_2 + \|\mathbf{z}_i\|_2 \leq B_{NCF}$ for NCF with concatenation, $\max_{(u,i)\in\mathcal{D}} \|\mathbf{z}_u + \mathbf{z}_i\|_2 \leq B_{NCF}$ for NCF with addition, and $\|\mathbf{W}_i\|_F \leq \lambda_i$ for $i = 1, \ldots, q$. Also suppose for MCF that $\big\|\mathbf{Z}_U \mathbf{Z}_I^{\intercal}\big\|_* \leq \lambda_{nuc}$. Then with probability at least $1 - \delta$ over the random splits of $\mathcal{D}$, for any $\big\{f^{NCF}(u,i) \,\big|\, (u,i) \in \mathcal{D}\big\}$ and $\gamma > 0$:*

$$Err_{\mathcal{D}_{test}}(f^{NCF}) \leq \sum_{(u,i)\in\mathcal{D}_{train}} 1\big[y_{u,i} f^{NCF}(u,i) \leq \gamma\big] +$$

$$\mathcal{O}\bigg(\frac{(1+\beta)\sqrt{q} B_{NCF} \prod_{i=1}^{q} \lambda_i}{\gamma\beta|\mathcal{D}_{train}|^{1/2}}\bigg) + \mathcal{O}\bigg(\sqrt{\frac{\log 1/\delta}{|\mathcal{D}_{test}|}}\bigg),$$

and for any $\big\{f^{MCF}(u,i) \,\big|\, (u,i) \in \mathcal{D}\big\}$ and $\gamma > 0$:

$$Err_{\mathcal{D}_{test}}(f^{MCF}) \leq \sum_{(u,i)\in\mathcal{D}_{train}} 1\big[y_{u,i} f^{MCF}(u,i) \leq \gamma\big] +$$

$$\mathcal{O}\bigg(\frac{(1+\beta)\sqrt[4]{\log|\mathcal{U}|}\sqrt{|\mathcal{I}|} \cdot \lambda_{nuc}}{\gamma\beta|\mathcal{D}_{train}|}\bigg) + \mathcal{O}\bigg(\sqrt{\frac{\log 1/\delta}{|\mathcal{D}_{test}|}}\bigg).$$

Theorem 3 gives the first NCF generalization result in the transductive setting, and while the generalization of matrix sensing has been studied in such as Srebro & Shraibman (2005) and Candès & Recht (2009), their results are either too loose, or require assuming a particular distribution over the entries. We leverage the more recent results from random matrix theory to derive more informative bounds (Bandeira et al., 2016; Tropp, 2015). The proofs are deferred to the appendix.

The critical implication of Theorem 3 is that MCF is the more effective method for transductive CF. Even though the overall asymptotic rate is still controlled by the slack term $\mathcal{O}\Big(\sqrt{\log\frac{1}{\delta}/|\mathcal{D}_{\text{test}}|}\Big)$ for both MCF and NCF, we observe a much faster linear decay with respect to $|\mathcal{D}_{\text{train}}|$ in the model complexity term (the second term on RHS of the generalization bounds) of MCF. Since we assume the parameter norms are controlled, it means the advantage of MCF will be more significant as we collect more observations for the same set of users and items. The superiority of MCF for transductive CF is also justified by the experiments, where it outperforms NCF in terms of all the ranking metrics as the gradient descent converges to zero training error (Figure 3). Our result also explains some of the previous success of MCF in such as movie recommendation where the task resembles transductive CF.

