Closed-form models for collaborative filtering with side-information

Closed-Form Models for Collaborative Filtering with Side-Information

OLIVIER JEUNEN and JAN VAN BALEN, Adrem Data Lab, University of Antwerp, Belgium
BART GOETHALS, Adrem Data Lab, University of Antwerp, Belgium and Monash University, Australia

Recent work has shown that, despite their simplicity, item-based models optimised through ridge regression can attain highly competitive results on collaborative filtering tasks. As these models are analytically computable and thus forgo the need for often expensive iterative optimisation procedures, they are an attractive choice for practitioners. We study the applicability of such closed-form models to implicit-feedback collaborative filtering when additional side-information or metadata about items is available. Two complementary extensions to the EASE^R paradigm are proposed, based on collective and additive models. Through an extensive empirical analysis on several large-scale datasets, we show that our methods can effectively exploit side-information whilst retaining a closed-form solution, and improve upon the state-of-the-art without increasing the computational complexity of the original EASE^R approach. Additionally, empirical results demonstrate that the use of side-information leads to more "long tail" items being recommended, benefiting the recommendations' coverage of the item catalogue.

ACM Reference Format:
Olivier Jeunen, Jan Van Balen, and Bart Goethals. 2020. Closed-Form Models for Collaborative Filtering with Side-Information. In *Fourteenth ACM Conference on Recommender Systems (RecSys '20), September 22–26, 2020, Virtual Event, Brazil.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3383313.3418480

1 INTRODUCTION

Most modern approaches to recommendation are based on some form of collaborative filtering [8]. As a consequence, the quest for more effective collaborative filtering algorithms is a very lively research area, where significant strides forward are being made every year. A long-going line of work has repetitively shown the competitiveness of simple linear models for collaborative filtering tasks [15, 19, 28, 32–35]. Most notably and recently, Embarrassingly Shallow Auto-Encoders (reversed: EASE^R) have been shown to yield highly competitive results with the state-of-the-art, in many cases outperforming complex neural network architectures whilst being much easier to implement, and much more efficient to compute [33]. The closed-form solution that is available for ridge regression models is at the heart of these major advantages; as EASE^R effectively optimises a regularised least-squares problem.

Several hurdles for recommender systems remain, such as the "long tail" (very few items account for the large majority of interactions) and "cold start" (new items do not have any interactions) issues [22, 27, 30]. It has become common practice to exploit item side-information or metadata to try and alleviate these problems, and several recent works show that they indeed succeed at this [3, 9]. In this work, we study the applicability of EASE^R-like models in the presence of such metadata. We present additive and collective EASE^R (ADD-EASE^R and CEASE^R), and show how these novel methods retain a closed-form solution whilst leveraging signals embedded in side-information to generate more effective recommendations. We show how these straightforward and complementary extensions of the EASE^R paradigm consistently outperform state-of-the-art approaches such as CVAE [3] and VLM [9]. Additionally, we empirically validate

Manuscript submitted to ACM

that ADD-EASE^R and CEASE^R are indeed able to soften the effect of the long tail, and are more likely to recommend different and less popular items than plain EASE^R. To summarise, the main contributions presented in this work are:

(1) We propose two natural extensions to the EASE^R paradigm: ADD-EASE^R and CEASE^R; and show how they retain closed-form solutions, without affecting EASE^R's computational complexity.
(2) Empirical results show that our proposed methods can improve upon EASE^R in terms of recommendation accuracy, most notably when the amount of training interactions is limited; additionally outperforming competing state-of-the-art approaches. An extensive empirical analysis shows that ADD-EASE^R and CEASE^R are more likely to recommend "long tail" items, and provide more catalogue coverage than EASE^R without side-information.[1]

2 BACKGROUND & RELATED WORK

Our use-case focuses on implicit-feedback data consisting of preference indications from users in $\mathcal{U}$ over items in $\mathcal{I}$, assumed from a set of interaction data $\mathcal{P} \subseteq \mathcal{U} \times \mathcal{I}$. These preferences can be represented in a binary user-item matrix $\boldsymbol{X} \in \{0,1\}^{|\mathcal{U}|\times|\mathcal{I}|}$, where $\boldsymbol{X}_{u,i} = 1$ if we have a click, view, purchase,... for user u and item i in $\mathcal{P}$, and 0 otherwise. Moreover, we are interested in the case where additional discrete side-information about items is available, such as release years, genres et cetera. We will refer to the set of all such tags as the *vocabulary* $\mathcal{V}$. In a similar fashion to the user-item matrix $\boldsymbol{X}$, a tag-item matrix $\boldsymbol{T} \in \mathbb{R}^{|\mathcal{V}|\times|\mathcal{I}|}$ is constructed.

