Matrix Completion in the Unit Hypercube via Structured Matrix Factorization

Emanuele Bugliarello[1*] , **Swayambhoo Jain**[2] and **Vineeth Rakesh**[2]

[1]Tokyo Institute of Technology
[2]Technicolor AI Lab

emanuele.bugliarello@nlp.c.titech.ac.jp, {Swayambhoo.Jain, Vineeth.Mohan}@technicolor.com

Abstract

Several complex tasks that arise in organizations can be simplified by mapping them into a matrix completion problem. In this paper, we address a key challenge faced by our company: predicting the efficiency of artists in rendering visual effects (VFX) in film shots. We tackle this challenge by using a two-fold approach: first, we transform this task into a constrained matrix completion problem with *entries bounded in the unit interval* $[0, 1]$; second, we propose two novel matrix factorization models that leverage our knowledge of the VFX environment. Our first approach, *expertise matrix factorization* (EMF), is an interpretable method that structures the latent factors as weighted user-item interplay. The second one, *survival matrix factorization* (SMF), is instead a probabilistic model for the underlying process defining employees' efficiencies. We show the effectiveness of our proposed models by extensive numerical tests on our VFX dataset and two additional datasets with values that are also bounded in the $[0, 1]$ interval.

1 Introduction

A variety of complex applications that emerge in several organizations can be translated to matrix completion problems. In this study, we address a key challenge faced by our company: predicting the efficiency of artists in rendering visual effects (VFX) in film shots. Effectively solving this problem is of crucial importance as project managers often rely on competency metrics, such as employee's efficiency, to best control and adjust the available assets. Figure 1 illustrates the hierarchical relationship between artists (i.e., employees) and job assignments in a VFX production framework. Here, a *job* for the organization just consists of a series of contributions from different *departments* and possibly involves numerous *employees*. Each department is led by a *manager* who divides the work in her department into multiple *tasks*, and each task is assigned to a single employee. Employees work on these tasks and submit partial results called *claims*. After assessing the quality of a claim, the manager decides whether to
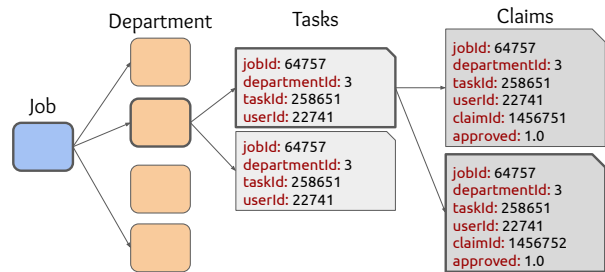

Figure 1: Hierarchical task allocation.

approve it or not. Hence, our goal is to exploit the latent attributes of each employee that explain their efficiency across various tasks to ensure products can be delivered on time.

The key contributions of our work are as follows: (i) transforming industrial setups into a sparse real-valued matrix with entries lying in the $[0, 1]$ interval (e.g., efficiencies of employees in departments), and (ii) proposing two novel structured matrix factorization (MF) models to effectively solve the resulting matrix completion problem. While our methods are inspired by the hierarchical task allocation shown in Figure 1, their formulation is very general and can easily be applied to a variety of scenarios. In this paper, in particular, we consider two additional applications: (a) recommendation of apps to the users of over-the-top (OTT) streaming devices, and (b) recommendation in online advertisement placement to maximize click-through rate (CTR). The details of these applications will be explained in later sections.

Although conventional matrix factorization algorithms have been proved effective to solve the matrix completion problem, recent studies show that constrained MF techniques outperform them on a variety of datasets, including MovieLens, Jester and BookCrossing [Fang *et al.*, 2017; Jawanpuria and Mishra, 2018]. Moreover, MF approaches tend to produce unstable and out-of-range predictions as the sparsity of the matrix increases [Jiang *et al.*, 2018]. While most recommender systems have entries bounded within a range of possible rating values (e.g., one to five in a five-star rating system), our data lies on the unit interval $[0, 1]$. While we could convert our data to an ordinal space, such as ratings between one and five, as shown in Figure 2, any monotonic transformation mapping the unit interval to a larger one does not simplify the prediction problem. Therefore, we propose two novel approaches for this scenario: *expertise matrix*
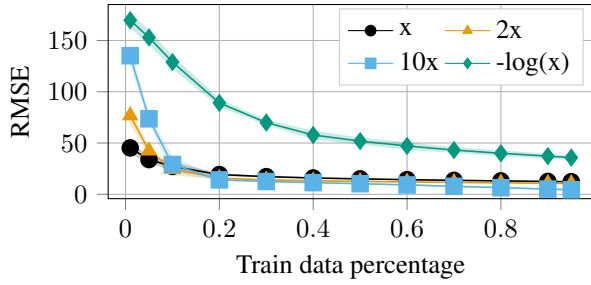
Figure 2: Prediction errors (mean $\pm$ standard error) on five 40×30 random matrices with entries in $[0, 1]$ when applying monotonic mappings at various training data percentages. These transformations do not ease the matrix factorization problem.

factorization (EMF), in which entries are approximated by a weighted correlation between user and item latent factors (i.e., the low-rank features), and *survival matrix factorization* (SMF), which uses a probabilistic model to capture the latent factors by modeling the procedure resulting in the average efficiency of an employee in a department. Both our approaches are structured matrix factorization methods with constrains derived by modeling the presented setup.

