Matrix Factorization for Collaborative Filtering Is Just Solving an Adjoint Latent Dirichlet Allocation Model After All

FLORIAN WILHELM, inovex GmbH, Germany

Matrix factorization-based methods are among the most popular methods for collaborative filtering tasks with implicit feedback. The most effective of these methods do not apply sign constraints, such as non-negativity, to their factors. Despite their simplicity, the latent factors for users and items lack interpretability, which is becoming an increasingly important requirement. In this work, we provide a theoretical link between unconstrained and the interpretable non-negative matrix factorization in terms of the personalized ranking induced by these methods. We also introduce a novel, latent Dirichlet allocation-inspired model for recommenders and extend our theoretical link to also allow the interpretation of an unconstrained matrix factorization as an adjoint formulation of our new model. Our experiments indicate that this novel approach represents the unknown processes of implicit user-item interactions in the real world much better than unconstrained matrix factorization while being interpretable.

CCS Concepts: • **Information systems → Recommender systems**; **Collaborative filtering**; • **Computing methodologies → Factorization methods**; **Latent variable models**; **Latent Dirichlet allocation**; **Learning latent representations**.

ACM Reference Format:
Florian Wilhelm. 2021. Matrix Factorization for Collaborative Filtering Is Just Solving an Adjoint Latent Dirichlet Allocation Model After All. In *Fifteenth ACM Conference on Recommender Systems (RecSys '21), September 27-October 1, 2021, Amsterdam, Netherlands.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3460231.3474266

1 INTRODUCTION

Since the Netflix competition from 2006 to 2009, matrix factorization (MF)-based methods have been and still are among the most popular approaches to collaborative filtering problems. Even the emergence of deep learning could not fundamentally change this, as recent publications have shown the effectiveness of MF and even simpler methods over neural network-based approaches [5, 30].

The question of why MF-based methods in particular are so effective in finding a personalized ranking of items based on implicit user feedback remains largely unanswered. What are the assumptions and user model behind a general MF approach? How can the latent factors, i.e., the learned parameters, be interpreted especially if no sign constraints, e.g., non-negativity, have been imposed? With more and more tasks taken over by algorithms, the call for interpretability is getting louder and louder. While there are approaches such as non-negative matrix factorization (NMF) that offer possibilities for interpretation, they often cannot rival the performance of general MF methods without these constraints on the factors [21]. In a well-received article, Rudin [31] pointed out that interpretable models could potentially exist in many different domains that are just as accurate as their non-interpretable counterparts. What could such an interpretable model look like in the context of collaborative filtering?

In this paper, we focus on the task of creating a user-specific ranking for a set of items based on the implicit feedback of a set of users. Our work thus relies heavily on the definition of the Bayesian pair-wise ranking (BPR) criterion

Manuscript submitted to ACM

by Rendle et al. [29], which formalizes the task of finding a user-specific item ranking. We begin with a review and discussion of several variants of MF before introducing a novel, latent Dirichlet allocation (LDA)-inspired model for recommendation tasks (LDA4Rec) that is easy to interpret and to reason about. After that, we prove that the factors of unconstrained MF can be transformed into NMF factors as well as into a factorization adjoint to the presented LDA4Rec model while keeping the MF-induced personalized ranking constant. Consequently, the presented transformations allow the interpretation of the results from the presented MF methods in the context of the LDA4Rec model. Although the personalized ranking remains constant under these transformations, the MF-based methods and LDA4Rec may not necessarily find the same optimal solution because the optimization losses are different. We evaluate this by running several experiments on two public datasets comparing the presented MF variants and the novel LDA4Rec model in terms of their results using mean reciprocal rank, precision, and recall as metrics.

2 RELATED WORK

Already in the original work on LDA, Blei et al. [3] evaluate recommender systems as a use case for LDA. Treating documents as users, topics as cohorts of users, and words as items, the LDA model is trained on a set of fully observed users while for each unobserved user all but one of the movies are shown. Then the likelihood of the held-out movie compared to the others is eventually used to derive a user-specific ranking. This direct application of LDA to the domain of recommender systems is also used by Xie et al. [36] in their LDA-inspired probabilistic method. Our work differs from these in that two additional vectors of parameters are introduced into the original LDA formulation, adding an inductive bias to support the use case. One vector for the popularity of the items regularizes the item preferences over the user cohorts while another vector weights this regularization for each user thus indicating the conformity of the user with the item popularity.

MF and LDA are often considered together to derive collaborative methods that also include content information. Wang and Blei [34] propose a collaborative topic regression (CTRlda) model that combines a textual LDA model for the content information and a probabilistic matrix factorization to jointly explain the observed content and user ratings, respectively. A similar approach is proposed by Nikolenko [23] while Rao et al. [28] extend CTRlda using the special words with background (SWB) model [4] instead of LDA. As these methods use additional content information, they differ from LDA4Rec in this aspect while also not including the aforementioned additional parameters.

Zhang et al. [38] emphasize the interpretability of NMF for collaborative filtering and regard the latent user vector as an additive mixture of different user communities, i.e., cohorts. A similar, but more probabilistic NMF approach, rendering the mixture a distribution over cohorts of users, is presented by Hernando et al. [12]. Compared to LDA4Rec, these works also lack the proposed additional parameters in LDA4Rec similar to the previously mentioned works that apply LDA directly. The lacking interpretability of MF without non-negativity constraints is also recognized by Datta et al. [6] and addressed with a different approach than NMF. The authors propose a shadow model that learns a mapping from interpretable auxiliary features to the latent factors of MF. Therefore, their approach cannot be considered as pure collaborative filtering like LDA4Rec, since additional content information is used.

3 NOTATION AND TERMINOLOGY

In this section, we formalize the problem and establish a common notation, which is to some extent based on the work of Rendle et al. [29]. Matrices are denoted by capital letters X, transposed matrices with X^t, vectors by bold letters $\mathbf{x}$, sets by calligraphic letters $\mathcal{X}$, and the cardinality of a set by $|\mathcal{X}|$. The scalar product of two vectors $\mathbf{x}$ and $\mathbf{y}$ is denoted by $\langle \mathbf{x}, \mathbf{y} \rangle := \sum_{i=1}^{n} x_i y_i$ and the l_1-norm is denoted by $\|\mathbf{x}\|_1 := \sum_{i=1}^{n} |x_i|$, where n is the dimension of the vector space.