Item-based collaborative filtering models aim to reconstruct the user-item matrix by approximating columns as a weighted sum of other columns: $\boldsymbol{X} \approx \boldsymbol{XS}$ [7, 26]. Ning and Karypis proposed to learn a sparse aggregation matrix $\boldsymbol{S}$, leading to the Sparse LInear Method (SLIM) [19]. SLIM optimises a least-squares regression model with elastic-net regularisation, constrained to positive weights. Many extensions of SLIM have been proposed in recent years, and it has become a widely used method for the collaborative filtering task [5, 6, 15, 20, 28, 33–35]. The efficiency of the original SLIM approach is a known impediment for its adoption in certain use-cases; related work has reported that hyper-parameter tuning took several weeks on large-scale datasets [17].[2]

Steck studied whether the restrictions of SLIM to only allow *positive* item-item weights and their l_1-regularisation-induced sparsity were necessary for the resulting model to remain competitive, and concluded that this was not always the case [33]. The resulting Tikhonov-regularised least-squares problem can then be formalised as

$$\boldsymbol{S}^* = \arg\min_{\boldsymbol{S}} \|\boldsymbol{X} - \boldsymbol{XS}\|_F^2 + \lambda \|\boldsymbol{S}\|_F^2 \text{, subject to } \mathrm{diag}(\boldsymbol{S}) = 0. \tag{1}$$

The restriction of the diagonal to zero, originally proposed in SLIM [19], avoids the trivial solution where $\boldsymbol{S} = \boldsymbol{I}$. The main advantage of simplifying the optimisation problem at hand, is that the well-known closed form solutions for Ordinary Least Squares (OLS) and ridge regression can now be adopted. Including the zero-diagonal constraint via Lagrange multipliers yields the Embarrassingly Shallow Auto-Encoder (EASE^R) model:

$$\hat{\boldsymbol{S}} = \boldsymbol{I} - \hat{\boldsymbol{P}} \cdot \mathrm{diagMat}(\vec{1} \oslash \mathrm{diag}(\hat{\boldsymbol{P}})), \text{ where } \hat{\boldsymbol{P}} \equiv (\boldsymbol{X}^\intercal \boldsymbol{X} + \lambda \boldsymbol{I})^{-1}. \tag{2}$$

As this model consists of a single regression problem to be solved and thus a single matrix inversion to be computed, its complexity is orders of magnitude smaller than that of the original SLIM variants. We refer the interested reader to [32, 33] for a full derivation of the model and additional information. Although inverting the regularised Gramian matrix still remains a bottleneck for large item catalogues, advantages of this closed-form expression over the traditional coordinate-descent optimisation procedure have been reported in terms of efficiency as well as recommendation accuracy [33, 35].

[1] To aid in the reproducibility of our work, we provide our source code at https://github.com/olivierjeunen/ease-side-info-recsys-2020/.
[2] It should be noted that the authors have since released a more performant coordinate-descent-based implementation of their method [21].

Collective SLIM (CSLIM) was proposed an extension of SLIM that exploits side-information [20]. CSLIM regularises the original SLIM objective with the side-information, solving the following optimisation problem:

$$S^* = \arg\min_S \frac{1}{2} \|X - XS\|_F^2 + \frac{\alpha}{2} \|T - TS\|_F^2 + \lambda \|S\|_F^2 + \beta \|S\|_1 \text{, subject to } S \geq 0 \text{, and } \text{diag}(S) = 0. \tag{3}$$

In this formulation, α is a hyper-parameter used to trade off the importance of the side-information in T with respect to the preference expressions in X. α, β and λ are typically optimised through a grid-search on a validation set, which will grow cubically with the number of possible values. Chen et al. further extended this framework to applications with high-dimensional side-information by incorporating dimensionality reduction techniques into the optimisation procedure [4].

Modern approaches often adopt Bayesian methods to model item metadata and exploit it for recommendation tasks, such as META-LDA [36], CTPF [10], CBVAE [16], CVAE [3] and VLM [9]. These latter two methods jointly model the user-item matrix and side-information through variational approximations, an approach that has been shown to be highly competitive for regular collaborative filtering tasks as well [17, 29]. As such, CVAE and VLM are the main competitors for our method, and the ones we will compare with in our experimental evaluation, together with SLIM and CSLIM. Note that while CVAE and VLM require libraries with automatic differentiation functionality such as Tensorflow or PyTorch [1, 23], EASE^R can be implemented in just a few lines of Python code.

3 CONTRIBUTION & METHODOLOGY

The aim of our work is to extend the EASE^R objective as presented in Equation 1 in order to incorporate side-information encoded in the tag-item matrix T. EASE^R's biggest advantage over competing methods is the fact that it is analytically computable and consists of solving a single regression problem, often leading to an efficiency advantage over competing methods. Naturally, we wish to retain this property in our extensions as well.