We validate the effectiveness of the proposed models using three datasets: (a) a proprietary dataset of VFX efficiencies, (b) a proprietary dataset of app usage in OTT device, and (c) a public dataset of CTR released by Outbrain. Using a rigorous series of numerical experiments, we show that the proposed models outperform popular MF techniques on scores such as RMSE, MAE, Precision@N and Recall@N.

2 Related Work

The allocation of resources is a relevant problem in Business Process Management to improve the performance of an organization [Dumas *et al.*, 2013]. Recently, different approaches in Process Mining [Van Der Aalst, 2011] have been proposed to extract useful knowledge from historical data [Arias *et al.*, 2018]. However, previous studies focused on particular process cases [Huang *et al.*, 2012; Conforti *et al.*, 2015]. In response, we investigate a generic framework that can be mapped to a variety of organizations. Our proposed approaches are inspired by a series of work on low-rank matrix completion, and their applications to recommendation systems [Hu *et al.*, 2008; Candès and Recht, 2009; Candès and Plan, 2010; Koren *et al.*, 2009; Funk, 2011; Jain *et al.*, 2013]. A class of approaches suitable for our task consists of bounded matrix factorization methods. Non-negative matrix factorization (NMF) is the most popular method in this class, and it only provides a lower bound of 0 for the predicted values. A more general approach is given by bounded matrix factorization (BMF) [Kannan *et al.*, 2014]: a low-rank approximation that leverages the fact that all ratings in a recommender system are bounded within a range $[x_{min}, x_{max}]$ Our first approach, expertise matrix factorization, is closely related to these approaches and to matrix completion models with additional structure on the latent factors [Soni *et al.*, 2016; Hoyer, 2004; Aharon *et al.*, 2006; Kannan *et al.*, 2014]. However, our approach departs from them in its unique structure given

by the unit interval constraint in our setup. To the best of our knowledge, this structure has not been specifically tackled in the existing literature of MF with explicit data. MF methods for implicit feedback data [Hu *et al.*, 2008; Johnson, 2014] work on the same interval, but with the major difference of relying on just matrix entries that are binary values. Our second approach, survival matrix factorization, is instead related to probabilistic matrix factorization methods [Mnih and Salakhutdinov, 2008; Salakhutdinov and Mnih, 2008]. While they also interpret matrix entries as probabilities, our method specifically models the end-to-end approval process in its formulation.

Notations. In this paper, all vectors are represented by bold lowercase letters and are column vectors (e.g., $\mathbf{p}$, $\mathbf{q}$). All matrices are represented by bold uppercase letters (e.g., $\mathbf{P}$, $\mathbf{Q}$). For a given matrix $\mathbf{A}$, $\mathbf{a}_i$ denotes its i-th row as a column vector and a_{ij} denotes its entry in row i and column j.

3 Problem Formulation

As described in Section 1, our proposed models are inspired by the framework depicted in Figure 1. This pipeline is motivated by the concrete application scenario of work allocation in movie production at Technicolor, where artists (employees) render visual effects in film shots across various disciplines (departments). Nonetheless, this framework is very general and other production pipelines can easily be mapped into this setting. For instance, if an organization logs employees' performance on pieces of work in a 100-point scale, then each score can be mapped to an approved or rejected claim by defining a threshold value of work quality (e.g., $90/100$).

Considering the work allocation scenario, a natural competency metric for employees' efficiencies is given by the ratio of number of claims accepted to the total number of claims submitted by the employee in the department. An accepted claim counts towards the employee's overall performance. In case of rejection, the employee submits a new claim when they believe it will meet the quality required. The latter scenario clearly results in a loss in performance for the entire organization as the employee might need to start over or the manager may decide to designate another employee. Let N_{dn} be the total number of claims by employee d in department n, and $a_{dn}^{(i)}$ denote whether the i-th claim by employee d in department n was accepted (1) or rejected (0). We then define the efficiency of employee d in department n, x_{dn}, as follows:

$$x_{dn} = \frac{\sum_{i=1}^{N_{dn}} a_{dn}^{(i)}}{N_{dn}}. \qquad (1)$$

It is straightforward to see that $x_{dn} \in [0, 1]$.

The goal of our study is to predict the efficiencies of employees in each department. We address this challenge by building an efficiency matrix $\mathbf{X} \in \mathbb{R}^{D \times N}$ whose entry $x_{dn} \in [0, 1]$ denotes the efficiency of employee d in department n. Only a few entries of this matrix, corresponding to the observation index set $\Omega \subset [D] \times [N]$, are known since most employees only work in a few departments throughout their careers. Hence, our goal of predicting employees' efficiencies now reduces to predicting the missing entries of $\mathbf{X}$.

4 Proposed Models

Our first proposed model, *expertise matrix factorization*, builds on low-rank models and introduces additional structure on the factors $\mathbf{W}$ and $\mathbf{Z}$ so that the entries of the resulting matrix $\mathbf{W} \cdot \mathbf{Z}^T$ lie in unit interval. Our second model, *survival matrix factorization*, instead follows a probabilistic modeling of the claim approval process defining the efficiency matrix.

4.1 Expertise Matrix Factorization

In this first approach, we model the efficiency matrix as a low-rank matrix $\mathbf{X} \approx \mathbf{W} \cdot \mathbf{Z}^T$. In our setup, we can think of the latent factors as the set of *skills* or *expertise* required to work in a given department. In particular, we assume each employee's latent factors (skills) to range from 0 to 1, while each department's latent factors to be non-negative and to sum to 1. On the one hand, employees have a certain level in each skill, where 0 indicates no ability and 1 proficiency. On the other hand, each department is assumed to have a given proportion of skills required to complete tasks in it.