The Hadamard element-wise vector multiplication and division are denoted by $\odot$ and $\oslash$, respectively. A concatenation of two vectors $\mathbf{x}, \mathbf{z}$ is denoted by $[\mathbf{x}, \mathbf{z}]$ and $\mathbf{1}$ is the vector of all ones. The i-th row vector of a matrix X is denoted by $\mathbf{x}_i$ whereas the j-th column vector is expressed with the help of the Kleene star as $\mathbf{x}_{*j}$. The symbol $\mathbb{R}_{\geq 0}$ is used for non-negative real numbers.

Let $\mathcal{U}$ be the set of all users and $\mathcal{I}$ the set of all items. With $\mathcal{S} \subset \mathcal{U} \times \mathcal{I}$ we denote the set of implicit feedback from users $u \in \mathcal{U}$ having interacted with items $i \in \mathcal{I}$. Following the definition of Rendle et al. [29], the task of personalized ranking is to provide each user u with a personalized total ranking $\succcurlyeq_u$ on $\mathcal{I}$. In particular, since we assume that $\succcurlyeq_u$ is a total order, we have for $i, j \in \mathcal{I}$ with $i \neq j$ that either $i >_u j$ or $i <_u j$.

4 MATRIX FACTORIZATION

MF-based methods for collaborative filtering share the idea of approximating the sparse matrix of user-item interactions $X \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{I}|}$ by the product of two low-rank matrices $W \in \mathbb{R}^{|\mathcal{U}| \times |\mathcal{K}|}$ and $H \in \mathbb{R}^{|\mathcal{I}| \times |\mathcal{K}|}$, i.e.,

$$X \approx \hat{X} := WH^t,$$

where $\mathcal{K} = \{1, \ldots, |\mathcal{K}|\}$ is the index set of the latent dimensions. Derived from this general form, we will define the personalized score of a user u for an item i as

$$\hat{x}_{ui} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle + b_i, \tag{1}$$

where $b_i \in \mathbb{R}$ is an item bias. Adding an explicit item bias term has been shown to improve MF-based models in many studies [17, 26] and can be interpreted as the popularity of an item independent of a user's preferences. The personalized scores of a user then induce the personalized ranking $\succcurlyeq_u$ by virtue of $\hat{x}_{ui} \geq \hat{x}_{uj}$ for $i, j \in \mathcal{I}$. Note that in our implicit feedback scenario, there is no need for a user bias term b_u as the personalized ranking $\succcurlyeq_u$ would not change by definition.

The actual approximation depends on the optimization loss $L(X, \hat{X})$ and over the years many were derived, most notably SVD++ [16], which was used to win the Netflix price, WR-MF [14, 24] and PMF [32]. Since we are ultimately interested in an optimal ranking $\succcurlyeq_u$ rather than an approximation of the original matrix, the Bayesian Personalized Ranking (BPR) method was proposed by Rendle et al. [29] to directly reflect this task in its loss L and can be considered a differentiable analogy to Area Under the ROC Curve (AUC) optimization [29].

Despite the simple formulation (1) of MF, the actual interpretation of the latent vectors $\mathbf{w}_u$ and $\mathbf{h}_i$ is not as easy. The latent elements of an item i, i.e., h_{ik}, $k \in \mathcal{K}$, might quantify the prevalence of some latent feature in an item while the corresponding element of a user u, i.e., w_{uk}, quantifies the user's preference for this feature. The problem with this notion becomes apparent when considering negative elements, since, for example, a strong negative prevalence together with a negative preference can lead to a large positive term in the scalar product. This observation motivates the usage of MF methods that demand non-negativity for $\mathbf{w}_u$ and $\mathbf{h}_i$.

4.1 Non-Negative Matrix Factorization

The non-negative Matrix Factorization (NMF) was introduced by Lee and Seung [19, 20] as a method to learn parts of objects, which can then be combined again to form a whole. NMF differs from MF in (1) only in that we have $\mathbf{w}_u \in \mathbb{R}_{\geq 0}^{|\mathcal{K}|}$, $\mathbf{h}_i \in \mathbb{R}_{\geq 0}^{|\mathcal{K}|}$ and $b \in \mathbb{R}_{\geq 0}$. Using the example of faces, Lee and Seung [20] showed that NMF is able to learn localized features, e.g., eye area, which can then be used again to form a whole face by an additive mixture. Although the notion

of feature prevalences within items and user preferences for certain features translates well to NMF, in many practical applications the results achieved with NMF, unfortunately, fall short of those of MF [21].

A mathematically more rigorous interpretation is that NMF finds a $|\mathcal{K}|$ clustering of the column vectors of X, i.e., $\mathbf{x}_{*i}$ for $i \in \mathcal{I}$ in the space of users u. Ding et al. [8] prove that NMF with least squares optimization is mathematically equivalent to the minimization of K-means clustering with some mild relaxation with respect to the orthogonality of H.

Due to the constraints of NMF, we also have that the approximation matrix $\hat{X}$ is non-negative. While the non-negativity constraint is naturally fulfilled by the interaction matrix X, this restriction might still be unnecessary in practice. As a middle ground between MF and NMF, a variation of NMF was proposed which requires the non-negativity only from the matrix H and thus also the restriction for $\hat{X}$ is lifted.

4.2 Semi Non-negative Matrix Factorization

The Semi Non-negative Matrix Factorization (SNMF) is proposed by Ding et al. [9] to lift the non-negativity condition on $\hat{X} = WH^t$ by setting the non-negativity constraint only on H. Analogously to NMF, they also show that SNMF is a relaxation of K-means clustering. We can interpret $W = (\mathbf{w}_{*1}, \ldots, \mathbf{w}_{*|\mathcal{K}|})$ as the cluster centroids and $H = (\mathbf{h}_1, \ldots, \mathbf{h}_{|\mathcal{I}|})$ as the cluster indicators for each data point $\mathbf{x}_{*i}$. To the best knowledge of the author, SNMF finds very little to no application in the field of recommender systems, despite its interpretability.

5 LATENT DIRICHLET ALLOCATION

The latent Dirichlet allocation (LDA) is a generative statistical model, most often used in the context of natural language processing (NLP). It is an instance of a *topic model* in that it explains the observations by assuming a set of unobserved groups or topics where the observations within an assigned group share some common features. In the context of NLP, LDA postulates that each document is a mixture of a number of latent topics and that the frequency of each word within the document then depends on the frequency of this word within a topic and the mixture of topics.