The generalization of inductive CF, on the other hand, requires taking account of the discrepancy between the hypothetical P_{unif} for the uniform exposure scenario, and the actual exposure distribution P_O. As we discussed earlier, using the samples from P_{unif} for testing essentially means we want the exposure to be made uniformly at random, so each (u, i) pair has an equal probability of being exposed. It eliminates the exposure bias during testing (evaluation), and is realistic for the modern recommender where the testing samples are often drawn randomly. Therefore, the empirical training risk needs to be adjusted such that it unbiasedly estimate the desired testing risk under P_{unif}. Otherwise, there

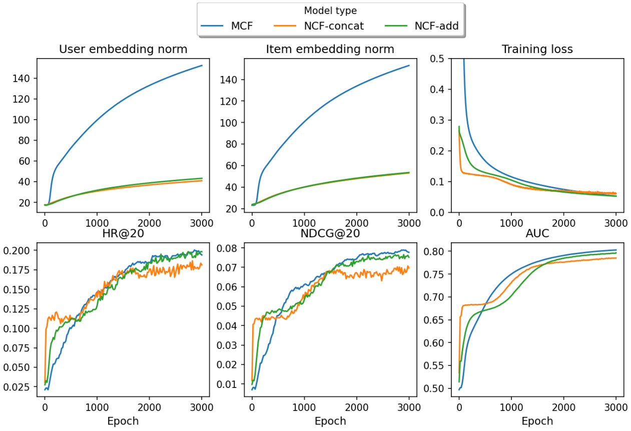

Figure 3: Analysis and model performances for transductive CF using the MovieLens data. We use: $d = 32$, $d_1 = 16$, initializations of $N(0, 0.1)$ and learning rate as 0.1. All the ranking metrics are computed on $\mathcal{D}_{\text{test}}$. The perturbations in the HR and NDCG plots are caused by averaging over ten repetitions. The details are provided in the appendix.

is no guarantee for generalization. Towards that end, we employ the reweighting method introduced in Section 3.1 that effectively corrects the exposure bias of the training data. The weight for each (u, i) pair is therefore given by $P_{\text{unif}}(u, i)/P(O_{u,i} = 1) = 1/P(O_{u,i} = 1)$, and the testing risk is given by: $\mathbb{E}_{(u,i)\sim P_{\text{unif}}} 1\big[y_{u,i} f(u, i) \le 0\big]$, which we denote by the shorthand: $\text{Err}_{P_{\text{unif}}}(f)$. We show the generalization bounds for MCF and NCF in the following two theorems. For brevity, we let $n = |\mathcal{D}_{\text{train}}|$.

Theorem 4 (**Inductive NCF**). *Assume the same setting from Theorem 3. Let $\mathcal{D}_{train}$ be the training data drawn i.i.d according to P_{unif}. Consider the Peasrson χ^2-divergence of two distributions P and Q: $D_2(P\|Q) = \int \big((\frac{dP}{dQ})^2 - 1\big)dQ$. Then with probability at least $1 - \delta$, for all $\gamma > 0$:*

$$Err_{P_{unif}}(f^{NCF}) \le \sum_{(u,i)\in\mathcal{D}_{train}} \frac{1}{P_O(u,i)} 1\big[y_{u,i} f^{NCF}(u,i) \le \gamma\big] + \mathcal{O}\bigg(\frac{\sqrt{q} B_{NCF} \prod_{i=1}^{q} \lambda_i \cdot \sqrt{D_2(P_{unif} \,\|\, P_O) + 1}}{\gamma n^{1/2}}\bigg) + \varepsilon,$$

where $\varepsilon = \sqrt{\frac{\log\log_2(4\sqrt{2}/\gamma)}{n}} + \sqrt{\frac{\log(1/\delta)}{n}}$.

Theorem 5 (**Inductive MCF**). *Following Theorem 4, we now consider another divergence between P and Q: $D_1(P\|Q) = \int \big(\frac{dP}{dQ}\big)dP$. For all $\gamma > 0$, it holds with probability at least $1 - \delta$ that:*

$$Err_{P_{unif}}(f^{MCF}) \le \sum_{(u,i)\in\mathcal{D}_{train}} \frac{1}{P_O(u,i)} 1\big[y_{u,i} f^{MCF}(u,i) \le \gamma\big] + \mathcal{O}\bigg(\frac{\sqrt{D_1(P_{unif} \,\|\, P_O)\big((|\mathcal{U}| + |\mathcal{I}|)\lambda_{nuc}\big)\log 9n}}{\gamma n^{1/2}}\bigg) + \varepsilon,$$

where $\varepsilon = \mathcal{O}\Big(\frac{\log(1/\delta) + (|\mathcal{U}| + |\mathcal{I}|)\lambda_{nuc}\log 9n}{n}\Big)$.