We assume a low-rank model based on the intuitive reasoning that a small number of skills may be required to complete tasks across different departments. With this model, the efficiency of employee d in department n is approximated by a weighted sum of the employee's skills, where the weights are given by the importance of each skill in the department. The structure in the latent vectors ensures that efficiency values under this model lie in $[0, 1]$. We further extend this model to accommodate user biases, representing the minimum proficiency of each employee in every skill. Note that it would be inconsistent to add a bias term for departments as not every skill is supposed to be useful in each discipline.

The resulting optimization problem is as follows:

$$\min_{\mathbf{W} \geq 0, \mathbf{Z} \geq 0, \boldsymbol{\beta} \geq 0} \quad \frac{1}{2|\Omega|} \sum_{(d,n) \in \Omega} \left[x_{dn} - \left(\beta_d + \mathbf{w}_d^T \cdot \mathbf{z}_n \right) \right]^2$$

$$+ \frac{\lambda_u}{2} \left(\|\mathbf{W}\|_F^2 + \|\boldsymbol{\beta}\|_2^2 \right) + \frac{\lambda_i}{2} \|\mathbf{Z}\|_F^2 \quad (2)$$

$$\text{subject to} \quad \beta_d + w_{dk} \leq 1, \quad \text{for } (d, k) \in [D] \times [K]$$

$$\sum_{k=1}^{K} z_{nk} = 1, \quad \text{for } n \in [N]$$

where $\boldsymbol{\beta} \in \mathbb{R}^D$ is the vector of employees' biases, Ω is the set of observed entries and $|\Omega|$ is its size, λ_u and λ_i control the extent of regularization in the users and items parameters, and $|| \cdot ||_F$ denotes the Frobenius norm. The first constraint $\beta_d + w_{dk} \leq 1$ enforces predictions in $[0, 1]$. In fact, $\hat{x}_{dn} = \beta_d + \mathbf{w}_d^T \cdot \mathbf{z}_n = (\beta_d \mathbf{1} + \mathbf{w}_d)^T \cdot \mathbf{z}_n \in [0, 1]$ if $\beta_d + w_{dk} \leq 1$.

The structure on the factor $\mathbf{Z}$ has also been considered in blind de-mixing applications [Lin *et al.*, 2015; Fu *et al.*, 2018], which, however, do not deal with missing entries and do not impose any structure on the factor $\mathbf{W}$. The optimization problem in Equation (2) is jointly non-convex in $\mathbf{W}, \boldsymbol{\beta}$ and $\mathbf{Z}$. However, for fixed values of $\mathbf{W}$ and $\boldsymbol{\beta}$, the problem is a convex quadratic program in $\mathbf{Z}$, and for fixed value $\mathbf{Z}$ the problem is a convex quadratic program in $\mathbf{W}$ and $\boldsymbol{\beta}$ [Boyd and Vandenberghe, 2004]. Hence, we rely on an algorithm based on alternating minimization to solve this problem, first

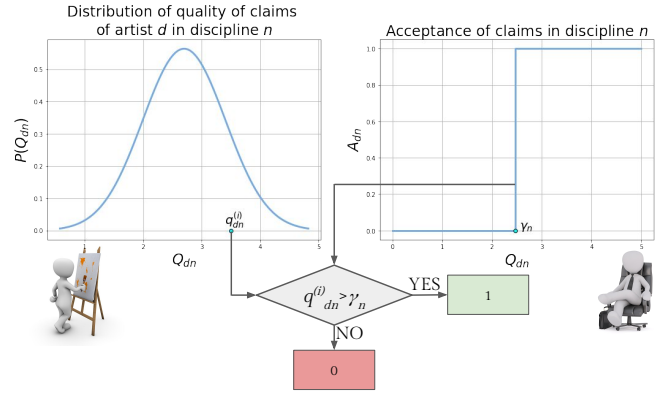Figure 3: Claims acceptance survival model.

solving for $\mathbf{W}$ and $\boldsymbol{\beta}$, and then for $\mathbf{Z}$. In particular, the former sub-problem can be solved by any quadratic program solver, while the latter by a projected gradient algorithm that maps each row of $\mathbf{Z}$ onto the probability simplex.

4.2 Survival Matrix Factorization

In this approach, we aim to derive a probabilistic model for the underlying process resulting in efficiency values. Recall from Equation (1) that each entry x_{dn} in the efficiency matrix $\mathbf{X}$ is defined as the empirical average efficiency of employee d in department n. Each claim submitted by an employee can be viewed as an i.i.d. sample from a Bernoulli random variable A_{dn}, and, by the Weak Law of Large Numbers, the matrix entry x_{dn} tends almost surely to the mean value of A_{dn}, which is equal to $\mathbb{P}(A_{dn} = 1)$.

Having introduced this probability framework, in order to define $\mathbb{P}(A_{dn} = 1)$, we model the process of claims acceptance. Claims represent pieces of work that employees deliver to managers, who assess the quality of each claim and accept them only if they are good enough. First, we assume employee d to submit claims to department n with a certain quality distribution. That is, each claim i has quality Q_{dn}, a random variable sampled from an underlying probability distribution of quality of work produced by employee d in department n. Second, we model the manager of department n by a single parameter, γ_n, representing their quality threshold: a claim in department n is only accepted if its quality is higher than the manager's threshold (as depicted in Figure 3).

