Collaborative Filtering Recommendation Method and SVD-based Incremental Approach

Winnie Nguyen
Computer Science Department at Earlham College
Richmond, Indiana, USA
zdnguyen18@earlham.edu

ABSTRACT

In recent years, the need for recommender systems as a tool to improve user interaction, provide personalized services, and enrich shopping potential to raise revenue has been increasing along with the sharp expansion of online shopping platforms. However, with the overload of huge amounts of customer data, recommender systems face challenges in processing data robustly and accurately. In this paper, I am using a matrix factorization technique called singular value decomposition (SVD) in item-based collaborative filtering. Despite the ability of quickly producing high quality recommendations, SVD has expensive matrix factorization steps - scalability matters that needs an updated algorithm solution. The paper will propose an Incremental SVD approach, expecting to make an improvement in prediction accuracy and running time as.

ACM Reference Format:
Winnie Nguyen. 2021. Collaborative Filtering Recommendation Method and SVD-based Incremental Approach. In *Proceedings of ACM Conference (Conference'17).* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

In the technology 4.0 era, especially after the COVID-19 pandemic reshaped our world, the growth of eCommerce has set a high bar as more people shift away from shopping in retail stores to doing online shopping [14]. According to eMarketer in the US Ecommerce Forecast 2021, US consumers are expected to spend $933.30 billion on eCommerce this year, up 17.9% year-over-year, and equalling 15.3% of total retail sales [7]. Along with the explosion of the eCommerce industry, a variety of recommendation methods have been proposed to help consumers discover the products to buy [6] and allow companies to maximize their return on investment (ROI) based on information gathered from customers through their experiences, behaviors, preferences, and interests [10]. Besides providing a better user experience and boosting sales, a recommendation system is a tool to help eCommerce companies enhance customer engagement and increase traffic to their website [18].

Researchers have developed various recommender systems with the aim of helping eCommerce companies utilize the large amount of customer data and ease the decision-making process for users. The three most common recommendation methods applied in Recommendation Systems are collaborative filtering, content-based filtering, and hybrid approach. The collaborative filtering approach uses the concept that "a set of users possessing similar features" will have similar interests to make recommendations. Meanwhile, content-based analysis makes a recommendation depending on the similarity measurement between item-feature and target-feature, rather than on the user's opinions [4]. Hybrids approach combines two or more techniques to maximize the benefits while covering the weakness of all chosen methods.

Among top eCommerce companies, Amazon and Netflix, use different recommender engines to help users navigate through the large product assortments, make decisions and overcome information overload. Netflix's successful content-based algorithm, Netflix Recommendation Engine (NRE), influences 80% of content watched on Netflix and helps the company save one billion dollars yearly in value from customer retention [15]. Meanwhile, Amazon, the world's leading online retailer, launched their item-based collaborative filtering in 1998, applied for millions of customers along with millions of products [19]. Amazon's revenue witnessed a 21.11% year-over-year growth, reaching more than $250B by the end of 2019 thanks to item-to-item collaborative filtering recommendation engines [8]. For two decades, Amazon's growth has served confirmation for the advantage of their collaborative filtering recommendation systems which are simplicity, scalability, explainability, adaptability, and relatively high-quality algorithms [19].

This proposal presents eCommerce collaborative filtering recommendation systems using matrix factorization algorithms called Singular Value Decomposition (SVD). Collaborative filtering methods face two challenges, especially with the tremendous growth of users and products. The first challenge is to improve the quality of recommendations to show customers their personalized preferences. The second challenge is that we also want to upgrade the scalability of the algorithms to handle millions of potential neighbors in real-time. Those two challenges are in conflicts in some ways as the less time an algorithm uses to search for neighbors, the more scalable it will be, but the worse its quality will be [17]. Therefore, besides the SVD-based approach, the paper will implement incremental SVD model-building with the aim of enhancing the scalability while providing faster and better predictive accuracy.