Collective EASE^R (CEASE^R). A first natural extension is to regularise the EASE^R objective to collectively solve the regression problem on X as well as T, analogous to CMF [31] and CSLIM [20]. This yields the objective shown in Equation 4, where α handles the trade-off between preference data and meta-data comparable to Equation 3. We will refer to this algorithm variant as Collective EASE^R (CEASE^R).

$$S^* = \arg\min_S \|X - XS\|_F^2 + \alpha \|T - TS\|_F^2 + \lambda \|S\|_F^2 \text{, subject to } \text{diag}(S) = 0 \tag{4}$$

From this formulation, it might seem non-trivial to obtain a closed-form solution for S. However, decomposing the l_2-norm clarifies its equivalence to solving a simpler objective that does indeed maintain it.

$$\|X - XS\|_F^2 + \alpha \|T - TS\|_F^2 = \|X - XS\|_F^2 + \left\|\sqrt{\alpha}T - \sqrt{\alpha}TS\right\|_F^2 = \left\|X' - X'S\right\|_F^2 \text{, where } X' = \begin{bmatrix} X \\ \sqrt{\alpha}T \end{bmatrix} \tag{5}$$

So, we only need to define X' by stacking the weighted user-item and tag-item matrices, and we can plug X' into EASE^R's Equation 2 to obtain its closed-form solution. Note that the regularisation strength α does not have to be a single scalar parameter that is equal for all tags, but that a different weight can be assigned to every tag or user, yielding a Weighted Linear Regression (WLS) problem as also remarked in [32]. The final CEASE^R objective and its closed-form solution are presented in Equations 6 and 7. The weight-matrix $W \in \mathbb{R}^{(|\mathcal{U}|+|\mathcal{V}|)\times(|\mathcal{U}|+|\mathcal{V}|)}$ is a diagonal matrix, where every weight $W_{u,u}$ corresponds to the relative *importance* a user or tag is given when solving the regression problem.

$$S^* = \arg\min_S \left\|\sqrt{W} \odot (X' - X'S)\right\|_F^2 + \lambda \|S\|_F^2 \text{, subject to } \text{diag}(S) = 0, \text{ where } X' = \begin{bmatrix} X \\ T \end{bmatrix} \tag{6}$$

$$\hat{\boldsymbol{S}} = \boldsymbol{I} - \hat{\boldsymbol{P}} \cdot \text{diagMat}(\vec{1} \oslash \text{diag}(\hat{\boldsymbol{P}})), \text{ where } \hat{\boldsymbol{P}} \equiv (\boldsymbol{X'}^{\intercal}\boldsymbol{W}\boldsymbol{X'} + \lambda\boldsymbol{I})^{-1} \tag{7}$$

In practice, the (weighted) Gram matrix $\boldsymbol{X'}^{\intercal}\boldsymbol{X'}$ will often be computed in a preprocessing step. Recent work has proposed efficient solutions to tackle this, most notably when $\boldsymbol{X'}$ is binary [14]. The bulk of the computational cost of the method then comes from the inversion of this matrix, which is dependent on $|\mathcal{I}|$, but neither on $|\mathcal{U}|$ nor $|\mathcal{V}|$. As such, the addition of side-information into the model comes without almost any added computational complexity when learning the actual model. It does not introduce any additional parameters, but rather updates the learning objective to reflect the information embedded in the meta-data. Intuitively, we can expect CEASE$^{\text{R}}$ to be most effective in cases where the linear modelling capacity of EASE$^{\text{R}}$ is sufficient to capture the underlying relation in the data. When the model capacity is constrained, introducing additional parameters might be more effective.

Additive EASE^{R} (ADD-EASE^{R}). Another option is to view the regression problem on the user-item matrix $\boldsymbol{X}$ and the one on the tag-item matrix $\boldsymbol{T}$ as two fully independent problems to solve in parallel, and combine the two resulting item-item weight matrices $\boldsymbol{S_X}$ and $\boldsymbol{S_T}$ in an additive fashion later down the line. We will refer to this model variant as Additive EASE$^{\text{R}}$ (ADD-EASE$^{\text{R}}$). The resulting objective is presented in Equation 8.

$$\begin{aligned}\boldsymbol{S}^{*} = \alpha \underset{\boldsymbol{S_X}}{\arg\min} \left(\left\| \sqrt{\boldsymbol{W_X}} \odot (\boldsymbol{X} - \boldsymbol{X}\boldsymbol{S_X}) \right\|_F^2 + \lambda_{\boldsymbol{X}} \left\| \boldsymbol{S_X} \right\|_F^2 \right) + (1-\alpha) \underset{\boldsymbol{S_T}}{\arg\min} \left(\left\| \sqrt{\boldsymbol{W_T}} \odot (\boldsymbol{T} - \boldsymbol{T})\boldsymbol{S_T} \right\|_F^2 + \lambda_{\boldsymbol{T}} \left\| \boldsymbol{S_T} \right\|_F^2 \right) \\ \text{Subject to } \text{diag}(\boldsymbol{S_X}) = \text{diag}(\boldsymbol{S_T}) = 0.\end{aligned} \tag{8}$$