Hence, each entry x_{dn} is approximated by:

$$x_{dn} \approx \mathbb{P}(A_{dn} = 1) = \mathbb{P}(Q_{dn} > \gamma_n)$$

$$= S_{Q_{dn}}(\gamma_n) = \int_{\gamma_n}^{+\infty} f_{Q_{dn}}(u) \, du, \quad (3)$$

where $S_{Q_{dn}}(\gamma_n)$ is the survival function, also known as complementary cumulative distribution function, of Q_{dn} at γ_n, and $f_{Q_{dn}}$ is its probability distribution. Note also that each manager is modeled by a single variable and they are then assumed to adhere to equal opportunity laws, accepting claims with no discrimination of race, color, sex, sexual orientation or any other status of the artists.

In this paper, we assume the employees' quality probability distributions to be normal and that the variance of the quality

of the claims is the same over all employees and departments:

$$f_{Q_{dn}}\left(x\big|\mu_{dn},\sigma^2\right) = \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{(x-\mu_{dn})^2}{2\sigma^2}}, \qquad (4)$$

where μ_{dn} is the mean of the distribution of quality of claims of employee d in department n and σ^2 is its variance.

Furthermore, we try to explain the means μ_{dn} by characterizing both artists and disciplines with vectors of factors inferred from patterns in the data such that employee-department interactions are modeled as inner products in that space; i.e., $\mu_{dn} \approx \mathbf{w}_d^T \cdot \mathbf{z}_n$. That is:

$$\hat{x}_{dn} = \int_{\gamma_n}^{+\infty} \frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{\left(u-\mathbf{w}_d^T\cdot\mathbf{z}_n\right)^2}{2\sigma^2}}\,du. \qquad (5)$$

The optimization problem we aim to solve is then given by:

$$\min_{\mathbf{W},\mathbf{Z},\gamma,\sigma} \frac{1}{2|\Omega|} \sum_{(d,n)\in\Omega} [x_{dn}-\hat{x}_{dn}]^2 + \frac{\lambda_u}{2}\|\mathbf{W}\|_F^2 + \frac{\lambda_i}{2}\|\mathbf{Z}\|_F^2. \qquad (6)$$

The objective function in the above problem is smooth in $\mathbf{W},\mathbf{Z},\gamma$ and σ, and stochastic gradient descent can be used to solve it. We use Leibniz's rule for differentiation under the integral sign to obtain the closed form expressions of the gradients with respect to $\mathbf{W},\mathbf{Z}$ and σ, and the fundamental theorem of calculus for the gradient with respect to γ.

5 Experimental Evaluations

In this section, we evaluate the empirical performance of our models on (a) a dataset of VFX rendering, (b) a dataset of OTT streams from Technicolor, and (c) a public dataset of click-through rates (CTR) in online advertising by Outbrain. Experimental results on public data are available on GitHub [1].

5.1 Methodology

Evaluation Metrics
The quality of a recommendation algorithm can be evaluated using different types of metrics. We use RMSE and MAE as statistical accuracy metrics, while Precision@N and Recall@N as decision support metrics, for $N \in \{2,3,5,10\}$. In the context of recommender systems, we are usually interested in recommending top-N items to the user. This is clearly the case for our framework too, items being either employees, apps or website categories. As usual, we consider as relevant items those that are already known in the datasets.

Cross-Validation
We use 3 rounds of *Monte Carlo cross-validation* on the movie production data (due to few non-missing entries) and *3-fold cross-validation* on the OTT and CTR data. We use the RMSE scores to find the best model via cross-validation.

Details of Training
We use a maximum number of 100 epochs and a tolerance defined by $\frac{\mathcal{L}^{(t)}-\mathcal{L}^{(t-1)}}{\mathcal{L}^{(t)}} < 10^{-6}$ as stopping criteria for training, where $\mathcal{L}^{(t)}$ is the objective cost at epoch t. In all SGD-based algorithms, we use batches of 8 entries on the VFX

---

[1]URL: https://github.com/e-bug/unit-mf.

---

data, and of 128 entries on the OTT and CTR data. The best number of latent factors K is searched over all possible values in the VFX data, while we use the common values of $K \in \{10,15,20\}$ in the larger OTT and CTR data. Each matrix factor is initialized with uniformly random numbers in $(0,1)$, and biases are initialized as zero vectors.

Learning Algorithms
The following algorithms are evaluated:

- **MF**. Stochastic gradient descent for MF with biases.

- **NMF**. Alternating non-negative least squares for NMF. Each block is updated by projected gradient descent.

- **BMF**. Bounded matrix factorization. To overcome the limitation requiring $x_{max} > 1$ in [Kannan *et al.*, 2014], we scale the randomly initialized factors $\mathbf{W}$ and $\mathbf{Z}$ by $\sqrt{\frac{1}{\max(\mathbf{W}\cdot\mathbf{Z}^T)}}$, ensuring feasible bounds at the first step.

- **PMF**. Probabilistic matrix factorization.

- **LMF**. Logistic matrix factorization. Differently from [Johnson, 2014], we use explicit data and model each entry in the data matrix to match our framework:

$$\begin{aligned}\hat{x}_{dn} &= \mathbb{P}\left(\text{claims by artist } d \text{ in discipline } n \text{ are accepted}\right)\\ &= \mathbb{P}\left(\text{manager of discipline } n \text{ accepts a claim}\right)\\ &\quad \mathbb{P}\left(\text{artist } d \text{ submits good claims in discipline } n\right)\\ &= \frac{1}{1+e^{-\gamma_n}}\frac{1}{1+e^{-\beta_d-\mathbf{w}_d^T\cdot\mathbf{z}_n}}.\end{aligned} \qquad (7)$$

- **EMF**. Expertise matrix factorization. Projections of each row in $\mathbf{Z}$ onto the probability simplex are obtained using the $\mathcal{O}\left(K\log K\right)$ efficient algorithm proposed in [Wang and Carreira-Perpiñán, 2013].

