Multi-Contextual Recommender using 3D Latent Factor Models and Online Tensor Decomposition

Ali Anaissi[1], Basem Suleiman[1], Muhammad Johan Alibasa[2], and Harrison Truong[1]

[1] School of Computer Science, University of Sydney, Australia
{basem.suleiman,ali.anaissi,harrison.truong}@sydney.edu.au
[2] School of Computing, Telkom University, Indonesia
alibasa@telkomuniversity.ac.id

Abstract. Traditional approaches to recommendation systems involve using collaborative filtering and content-based techniques which make use of the similarities between users and items respectively. Such approaches evolved to encapsulate model-based latent factor (LF) algorithms that use matrix decomposition to ingest a user-item matrix of ratings to generate recommendations. In this paper, we propose a novel approach based on 3D LF model and tensor decomposition method for devising personalized recommendations driven from additional contextual features. We also present our stacking method for a tensor generation prior to incorporating LF models. We validate our proposed personalized recommender using two real-world datasets. Our experimental results showed that additional contextual features can help to personalize recommendations while maintaining similar or better performance compared to conventional 2D LF methods. Furthermore, our results demonstrate the importance of the quality of the contextual features to be used in 3D LF models. In addition, our experiments show the effective performance of our new stacking method on computation time and accuracy of the proposed recommender.

1 Introduction

Data on interactions between users and products is a major resource that is increasingly collected and used by all online platforms today. The development of recommendation systems has been effective in assisting consumers in sifting through seemingly limitless options to more efficiently decide on products or services to consume [9]. Recommendation systems have proliferated throughout various industries, emerging in products and applications such as Netflix, Spotify, and Amazon with an abundance of literature released by both academia and industry [4]. As a result, there have been constant advancements to more accurately predict user preferences whilst optimizing for scalability and efficiency [3, 22]. Since the Netflix Prize Competition in 2006 [13], there has been a substantial increase in the use of 2D Latent Factor models to generate recommendations based on each user's explicit ratings.

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

In this paper, we build upon the use of 2D latent factor models in recommendation systems. Specifically, we aim to create 3D LF models that are capable of generating more personalized recommendations by incorporating additional information in the form of contextual features whilst maintaining benchmarked performances in accuracy. Due to advancements in machine learning techniques, the availability of more data and processing power, higher rank-decomposition methods on higher-order matrices (i.e. tensors) are now possible, where n-rank tensors can be decomposed using methods such as CP decomposition and Tucker decomposition. In this paper, we introduce a 3D LF model and CP decomposition method to help produce personalized recommendations. Furthermore, we investigate the performance of our approach to determine whether incorporating a third dimension, referred to as context can result in similar or better accuracy compared to conventional 2D methods. As such, the contributions made by this paper are: (1) A novel approach based on 3D Latent Factor model and tensor decomposition method for devising personalized recommendations driven from additional contextual information. (2) A novel stacking method for tensor generation prior to incorporating LF models. (3) Empirical validation of the performance of our 3D LF models and its dependency on the contextual features chosen using two real-world datasets

2 Related Work

In recommender systems, Content-Based Filtering (CB) method is based solely on how a user interacts with specific content and involves creating a user profile using past behaviour to classify content on whether a user would or would not consume the content [14]. Collaborative Filtering (CF), on the other hand, groups together users who interact similarly with content (e.g., provide the same ratings for the same movies) and generates recommendations based on content others in the same group have also interacted with [18]. A pure CB system does not require other user information compared to CF and thus can cater specifically to the niche interests and viewing behaviours of the user. However, it requires an extra amount of data on the content itself to compute similarities to make effective recommendations. CB recommenders also tend to overspecialize, resulting in recommendations that only cater for a user's historical preferences but do not facilitate the discovery of different content.

To overcome the above disadvantages, hybrid recommender approaches were proposed to provide a more optimal solution [6]. Hybrid systems can be effectively initiated by utilizing the user profile data gained from a CB system, thereby overcoming the cold start problem experienced by CF systems. With visual items, the continued collection of user usage information will also allow content to be filtered out (if users have watched a video already) and recommended (if they want to continue watching) [4]. Over-specialization of CB systems is also negated as the discovery of new content is facilitated by CF systems.