ADD-EASE$^{\text{R}}$ doubles the amount of parameters used by EASE$^{\text{R}}$ and CEASE$^{\text{R}}$, increasing its degrees of freedom at learning time at the cost of having to solve two regression problems instead of one. Note, however, that these are fully independent and can be computed in parallel. Equation 9 shows the analytical formulas to obtain the two independent models, and combine it with a blending parameter $0 \leq \alpha \leq 1$.

$$\begin{aligned}\hat{\boldsymbol{S}}_{\boldsymbol{X}} &= \boldsymbol{I} - \hat{\boldsymbol{P}}_{\boldsymbol{X}} \cdot \text{diagMat}(\vec{1} \oslash \text{diag}(\hat{\boldsymbol{P}}_{\boldsymbol{X}})), \text{ where } \hat{\boldsymbol{P}}_{\boldsymbol{X}} \equiv (\boldsymbol{X}^{\intercal}\boldsymbol{W}_{\boldsymbol{X}}\boldsymbol{X} + \lambda_{\boldsymbol{X}}\boldsymbol{I})^{-1} \\ \hat{\boldsymbol{S}}_{\boldsymbol{T}} &= \boldsymbol{I} - \hat{\boldsymbol{P}}_{\boldsymbol{T}} \cdot \text{diagMat}(\vec{1} \oslash \text{diag}(\hat{\boldsymbol{P}}_{\boldsymbol{T}})), \text{ where } \hat{\boldsymbol{P}}_{\boldsymbol{T}} \equiv (\boldsymbol{T}^{\intercal}\boldsymbol{W}_{\boldsymbol{T}}\boldsymbol{T} + \lambda_{\boldsymbol{T}}\boldsymbol{I})^{-1} \\ \hat{\boldsymbol{S}} &= \alpha\hat{\boldsymbol{S}}_{\boldsymbol{X}} + (1-\alpha)\hat{\boldsymbol{S}}_{\boldsymbol{T}}\end{aligned} \tag{9}$$

This blending parameter α is computationally much more efficient to tune than with the CEASE$^{\text{R}}$ variant, as there is no need for model retraining when evaluating different values. Intuitively, we can expect ADD-EASE$^{\text{R}}$ to be most effective in cases where the modelling capacity of EASE$^{\text{R}}$ is *insufficient* to capture the underlying relation in the data - in contrast to CEASE$^{\text{R}}$. Indeed, introducing additional parameters to a model will be most effective when the original model's modelling capacity is already saturated . As such, the two approaches we introduce in this work are complementary and we expect them to excel in different settings, which is confirmed by our empirical observations.

4 EXPERIMENTAL EVALUATION

The research questions we wish to answer in this work are the following:

RQ1 Are CEASE$^{\text{R}}$ and ADD-EASE$^{\text{R}}$ competitive with the state-of-the art in collaborative filtering with side-information?
RQ2 How do CEASE$^{\text{R}}$ and ADD-EASE$^{\text{R}}$ behave when training data becomes scarce, compared to other methods?
RQ3 Are CEASE$^{\text{R}}$ and ADD-EASE$^{\text{R}}$ more likely to recommend "long tail" items from the training set than vanilla EASE$^{\text{R}}$?
RQ4 Are CEASE$^{\text{R}}$ and ADD-EASE$^{\text{R}}$ more likely to diversify recommendations over all items than vanilla EASE$^{\text{R}}$?

Table 1. Training datasets used for our experimental evaluation, after a train-validation-test split has occurred.

Name	nnz(**X**)	$\lvert\mathcal{U}\rvert$	$\lvert\mathcal{I}\rvert$	$\overline{\lvert\mathcal{U}_i\rvert}$	$\overline{\lvert\mathcal{I}_u\rvert}$	nnz(**T**)	$\lvert\mathcal{V}\rvert$	$\overline{\lvert\mathcal{V}_i\rvert}$	Metadata
MovieLens-20M	9M	117k	21k	411	73	111k	11k	5	*Genre, year, director, writers*
Netflix	47M	383k	18k	2.6k	123	78k	6k	4	*Genre, year, director, writers*
Million Song Dataset	28M	471k	41k	675	59	1.5M	51k	37	*Genre, artist, tags*
Yahoo! Movies	150k	7k	10k	20	15	72k	11k	7	*Genre, year, director, writers*
Amazon Video Games	118k	16k	11k	11	7	898k	21k	85	*Title, brand, description*
Amazon Sports & Outdoors	170k	27k	18k	9	6	1.1M	25k	62	*Title, brand, description*