The generalization results for correcting distribution shift via reweighting are novel and of interest beyond this paper. The proof of Theorem 4 resembles what we show in our concurrent work of Xu et al. (2021b), and we employ the classical "double sampling" trick and a covering number argument to prove Theorem 5. Note that we use different types of divergence in the bounds of NCF and MCF, which is a matter of technical necessity. Nevertheless, they lead to the same conclusions as we discuss below.

Remark 1. *The bounds in Theorem 3 and 4 are tight up to constants, and the $\log n$ rate in Theorem 5 might be improved, which we discuss in detail in the appendix. The takeaway is that those bounds provide a sound theoretical basis for comparing the models' generalization guarantees. However, we point out that the bounds are not comparable across transductive and inductive setting because the meanings of generalization are different. It is possible to obtain post-hoc generalization bounds, i.e. plug in the trained parameters' norms, using the technique in Koltchinskii et al. (2002), which is beyond the scope of this paper.*

We first observe from Theorem 4 and 5 that in the inductive setting, the rates of generalization are different for NCF and MCF in the asymptotic regime: the NCF has a more favorable $\sqrt{\frac{1}{n}}$ rate. Crucially, in the more interested finite-sample regime, MCF still has an explicit dependency on $|\mathcal{U}|$ and $|\mathcal{I}|$ even when λ_{nuc} is controlled. Therefore, regardless of the rate, MCF's model complexity still exceeds NCF by the factor of least $\sqrt{|I|}$, which is significant for modern applications. Since MCF and NCF can achieve a similar empirical loss term (the first term on RHS) as they are both overparameterized, NCF will provide better generalization performance in the inductive setting.

Furthermore, both Theorem 4 and Theorem 5 reveals the divergence between P_{unif} and P_O as an amplifying factor of the model complexity. In other words, the model with a higher complexity will be more difficult to generalize when the target distribution P_{unif} is further apart from P_O. It also implies that the generalization performance is sensitive to the exposure, emphasising the importance of using the correct P_O for evaluation. If the exposure is misspecified by such as using P_n, i.e. not correcting for the exposure bias at all, the evaluation results may be misleading. In what follows, we carry out semi-synthetic experiments by controlling the exposure mechanism to illustrate our conclusions.

To preserve the inductive bias of the MovieLens dataset when generating synthetic data, we use the user and item embeddings learnt from the original dataset for the designed exposure and relevance model. In particular, we generate the click data according to : $p(Y_{u,i} = 1) = p(R_{u,i} = 1) \cdot p(O_{u,i} = 1)$, where $R_{u,i}$ indicates the rele-

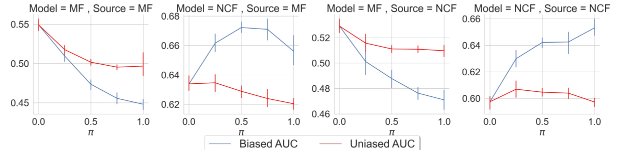

Figure 4: The ranking AUC on the testing data when evaluated using the correct exposure (unbiased evaluation), and treating P_{unif} as the exposure (biased evaluation). We experiment with $\pi \in \{0, 0.25, 0.5, 0.75, 1\}$. In the plot titles, the *source* indicates the model type of g_{exp} and g_{rel}, and the *model* indicates which CF model we use. The patterns of the unbiased AUC justifies our theoretical argument that it is more difficult to generalize for both NCF and MCF when π gets larger, i.e. the discrepancy increases. The experiment details are provided in appendix.

vance. Since we have access to the explicit rating scores from the MovieLens dataset, we first train the relevance model $g_{\text{rel}}(u, i)$ as a regression task. We then train the exposure model $g_{\text{exp}}(u, i)$ by treating each rating event as an exposure, and let the designed exposure model be given by: $P_{\text{design}} : p(O_{u,i} = 1) = \sigma\big(g_{\text{exp}}(u, i)\big)$. Finally, we tune the relevance distribution via: $p(R_{u,i} = 1) = \sigma\big(g_{\text{rel}}(u, i) - \mu\big)^{\rho}$, and the goal is to finding the μ and ρ such that the populational relevance matches that of the original dataset. We end up with $\mu = 3$, $\rho = 2$.