Other techniques for improving recommendation systems involve the use of 2D LF Latent Factor (LF) models. LF Models are a branch of collaborative filtering that ingest a user-item matrix of ratings to predict unknown ratings

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

via dimensionality reduction. It was first proposed as a case study on a movie dataset and an e-commerce dataset [17]. The success of LF models was proven during the Netflix Prize competition [13]. Since then, this area of research has proliferated, with accuracy and performance continuously optimized by adjusting the methodologies that applied Singular Value Decomposition (SVD) to the user-item matrix. This included different SVD methods such as the development of Regularised SVD [13] and the use of neighboring computational techniques alongside SVD such as Boltzmann Machines with SVD [16].

Other studies investigated recommendation systems and tensor decomposition where it is suggested that adding context, or additional features, could improve the accuracy of model-based recommender systems [8, 19]. For example, when watching movies, other contextual features such as the "time of day watched" or "whom you watch with i.e. with a group or solely with your partner" may contain valuable information that will affect the ratings given by a user. Further research builds on this idea of contextually-aware recommender systems and has proposed other techniques that ingest higher level matrices beyond the user-item matrix. This includes models that deal with explicit feedback using factorisation machines [15] and multi-layer tensor factorisation methods [20].

3 Methodology

Datasets: In this paper, two datasets will be used to address the research problem. The first dataset is the 100K MovieLens dataset, which describes ratings (ranging from 1 to 5), given by 943 users across 1682 movies. The data is a sparse matrix where not all users have rated all movies. The sparsity for this dataset is 6.3%. The second dataset is the Book Crossing dataset, which describes ratings from 1 to 10, given by 278,858 users across 271,379 books with 1,149,780 ratings that are both implicit and explicit. The rating data becomes a user-item matrix with a sparsity of 0.00152%. The dataset was cleaned by removing users from the dataset that had missing/incorrect age and year of publication values.

To overcome the scalability issues when implementing the latent factor algorithm for ratings predictions, a subset of the dataset is extracted. The Book Crossing dataset, with over 1 million ratings, is substantially larger than the 100,000 ratings provided by the MovieLens dataset. After basic cleansing of data, the following steps were taken to create a subset of the Book Crossing dataset. All zero ratings were removed (handling implicit feedback is beyond the scope of this paper). The top 1000 most actively rated users was identified. From these users, the 2000 most rated books was extracted. From this list, all users who have less than 10 ratings in the new subset of data were removed.

Application of 2D LF Models using SVD in Recommendation Context: Before describing the challenge of sparsity in the dataset and the following experiments, an overview of 2D and 3D matrix and tensor decomposition will be given respectively. The 2D SVD (Singular Value Decomposition) in a recommendation context is initiated by the generation of a user-item matrix from a list of ratings provided by users [3, 22]. Since not every user can consume or rate every available item, several missing values in the user-item matrix will need to

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

be filled before prediction. These null values will ultimately correspond to the ratings that will be predicted by the recommendation algorithm.

To apply SVD, imputation methods are used to fill in the sparse user-item matrix. It is evident that imputation methods can impact the quality of prediction and will consequently improve upon the final predicted ratings of each user to each item [3]. Once imputed, SVD is applied to deconstruct the original matrix into smaller matrices describing a series of latent factors (consisting of eigenvalues and eigenvectors). A subset of these latent factors will then be selected to reconstruct a new matrix. The newly reconstructed matrix will house the predicted ratings of each user for each particular item.

It should be noted that SVD has been investigated in the related literature for different purposes [3, 22]. This includes using (a) different types of SVD, (b) different rank values to select the best subset of latent factors for matrix reconstruction, and (c) different imputation methods to fill in missing values. This is important as seen from the Netflix Prize in attaining optimal results.

Application of 3D Latent Factor Models in a Recommendation Context: 3D tensor factorization in a recommendation context, unsurprisingly, shares a similar process compared to 2D tensor factorization. However, there are some notable differences. First, the requirement for datasets with additional data (besides ratings) to be used as the third dimension (contextual feature). This may include dimensions such as time of consumption, gender, age, or demographic information. Theoretically, any dimension that has the potential to discover latent relationships between users, items, and their ratings can be used as the third dimension. The second difference is the type of tensor decomposition technique used. The two common types of tensor decomposition techniques are CP Decomposition [10] and Tucker Decomposition [21]. CP Decomposition will be used in our method because it was identified that no one had used CP Decomposition in this type of application previously. The difference is the imputation method. Regarding the imputation step, which is found also in 2D SVD, imputation may potentially be required twice during 3D tensor factorization. Once, when generating the user-item matrix, and the second time when creating the user-item-context matrix. This represents an additional challenge and will be addressed in one of the experiments undertaken during this paper.