We adopt the same evaluation procedure as Liang et al. [17] and subsequent works [9, 29, 33], focusing on the *strong generalisation* setting where *users* are split into disjoint training/validation/test sets. Because of space limitations, we refer to these works for further details. As the advantages of incorporating item side-information into the collaborative filtering process are most tangible when training user-item interactions are limited, we additionally include three smaller datasets in our experiments. Models on these smaller datasets are evaluated through the widely-used leave-one-out protocol: for every user, we randomly sample two items to be held-out and used for the validation and test sets respectively. It has been correctly noted that this random splitting procedure violates the sequential ordering of user-item interactions in the data, which can prohibit effective offline evaluation [12, 13, 25]; the performance of these methods in sequential settings warrants further investigation that falls outside of the scope of this work [24]

Table 1 provides an overview of the datasets we use throughout this work, along with basic statistics about sparsity and the metadata used. All datasets are binarised and interpreted as implicit feedback. For the MovieLens-20M [11], Netflix [2], Yahoo! Movies, Amazon Video Games and Amazon Sports & Outdoors datasets [18], we removed all ratings lower than 4 and retained only users who had at least 5 rated items left. Additional publicly available metadata for the movie datasets was obtained through IMDB and matched using fuzzy techniques.

We compare with the state-of-the-art CVAE [3],VLM [9], SLIM and CSLIM [21] approaches, and further report results for an item-kNN method using cosine similarity (COS) [26] and the original EASE$^{\text{R}}$ formulation [33]. All methods' hyperparameters were tuned through a grid-search on the validation set, best performers were trained until convergence. The advantages of EASE$^{\text{R}}$ over vanilla SLIM in terms of recommendation accuracy as well as efficiency on large datasets have been studied in recent work [33, 35]; these effects will be exacerbated by CSLIM, as the efficiency of its optimisation procedure directly depends on the dimensionality of the tag-item matrix, and CSLIM's three hyper-parameters are costly to tune properly. As our computational resources are limited, we only include the SLIM and CSLIM baselines on the smaller datasets. Metadata can be seen as an alternative source of information to learn similarity between items when their mutual information in the user-item matrix $\boldsymbol{X}$ is scarce. Two movies might never co-occur in the set of training interactions, but if we know that they are both sci-fi movies from the same year, we can still infer some valuable signal. From this perspective, it comes naturally that side-information can be exploited most effectively when the training set of user-item interactions is limited. This effect also emerges in the empirical results reported by Elahi et al. [9], where the introduction of side-information for MovieLens-20M only affects a single metric for 0.002. Because of this, we report results for models trained on subsets of the training data by subsampling users and their interactions, as also done in [35], in addition to models trained on the full datasets.

Key observations from the results presented in Table 2 include: (1) EASE$^{\text{R}}$, despite its simplicity, outperforms the Bayesian approaches on every setting, most notably when the amount of available training data is scarce. (2) The CVAE and VLM approaches are greatly impacted in terms of performance when training data is subsampled, even being outperformed

Table 2. Experimental results for the collaborative filtering task on real-world datasets, with subsampled training users. The best performing result for every setting is shown **bold**. The dashed line divides methods that learn from metadata with those that do not.

		MovieLens-20M			*Netflix*			*Million Song Dataset*		
$\lvert\mathcal{U}_{\text{train}}\rvert$	**Model**	Recall @20	Recall @50	NDCG @100	Recall @20	Recall @50	NDCG @100	Recall @20	Recall @50	NDCG @100
1%	COS	0.266	0.370	0.306	0.202	0.267	0.240	0.169	0.236	0.215
	VLM	0.268	0.371	0.294	0.270	0.351	0.306	0.115	0.165	0.146
	EASE^{R}	0.321	0.434	0.357	0.314	0.389	0.350	0.212	0.282	0.258
	CVAE	0.170	0.270	0.214	0.163	0.236	0.208	0.044	0.072	0.062
	VLM$_{\text{SIDE}}$	0.278	0.385	0.304	0.270	0.352	0.306	0.117	0.169	0.149
	CEASE^{R}	0.328	0.443	0.368	**0.317**	**0.393**	**0.353**	**0.246**	**0.330**	**0.294**
	ADD-EASE^{R}	**0.332**	**0.449**	**0.371**	0.315	0.392	0.352	0.242	0.326	0.292
100%	COS	0.280	0.382	0.321	0.202	0.268	0.241	0.236	0.319	0.294
	VLM	0.378	0.514	0.419	0.329	0.420	0.371	– did not finish after 24h –		
	EASE^{R}	**0.387**	0.516	0.429	**0.362**	**0.444**	**0.399**	**0.331**	**0.425**	0.390
	CVAE	0.314	0.450	0.362	0.289	0.370	0.332	0.201	0.264	0.223
	VLM$_{\text{SIDE}}$	0.377	0.514	0.419	0.329	0.420	0.372	– did not finish after 24h –		
	CEASE^{R}	**0.387**	0.516	0.429	**0.362**	**0.444**	**0.399**	**0.331**	**0.425**	**0.391**
	ADD-EASE^{R}	**0.387**	**0.517**	**0.430**	**0.362**	**0.444**	**0.399**	**0.331**	**0.425**	**0.391**
		Yahoo! Movies			*Amazon Video Games*			*Amazon Sports*		
100%	COS	0.445	0.559	0.266	0.116	0.185	0.074	0.042	0.070	0.030
	VLM	0.502	0.635	0.297	0.156	0.241	0.099	0.062	0.099	0.042
	SLIM	0.515	0.637	0.309	0.185	0.272	0.117	0.090	0.130	0.058
	EASE^{R}	0.522	0.646	0.310	0.184	0.270	0.116	0.091	0.131	0.058
	CVAE	0.362	0.525	0.205	0.064	0.119	0.047	0.029	0.056	0.022
	VLM$_{\text{SIDE}}$	0.500	0.637	0.297	0.155	0.238	0.098	0.062	0.101	0.042
	CSLIM	0.529	0.656	0.317	0.185	0.272	0.117	0.092	0.135	0.061
	CEASE^{R}	**0.542**	**0.670**	**0.321**	**0.187**	**0.277**	**0.118**	0.100	0.149	0.065
	ADD-EASE^{R}	0.530	0.660	0.317	0.186	0.276	**0.118**	**0.105**	**0.160**	**0.068**