We adapt the generative process of a smoothed LDA from Blei et al. [3] to our notation and context of users interacting with items. Given a set of items $i \in \mathcal{I}$ and $|\mathcal{K}|$ cohorts of users, each user $u \in \mathcal{U}$ has $\mathcal{S}_u = \{1, \ldots, |\mathcal{S}_u|\}$ interactions, assuming the following generative process:

1. Choose $\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$ for $u \in \mathcal{U}$.
2. Choose $\boldsymbol{\varphi}_k \sim \text{Dirichlet}(\boldsymbol{\beta})$ for $k \in \mathcal{K}$.
3. For each user $u \in \mathcal{U}$ and his or her interactions $s \in \mathcal{S}_u$:
   (a) Choose a cohort $z_{us} \sim \text{Categorical}(\boldsymbol{\theta}_u)$.
   (b) Choose an item $i_{us} \sim p(i_{us}|\boldsymbol{\varphi}_{z_{us}}) := \text{Categorical}(\boldsymbol{\varphi}_{z_{us}})$.

The hyperparameters $\boldsymbol{\alpha} \in \mathbb{R}_{>0}^{|\mathcal{K}|}$ and $\boldsymbol{\beta} \in \mathbb{R}_{>0}^{|\mathcal{I}|}$ are typically chosen to be sparse, i.e., components smaller than 1, in order to favor users belonging mainly to a single or only a few cohorts and cohorts with an item distribution that has low entropy, respectively. Also note that the actual number of user interactions S_u is not part of the generation process and is therefore taken as given. The subscript is dropped for the ease of notation where it seems reasonable. The graphical model corresponding to the generative process is depicted in Figure 1.

With respect to interpretability, the presented generative process matches the intuition about cohorts of users, i.e., topics in the original LDA, sharing similar item preferences quite well. Each user is then probabilistically assigned to some cohorts, assuming a high probability for a single cohort. With regard to the modeling of preferences in cohorts, some caveats arise.

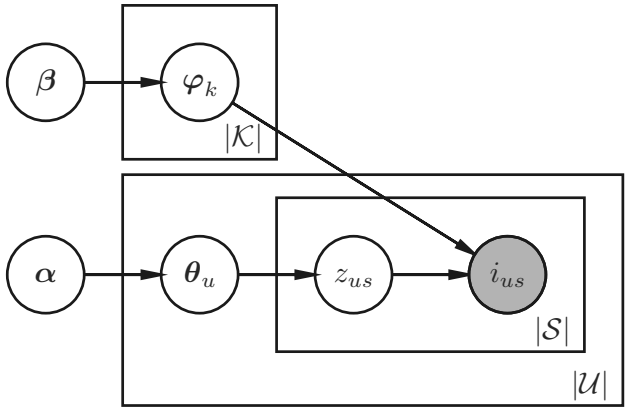

Fig. 1. Graphical model representation of the smoothed LDA model adapted to a collaborative filtering problem [3]. The boxes are called *plates* representing replicates. The upper box represents the latent cohorts whereas the lower outer plate represents the users, while the inner plate represents the repeated choice of cohorts and items within the interactions of a user.

First of all, the item preferences within the cohorts are only connected by the allocation of the users and not directly by the items. Taking up the point made for the item bias in Section 4, we argue that the probability of an item within different cohorts should also depend on the item itself. According to intuition, the movie Pulp Fiction, for example, should exhibit within each user cohort k a relatively high probability φ_{ki} of being interacted with, compared to other movies within the same cohort due to its popularity. This limitation can be remedied by including an explicit item bias in the generative process that changes the categorical distribution in Step 3b accordingly.

Another caveat becomes clearer after considering the relationship of LDA and NMF. NMF with Kullback-Leibler divergence is an approximation of the evidence lower bound (ELBO) of LDA with symmetric Dirichlet priors [7]. The connection becomes also apparent, in a mathematically somewhat imprecise way, when we consider $\mathbf{w}_u$ and $\mathbf{h}_{*k}$ of NMF as not l_1-normalized $\boldsymbol{\theta}_u$ and not l_1-normalized $\boldsymbol{\varphi}_k$ in LDA, respectively. The fact that $\mathbf{w}_u$ in NMF is not normalized allows cohort preferences to be weighted against item popularity depending on the user u. Intuitively, we demand this flexibility from a model as users differ in whether they are more conformist or more individual in their perception of the popularity of items. To take this into account, we will introduce a user-specific weighting factor for the popularity of items, thus also resolving this caveat.

In total, we have identified two inductive biases that traditional LDA is missing and present a modified LDA model for recommender systems that incorporates these.

5.1 LDA for Recommender Systems

The generative process of a traditional LDA is modified by incorporating the item popularity δ_i and the user's conformity λ_u. This results in the generative process of an LDA for recommender systems (LDA4Rec):

1. Choose $\boldsymbol{\theta}_u \sim \text{Dirichlet}(\boldsymbol{\alpha})$ and $\lambda_u \sim \text{LogNormal}(\mu_\lambda, \sigma_\lambda^2)$ for $u \in \mathcal{U}$.
2. Choose $\delta_i \sim \text{LogNormal}(\mu_\delta, \sigma_\delta^2)$ for $i \in \mathcal{I}$.
3. Choose $\varphi_{ki} \sim \text{LogNormal}(\mu_\varphi, \sigma_\varphi^2)$ for $k \in \mathcal{K}$ and $i \in \mathcal{I}$.

4. For each user $u \in \mathcal{U}$ and his or her interactions $s \in \mathcal{S}_u$:
   (a) Choose a cohort $z_{us} \sim \text{Categorical}(\boldsymbol{\theta}_u)$.
   (b) Choose an item $i_{us} \sim p(i_{us}|\boldsymbol{\varphi}_{z_{us}}, \delta_i, \lambda_u) := \text{Categorical}(\|\mathbf{c}\|_1{}^{-1}\mathbf{c})$ with $\mathbf{c} = \boldsymbol{\varphi}_{z_{us}} + \lambda_u \cdot \boldsymbol{\delta}$.

The hyperparameters μ_*, σ_*^2 can be used to incorporate prior knowledge about the relations of $\boldsymbol{\lambda}$, $\boldsymbol{\delta}$ and $\boldsymbol{\varphi}_k$. In Step 4b, we see that the probability of a user interacting with an item not only depends on the preference assigned to the item by the cohort, i.e., $\boldsymbol{\varphi}_{z_{us}}$, but also the popularity δ_i of the item and the conformity λ_u of the user to the general popularity. The graphical model of LDA4Rec is illustrated in Figure 2.