For fair comparisons, we experiment using both f^{MCF} and f^{NCF} for training g_{rel} and g_{exp} so the generating mechanism would not bias toward one of the CF methods. We introduce the hyper-parameter $\pi \in [0, 1]$ to control the deviation from P_O to P_{unif} by using the mixture of:

$$P_O = \pi P_{\text{design}} + (1 - \pi) P_{\text{unif}}$$

The idea is to use π as a proxy to examine the outcome's sensitivity to $D(P_{\text{unif}} \,\|\, P_O)$ and the degree of misspecification for the exposure model. The various evaluation results are provided in Figure 4. We see that in the biased evaluation where the reweighting is not used, the testing performances drastically change as we increase π. We point out that this scenario often happens in practice where the exposure bias is not corrected during testing. The relative lift of NCF to MCF, which is shown in Figure 5, also suffers from significant perturbations as we increase π. In contrast, when using the unbiased estimation under the correct reweighting, both the testing metrics and model comparisons are much more consistent, as we show in both Figure 4 and 5. We conclude from the results that:

1. in the inductive CF setting that we consider, NCF outperforms MCF as suggested by Theorem 4 and 5;

2. when P_O is misspecified by such asP_{unif}, the biased evaluation is very sensitive to the their divergence so the model comparison results can be misleading.

In this section, we rigorous show for MCF and NCF their generalization behaviors for the transductive and inductive CF. Our results suggest MCF is favorable to transductive CF, while NCF exhibits better guarantees for inductive CF. We further reveal exposure as a critical factor for correctly evaluating and comparing models in the inductive setting. All the conclusions are justified by the numerical results.

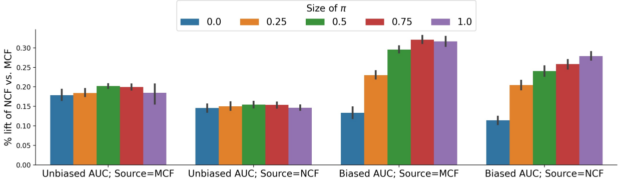

Figure 5: Following Figure 4, we compute the relative percentage lift of the NCF model over MCF, with respect to the unbiased and biased ranking AUC. In the xlabel, we specify the evaluation type, and use *source* to indicate the model type of g_{exp} and g_{rel}. The lift of NCF w.r.t MCF is much more consistent under the unbiased evaluation.

7. Summary, Scope and Limitation

We revisit the critical comparison between MCF and NCF by studying the underlying mechanisms of model expressivity, optimization and generalization. We show that the comparisons are better characterized by the optimization paths rather than model expressivity, as they lead to specific norm-constraint solutions that exhibit meaningful generalization patterns under different learning settings. We further emphasize the crucial role of exposure for model evaluation in inductive CF. The fact that they receive less attention in the relevant literature might explain some of the controversial outcomes.

In the concurrent work of Xu et al. (2021a), the authors primarily study the theoretical perspectives of user and item embeddings learnt by the skip-gram negative sampling algorithm, which deviates from the scope of regular CF. On the other hand, some of our analysis is carried out under conditions C1-C3, which may restrict our result to generalize to the more sophisticated NCF architectures. Also, we do not study how the various tricks of training and data manipulation could affect our conclusions. Finally, while our results strongly advocate for unbiased evaluation in the inductive setting, it is nevertheless very challenging to implement the inverse weighting approach in practice. It is pointed out by Xu et al. (2020) that there often exists identifiability issues, and more importantly, the inverse propensity weighting is not applicable when the exposure is deterministic. We leave to future work to study how to leverage reweighting for recommender systems more effectively.