- **SMF**. Survival matrix factorization. σ is initialized to 1.

5.2 Movie Production Data
Our analysis is driven by data collected at Technicolor, where different disciplines (departments) are responsible for generating VFX in movies (jobs) and employees are referred as artists. All data was collected in accordance with appropriate end user agreements and privacy policies.

Our movie production dataset consists of claim records, each with fields: *jobId* (int), *disciplineId* (int), *taskId* (int), *userId* (int), *claimId* (int) and *approved* (bool). To ensure that each average efficiency is sufficiently representative of the true efficiency of an artist in a discipline, we remove all the entries that result from averaging less than 10 claims. This also ensures that, according to Hoeffding's theorem [Hoeffding, 1963], the approximation introduced in Equation (3) holds true with high probability. Moreover, to alleviate the cold-start problem, we remove disciplines that have claims from less than 10 artists and also drop artists with less than 3 non-missing entries. At the end of these preprocessing steps, we are left with a 312×25 matrix and $1,026$ non-missing entries. Not only is this matrix very sparse (86.85%) but there are also few ratings per user, with most of the artist having worked on three disciplines only. The distribution of efficiencies increases exponentially (Figure 4).
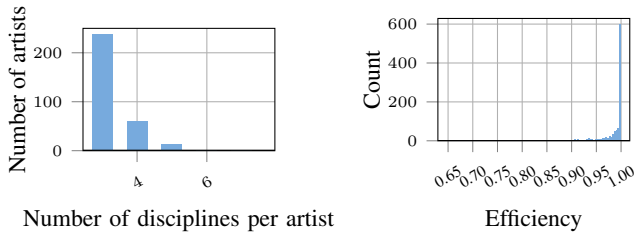
Figure 4: Statistics about movie production data.

Figure 5: Statistics about over-the-top data.

Method	K	RMSE $[\times 10^{-2}]$ Mean (standard error)		MAE $[\times 10^{-2}]$ Mean (standard error)	
		Training	Test	Training	Test
MF	6	2.50 (0.045)	2.66 (0.019)	1.42 (0.021)	1.79 (0.078)
NMF	1	2.05 (0.036)	2.73 (0.016)	1.19 (0.016)	1.93 (0.020)
BMF	1	2.17 (0.069)	2.64 (0.010)	1.19 (0.027)	1.78 (0.008)
PMF	24	3.53 (0.018)	3.18 (0.017)	1.94 (0.010)	1.99 (0.001)
LMF	21	5.08 (0.015)	5.07 (0.043)	4.42 (0.014)	4.39 (0.016)
EMF	20	**0.46 (0.004)**	2.62 (0.007)	**0.36 (0.002)**	1.60 (0.011)
SMF	14	3.53 (0.079)	**1.54 (0.017)**	3.15 (0.026)	**1.50 (0.059)**

Table 1: Prediction errors on the movie production dataset.

Table 1 reports prediction errors of each algorithm on this dataset. Precision and recall values are listed in Table 2.

5.3 Over-The-Top Data

Our second application consists of apps usage on over-the-top devices where the goal is to recommend apps to customers. Our OTT dataset consists of number of views of a given app by any user. This data is mapped to users' watching rates $\in [0, 1]$ by dividing by the user's maximum number of views. Similarly to what we did for the VFX data, we remove apps that have less than 15 viewers and users who watched less than 10 apps. After preprocessing, we are left with 934 users and 140 apps and a matrix that is extremely sparse (99.91%). The distribution of watching rates is shown in Figure 5.

Tables 3 and 4 show precision and recall values for K equal to 15 and 20, respectively. Due to space constraints, results for $K = 10$ are omitted but followed similar patterns.

5.4 Click-Through Rate Data

In our third application, we consider the task of placing online advertisements in categories of websites (e.g., Entertainment, Finance, etc.) to maximize their click-through rate.

To simulate this scenario, we use publicly available data from a Kaggle competition [2] run by the advertisement company Outbrain. This dataset contains users' webpage views and clicks on multiple publisher sites in the United States in a two-week period in June 2016. By using additional semantic attributes of the webpages, we can formulate the problem of maximizing the CTR of an advertisement by finding the best categories to display it. After removing advertisements displayed in less than 5 categories and categories with less than 10 advertisements, we are left with $15,647$ advertisements and 85 webpage categories, resulting in a 79.99%-sparse matrix and a distribution of CTR skewed to the right (Figure 6).

Tables 5, 6 and 7 show precision and recall values for number of latent factors K equal to 10, 15 and 20, respectively.
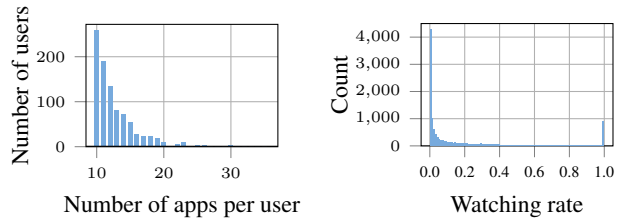
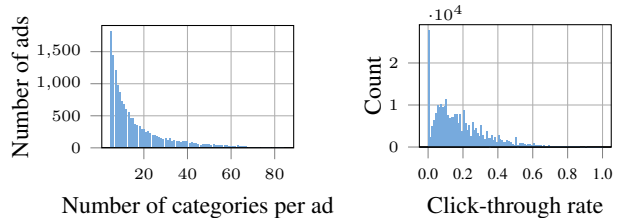[2]URL: https://www.kaggle.com/c/outbrain-click-prediction.