by the COS baseline in some settings. (3) ADD-EASE^{R} and CEASE^{R} are effective in exploiting side-information for enhanced recommendation accuracy. The effects are most palpable when training data is limited, but consistent nevertheless. Furthermore, they retain EASE^{R}'s biggest advantage: a closed-form solution that is many times more efficient than the deep learning alternatives. Whereas VLM needs hours of training on a GPU for the large datasets, a CEASE^{R} model needs less than 20 minutes for the Million Song Dataset, and less than 3 for Netflix and MovieLens. The key observations from the experiments on the large datasets also hold for the smaller datasets, but the effects of introducing metadata to EASE^{R} are more explicit. Naturally, these results are all highly dependent on the quality of the side-information that is used.

Lifting the Long Tail. Aside from purely looking at recommendation accuracy in terms of being able to correctly identify held-out items, investigating the actual recommendations that are generated has its merits as well. We visualise the results of our analysis on the two most extreme datasets in terms of catalogue size: Yahoo! Movies and the Million Song Dataset (trained on 1% of the users, corresponding to the top rows of Table 2).

Figures 1(a,b) show the cumulative distribution function for the probability that an item appears in the top-100 recommendations that were generated for test users, with the catalogue size on the x-axis sorted based on the popularity of an item in the training set. We can observe that, for EASE^{R} and MSD, roughly 90% of the recommendations consist of

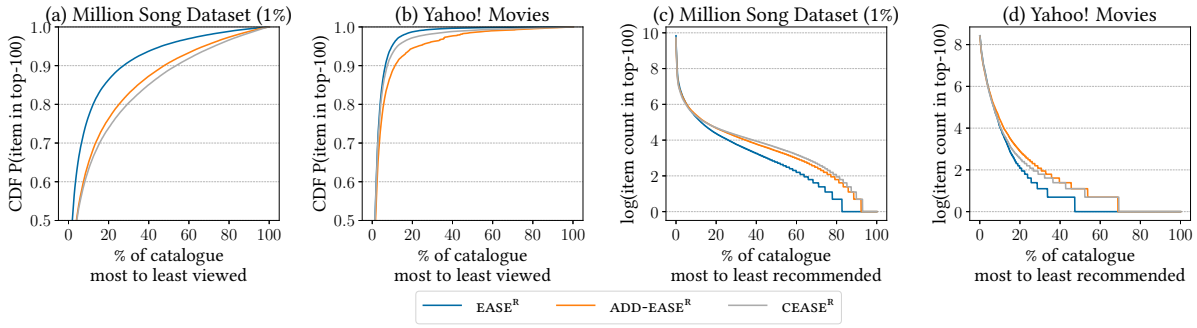

Fig. 1. When analysing the recommended items for EASE^R and its variants that exploit side-information, we see that (a,b) side-information helps to allocate more recommendations to items that were less popular in the training data and (c,d) side-information helps to diversify over the entire item catalogue and recommend long-tail items more often.

just the 25% most popular training items. By using side-information, CEASE^R and ADD-EASE^R bring this down to roughly 80%, effectively doubling the exposure for 75% of the item catalogue. With the Yahoo! Movies dataset, the absolute effects are smaller but the relative effects are larger. Here, the proportion of exposure for the 75% long tail items can be increased by a factor of 2.6 and 5.4 with CEASE^R and ADD-EASE^R respectively.