Imputation Methods: The datasets used in our study have a high level of sparsity as a result of not every user being able to consume every available item. Imputation methods are, therefore, required to fill these missing values and allow the application of decomposition techniques. The imputation methods adopted in our approach are: (1) *Filling with Zeros* where all missing values are filled with zeros. This method is typically the most basic and least effective imputation method. Due to its simplicity, it will be used as the simplest benchmark to compare results between 2D SVD and 3D CP Decomposition. (2) *Filling with Item Averages* where missing values of each item will be filled in using the average of all ratings received for that item (movie or book). (3) *Filling with User Averages* where the missing values of each user will be filled in using the average of all ratings given by that user. These imputation methods are chosen to

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

show how robust the performance of a 3D latent factor model is when predicting ratings and compared to a 2D latent factor model.

Contextual Features: The choice of contextual features is an independent variable that will affect the results of the experiments conducted in this paper. For the MovieLens dataset, the two contextual features used are Age and Gender as they will form the context component of the user-item-content matrix and will allow the 3D latent factor model to generate more personalized predictions. The contextual features used for the Book Crossing dataset are Age and Year of publication which can be important to personalize recommendations.

Nesterov CP Decomposition (NeCPD): Here, we present our tensor decomposition method which adopts NeCPD. Given an N^{th}-order tensor $\mathcal{X} \in \mathbb{R}^{I_1 \times \cdots \times I_N}$, NeCPD initially divides the CP problem into a convex N sub-problems since its loss function L is non-convex problem which may have many local minima. For simplicity, we present our method based on three-way tensor data. However, it can be naturally extended to handle a higher-order tensor.

In a three-way tensor $\mathcal{X} \in \Re^{I \times J \times K}$, $\mathcal{X}$ can be decomposed into three matrices $A \in \Re^{I \times R}$, $B \in \Re^{J \times R}$and $C \in \Re^{K \times R}$, where R is the latent factors. Following the SGD, we need to calculate the partial derivative of the loss function L defined in Equation 1 with respect to the three modes A, B and C alternatively.

The rational idea of the SGD algorithm depends only on the gradient information of $\frac{\partial L}{\partial w}$. In such non-convex setting, this partial derivative may encounter data points with $\frac{\partial L}{\partial w} = 0$ even though it is not at a global minimum. These data points are known as saddle points which may detente the optimization process to reach the desired local minimum if not escaped [1]. These saddle points can be identified by studying the second-order derivative (aka Hessian) $\frac{\partial L}{\partial w}^2$. Theoretically, when the $\frac{\partial L}{\partial w}^2(x; w) \succ 0$, x must be a local minimum; if $\frac{\partial L}{\partial w}^2(x; w) \prec 0$, then we are at a local maximum; if $\frac{\partial L}{\partial w}^2(x; w)$ has both positive and negative eigenvalues, the point is a saddle point. The second-order methods guarantee convergence, but the computing of Hessian matrix $H^{(t)}$ is high, which makes the method infeasible for high dimensional data and online learning. Ge *et al.*[1] show that saddle points are very unstable and can be escaped if we slightly perturb them with some noise. Based on this, we use the perturbation approach which adds Gaussian noise to the gradient. This reinforces the next update step to start moving away from that saddle point toward the correct direction. After a random perturbation, it is highly unlikely that the point remains in the same band and hence it can be efficiently escaped (i.e., no longer a saddle point). Since we are interested in fast optimization process due to online settings, we further incorporate Nesterov's method into the perturbed-SGD algorithm to accelerate the convergence rate.

$$L(\mathcal{X}, A, B, C) = \min_{A,B,C} \|\mathcal{X} - \sum_{r=1}^{R} A_{ir} \circ B_{jr} \circ C_{kr}\|_f^2, \tag{1}$$

$$A^{(t+1)} := A^{(t)} + \eta^{(t)} \nu^{(A,t)} + \epsilon - \beta ||A||_{L_1} \tag{2}$$

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

where

$$\nu^{(A,t)} := \gamma\nu^{(A,t-1)} + (1-\gamma)\frac{\partial L}{\partial A}(X^{(1,t)}, A^{(t)}) \tag{3}$$

Recently, Nesterov's Accelerated Gradient (NAG) [11] has received much attention for solving convex optimization problems [5, 12, 2]. It introduces a smart variation of momentum that works slightly better than standard momentum. This technique modifies the traditional SGD by introducing velocity ν and friction γ, which tries to control the velocity and prevents overshooting the valley while allowing faster descent.