Figure 6: Statistics about click-through rate data.

5.5 Discussion

In our VFX data, SMF achieves the lowest prediction errors and EMF outperforms every other method in precision and recall with values three times higher than the second best model. Here, the hypotheses of EMF closely match artists in our VFX framework, while they do not entirely hold for ads in CTR and users in OTT. One of the strongest points of PMF is its ability to generalize considerably well for users with very few ratings. However, here – where the maximum number of entries per user is just 7 – we see that EMF outperforms PMF in both prediction accuracy and quality of recommendations. In Table 1, we also notice an interesting empirical finding: constrained models are capable of better exploiting the underlying structure relying on a higher latent dimensionality.

Results in OTT and CTR datasets instead show how SMF is a very general model and outperforms every other algorithm in terms of precision and recall, regardless of the chosen number of latent factors K. In OTT, the gap is wider for smaller values of K; in CTR, EMF consistently achieves the second highest precision. We believe that SMF underperforms in the VFX data due to the very few available entries rather than its sparsity (OTT is much sparser). Here, SMF's performance is comparable to other bounded models such as BMF and LMF.

6 Conclusion

In this paper, we introduced *EMF* and *SMF*: two novel structured MF methods to predict entries in the unit interval. We showed that these models provide better recommendations than popular MF techniques on three real-world datasets. Hence, it is beneficial to explicitly model entries in the unit interval, motivating further research in this domain.

In addition, SMF is more than just one method: it is the simplest in a class of probabilistic models. In this paper, we assumed users' quality of work to be normally distributed and managers to be modeled by a scalar threshold variable. In future work, we plan to extend this model to include more complex assumptions depending on the application domains.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.67 (0.544)	3.11 (0.363)	4.80 (0.377)	5.47 (0.393)	0.19 (0.065)	0.37 (0.037)	1.01 (0.042)	2.27 (0.268)
NMF	1.33 (0.544)	2.22 (0.363)	3.47 (0.435)	4.67 (0.435)	0.10 (0.042)	0.57 (0.190)	1.42 (0.265)	2.67 (0.243)
BMF	0.00 (0.000)	0.89 (0.726)	2.13 (0.576)	2.67 (0.109)	0.00 (0.000)	0.05 (0.037)	0.53 (0.038)	1.40 (0.010)
PMF	1.33 (0.544)	3.11 (0.363)	3.47 (0.218)	3.20 (0.189)	0.13 (0.059)	0.37 (0.026)	0.76 (0.116)	1.52 (0.021)
LMF	1.33 (0.544)	0.89 (0.363)	2.67 (0.576)	4.67 (0.288)	0.24 (0.099)	0.24 (0.099)	0.88 (0.044)	2.95 (0.226)
EMF	**4.67 (0.544)**	**6.22 (0.363)**	**10.40 (0.998)**	**19.07 (1.039)**	**0.52 (0.125)**	**1.14 (0.091)**	**2.34 (0.132)**	**8.39 (0.636)**
SMF	4.00 (0.000)	2.67 (0.000)	2.13 (0.218)	2.93 (0.288)	0.29 (0.009)	0.29 (0.009)	0.53 (0.093)	2.82 (0.204)

Table 2: Precision and Recall (mean and standard error) on the movie production dataset.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.02 (0.194)	2.22 (0.130)	2.48 (0.216)	3.17 (0.103)	0.20 (0.049)	0.29 (0.044)	0.46 (0.055)	1.17 (0.118)
NMF	2.74 (0.257)	2.78 (0.467)	3.71 (0.294)	3.21 (0.202)	0.24 (0.024)	0.35 (0.055)	0.70 (0.034)	1.17 (0.074)
BMF	0.95 (0.097)	1.75 (0.130)	2.67 (0.318)	3.24 (0.395)	0.03 (0.010)	0.11 (0.011)	0.27 (0.010)	0.69 (0.067)
PMF	0.24 (0.097)	0.24 (0.112)	0.52 (0.104)	1.48 (0.103)	0.03 (0.021)	0.03 (0.019)	0.09 (0.023)	0.47 (0.016)
LMF	0.24 (0.097)	0.48 (0.194)	0.52 (0.039)	0.90 (0.136)	0.00 (0.003)	0.04 (0.017)	0.08 (0.013)	0.35 (0.073)
EMF	0.71 (0.000)	1.03 (0.171)	2.00 (0.178)	2.81 (0.248)	0.08 (0.021)	0.12 (0.009)	0.26 (0.054)	0.64 (0.064)
SMF	**32.62 (0.350)**	**30.71 (0.194)**	**25.48 (0.433)**	**18.60 (0.402)**	**2.69 (0.063)**	**3.40 (0.087)**	**4.18 (0.109)**	**5.70 (0.130)**