References

Allen-Zhu, Z., Li, Y., and Song, Z. A convergence theory for deep learning via over-parameterization. In *International Conference on Machine Learning*, pp. 242–252. PMLR, 2019.

Arora, S., Cohen, N., Hu, W., and Luo, Y. Implicit regularization in deep matrix factorization. In *Advances in Neural Information Processing Systems*, pp. 7413–7424, 2019a.

Arora, S., Du, S. S., Hu, W., Li, Z., Salakhutdinov, R., and Wang, R. On exact computation with an infinitely wide neural net. *arXiv preprint arXiv:1904.11955*, 2019b.

Arora, S., Du, S. S., Hu, W., Li, Z., and Wang, R. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. *arXiv preprint arXiv:1901.08584*, 2019c.

Bandeira, A. S., Van Handel, R., et al. Sharp nonasymptotic bounds on the norm of random matrices with independent entries. *Annals of Probability*, 44(4):2479–2506, 2016.

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

Candès, E. J. and Recht, B. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717–772, 2009.

Chang, C.-C. and Lin, C.-J. Libsvm: a library for support vector machines. *ACM transactions on intelligent systems and technology (TIST)*, 2(3):1–27, 2011.

Chen, J., Dong, H., Wang, X., Feng, F., Wang, M., and He, X. Bias and debias in recommender system: A survey and future directions. *arXiv preprint arXiv:2010.03240*, 2020.

Chizat, L. and Bach, F. A note on lazy training in supervised differentiable programming. In *32nd Conf. Neural Information Processing Systems (NeurIPS 2018)*, 2019.

Chizat, L. and Bach, F. Implicit bias of gradient descent for wide two-layer neural networks trained with the logistic loss. *arXiv preprint arXiv:2002.04486*, 2020.

Daniely, A., Frostig, R., and Singer, Y. Toward deeper understanding of neural networks: The power of initialization and a dual view on expressivity. In *Advances In Neural Information Processing Systems*, pp. 2253–2261, 2016.

Drucker, H., Burges, C. J., Kaufman, L., Smola, A., Vapnik, V., et al. Support vector regression machines. *Advances in neural information processing systems*, 9:155–161, 1997.

Du, S., Lee, J., Li, H., Wang, L., and Zhai, X. Gradient descent finds global minima of deep neural networks. In *International Conference on Machine Learning*, pp. 1675–1685. PMLR, 2019.

Gilotte, A., Calauzènes, C., Nedelec, T., Abraham, A., and Dollé, S. Offline a/b testing for recommender systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, pp. 198–206, 2018.

Grant, M. and Boyd, S. Cvx: Matlab software for disciplined convex programming, version 2.1, 2014.

Gunasekar, S., Lee, J. D., Soudry, D., and Srebro, N. Implicit bias of gradient descent on linear convolutional networks. In *Advances in Neural Information Processing Systems*, pp. 9461–9471, 2018a.

Gunasekar, S., Woodworth, B., Bhojanapalli, S., Neyshabur, B., and Srebro, N. Implicit regularization in matrix factorization. In *2018 Information Theory and Applications Workshop (ITA)*, pp. 1–10. IEEE, 2018b.

He, X., Liao, L., Zhang, H., Nie, L., Hu, X., and Chua, T.-S. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

Jacot, A., Gabriel, F., and Hongler, C. Neural tangent kernel: Convergence and generalization in neural networks. In *Advances in neural information processing systems*, pp. 8571–8580, 2018.

Koltchinskii, V., Panchenko, D., et al. Empirical margin distributions and bounding the generalization error of combined classifiers. *Annals of statistics*, 30(1):1–50, 2002.

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

Koren, Y., Bell, R., and Volinsky, C. Matrix factorization techniques for recommender systems. *Computer*, 42(8): 30–37, 2009.

Li, Y., Ma, T., and Zhang, H. Algorithmic regularization in over-parameterized matrix sensing and neural networks with quadratic activations. In *Conference On Learning Theory*, pp. 2–47. PMLR, 2018.

Liang, D., Charlin, L., McInerney, J., and Blei, D. M. Modeling user exposure in recommendation. In *Proceedings of the 25th international conference on World Wide Web*, pp. 951–961, 2016.