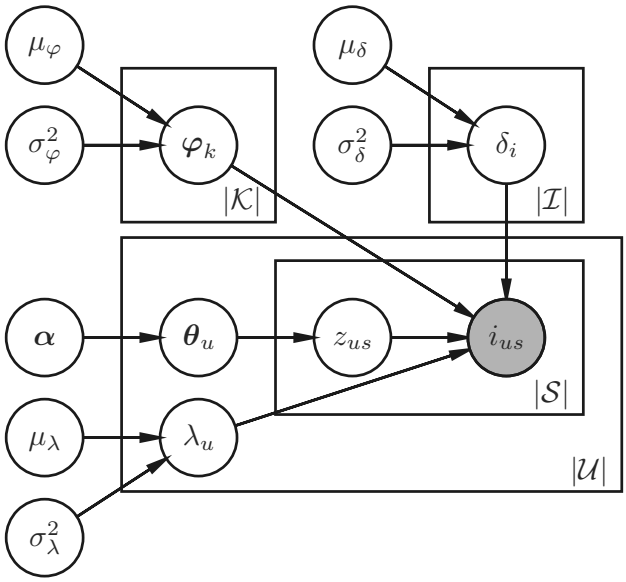

Fig. 2. Graphical model representation of the proposed LDA4Rec model that also incorporates the item popularity δ_i as well as the user's conformity λ_u to the general popularity.

We show now that MF, as introduced in Section 4, has an adjoint formulation that corresponds to the parameters $\boldsymbol{\varphi}_k$, $\boldsymbol{\theta}_u$, δ_i and λ_u of LDA4Rec. Finally, this allows us to intuitively interpret the latent factors of MF.

5.2 Adjoint LDA4Rec Formulation of Matrix Factorization

We derive the adjoint LDA4Rec formulation of MF using two steps. First, we show that the personalized ranking given by MF can be reformulated as NMF. Assuming an NMF, we then transform the latent vectors of users and items into an l_1-normalized representation, which can be interpreted as parameters of the categorical distributions in Step 4b of the generating process of LDA4Rec.

Lemma. *Given personalized ranking scores $\hat{x}_{ui} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle + b_i$ for users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ with $\mathbf{w}_u \in \mathbb{R}^{|\mathcal{K}|}$, $\mathbf{h}_i \in \mathbb{R}^{|\mathcal{K}|}$ and $b_i \in \mathbb{R}$ that induce a total ranking $\geqslant_u$ for all users. Then there exists $x'_{ui} = \langle \mathbf{w}'_u, \mathbf{h}'_i \rangle + b'_i$ with $\mathbf{w}'_u \in \mathbb{R}_{\geq 0}^{|\mathcal{K}'|}$, $\mathbf{h}'_i \in \mathbb{R}_{\geq 0}^{|\mathcal{K}'|}$ and $b'_i \in \mathbb{R}_{\geq 0}$ that induce the same total ranking $\geqslant_u$ for all users.*

Proof. We define $\mathbf{w}'_u = \left[\mathbf{w}^+{}_u, \mathbf{w}^-{}_u\right]$ where

$$\mathbf{w}^+{}_{uk} = \begin{cases} \mathbf{w}_{uk} & \text{if } \mathbf{w}_{uk} \geq 0 \\ 0 & \text{otherwise} \end{cases} \qquad \text{and} \qquad \mathbf{w}^-{}_{uk} = \begin{cases} -\mathbf{w}_{uk} & \text{if } \mathbf{w}_{uk} < 0 \\ 0 & \text{otherwise} \end{cases},$$

for $k \in \mathcal{K}$. Also, we define analogously $\mathbf{h}'_i = [\mathbf{h}_i + \mathbf{s}, -\mathbf{h}_i + \mathbf{s}]$ with $\mathbf{s} = (s_i)_{i \in \mathcal{I}}$, $s_i = \max_{k \in \mathcal{K}} |h_{ik}|$ and $b'_i = b_i + \max_{i \in \mathcal{I}} |b_i|$. By construction, we have $\mathbf{w}'_u \in \mathbb{R}^{|\mathcal{K}'|}_{\geq 0}$, $\mathbf{h}'_i \in \mathbb{R}^{|\mathcal{K}'|}_{\geq 0}$ and $b'_i \in \mathbb{R}_{\geq 0}$ with $\mathcal{K}' = \{1, \ldots, 2|\mathcal{K}|\}$. We also have by construction that $\langle \mathbf{w}'_u, \mathbf{h}'_i \rangle = \langle \mathbf{w}^+_u, \mathbf{h}_i \rangle + \langle \mathbf{w}^+_u, \mathbf{s} \rangle - \langle \mathbf{w}^-_u, \mathbf{h}_i \rangle + \langle \mathbf{w}^-_u, \mathbf{s} \rangle = \langle \mathbf{w}^+_u - \mathbf{w}^-_u, \mathbf{h}_i \rangle + \langle \mathbf{w}^+_u + \mathbf{w}^-_u, \mathbf{s} \rangle = \langle \mathbf{w}_u, \mathbf{h}_i \rangle + \langle \mathbf{w}^+_u + \mathbf{w}^-_u, \mathbf{s} \rangle$. This is now applied to conclude that

$$\begin{aligned} x'_{ui} \geq x'_{uj} &\iff \langle \mathbf{w}'_u, \mathbf{h}'_i \rangle + b'_i \geq \langle \mathbf{w}'_u, \mathbf{h}'_j \rangle + b'_j \\ &\iff \langle \mathbf{w}_u, \mathbf{h}_i \rangle + \langle \mathbf{w}^+_u + \mathbf{w}^-_u, \mathbf{s} \rangle + b_i + \max_{i \in \mathcal{I}} |b_i| \geq \langle \mathbf{w}_u, \mathbf{h}_j \rangle + \langle \mathbf{w}^+_u + \mathbf{w}^-_u, \mathbf{s} \rangle + b_j + \max_{i \in \mathcal{I}} |b_i| \\ &\iff \langle \mathbf{w}_u, \mathbf{h}_i \rangle + b_i \geq \langle \mathbf{w}_u, \mathbf{h}_j \rangle + b_j \\ &\iff \hat{x}_{ui} \geq \hat{x}_{uj}. \end{aligned}$$