This paper starts by introducing collaborative filtering recommender systems and the challenges in applying this method. Then the paper focuses on related work about both SVD and incremental SVD algorithms. Following this is the design of proposed systems,

Conference'17, July 2017, Washington, DC, USA

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM…$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

explained in detail. This section not only elaborates on the frameworks but also covers each component of the SVD-based recommender system algorithms. Then, I discuss the verification method of the systems, test plans, and major risks, followed by the timeline section.

2 BACKGROUND & RELATED WORK

This section explains in detail the collaborative filtering method and the challenges we are facing. Then I focus on SVD algorithms including related researched about the incremental SVD-based algorithm. The datasets used in related pieces of research will also be introduced. The section focuses on details of each technique, the advantages, and the theoretical nature of the research. Additionally, the analysis and comparison among those updated techniques used in each algorithm are presented.

2.1 Collaborative Filtering

A collaborative filtering model is built by collecting users' interactions on different items, then creating embeddings for every user and item [16]. It makes a recommendation to a particular user based on the reactions of other users who share similar taste. Users' interactions have two main types: explicit and implicit. Explicit interactions are the input of users regarding to their interest in an item. It is often measured by ratings or ranking provided by each user using one or more ordinal or qualitative scales [11]. Meanwhile, implicit interactions are information produced after observing users' behavior. There is no requirement for users to participate or gather this data by themselves as the system will automatically track users' preferences by monitoring the performed actions including which items they visited, clicked, or bought [1]. For collaborative methods and the algorithms used in this paper, the explicit feedback can reflect users' preferences more accurately than implicit ones, as there is less noise in the data. In datasets used in related pieces of research and this papers, explicit users' preference information is shown as users' rating column.

Traditional collaborative filtering algorithms include memory-based (User-User or Item-Item-filtering) and model-based methods. Among model-based methods, the most popular model applied to collaborative filtering is matrix factorization, which provides a decomposition of a rating matrix into two matrices representing users and items in a latent factor space [12]. The algorithm is used to predict the expected rating for an user who hasn't rated or buy the item yet. To make a rating prediction for an item, we look at the previous item's rating given by users who share the similar taste to the given user. In detail, we multiply two matrices, items' and users' entities, to predict the relationship between them - how the given users would rate the items [5].

However, with the massive growth of customers and products data, the matrix factorization approach to collaborative filtering still faces three main problems to making accurate recommendations [2, 17]:

- `Responding time`: the neccessity of improving algorithms responding time applied in a huge amount of data
- `Sparsity`: the missing values in users-items matrix because many users will only have rated a small subset of the overall database. Thus, even the most popular items have very few ratings, and even users that are very active rate just a few items compared to the total number
- `Scalability`: the collaborative filtering fails to scale up the computing time with the massive growth of both number of new users and items when making accurate recommendations

2.2 Singular Value Decomposition (SVD)

Zhou et al. [21] states that applying data dimension reduction methods, specifically SVD, is one of the popular solutions for the sparsity problems. SVD is a matrix factorization algorithm that can extract characteristics of the dataset's features by splitting the original user-item ratings matrix into three smaller matrix multiplications. Given a *mx* matrix A (N is the number of items, M is the number of users) with rank(A) = r, the $SVD(A)$ is defined as:

$$SVD(A) = U \times S \times V^T \tag{1}$$

where U, S, V are dimensions $m \times r$, $r \times r$, $r \times n$. While the middle matrix S is a diagonal matrix with r nonzero entries, which are the singular values of A, matrices U and V are orthogonal [21]. U and V are known as the $left$ and $right$ singular vectors, respectively with the first r columns of U corresponding to the nonzero singular values span the *columnspace*, and the first r columns of V span the $rowspace$ of the matrix A [17].

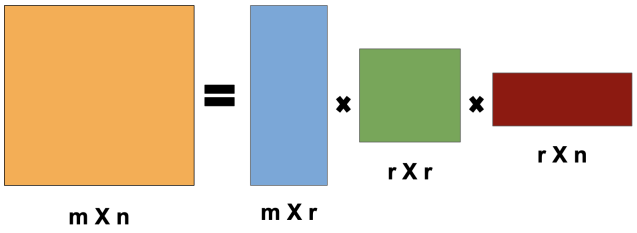

Figure 1: SVD Matrix Factorization

The accuracy of the SVD recommender system is evaluated through two popular measures: Root Mean Square Error (RMSE) and Mean Absolute Error (MAE). The lower value of both metrics, the better performance of the recommendation algorithms. SVD decreases the dimension of the utility matrix and be the best low-rank linear approximation of the original matrix A using three matrices multiplication [17].