Lyu, K. and Li, J. Gradient descent maximizes the margin of homogeneous neural networks. *arXiv preprint arXiv:1906.05890*, 2019.

Mei, S., Misiakiewicz, T., and Montanari, A. Mean-field theory of two-layers neural networks: dimension-free bounds and kernel limit. *arXiv preprint arXiv:1902.06015*, 2019.

Neal, R. M. Bayesian leaning for neural networks, 1996.

Olivier, C., Bernhard, S., and Alexander, Z. Semi-supervised learning. In *IEEE Transactions on Neural Networks*, volume 20, pp. 542–542. 2006.

Paterek, A. Improving regularized singular value decomposition for collaborative filtering. In *Proceedings of KDD cup and workshop*, volume 2007, pp. 5–8, 2007.

Rendle, S. Factorization machines. In *2010 IEEE International Conference on Data Mining*, pp. 995–1000. IEEE, 2010.

Rendle, S. Evaluation metrics for item recommendation under sampling. *arXiv preprint arXiv:1912.02263*, 2019.

Rendle, S., Krichene, W., Zhang, L., and Anderson, J. Neural collaborative filtering vs. matrix factorization revisited. *arXiv preprint arXiv:2005.09683*, 2020.

Saito, Y., Yaginuma, S., Nishino, Y., Sakata, H., and Nakata, K. Unbiased recommender learning from missing-not-at-random implicit feedback. In *Proceedings of the 13th International Conference on Web Search and Data Mining*, pp. 501–509, 2020.

Schnabel, T., Swaminathan, A., Singh, A., Chandak, N., and Joachims, T. Recommendations as treatments: Debiasing learning and evaluation. In *international conference on machine learning*, pp. 1670–1679. PMLR, 2016.

Shawe-Taylor, J., Cristianini, N., et al. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

Soudry, D., Hoffer, E., Nacson, M. S., Gunasekar, S., and Srebro, N. The implicit bias of gradient descent on separable data. *The Journal of Machine Learning Research*, 19(1):2822–2878, 2018.

Srebro, N. and Shraibman, A. Rank, trace-norm and max-norm. In *International Conference on Computational Learning Theory*, pp. 545–560. Springer, 2005.

Srebro, N., Rennie, J., and Jaakkola, T. Maximum-margin matrix factorization. *Advances in neural information processing systems*, 17:1329–1336, 2004.

Tropp, J. A. An introduction to matrix concentration inequalities. *Foundations and Trends in Machine Learning*, 8(1-2):1–230, 2015.

Vapnik, V. Estimation of dependences based on empirical data: Springer series in statistics (springer series in statistics), 1982.

Wei, C., Lee, J. D., Liu, Q., and Ma, T. Regularization matters: Generalization and optimization of neural nets vs their induced kernel. In *Advances in Neural Information Processing Systems*, pp. 9712–9724, 2019.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. Adversarial counterfactual learning and evaluation for recommender system. *Advances in Neural Information Processing Systems*, 33, 2020.

Xu, D., Ruan, C., Korpeoglu, E., Kumar, S., and Achan, K. Theoretical understandings of product embedding for e-commerce machine learning. In *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, pp. 256–264, 2021a.

Xu, D., Ye, Y., and Ruan, C. Understanding the role of importance weighting for deep learning. In *ICLR*. OpenReview.net, 2021b.

Yang, G. Scaling limits of wide neural networks with weight sharing: Gaussian process behavior, gradient independence, and neural tangent kernel derivation. *arXiv preprint arXiv:1902.04760*, 2019.

Yang, L., Cui, Y., Xuan, Y., Wang, C., Belongie, S., and Estrin, D. Unbiased offline recommender evaluation for missing-not-at-random implicit feedback. In *Proceedings of the 12th ACM Conference on Recommender Systems*, pp. 279–287, 2018.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. Understanding deep learning requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016.

Zhang, S., Yao, L., Sun, A., and Tay, Y. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys (CSUR)*, 52(1):1–38, 2019.