Subsequently, x'_{ui} induces the same total ranking $\geqslant_u$ as $\hat{x}_{ui}$. □

Theorem. *Given personalized ranking scores $\hat{x}_{ui} = \langle \mathbf{w}_u, \mathbf{h}_i \rangle + b_i$ for users $u \in \mathcal{U}$ and items $i \in \mathcal{I}$ with $\mathbf{w}_u \in \mathbb{R}^{|\mathcal{K}|}$, $\mathbf{h}_i \in \mathbb{R}^{|\mathcal{K}|}$ and $b_i \in \mathbb{R}$ that induce a total ranking $\geqslant_u$ for all users. Then there exists $x'_{ui} = \langle \mathbf{v}_u, \mathbf{g}_i(u) \rangle$ where $\mathbf{v}_u \in \mathbb{R}^{|\mathcal{K}|}_{\geq 0}$ and $\mathbf{g}_i(u) \in \mathbb{R}^{|\mathcal{K}|}_{\geq 0}$ with $\|\mathbf{v}_u\|_1 = 1$ and $\|\mathbf{g}_i(u)\|_1 = 1$, such that the same total ranking $\geqslant_u$ is induced for all users.*

Proof. Without loss of generality, we assume $\mathbf{w}_u \in \mathbb{R}^{|\mathcal{K}|}_{\geq 0}$, $\mathbf{h}_i \in \mathbb{R}^{|\mathcal{K}|}_{\geq 0}$ and $b_i \in \mathbb{R}_{\geq 0}$ by virtue of the lemma. We define

$$\mathbf{v}_u = c_u^{-1} \mathbf{w}_u \odot \mathbf{n}_\mathbf{u}, \qquad \mathbf{g}_i(u) = (\mathbf{h}_i + t_u^{-1} b_i \mathbf{1}) \oslash \mathbf{n}_u,$$

where $\mathbf{n}_u = (n_{uk})_{k \in \mathcal{K}}$ with $n_{uk} = \sum_{i \in \mathcal{I}} h_{ik} + t_u^{-1} b_i$, $t_u = \|\mathbf{w}_u\|_1$ and $c_u = \langle \mathbf{w}_u, \mathbf{n}_u \rangle$. We can neglect the pathological cases, i.e., $t_u = 0$ and $n_{uk} = 0$, as in the former case we have a trivial solution and $\geqslant_u$ only depends on $\mathbf{b}$ whereas in the latter case, we have $\sum_{i \in \mathcal{I}} h_{ik} = 0$ and thus the latent vector $(h_{ik})_{i \in \mathcal{I}}$ could just be removed. By construction, we have now that $\|\mathbf{v}_u\|_1 = 1$ and $\|\mathbf{g}_i(u)\|_1 = 1$. We can thus conclude that

$$\begin{aligned} x'_{ui} \geq x'_{uj} &\iff \langle \mathbf{v}_u, \mathbf{g}_i(u) \rangle \geq \langle \mathbf{v}_u, \mathbf{g}_j(u) \rangle \\ &\iff \langle c_u^{-1} \mathbf{w}_u \odot \mathbf{n}_\mathbf{u}, (\mathbf{h}_i + t_u^{-1} b_i \mathbf{1}) \oslash \mathbf{n}_u \rangle \geq \langle c_u^{-1} \mathbf{w}_u \odot \mathbf{n}_\mathbf{u}, (\mathbf{h}_j + t_u^{-1} b_j \mathbf{1}) \oslash \mathbf{n}_u \rangle \\ &\iff \langle \mathbf{w}_u, \mathbf{h}_i + t_u^{-1} b_i \mathbf{1} \rangle \geq \langle \mathbf{w}_u, \mathbf{h}_j + t_u^{-1} b_j \mathbf{1} \rangle \\ &\iff \langle \mathbf{w}_u, \mathbf{h}_i \rangle + \langle \mathbf{w}_u, t_u^{-1} b_i \mathbf{1} \rangle \geq \langle \mathbf{w}_u, \mathbf{h}_j \rangle + \langle \mathbf{w}_u, t_u^{-1} b_j \mathbf{1} \rangle \\ &\iff \langle \mathbf{w}_u, \mathbf{h}_i \rangle + b_i \geq \langle \mathbf{w}_u, \mathbf{h}_j \rangle + b_j \\ &\iff \hat{x}_{ui} \geq \hat{x}_{uj}. \end{aligned}$$

Consequently, x'_{ui} induces the same total ranking $\geqslant_u$ as $\hat{x}_{ui}$. □

In the light of these constructive proofs and noting that SNMF was an intermediate step in the proof of the lemma, we can also make a statement about the expressive power of MF, NMF, SNMF, and LDA4Rec. In particular, we have seen that each latent dimension indexed by k of MF is split up into two corresponding dimensions in the NMF representation. Following our previous interpretation, those dimensions stand for cohorts having complementary item preferences.

Corollary. *The expressive power, i.e., the number of possible total rankings $\geqslant_u$ that can be encoded, of MF is twice as high as in the case of NMF for a given latent vector length $|\mathcal{K}|$. LDA4Rec has the same expressive power as NMF and the expressive power of MF is equivalent to the expressive power of SNMF.*

It is important to note here that we have only proved that the personalized ranking $\geqslant_u$ remains constant under some transformations that allow us to express an MF as NMF or an adjoint LDA4Rec formulation. Since $\geqslant_u$ is eventually the result of an optimization problem with some loss function L, e.g., BPR for MF or likelihood for LDA4Rec, we make no statement about maintaining the optimality of some solution under these transformations.

6 EVALUATION

To support our theoretical considerations with empirical results, several experiments were conducted with real-world datasets. The source code of our implementation and the detailed results of all experiments are available on Github[1].

6.1 Datasets & Evaluation Metrics

For our experiments, three different datasets were used. MovieLens-1M encompasses approximately 1 million movie ratings across 6,040 users and 3,706 movies while Movielens-100K has roughly 100 thousand interactions across 610 users and 9,724 movies [11]. Goodbooks has approximately 6 million interactions across 53,425 users and 10,001 books [37]. We split these datasets randomly into train, validation, and test sets using 90% of interactions for training, and 5% each for validation and testing. The explicit feedback of these datasets was treated as implicit, i.e., the various user ratings were converted to 1 representing an interaction while no rating means no interaction. Also, we limited the maximum number of interactions that a single user might have to 200 to avoid results that are skewed towards users with a high number of interactions due to our random split. This reduces the number of interactions in MovieLens-1M to approximately 661 thousand and in MovieLens-100K to 60 thousand while the number of interactions in Goodbooks is unaffected.