An algorithm that consistently recommends the 100 least popular items from the training set would look even better on such a visualisation, whilst of course not being desirable. Another interesting thing to look at is the frequency with which items occur in the top-100 recommendations generated for test users. The logarithm of these counts is shown in Figures 1(c,d); a similar analysis was done in [33]. These figures show that 17% of the MSD item catalogue is never recommended in the top-100 by EASE^R. ADD-EASE^R and CEASE^R effectively bring this number down to 7%. The results are even more dramatic for Yahoo! Movies: from 52% to 30%. Furthermore, we can observe a general trend that the distribution of recommendations over the item catalogue is less *heavy-tailed* when using side-information. More formally, the entropy of the distribution of recommendations over the item catalogue increases, along with coverage.

5 CONCLUSION

We introduced two extensions to the EASE^R algorithm for collaborative filtering, in order to naturally handle item side-information or metadata. We have shown how our proposed CEASE^R and ADD-EASE^R models retain a closed-form solution, which is arguably EASE^R's biggest advantage over competing methods. In an extensive empirical evaluation on six publicly available real-world datasets, we have validated that they can effectively improve upon EASE^R's ranking accuracy, most notably in those cases where the amount of training data is limited. Furthermore, we have demonstrated that side-information helps the model to be (1) less prone to popularity bias in the training data, and (2) more likely to recommend items spanning the entire item catalogue, improving on coverage. We have released the source code needed to reproduce our experiments in full. For simplicity and brevity, we set the weight matrix $\boldsymbol{W}$ for CEASE^R and ADD-EASE^R to the identity matrix $\boldsymbol{I}$, while tuning a single scalar hyper-parameter α. Preliminary results show that the choice of weights can have a high impact on the recommendation accuracy of the resulting model. Heuristics or principled approaches to learn a heterogeneous $\boldsymbol{W}$ can undoubtedly boost the performance of the methods we propose in this work.

ACKNOWLEDGMENTS

This research received funding from the Flemish Government (AI Research Program).

REFERENCES

[1] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D. G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng. 2016. TensorFlow: A System for Large-Scale Machine Learning. In *Proc. of the 12th USENIX Conference on Operating Systems Design and Implementation (OSDI'16)*. 265–283.

[2] J. Bennett, S. Lanning, et al. 2007. The Netflix prize. In *Proc. of the KDD cup and workshop*, Vol. 2007. 35.

[3] Y. Chen and M. de Rijke. 2018. A Collective Variational Autoencoder for Top-N Recommendation with Side Information. In *Proc. of the 3rd Workshop on Deep Learning for Recommender Systems, DLRS@RecSys*. 3–9.

[4] Y. Chen, X. Zhao, and M. de Rijke. 2017. Top-N Recommendation with High-Dimensional Side Information via Locality Preserving Projection. In *Proc. of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '17)*. ACM, 985–988.

[5] E. Christakopoulou and G. Karypis. 2014. HOSLIM: Higher-Order Sparse LInear Method for Top-N Recommender Systems. In *Advances in Knowledge Discovery and Data Mining*. Springer International Publishing, 38–49.

[6] E. Christakopoulou and G. Karypis. 2016. Local Item-Item Models For Top-N Recommendation. In *Proc. of the 10th ACM Conference on Recommender Systems (RecSys '16)*. ACM, 67–74.

[7] M. Deshpande and G. Karypis. 2004. Item-Based Top-N Recommendation Algorithms. *ACM Trans. Inf. Syst.* 22, 1 (Jan. 2004), 143–177.

[8] M. D. Ekstrand, J. T. Riedl, and J. A. Konstan. 2011. Collaborative Filtering Recommender Systems. *Found. Trends Hum.-Comput. Interact.* 4, 2 (Feb. 2011), 81–173.

[9] E. Elahi, W. Wang, D. Ray, A. Fenton, and T. Jebara. 2019. Variational Low Rank Multinomials for Collaborative Filtering with Side-information. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 340–347.

[10] P. K. Gopalan, L. Charlin, and D. Blei. 2014. Content-based recommendations with Poisson factorization. In *Advances in Neural Information Processing Systems 27*. 3176–3184.

[11] F. M. Harper and J. A. Konstan. 2015. The MovieLens Datasets: History and Context. *ACM Transactions on Interactive Intelligent Systems* 5, 4, Article 19 (2015), 19 pages.

[12] O. Jeunen. 2019. Revisiting Offline Evaluation for Implicit-feedback Recommender Systems. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 596–600.

[13] O. Jeunen, K. Verstrepen, and B. Goethals. 2018. Fair Offline Evaluation Methodologies for Implicit-feedback Recommender Systems with MNAR Data. In *Proc. of the REVEAL 18 Workshop on Offline Evaluation for Recommender Systems (RecSys '18)*.