Table 3: Precision and Recall (mean and standard error) on the OTT dataset for $K = 15$.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.02 (0.424)	2.06 (0.282)	3.00 (0.282)	3.43 (0.221)	0.15 (0.035)	0.22 (0.033)	0.59 (0.067)	1.34 (0.165)
NMF	1.07 (0.292)	1.51 (0.259)	1.43 (0.178)	1.45 (0.237)	0.11 (0.034)	0.22 (0.038)	0.35 (0.047)	0.67 (0.121)
BMF	1.90 (0.097)	2.70 (0.282)	4.14 (0.486)	4.26 (0.051)	0.09 (0.029)	0.21 (0.028)	0.47 (0.035)	1.08 (0.049)
PMF	0.12 (0.097)	0.56 (0.065)	0.90 (0.039)	1.36 (0.067)	0.02 (0.015)	0.04 (0.023)	0.17 (0.034)	0.47 (0.059)
LMF	0.36 (0.168)	0.56 (0.282)	0.95 (0.237)	1.29 (0.154)	0.02 (0.011)	0.10 (0.044)	0.17 (0.062)	0.50 (0.058)
EMF	1.43 (0.292)	1.51 (0.343)	2.62 (0.383)	2.79 (0.243)	0.14 (0.032)	0.16 (0.036)	0.34 (0.067)	0.69 (0.080)
SMF	**4.52 (1.029)**	**5.56 (0.394)**	**5.67 (0.433)**	**5.74 (0.159)**	**0.29 (0.064)**	**0.49 (0.038)**	**0.74 (0.062)**	**1.51 (0.049)**

Table 4: Precision and Recall (mean and standard error) on the OTT dataset for $K = 20$.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.94 (0.000)	4.31 (0.640)	5.18 (0.293)	6.31 (0.115)	0.01 (0.001)	0.01 (0.002)	0.02 (0.002)	0.05 (0.002)
NMF	0.00 (0.000)	0.00 (0.000)	0.08 (0.064)	0.20 (0.064)	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)
BMF	0.78 (0.424)	0.52 (0.282)	0.55 (0.169)	1.06 (0.192)	0.00 (0.001)	0.00 (0.001)	0.00 (0.001)	0.01 (0.003)
PMF	1.57 (0.577)	1.96 (0.489)	2.98 (0.420)	4.31 (0.401)	0.02 (0.006)	0.02 (0.008)	0.04 (0.013)	0.09 (0.015)
LMF	3.14 (0.160)	3.01 (0.107)	4.08 (0.231)	6.59 (0.508)	0.01 (0.003)	0.03 (0.002)	0.06 (0.013)	0.16 (0.032)
EMF	3.14 (0.424)	5.10 (0.320)	5.88 (0.111)	6.63 (0.115)	0.01 (0.001)	0.01 (0.001)	0.02 (0.002)	0.05 (0.001)
SMF	**7.06 (0.277)**	**7.58 (0.427)**	**8.78 (0.279)**	**9.37 (0.210)**	**0.04 (0.009)**	**0.05 (0.013)**	**0.10 (0.008)**	**0.16 (0.009)**

Table 5: Precision and Recall (mean and standard error) on the CTR dataset for $K = 10$.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.94 (0.277)	3.79 (0.107)	5.02 (0.231)	6.35 (0.096)	0.00 (0.001)	0.01 (0.001)	0.02 (0.001)	0.05 (0.002)
NMF	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)	0.24 (0.055)	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)	0.00 (0.001)
BMF	1.18 (0.277)	1.96 (0.370)	2.12 (0.222)	2.47 (0.242)	0.00 (0.001)	0.00 (0.001)	0.01 (0.001)	0.02 (0.002)
PMF	2.94 (0.555)	3.53 (0.185)	3.45 (0.128)	4.78 (0.547)	0.01 (0.001)	0.01 (0.001)	0.03 (0.007)	0.10 (0.025)
LMF	3.53 (0.000)	3.66 (0.213)	4.55 (0.256)	7.18 (0.640)	0.02 (0.007)	0.04 (0.006)	0.06 (0.013)	0.15 (0.033)
EMF	3.14 (0.424)	5.10 (0.320)	5.80 (0.064)	6.75 (0.032)	0.01 (0.001)	0.01 (0.001)	0.02 (0.001)	0.05 (0.003)
SMF	**8.24 (0.000)**	**9.15 (0.282)**	**9.96 (0.279)**	**11.10 (0.378)**	**0.04 (0.003)**	**0.05 (0.007)**	**0.09 (0.011)**	**0.17 (0.024)**

Table 6: Precision and Recall (mean and standard error) on the CTR dataset for $K = 15$.

Method	Precision				Recall			
	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$	@2 $[\times 10^{-2}]$	@3 $[\times 10^{-2}]$	@5 $[\times 10^{-2}]$	@10 $[\times 10^{-2}]$
MF	2.75 (0.160)	3.40 (0.213)	5.65 (0.192)	6.59 (0.055)	0.00 (0.001)	0.01 (0.001)	0.02 (0.002)	0.05 (0.003)
NMF	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)	0.20 (0.000)	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)	0.00 (0.000)
BMF	1.96 (0.320)	1.83 (0.213)	2.35 (0.293)	3.25 (0.390)	0.00 (0.001)	0.00 (0.001)	0.00 (0.001)	0.02 (0.001)
PMF	2.35 (0.000)	3.40 (0.700)	4.31 (0.128)	4.71 (0.364)	0.02 (0.007)	0.03 (0.008)	0.04 (0.007)	0.07 (0.012)
LMF	2.94 (0.000)	3.66 (0.213)	5.25 (0.279)	7.88 (0.454)	0.01 (0.006)	0.04 (0.006)	0.06 (0.012)	0.17 (0.032)
EMF	3.14 (0.424)	5.10 (0.320)	5.80 (0.064)	6.63 (0.115)	0.01 (0.001)	0.01 (0.001)	0.02 (0.001)	0.05 (0.001)
SMF	**9.41 (0.734)**	**9.67 (0.565)**	**10.20 (0.256)**	**10.16 (0.250)**	**0.06 (0.007)**	**0.07 (0.006)**	**0.12 (0.013)**	**0.19 (0.016)**

Table 7: Precision and Recall (mean and standard error) on the CTR dataset for $K = 20$.