According to Sarwar et al. [17] dimensionality reduction is a popular approach in SVD to help customers who rated the similar products (not exactly the same products) can be mapped into the space spanned by the same eigenvectors. It is possible to keep the first k largest singular values in diagonal matrix S only $k \ll r$ and the remaining smaller ones set to zero by discard other entries . The reduced matrix is denoted as S_k. Simultaneously, by deleting the corresponding $(r-k)$ columns from matrix U and $(r-k)$ rows from matrix V, we produce two reduced matrices U_k and V_k . The reconstructed SVD is represented as:

$$SVD(A_k) = U_k \times S_k \times V_k^T \tag{2}$$

which is the rank-k matrix closet approximation to the original matrix A. However, in my paper, I apply the new incremental SVD

technique in the original matrix A instead of the reduced version A_k as in Sarwar's research.

2.3 Related Work - Incremental SVD-based algorithm

The time complexity of the SVD algorithm is calculated by batch and equals $O(m^2n + n^3)$ (where m,n are, respectively, row size and column size of the matrix) [21]. As SVD requires all the data be processed simultaneously, it has a challenge in not only dealing with a large dataset but also the expensive computing time when a new user or item is added to the system. Due to the limitation in running time with massive data of SVD-based algorithm, we need to experiment with an incremental model-building technique for SVD to improve the scalability of recommender system.

Zhou et al. [21] proposed an incremental algorithm called Incremental ApproSVD - the combination of Incremental SVD and Approximating SVD algorithm - to improve running time and accuracy of predicting new items entered dynamically. Compared to other clustering or data dimensionality reduction methods which solve the massive amount of data with quick response time and the sparsity problem as offline computation, Incremental ApproSVD can handle online and dynamic issues more efficiently. For the dataset, they used MovieLens and Flixster. The essential technique of Incremental ApproSVD is choosing column sampling probabilities, specifically adopting column sampling to reduce the column number then the size of the original matrix. The evaluation showed that the prediction model could predict unknown ratings when new items enter dynamically with the lower value of both RMSE and MAE, plus be a suboptimal approximation with less running time. Moreover, the paper provided an updated mathematical error analysis between the actual ratings and the predicted ones generated by the Incremental ApproSVD algorithm.

Sarwar et al. [17] address this problem by designing an incremental algorithm known as folding-in SVD literature, which allows new users and items to be added without affecting the existing ones. The model will use Latent Semantic Indexing (LSI) to reduce the dimensionality before applying the incremental technique folding in. Therefore, when a new user is incrementally folded-in the space, the user-item matrix is already reduced the size. As SVD decomposition using existing users and items is pre-computed, folding-in technique will take advantages to create a more scalable recommender system. Applied to the MovieLens dataset, the result shows that incremental algorithm speeds up computational time while provide comparable prediction accuracy.

Brand [3] applies an incremental SVD technique in incomplete data with the aim of solving the uncertain new data with missing values and/or contaminated with correlated noise. When having missing or untrusted adding rows and/or columns of data in the system, the incremental technique produces factoring of lower rank and residual than the highly optimized batch algorithm - Matlab's SVD algorithm for example. Brand shows the applications in computer vision and audio feature extraction and gets the results with better time and space complexity.