As evaluation metrics, we use the mean reciprocal rank (MRR), precision at 10 (Prec@10), and recall at 10 (Recall@10) to measure the quality of our models. We define Prec@10 as the fraction of known positives in the first 10 positions of the ranked list of results and Recall@10 as the number of known positives in the first 10 positions divided by the total number of known positives.

6.2 Experiments

6.2.1 Comparison of the Different Variants of Matrix Factorization. Despite the theoretical results from Subsection 5.2, which allow us to transform a personalized ranking solution found through MF into an NMF formulation, this does not necessarily mean that a solution found through direct application of NMF has the same quality. For this reason, we implemented MF, NMF, and SNMF using BPR as loss function and the Adam optimizer [15]. The implementations of NMF and SNMF differ from MF only in that they restrict the corresponding parameters to non-negative values using the sigmoid function. Our implementation heavily relies on Spotlight [18] and PyTorch [25]. In our experiment various batch sizes and at last 3 different seeds as random initialization are used. In order to provide a baseline, we also implemented a purely popularity-based recommender (Pop).

[1] https://github.com/florianwilhelm/lda4rec

6.2.2 Transformation of Matrix Factorization to the Adjoint LDA4Rec Formulation. Although we have mathematically proven in Subsection 5.2 that an MF solution can be transformed to NMF and subsequently also to an adjoint LDA formulation, floating-point arithmetic may pose challenges in a practical application. To follow up on this, we implemented the presented transformations as an optional preprocessing step before the evaluation. This allows us to evaluate a solution obtained from MF directly and after the transformation in order to compare the resulting personalized rankings $\geqslant_u$, which should be equivalent in theory.

6.2.3 Comparison of LDA4Rec to Matrix Factorization. We implemented the LDA4Rec model as presented in Subsection 5.1 with the help of the Pyro deep universal probabilistic programming framework [2]. To cope with the high dimensionality low-sample size setting, we decided for a stochastic variational inference (SVI) [13] approach using the Adam optimizer [15]. Due to the presence of discrete latent variables, a trace implementation of ELBO-based SVI [27, 35] with exhaustive enumeration over discrete sample sites was chosen. To predict the personalized ranking scores $\hat{x}_{ui}$, we sampled items from the posterior predictive distribution [10] and counted the occurrences to obtain a personalized ranking. Ties were broken by adding a small non-negative random number.

Our implementation and thus the evaluation of LDA4Rec turned out to be several orders of magnitude slower than the MF-based methods. For this reason, the experiments comparing LDA4Rec to MF-based methods were performed on the smaller MovieLens-100K dataset. While the variational inference makes the training process quite fast, the bottleneck of LDA4Rec is the prediction of the personalized rankings for which we need a high number of samples per user to compute a stable ranking. Thus for each user 10,000 items were sampled.

6.3 Results

6.3.1 Comparison of the Different Variants of Matrix Factorization. Table 1 shows the results of our first experiment comparing various variants of matrix factorization. For each metric, we report the value on the test set corresponding to the best value on the validation set for a model and its set of hyperparameters. Comparing the results with a fixed latent dimensionality of $|\mathcal{K}|$, we see that MF outperforms SNMF and NMF for low $|\mathcal{K}|$. Starting at $|\mathcal{K}| = 32$, SNMF surpasses MF and achieves the overall best results at $|\mathcal{K}| = 64$ by a small margin. Thus, the non-negativity constraints on H of SNMF that make it interpretable appear to cause a positive regularization effect.

For NMF we have a completely different picture, here we see results worse and slightly better than the popularity baseline for low values and high values of $|\mathcal{K}|$, respectively. From the theoretical results established in Subsection 5.2, we know that an MF result of latent dimensionality $|\mathcal{K}|$ could be represented as an NMF result with twice the latent dimensionality but our experiments show that a solution of the same quality is not found.

6.3.2 Transformation of Matrix Factorization to the Adjoint LDA4Rec Formulation. Our evaluations show that the personalized ranking scores of a given user obtained from MF and from its adjoint LDA4Rec formulation lead to equivalent rankings except for a small number of inversions due to numerical reasons. The rounding errors of floating-point arithmetic cause on some occasions that if $\hat{x}_{ui} > \hat{x}_{uj}$ where $|\hat{x}_{ui} - \hat{x}_{uj}| \leq \epsilon$, we have $x'_{ui} < x'_{uj}$ with $|x'_{ui} - x'_{uj}| \leq \epsilon$ for some small value ϵ using the notation established in Subsection 5.2. Statistically, across all users, these inversions have no impact on the results as the overall metrics MRR@10, Prec@10, and Recall@10 are equivalent in accordance with our theoretical results.

6.3.3 Comparison of LDA4Rec to Matrix Factorization. The benchmark results in Table 2 show that LDA4Rec outperforms MF in all but one metric, requiring a much lower latent dimensionality. Although the expressiveness of MF is

Table 1. Comparison of different variants of matrix factorization with varying number of latent parameters $|\mathcal{K}|$.

		Goodbooks			MovieLens-1M		
$\|\mathcal{K}\|$	Model	MRR@10	Prec@10	Recall@10	MRR@10	Prec@10	Recall@10
	Pop	0.023918	0.027079	0.047867	0.033488	0.033084	0.065908
4	NMF	0.014388	0.022056	0.038420	0.032927	0.033031	0.066281
	SNMF	0.036186	0.040912	0.072584	0.046642	0.046211	0.097219
	MF	**0.038901**	**0.044045**	**0.079124**	**0.050495**	**0.048702**	**0.103090**
8	NMF	0.015121	0.019261	0.033310	0.033445	0.033191	0.066516
	SNMF	0.042435	0.047185	0.085326	0.053639	0.052028	0.108542
	MF	**0.044683**	**0.049835**	**0.090115**	**0.058240**	**0.057044**	**0.119924**
16	NMF	0.019945	0.026436	0.046623	0.033461	0.033351	0.066191
	SNMF	0.049671	0.055800	0.101652	0.062695	0.061526	0.130733
	MF	**0.050875**	**0.057127**	**0.103747**	**0.063849**	**0.062131**	**0.131973**
32	NMF	0.028766	0.028268	0.050012	0.033223	0.033155	0.064652
	SNMF	0.055048	0.062215	0.113179	**0.068511**	**0.066453**	**0.141667**
	MF	**0.056080**	**0.062841**	**0.114629**	0.064506	0.064888	0.138600
48	NMF	0.032190	0.033548	0.060065	0.032996	0.033084	0.066660
	SNMF	0.058595	**0.066861**	**0.122292**	**0.068369**	**0.068143**	**0.146653**
	MF	**0.058730**	0.066321	0.121440	0.067427	0.065777	0.143905
64	NMF	0.034171	0.039272	0.070492	0.032925	0.032978	0.066924
	SNMF	**0.061561**	**0.070156**	**0.128119**	**0.069775**	**0.069050**	**0.151497**
	MF	0.060261	0.068837	0.126254	0.067474	0.066489	0.145744

Table 2. Comparison of MF to LDA4Rec with varying number of latent parameters $|\mathcal{K}|$.