References

[Aharon *et al.*, 2006] Michal Aharon, Michael Elad, Alfred Bruckstein, et al. K-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on signal processing*, 54(11):4311, 2006.

[Arias *et al.*, 2018] Michael Arias, Rodrigo Saavedra, Maira R Marques, Jorge Munoz-Gama, and Marcos Sepúlveda. Human resource allocation in business process management and process mining: A systematic mapping study. *Management Decision*, 56(2):376–405, 2018.

[Boyd and Vandenberghe, 2004] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge university press, 2004.

[Candès and Plan, 2010] Emmanuel J Candès and Yaniv Plan. Matrix completion with noise. *Proceedings of the IEEE*, 98(6):925–936, 2010.

[Candès and Recht, 2009] Emmanuel J Candès and Benjamin Recht. Exact matrix completion via convex optimization. *Foundations of Computational mathematics*, 9(6):717, 2009.

[Conforti *et al.*, 2015] Raffaele Conforti, Massimiliano de Leoni, Marcello La Rosa, Wil MP van der Aalst, and Arthur HM ter Hofstede. A recommendation system for predicting risks across multiple business process instances. *Decision Support Systems*, 69:1–19, 2015.

[Dumas *et al.*, 2013] Marlon Dumas, Marcello La Rosa, Jan Mendling, Hajo A Reijers, et al. *Fundamentals of business process management*, volume 1. Springer, 2013.

[Fang *et al.*, 2017] Huang Fang, Zhen Zhang, Yiqun Shao, and Cho-Jui Hsieh. Improved bounded matrix completion for large-scale recommender systems. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1654–1660. AAAI Press, 2017.

[Fu *et al.*, 2018] Xiao Fu, Kejun Huang, and Nicholas D Sidiropoulos. On identifiability of nonnegative matrix factorization. *IEEE Signal Processing Letters*, 25(3):328–332, 2018.

[Funk, 2011] Simon Funk. Netflix update: Try this at home, 2006, 2011.

[Hoeffding, 1963] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American statistical association*, 58(301):13–30, 1963.

[Hoyer, 2004] Patrik O Hoyer. Non-negative matrix factorization with sparseness constraints. *Journal of machine learning research*, 5(Nov):1457–1469, 2004.

[Hu *et al.*, 2008] Yifan Hu, Yehuda Koren, and Chris Volinsky. Collaborative filtering for implicit feedback datasets. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 263–272. Ieee, 2008.

[Huang *et al.*, 2012] Zhengxing Huang, Xudong Lu, and Huilong Duan. A task operation model for resource allocation optimization in business process management. *IEEE Transactions on Systems, man, and cybernetics-part a: systems and humans*, 42(5):1256–1270, 2012.

[Jain *et al.*, 2013] Prateek Jain, Praneeth Netrapalli, and Sujay Sanghavi. Low-rank matrix completion using alternating minimization. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 665–674. ACM, 2013.

[Jawanpuria and Mishra, 2018] Pratik Jawanpuria and Bamdev Mishra. A unified framework for structured low-rank matrix learning. In *International Conference on Machine Learning*, pages 2259–2268, 2018.

[Jiang *et al.*, 2018] Shuai Jiang, Kan Li, Da Xu, and Richard Yi. Magnitude bounded matrix factorisation for recommender systems. *arXiv preprint arXiv:1807.05515*, 2018.

[Johnson, 2014] Christopher C Johnson. Logistic matrix factorization for implicit feedback data. 2014.

[Kannan *et al.*, 2014] Ramakrishnan Kannan, Mariya Ishteva, and Haesun Park. Bounded matrix factorization for recommender system. *Knowledge and information systems*, 39(3):491–511, 2014.

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

[Lin *et al.*, 2015] Chia-Hsiang Lin, Wing-Kin Ma, Wei-Chiang Li, Chong-Yung Chi, and ArulMurugan Ambikapathi. Identifiability of the simplex volume minimization criterion for blind hyperspectral unmixing: The no-pure-pixel case. *IEEE Transactions on Geoscience and Remote Sensing*, 53(10):5530–5546, 2015.

[Mnih and Salakhutdinov, 2008] Andriy Mnih and Ruslan R Salakhutdinov. Probabilistic matrix factorization. In *Advances in neural information processing systems*, pages 1257–1264, 2008.

[Salakhutdinov and Mnih, 2008] Ruslan Salakhutdinov and Andriy Mnih. Bayesian probabilistic matrix factorization using markov chain monte carlo. In *Proceedings of the 25th International Conference on Machine Learning*, ICML '08, pages 880–887, New York, NY, USA, 2008. ACM.

[Soni *et al.*, 2016] Akshay Soni, Swayambhoo Jain, Jarvis Haupt, and Stefano Gonella. Noisy matrix completion under sparse factor models. *IEEE Transactions on Information Theory*, 62(6):3636–3661, 2016.

[Van Der Aalst, 2011] Wil Van Der Aalst. *Process mining: discovery, conformance and enhancement of business processes*, volume 2. Springer, 2011.

[Wang and Carreira-Perpinán, 2013] Weiran Wang and Miguel A Carreira-Perpinán. Projection onto the probability simplex: An efficient algorithm with a simple proof, and an application. *arXiv preprint arXiv:1309.1541*, 2013.