In this paper, to handle the scalability problem and speeding up recommendation formulation, the new applied incremental SVD technique closely follows both Zhou et al.'s work and Sarwar et al.'s work. However, instead of using incremental technique after applying approximating SVD algorithm called ApproSVD as Zhou's research or after reducing the dimensonality of users-items matrix as Sarwan's research, the incremental SVD is applied directly to the standard SVD algorithm in the execution steps, after the building the model. Despite both Zhou and Sarwan's methods improved the performance of the model, they reduce the quality of recommendation quality in some ways. ApproSVD may have slower computing time, specifically, time needed to build the model. It is because the ApproSVD algorithm takes the SVD of a $k(\varepsilon^2 \times k)\varepsilon^2$ matrix, which requires computation that is cubic in k [20]. Meanwhile, the drawback of Sarwan's reduction approach is reducing the accuracy due to the loss of information [13].

2.4 Datasets

This sub-section introduces the datasets used in collaborative filtering recommender systems. While different papers have different ways of approaching the problem and make different assumptions, they have the same objective: to enhance the performance of recommender systems in terms of accuracy or running time.

MovieLens [8] is the most common dataset among the research papers examined for this work. The MovieLens dataset is provided by GroupLens, a web-based research recommender system with over 20 million movie ratings and tagging activities, released 4/2015; updated 10/2016. All users in this dataset rated at least 20 movies which proved their active interactions with items. The rating scale ranges from 1 to 5, where 1 represents dislike, and 5 illustrates a strong preference. No demographic data is included; movies and users are represented by id numbers.

Other datasets are also used. Zhou et al. [21] used the Flixster dataset, containing more than 8M ratings from 786,936 users for 48.794 movies in Flixster. Brand [3] factored and eigen-coded a 664932 x 31 matrix containing a 31-band constant-Q spectrogram of roughly 2 hours of audio for a music classification study. Brand also applies the incremental SVD in face recognition to track point on a face in 150 frames of video from 2D motions into 3D reconstructions.

3 DESIGN & IMPLEMENTATION

As illustrated in Figure 1, in recommender system, the SVD algorithm works in two independent steps: offline process and online process.

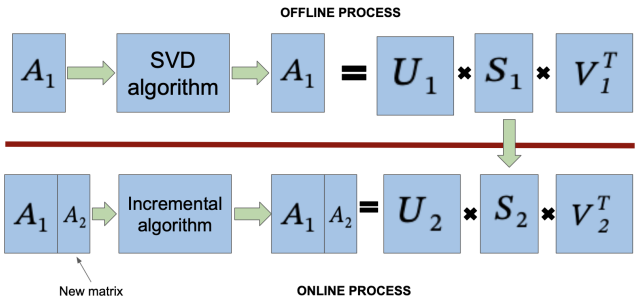

Figure 2: Flow chart of offline and online process of SVD

In the offline process, the model is built by computing the user-user similarity or item-item similarity. Offline computation is always expensive and time-consuming, however, it computes less frequently than the online process. Meanwhile, online process executes when the model starts making prediction when new user or item is added.

To handle the expensive computation of SVD offline stage, we use the incremental algorithm to ensure the highly scalable overall performance with new adding user or item into the user-item matrix. In Figure 2, after performing SVD algorithm on A_1 in the offline process with three matrices U_1, S_1, and V_1, the online process applies the incremental algorithm whenever we have new matrix A_2, producing three updated matrices U_2, S_2 and V_2

3.1 Incremental SVD algorithm

The incremental SVD algorithm computes the SVD of standard matrix A_1 by adding one column or fully observed matrix A_2 at a time. The size of U_1, V_1, and the diagonal matrix of singular values S_1 grows as A_2 is added. The process of SVD-updating using incremental algorithm happens when new user or item incorporates into an existing recommender system matrix using three factor matrices. The previous singluar values and singular vector of the original user-item matrix A_1 is used as an alternative to recompute the SVD of the updated matrix when adding matrix A_2. The update procedure of the Incremental SVD algorithm is shown in Algorithm 1.