Algorithm 1: NeCPD algorithm

NeCPD
Input: Tensor $X \in \Re^{I\times J\times K}$, number of components R
Output: Matrices $A \in \Re^{I\times R}$, $B \in \Re^{J\times R}$ and $C \in \Re^{K\times R}$
1: Initialize A, B, C
2: Repeat
3: Compute the partial derivative of A, B and C
3: Compute ν of A, B and C using Equation 3
4: Update the matrices A, B and C using Equation 2
6: until convergence

Our idea behind Nesterov's [11] is to calculate the gradient at a position that we know our momentum is about to take us instead of calculating the gradient at the current position. In practice, it performs a simple step of gradient descent to go from $w^{(t)}$ to $w^{(t+1)}$, and then it shifts slightly further than $w^{(t+1)}$ in the direction given by $\nu^{(t-1)}$. In this setting, we model the gradient update step with NAG as shown in Equation 2 and 3 where ϵ is a Gaussian noise, $\eta^{(t)}$ is the step size, and $||A||_{L_1}$ is the regularization and penalization parameter into the L_1 norms to achieve smooth representations of the outcome and thus bypassing the perturbation surrounding the local minimum problem. The updates for $(B^{(t+1)}, \nu^{(B,t)})$ and $(C^{(t+1)}, \nu^{(C,t)})$ are similar to the aforementioned ones. With NAG, our method achieves a global convergence rate of $O(\frac{1}{T^2})$ comparing to $O(\frac{1}{T})$ for traditional gradient descent. Based on the above models, we present our NeCPD algorithm 1.

Experiment 1 Setup: the aim of this is to evaluate whether our proposed 3D latent factor model using CP Decomposition can generate personalized predictions (via the incorporation of the contextual feature) whilst maintaining rating prediction accuracy compared to a 2D SVD approach. To predict ratings using CP Decomposition, a 3D tensor describing the user-item-context matrix was prepared from the standard user-item dataset. This involved adding the contextual feature along the third axis of the user-item matrix. For the Age contextual feature of the MovieLens dataset, different age buckets were tested to

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

determine whether this would affect performance. These age buckets were 0 – 18, 19 – 50, and > 50 [7]. It was kept at 3, keeping the user-item-context matrix to a dimension of 3 along the context axis.

Once the 3D tensors were generated, the train and test sets were created using five fold cross-validation. The missing values of the train set were first imputed using an imputation methodology before CP Decomposition was applied. Choosing different rank values (denoted as K in the Results), the resulting two matrices were then reconstructed to obtain the predicted values for each user-item combination. The predicted values were then compared to their corresponding values in the test set and the mean absolute error was calculated. The experiment is repeated for the different imputation methods (zeros, item average, user average) and for different rank values to identify the combination of variables that result in the optimal performance for ratings prediction. The rank values indicate the degree of information retained during the decomposition process. The results obtained from Experiment 1 are then finally compared to the results observed by Ghazanfar et al. [3], who performed a modified version of 2D SVD on the same dataset to predict ratings.

Experiment 2 Setup: the aim of this is to compare the effectiveness of using different contextual features on rating prediction using CP Decomposition. In this experiment, both datasets were used to compare the impact of using Age, Gender and Rating Avidness (MovieLens) and Age and Year of Publication (Book Crossing). Before experimenting with the Book Crossing dataset, a subset of the original Book Crossing data was created to overcome the scalability issues encountered when using the full dataset. The process of sub-setting the dataset is described in the *Datasets*.

Using the new subset of data for Book Crossing (or complete data for MovieLens), a user-item-context matrix was created similar to Experiment 1 by splitting the contextual factor into three different buckets of values. These groupings are shown in the Results section. With regards to the imputation method and rank values, only the item-average imputation method and a rank value of 15 are used respectively. This is based on the results obtained from Experiment 1, which depicted these methods as being the most optimal in achieving best MAE when predicting ratings. After the user-item-context matrices are generated, they are further split into a train and test set for five-fold cross validation. We applied our CP Decomposition on the train set in order to generate predictions and the results are compared to the actual ratings stored in the test set. The results are then recorded for further discussion and evaluation. Experiment 2 is then repeated, with either different Age groups or different contextual features.

Experiment 3 Setup: upon the analysis of results obtained from prior experiments (experiment 1 and 2), a different, our novel approach to applying tensor decomposition to the user-item-context tensor was tested. Specifically, the user-item-context was generated differently, with the aim of this experiment to identify whether this novel method was capable of delivering better MAE compared to the method used in the prior experiments. To generate the new user-item-context tensor, the following actions were performed: (1) the number

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

of users belonging to each context was made equal and (2) each layer of users was then stacked behind each other.