[14] O. Jeunen, K. Verstrepen, and B. Goethals. 2019. Efficient Similarity Computation for Collaborative Filtering in Dynamic Environments. In *Proc. of the 13th ACM Conference on Recommender Systems (RecSys '19)*. ACM, 251–259.

[15] M. Levy and K. Jack. 2013. Efficient top-N recommendation by linear regression. In *RecSys Large Scale Recommender Systems Workshop*.

[16] X. Li and J. She. 2017. Collaborative Variational Autoencoder for Recommender Systems. In *Proc. of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '17)*. 305–314.

[17] D. Liang, R. G. Krishnan, M. D Hoffman, and T. Jebara. 2018. Variational autoencoders for collaborative filtering. In *Proc. of the 2018 World Wide Web Conference (WWW '18)*. International World Wide Web Conferences Steering Committee, ACM, 689–698.

[18] J. McAuley, C. Targett, Q. Shi, and A. van den Hengel. 2015. Image-Based Recommendations on Styles and Substitutes. In *Proc. of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '15)*. ACM, 43–52.

[19] X. Ning and G. Karypis. 2011. SLIM: Sparse Linear Methods for Top-N Recommender Systems. In *Proc. of the 2011 IEEE 11th International Conference on Data Mining (ICDM '11)*. IEEE Computer Society, 497–506.

[20] X. Ning and G. Karypis. 2012. Sparse Linear Methods with Side Information for Top-n Recommendations. In *Proc. of the 6th ACM Conference on Recommender Systems (RecSys '12)*. 155–162.

[21] X. Ning, A. N. Nikolakopoulos, Z. Shui, M. Sharma, and G. Karypis. 2019. *SLIM Library for Recommender Systems*. https://github.com/KarypisLab/SLIM

[22] Y. Park and A. Tuzhilin. 2008. The Long Tail of Recommender Systems and How to Leverage It. In *Proc. of the 2008 ACM Conference on Recommender Systems (RecSys '08)*. 11–18. https://doi.org/10.1145/1454008.1454012

[23] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. 8026–8037.

[24] M. Quadrana, P. Cremonesi, and D. Jannach. 2018. Sequence-Aware Recommender Systems. *ACM Comput. Surv.*, Article Article 66 (July 2018), 36 pages.

[25] N. Sachdeva, G. Manco, E. Ritacco, and V. Pudi. 2019. Sequential Variational Autoencoders for Collaborative Filtering. In *Proc. of the 12th ACM International Conference on Web Search and Data Mining (WSDM '19)*. 600–608.

[26] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl. 2001. Item-based Collaborative Filtering Recommendation Algorithms. In *Proc. of the 10th International Conference on World Wide Web (WWW '01)*. ACM, 285–295.

[27] Andrew I. Schein, Alexandrin Popescul, Lyle H. Ungar, and David M. Pennock. 2002. Methods and Metrics for Cold-start Recommendations. In *Proc. of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02)*. ACM, 253–260.

[28] S. Sedhain, A. K. Menon, S. Sanner, and D. Braziunas. 2016. On the Effectiveness of Linear Models for One-Class Collaborative Filtering. In *Proc. of the 30th AAAI Conference on Artificial Intelligence (AAAI'16)*. 229–235.

[29] I. Shenbin, A. Alekseev, E. Tutubalina, V. Malykh, and S. I. Nikolenko. 2020. RecVAE: A New Variational Autoencoder for Top-N Recommendations with Implicit Feedback. In *Proc. of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 528–536.

[30] Y. Shi, M. Larson, and A. Hanjalic. 2014. Collaborative Filtering beyond the User-Item Matrix: A Survey of the State of the Art and Future Challenges. *ACM Comput. Surv.* 47, 1, Article Article 3 (May 2014), 45 pages.

[31] A. P. Singh and G. J. Gordon. 2008. Relational Learning via Collective Matrix Factorization. In *Proc.of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD '08)*. 650–658.

[32] H. Steck. 2019. Collaborative Filtering via High-Dimensional Regression. *CoRR* abs/1904.13033 (2019). arXiv:1904.13033

[33] H. Steck. 2019. Embarrassingly Shallow Autoencoders for Sparse Data. In *The World Wide Web Conference (WWW '19)*. 3251–3257.

[34] H. Steck. 2019. Markov Random Fields for Collaborative Filtering. In *Advances in Neural Information Processing Systems 32*. 5473–5484.

[35] H. Steck, M. Dimakopoulou, N. Riabov, and T. Jebara. 2020. ADMM SLIM: Sparse Recommendations for Many Users. In *Proc. of the 13th International Conference on Web Search and Data Mining (WSDM '20)*. 555–563.

[36] H. Zhao, L. Du, W. Buntine, and G. Liu. 2017. MetaLDA: A Topic Model that Efficiently Incorporates Meta Information. In *2017 IEEE International Conference on Data Mining (ICDM)*. 635–644.