Model	$\|\mathcal{K}\|$	MRR@10	Prec@10	Recall@10
Pop		0.037069	0.031148	0.072635
MF	2	0.041071	0.033151	0.075098
	4	**0.049304**	0.041530	0.099996
	8	0.047378	0.042987	0.110723
	16	0.046867	0.045173	0.111719
	32	0.048462	**0.049909**	**0.116455**
LDA4Rec	2	0.052898	0.040984	0.105045
	4	**0.066236**	**0.048816**	**0.130728**
	8	0.053524	0.045902	0.119682
	16	0.058407	0.046995	0.125783
	32	0.058738	0.044991	0.115292

twice as high as of LDA4Rec given the same latent dimensionality, the results of LDA4Rec for $|\mathcal{K}| = 4$ are an indication that the LDA4Rec model better represents reality due to its inductive biases.

7 CONCLUSION

From a theoretical point of view, we have discussed several variants of matrix factorization, i.e., MF, SNMF, NMF, and introduced the novel and interpretable LDA4Rec model, which extends the traditional LDA by incorporating parameters

for the popularity of items and conformity of users. We have proven that the personalized ranking induced by MF can be transformed so that the same personalized ranking is induced by NMF as well as by an adjoint formulation corresponding to the parameters of LDA4Rec. The adjoint LDA4Rec formulation of an MF allows easy interpretation of its parameters without sacrificing accuracy.

In several experiments, we have shown that SNMF performs slightly better than MF in same cases and is interpretable at the same time. Our evaluations also show that the result obtained by directly solving LDA4Rec outperforms MF with BPR loss while being more interpretable. Our empirical results combined with the derivation of LDA4Rec as a mathematical model suggest that its generative process represents reality well and and thus provides means to interpret the results of traditional MF-based methods.

Following on from this and assuming that the unknown, real-world process behind implicit user feedback is actually well represented by LDA4Rec, some conclusion about the effectiveness of Neural Collaborative Filtering (NCF) can also be drawn. NCF replaces the scalar product of MF with a learned similarity, e.g., using a multi-layer perceptron (MLP). Rendle et al. [30] show in a reproducibility paper that the scalar product outperforms several NCF-based methods and that it should thus be the default choice for combining embeddings, i.e., vectors of the latent factors. Similar results demonstrating the effectiveness of simple factorization-based models are shown by Dacrema et al. [5]. Our work underpins these findings as the scalar product of embeddings can be interpreted as a mixture of several preferences thus explaining its effectiveness. Since learning a multiplication and also a scalar product is possible in theory [22] but proves difficult in practice [1, 30, 33] for an MLP, MF-based methods will continue to have an advantage over NCF under this assumption.

REFERENCES

[1] Alex Beutel, Paul Covington, Sagar Jain, Can Xu, Jia Li, Vince Gatto, and Ed H. Chi. 2018. Latent Cross: Making Use of Context in Recurrent Recommender Systems. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, Marina Del Rey CA USA, 46–54. https://doi.org/10.1145/3159652.3159727

[2] Eli Bingham, Jonathan P. Chen, Martin Jankowiak, Fritz Obermeyer, Neeraj Pradhan, Theofanis Karaletsos, Rohit Singh, Paul Szerlip, Paul Horsfall, and Noah D. Goodman. 2018. Pyro: Deep Universal Probabilistic Programming. *arXiv:1810.09538 [cs, stat]* (Oct. 2018). http://arxiv.org/abs/1810.09538 arXiv: 1810.09538.

[3] David M Blei, Andres Y. Ng, and Michael I Jordan. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research* 3 (2003), pp. 993–1022. Issue 4-5.

[4] Chaitanya Chemudugunta, Padhraic Smyth, and Mark Steyvers. 2007. Modeling General and Specific Aspects of Documents with a Probabilistic Topic Model. *Advances in Neural Information Processing Systems* 19 (2007). https://proceedings.neurips.cc/paper/2006/file/ec47a5de1ebd60f559fee4afd739d59b-Paper.pdf

[5] Maurizio Ferrari Dacrema, Paolo Cremonesi, and Dietmar Jannach. 2019. Are We Really Making Much Progress? A Worrying Analysis of Recent Neural Recommendation Approaches. *Proceedings of the 13th ACM Conference on Recommender Systems - RecSys '19* (2019), 101–109. https://doi.org/10.1145/3298689.3347058 arXiv: 1907.06902.

[6] Anupam Datta, Sophia Kovaleva, Piotr Mardziel, and Shayak Sen. 2018. Latent Factor Interpretations for Collaborative Filtering. *arXiv:1711.10816 [cs]* (April 2018). http://arxiv.org/abs/1711.10816 arXiv: 1711.10816.

[7] Thiago de Paulo Faleiros and Alneu de Andrade Lopes. 2016. On the equivalence between algorithms for Non-negative Matrix Factorization and Latent Dirichlet Allocation. In *24 th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*.

[8] Chris Ding, Xiaofeng He, and Horst D. Simon. 2005. On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. In *Proceedings of the 2005 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 606–610. https://doi.org/10.1137/1.9781611972757.70

[9] C.H.Q. Ding, Tao Li, and M.I. Jordan. 2010. Convex and Semi-Nonnegative Matrix Factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32, 1 (Jan. 2010), 45–55. https://doi.org/10.1109/TPAMI.2008.277

[10] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. 2004. *Bayesian Data Analysis* (2nd ed. ed.). Chapman and Hall/CRC.