To illustrate the difference, if there were 900 users and 100 items and 3 contextual values. Method 1 (used in Experiment 1 and 2) would generate a user-item-context tensor that has 900 rows, 100 items and 3 layers. The new method used in this experiment will generate a user-item-context tensor that has 300 rows, 100 items and 3 layers. During the generation of the matrix, the imputation method (item averages) and context (Age) were kept consistent for the two datasets. Experiment 3 was then repeated 3 times, with a randomised order for each user within each layer to overcome bias. Results were recorded at the end of each activity for further discussion and evaluation.

Table 1: Results from Experiment 1 – Part 1

MovieLens Dataset							
Imputation Method	Context	K = 2	K = 5	K = 8	K = 10	K = 12	K = 15
Zeros	Age	2.712	2.525	2.513	2.406	2.393	2.382
Zeros	Gender	2.697	2.471	2.401	2.351	2.331	2.313
Item Average	Age	0.927	0.793	0.786	0.782	0.781	0.779
Item Average	Gender	0.803	0.791	0.787	0.78	0.776	0.763
User Average	Age	0.94	0.795	0.912	0.788	0.786	0.784
User Average	Gender	0.824	0.793	0.789	0.785	0.78	0.769

Table 2: Results from Experiment 1 – Part 2

MovieLens Dataset						Ghazanfar et al Results [3]		
Imputation Method	Context	K=8	K=10	K=12	K=15	Imputation Method	K	MAE
Zeros	Gender	2.401	2.351	2.331	2.313	Zeros	12	2.321
Item Average	Gender	0.787	0.78	0.776	0.763	Item Average	10	0.774
User Average	Gender	0.789	0.785	0.78	0.769	User Average	8	0.778

Evaluation Methodology: The evaluation methodology used in this paper consists of five-fold cross validation with a train test split ratio of 0.8 and 0.2 to each user's ratings. It must be highlighted that instead of the conventional removal of 20% of all users from the full dataset, 20% of a user's ratings are retained in a test set instead. This means that all users will be present in both train and test sets. This is required due to the nature of CP Decomposition, which requires the same dimension of users and items for prediction (you can not predict a user's ratings if they are not in the matrix at the point of decomposition).

Once the train set has been deconstructed using CP decomposition with the chosen rank value, the resulting matrices are then reconstructed to generate predictions for each user and each item. The Mean Absolute Error (MAE) is

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

calculated by averaging all corresponding values in the test set. This is then repeated using each fold during cross validation to calculate the average MAE.

4 Results

Experiment 1 Results: from Table 1, it can be seen that Gender slightly outperformed Age in mean absolute error across almost all iterations. Similarly, Item Averages outperforms Zeros and User Average imputation methods. There is a substantial difference between Zeros and the item/user average results. This is expected, as the zeros imputation method is understood across the literature to be the most basic method and used for benchmarking purposes only.

From Table 2, it can be observed that the application of our CP Decomposition algorithm is capable of achieving slightly better results than the 2D results reported by Ghazanfar et al. [3]. The use of Gender outperformed the benchmarked results, delivering an improvement of 0.1 in MAE across all imputation methods. Specifically, this result is observed at a rank value of 15 whereas 2D SVD reported optimal results at 8, 10, and 12 for the different imputation methods (zeros, item average, and user average respectively). It can be noted that using a higher rank value is more beneficial to 3D tensor factorization because there is more information to retain and take advantage of compared to 2D SVD approaches. It should be noted, however, that in Ghazanfar et al's research [3], it was reported that a large range of rank values was tested, and the reported rank values resulted in the best performance.

Table 3: Results from Experiment 2 – MovieLens dataset

MovieLens Dataset							
Context	Buckets	K = 2	K = 5	K = 8	K = 10	K = 12	K = 15
Age	0 - 18	0.927	0.793	0.786	0.782	0.781	0.779
	19 - 50						
	50						
Age	0 - 26	0.931	0.794	0.786	0.783	0.782	0.782
	27 - 38						
	38						
Gender	M / F	0.803	0.791	0.787	0.78	0.776	0.763
Avidness of Ratings	1 – 40	0.804	0.812	0.789	0.782	0.78	0.779
	41 – 120						
	120						

Experiment 2 Results: similar to Experiment 1 results, Table 3 shows that Gender continues to be the most effective contextual feature, followed by Age and Avidness. What is more interesting to observe across Table 3 and 4 is the performance between the two age buckets, with the context using Ages and a threshold of 18 and 50 outperforming the Age context that equally distributes the number of ratings across each bucket. The two extra contexts, Avidness and

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

Year of Publication perform relatively poorly compared to the other features, which is an indication that more fine-tuning is required when choosing contextual features. These results show that the choice of contextual feature is highly important when using tensor decomposition to predict ratings.

Experiment 3 Results: the results from Tables 5 and 6 show that at a rank value of 2, an improved MAE is observed from both datasets and converge to the initial method as the rank value increases. Consequently, our proposed method shows better performance at a lower rank value which may prove beneficial during implementation as lower rank values require less computational time and effort to run. At a high level, it can be concluded that our proposed method of tensor generation can potentially be used to predict ratings, however, there are a number of limitations to consider as discussed in section 5.

Table 4: Results from Experiment 2 – Book Crossing Dataset

Book Crossing Dataset						
Context	Buckets	K = 2	K = 5	K = 8	K = 10	K = 15
Age	0 - 18	1.521	1.326	1.320	1.319	1.315
	19 - 50					
	50					
Age	0 - 29	2.891	1.328	1.323	1.322	1.315
	30 - 37					
	38					
Year of Publication	1995	2.832	2.843	2.834	2.836	2.841
	1995 - 2000					
	2000					

5 Discussion

Experiment 1: Experiment 1 was performed by changing a number of different variables. These included two different contextual features (Age and Gender), three imputation methods (Zeros, Item Average and User Average), and a range of rank values (2, 5, 8, 10, 15). Experimentation using these iterations surfaced a number of interesting results.

Firstly, Gender slightly outperformed Age in mean absolute error across almost all iterations. By analyzing the distribution of ratings as a percentage of total ratings, a Male/Female split shows that Males provide a higher concentration of 3s and 4s compared to Females, who in turn tend to provide more extreme results in the form of 1s and 5s. Looking at the Rating Distribution by Age buckets (<18, $19-50$, >50), we see the younger audience giving the more extreme values (1s and 5s), the middle-aged audience giving more 3s, and the older audience giving more 4s. The above variations in ratings may contribute to the reason why Gender outperforms Age, however, a few caveats should be noted. The trends highlighted above are very slight and are not apparent when

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

Table 5: Results from Experiment 3 – MovieLens Dataset

Book Crossing Dataset						
Tensor Generation Method	Context (Age)	K = 2	K = 5	K = 8	K = 10	K = 15
Conventional Method	0 - 29	0.927	0.793	0.786	0.782	0.779
	30 - 37					
	37					
Stacking Method	0 - 29	0.901	0.802	0.792	0.788	0.786
	30 - 37					
	37					

Table 6: Results from Experiment 3 – Book Crossing Dataset

Book Crossing Dataset						
Tensor Generation Method	Context (Age)	K = 2	K = 5	K = 8	K = 10	K = 15
Conventional Method	0 - 29	3.521	1.323	1.321	1.319	1.31
	30 - 37					
	37					
Stacking Method	0 - 29	1.323	1.32	1.321	1.312	1.311
	30 - 37					
	37					

observing the rating distribution. The Gender buckets and Age buckets have a different number of raters within each group and are imbalanced. For example, the rating distribution shows that there are more males than females. Similarly, the younger age group has 54, the middle group has 784 and the older group has 105. It is also unclear how dominant the effects are of having a heavily skewed middle group. This relationship is further explored in Experiment 2. Since only one model (CP Decomposition) is in scope for this paper, it is unclear whether the model itself has a bias towards groups that skew towards a certain rating. For example, it is possible that the model has a bias towards ratings with higher concentrations of 3s and 4s as opposed to more extreme values. As this relationship is better surfaced by the Gender context, is it possible that the model thus naturally tends towards this relationship when predicting unknown ratings? This is another possible avenue for future work.

An important conclusion to draw from this, however, is that the choice of contextual feature is important when exploring the use of 3D tensor factorization. This is further explored in Experiment 2. Lastly, the optimal imputation method out of the three explored was item averages. This was consistent with the findings of Ghazanfar et al [3], who also found that item average performed the best when compared to user averages and zeros. Due to this observation, the item averages imputation method was adopted for all following experiments in this paper.