[11] F. Maxwell Harper and Joseph A. Konstan. 2016. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4 (Jan. 2016), 1–19. https://doi.org/10.1145/2827872

[12] Antonio Hernando, Jesús Bobadilla, and Fernando Ortega. 2016. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. *Knowledge-Based Systems* 97 (April 2016), 188–202. https://doi.org/10.1016/j.knosys.2015.12.018

[13] Matthew D. Hoffman, David M. Blei, Chong Wang, and John Paisley. 2013. Stochastic Variational Inference. *Journal of Machine Learning Research* 14, 4 (2013), 1303–1347. http://jmlr.org/papers/v14/hoffman13a.html

[14] Yifan Hu, Yehuda Koren, and Chris Volinsky. 2008. Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*. Ieee, IEEE, Pisa, Italy, 263–272. https://doi.org/10.1109/ICDM.2008.22

[15] Diederik Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. *International Conference on Learning Representations* (12 2014).

[16] Yehuda Koren. 2009. The bellkor solution to the netflix grand prize. *Netflix prize documentation* 81, 2009 (2009), 1–10.

[17] Yehuda Koren and Robert Bell. 2015. Advances in Collaborative Filtering. In *Recommender Systems Handbook*, Francesco Ricci, Lior Rokach, and Bracha Shapira (Eds.). Springer US, Boston, MA, 77–118. https://doi.org/10.1007/978-1-4899-7637-6_3

[18] Maciej Kula. 2017. Spotlight. https://github.com/maciejkula/spotlight.

[19] Daniel Lee and Hyunjune Seung. 2001. Algorithms for Non-negative Matrix Factorization. *Adv. Neural Inform. Process. Syst.* 13 (02 2001).

[20] Daniel D. Lee and H. Sebastian Seung. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature* 401, 6755 (Oct. 1999), 788–791. https://doi.org/10.1038/44565

[21] Joonseok Lee, Mingxuan Sun, and Guy Lebanon. 2012. A Comparative Study of Collaborative Filtering Algorithms. *arXiv:1205.3193 [cs, stat]* (May 2012). http://arxiv.org/abs/1205.3193 arXiv: 1205.3193.

[22] Henry W. Lin, Max Tegmark, and David Rolnick. 2017. Why does deep and cheap learning work so well? *Journal of Statistical Physics* 168, 6 (Sept. 2017), 1223–1247. https://doi.org/10.1007/s10955-017-1836-5 arXiv: 1608.08225.

[23] Sergey Nikolenko. 2015. SVD-LDA: Topic Modeling for Full-Text Recommender Systems. In *Advances in Artificial Intelligence and Its Applications*, Obdulia Pichardo Lagunas, Oscar Herrera Alcántara, and Gustavo Arroyo Figueroa (Eds.). Vol. 9414. Springer International Publishing, Cham, 67–79. https://doi.org/10.1007/978-3-319-27101-9_5 Series Title: Lecture Notes in Computer Science.

[24] Rong Pan, Yunhong Zhou, Bin Cao, Nathan N. Liu, Rajan Lukose, Martin Scholz, and Qiang Yang. 2008. One-Class Collaborative Filtering. In *2008 Eighth IEEE International Conference on Data Mining*. IEEE, Pisa, Italy, 502–511. https://doi.org/10.1109/ICDM.2008.16

[25] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Köpf, Edward Yang, Zach DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. arXiv:1912.01703 [cs.LG]

[26] Arkadiusz Paterek. 2007. Improving regularized singular value decomposition for collaborative filtering. *Proceedings of KDD cup and workshop* vol. 2007 (2007), pp. 5–8.

[27] Rajesh Ranganath, Sean Gerrish, and David M. Blei. 2013. Black Box Variational Inference. *arXiv:1401.0118 [cs, stat]* (Dec. 2013). http://arxiv.org/abs/1401.0118 arXiv: 1401.0118.

[28] Vidyadhar Rao, KV Rosni, and Vineet Padmanabhan. 2017. Divide and Transfer: Understanding Latent Factors for Recommendation Tasks. In *RecSysKTL*. 1–8.

[29] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian Personalized Ranking from Implicit Feedback. (01 2009), 452–461.

[30] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. 2020. Neural Collaborative Filtering vs. Matrix Factorization Revisited. In *Fourteenth ACM Conference on Recommender Systems* (Virtual Event, Brazil) *(RecSys '20)*. Association for Computing Machinery, New York, NY, USA, 240–248. https://doi.org/10.1145/3383313.3412488

[31] Cynthia Rudin. 2019. Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead. *Nature Machine Intelligence* 1, 5 (May 2019), 206–215. https://doi.org/10.1038/s42256-019-0048-x

[32] Ruslan Salakhutdinov and Andriy Mnih. 2007. Probabilistic Matrix Factorization. In *Proceedings of the 20th International Conference on Neural Information Processing Systems* (Vancouver, British Columbia, Canada) *(NIPS'07)*. Curran Associates Inc., Red Hook, NY, USA, 1257–1264.

[33] Andrew Trask, Felix Hill, Scott E Reed, Jack Rae, Chris Dyer, and Phil Blunsom. 2018. Neural Arithmetic Logic Units. (2018), 10.

[34] Chong Wang and David M. Blei. 2011. Collaborative topic modeling for recommending scientific articles. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11*. ACM Press, San Diego, California, USA, 448. https://doi.org/10.1145/2020408.2020480

[35] David Wingate and Theophane Weber. 2013. Automated Variational Inference in Probabilistic Programming. *arXiv:1301.1299 [cs, stat]* (Jan. 2013). http://arxiv.org/abs/1301.1299 arXiv: 1301.1299.

[36] WenBo Xie, Qiang Dong, and Hui Gao. 2014. A Probabilistic Recommendation Method Inspired by Latent Dirichlet Allocation Model. *Mathematical Problems in Engineering* 2014 (2014), 1–10. https://doi.org/10.1155/2014/979147

[37] Zygmunt Zajac. 2017. Goodbooks-10k: a new dataset for book recommendations. http://fastml.com/goodbooks-10k. *FastML* (2017).

[38] Sheng Zhang, Weihong Wang, James Ford, and Fillia Makedon. 2006. Learning from Incomplete Ratings Using Non-negative Matrix Factorization. In *Proceedings of the 2006 SIAM International Conference on Data Mining*. Society for Industrial and Applied Mathematics, 549–553. https://doi.org/10.1137/1.9781611972764.58