The second part of analysis for Experiment 1 results was the comparison of results to Ghazanfar et al's research [3], who looked into the optimal results of 2D SVD using different imputation methods. Analyzing the results from Table 2, it is clear that our 3D tensor factorization is capable of producing an overall im-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

proved result compared to 2D SVD. There are, however, practical implications of these results to be discussed. An improvement in MAE of 0.1 in the context of a $1-5$ rating can be arguable insignificant when building a recommendation system. As a result, it is identified that further experimentation in the form of different contextual features may be required to realize larger improvements for 3D tensor factorization in its current form to be beneficial from a practical perspective. During experimentation, it was also observed that increasing the rank value also increased computational time. When implementing such a solution in the real world, a trade-off between computational time and performance must be considered and could be overcome with more powerful machines and processing power. The results from Experiment 1 as a whole confirm that our 3D tensor factorization has the potential to incorporate more information to deliver results that are arguably more personalized whilst maintaining the performance in accuracy compared to 2D methods. With this confirmation, Experiment 2 was undertaken to critically evaluate the effectiveness of different contextual features.

Experiment 2: The aim of Experiment 2 was to critically evaluate the effectiveness of using different contextual features to predict ratings using CP decomposition. Different variations of experiments were performed on two datasets, MovieLens and Book Crossing. Specific to the MovieLens results, the results show that Gender continues to be the best contextual feature out of those tested, with a consistently lower MAE across most rank values compared to Age and Rating Avidness. However, it is the comparison between the different Age buckets that is more interesting, with the 18 and 50 buckets consistently outperforming the "equal distribution" buckets across both datasets. This raises an interesting hypothesis as to whether it is more effective to split buckets into an equal number of ratings per bucket or rather to split buckets based on their inherent relationships (such as users who tend to rate lower, the rate at extremes, or the rate at averages). Based on theoretical discussions and the results recorded from Experiment 2, it is suggested that the latter is more effective in providing better rating prediction performance. The use (and poor MAE performance) of Avidness and Year of Publication can also be used to support the previous argument and to conclude the results of Experiment 2. Both contextual features were split based on an equal distribution of ratings and showed poor performance compared to the other features. Additionally, it is clear upon analysis of all results in Experiment 2 that the choice of the contextual feature remains a highly important decision that will impact the overall performance of the MAE.

Experiment 3: Experiment 3 was performed in order to investigate an alternative method of creating the user-item-context matrix without requiring a second imputation step during tensor generation (and subsequent) factorization. In order to reduce the time taken to perform this experiment, item averages were used as the imputation method, similar to Experiment 2. To maintain robustness, different rank values were tested across the two available datasets. As mentioned in the Results section, the results show that at a rank value of 2, an improved MAE is observed from both datasets and converges to the initial method as the rank value increases. There are, however, theoretical and prac-

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

tical inconsistencies with this method, regardless of the positive performance observed. As described in the Methodology Section, the method stacks users behind each other, which creates an artificial relationship between users along the third (contextual) dimension. For example, if user 4 is stacked behind user 1, then tensor factorization will create a component matrix for users that will attempt to capture that information. Secondly, there is a practical limitation to applying this method. In order to stack users behind each other, the number of users belonging to each contextual value or layer must be the same. In the real world, it is highly unlikely that the numbers of users belonging to each contextual feature will be the same. Based on these observations, it can be seen that there are a number of limitations to the alternate method of generating user-item-context tensors. If these limitations can be overcome, there is potential for this method to be further revised and improved based on the initial results seen from Experiment 3. These limitations are further described in the next section under limitations and future works.

Limitations and future works: The scalability of the CP Decomposition algorithm was found to be limited in its ability to process large datasets. The complete Book Crossing dataset was too large for the algorithm to handle and therefore, a subset of the dataset was used in this paper. Improvements to the algorithm or the use of more computational power may have influenced the results of the Book Crossing dataset and may have allowed for different observations.

Exploring the use of implicit feedback to improve rating prediction would be a very big area of future work for this paper. In the Book Crossing dataset, implicit feedback (whether a user consumed the item but did not rate it) was given. An interesting avenue for exploration would be to use implicit feedback as input for improved recommendations. Exploring this area as a future option would be extremely beneficial as it is vastly more likely in the real world to capture implicit feedback rather than explicit feedback. By developing an algorithm that can incorporate implicit feedback, the number of scenarios where this algorithm could be applied and the amount of training data would increase dramatically.

Finally, evaluation of the paper's experiments without using Mean Absolute Error should also be incorporated in future work. In this paper, due to the number of uncertainties with the new technique, the methodology, evaluation methods, and datasets, it was difficult to incorporate a different metric of evaluation. MAE is highly utilized in the academic literature and therefore well documented, but is not a practical method for evaluating recommendation systems that are implemented in real work. A very strong improvement to this paper would be to explore the qualitative performance of 3D tensor factorization compared to conventional 2D conventional methods.

References

1. Ge, R., Huang, F., Jin, C., Yuan, Y.: Escaping from saddle points—online stochastic gradient for tensor decomposition. In: Learning Theory. pp. 797–842 (2015)
2. Ghadimi, S., Lan, G.: Accelerated gradient methods for nonconvex nonlinear and stochastic programming. Mathematical Programming **156**(1-2), 59–99 (2016)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20

3. Ghazanfar, M.A., Prügel-Bennett, A.: The advantage of careful imputation sources in sparse data-environment of recommender systems: Generating improved svd-based recommendations. Informatica (Slovenia) **37**, 61–92 (2013)
4. Gomez-Uribe, C.A., Hunt, N.: The netflix recommender system: Algorithms, business value, and innovation. ACM Transactions on Management Information Systems (TMIS) **6**(4), 1–19 (2015)
5. Guan, N., Tao, D., Luo, Z., Yuan, B.: Nenmf: An optimal gradient method for nonnegative matrix factorization. Trans. Signal Processing **60**(6), 2882–2898 (2012)
6. Isinkaye, F.O., Folajimi, Y.O., Ojokoh, B.A.: Recommendation systems: Principles, methods and evaluation. Egyptian informatics journal **16**(3), 261–273 (2015)
7. Karatzoglou, A., Amatriain, X., Baltrunas, L., Oliver, N.: Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In: Proc. of the 4th ACM conf. on Recommender systems. pp. 79–86 (2010)
8. Kolda, T.G., Sun, J.: Scalable tensor decompositions for multi-aspect data mining. In: IEEE int. conf. on data mining. pp. 363–372. IEEE (2008)
9. Kumar, P., Thakur, R.S.: Recommendation system techniques and related issues: a survey. International Journal of Information Technology **10**(4), 495–501 (2018)
10. Lebedev, V., Ganin, Y., Rakhuba, M., Oseledets, I., Lempitsky, V.: Speeding-up convolutional neural networks using fine-tuned cp-decomposition. arXiv:1412.6553 (2014)
11. Nesterov, Y.: Introductory lectures on convex optimization: A basic course, vol. 87. Springer Science & Business Media (2013)
12. Nitanda, A.: Stochastic proximal gradient descent with acceleration techniques. In: Advances in Neural Information Processing Systems. pp. 1574–1582 (2014)
13. Paterek, A.: Improving regularized singular value decomposition for collaborative filtering. In: Proceedings of KDD cup and workshop. vol. 2007, pp. 5–8 (2007)
14. Pazzani, M.J., Billsus, D.: Content-based recommendation systems. In: The adaptive web, pp. 325–341. Springer (2007)
15. Rendle, S.: Factorization machines. In: 2010 IEEE International conference on data mining. pp. 995–1000. IEEE (2010)
16. Salakhutdinov, R., Mnih, A., Hinton, G.: Restricted boltzmann machines for collaborative filtering. In: Proc. of int. conf. on Machine learning. pp. 791–798 (2007)
17. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Item-based collaborative filtering recommendation algorithms. In: Proceedings of the 10th international conference on World Wide Web. pp. 285–295 (2001)
18. Schafer, J.B., Frankowski, D., Herlocker, J., Sen, S.: Collaborative filtering recommender systems. In: The adaptive web, pp. 291–324. Springer (2007)
19. Sun, J., Tao, D., Papadimitriou, S., Yu, P.S., Faloutsos, C.: Incremental tensor analysis: Theory and applications. ACM Transactions on Knowledge Discovery from Data (TKDD) **2**(3), 11 (2008)
20. Tang, X., Bi, X., Qu, A.: Individualized multilayer tensor learning with an application in imaging analysis. J. of American Stat. Assoc. **115**(530), 836–851 (2020)
21. Yang, L., Fang, J., Li, H., Zeng, B.: An iterative reweighted method for tucker decomposition of incomplete tensors. Signal Processing **64**(18), 4817–4829 (2016)
22. Zhou, X., He, J., Huang, G., Zhang, Y.: Svd-based incremental approaches for recommender systems. J. of Computer and Sys. Sciences **81**(4), 717–733 (2015)

ICCS Camera Ready Version 2022
To cite this paper please use the final published version:
DOI: 10.1007/978-3-031-08751-6_20