Algorithm 1 Incremental SVD

Input: matrices $A_t \in \mathbb{R}^{m \times n}$ with null matrix U_t, S_t, V_t;
Start: Set t := 0
Repeat:

1: Given new column vector v_t
2: Define $w_t := argmin_w ||U_t w - v_t||_2^2 = U_t^T v_t$
3: Define $p_t := U_t w_t; r_t := v_t - p_t$; Set $r_t := v_0$ when $t = 0$
4: When adding v_t to standard matrix A_t, we have

$$\begin{bmatrix} U_t S_t V_t^T & v_t \end{bmatrix} = \begin{bmatrix} U_t & \frac{r_t}{||r_t||} \end{bmatrix} \begin{bmatrix} S_t & w_t \\ 0 & ||r_t|| \end{bmatrix} \begin{bmatrix} V_t & 0 \\ 0 & 1 \end{bmatrix}^T \quad (3)$$

5: Compute the SVD of update matrix:

$$\begin{bmatrix} S_t & w_t \\ 0 & ||r_t|| \end{bmatrix} = \hat{U}\hat{S}\hat{V}^T \quad (4)$$

6: Set $U_{t+1} := \begin{bmatrix} U_t & \frac{r_t}{||r_t||} \end{bmatrix} \hat{U}$; $S_{t+1} := \hat{S}$; $V_{t+1} := \begin{bmatrix} V_t & 0 \\ 0 & 1 \end{bmatrix}^T \hat{V}^T$ when $t := t + 1$

Until: termination

4 EXPERIMENTAL EVALUATIONS

This section describes the experimental platform and verification of the incremental SVD algorithm. First, I propose the dataset, evaluation metrics, and the computational environment. Then I explain the experimental procedure plan, following by the timeline for the Capstone Project in CS488.

4.1 Datasets

I use Amazon Beauty product provided by Amazon Product Data to conduct the experiment [9]. The data is collected from May 1996 and the latest version is updated in 2018. This Amazon beauty product dataset related to over 2 Million customer reviews and ratings of Beauty related products sold on the website with 4 columns including: the unique UserId (Customer Identification), the product ASIN (Amazon's unique product identification code for each product), ratings (ranging from 1-5 based on customer satisfaction) and the timestamp of the rating (in UNIX time). I only consider the users that had rated 10 or more products then convert the dataset into user-item matrix where rows represent users and columns represent items. For the experiment, I randomly divide dataset into a training (80% data) and test portion (20 % data).

4.2 Evaluation Metrics

I evaluate the performance of the proposed recommendation algorithms according to response time and accuracy metrics:

4.2.1 Response Time: Time required by the algorithm to find out the items to recommend

4.2.2 Accuracy: The fraction number of items an algorithm recommends, to the number of items that are recommended by an algorithm that takes into consideration the whole dataset available. I use a popular statistical accuracy measurement named Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE) to show the closeness of predicted rating to the true ratings. The smaller values of both metrics are, the higher accuracy of the recommender system. If r_{ui} represents the true rating on item i by user u, and $\hat{r}_{ui}$ shows the predicted rating on item i by user u, RSME and MAE of N correspnding rating prediction pairs are define:

$$RSME = \sqrt{\frac{\Sigma_{i=1}^{N}(r_{ui} - \hat{r}_{ui})^2}{N}} \quad (5)$$

$$MAE = \frac{\Sigma_{i=1}^{N}|r_{ui} - \hat{r}_{ui}|}{N} \quad (6)$$

4.3 Environment

All codes will be written in Python and runned in Google Colab Pro or Slumr scheduler.

4.4 Experimental Procedure

For the experiment, I compute the standard SVD model for all users, then apply the incremental technique to compute the SVD model with additonal users into the user-item matrix. From that, I can make an evaluation of the performance implications of incremental technqiue. Moreover, I use 10-fold cross validation by selecting random training and testing data for all my experiments.

4.5 Timeline

This section show the detail timeline in Spring semester 2022 (15 weeks schedule) to propose the Capstone Project from theory to practice, expecting to complete the paper, applied software, and poster.

Table 1: Timeline & Plan for CS488 Spring '22

Non-English or Math	Frequency	Comments
Jan 31, 2022	1	Review the work in CS388
Feb 7, 2022	2	
Feb 14, 2022	3	First draft of paper due
Feb 21, 2022	4	Test the idea of having software
Feb 28, 2022	5	Second draft of paper due
Mar 7, 2022	6	Second draft of software
Mar 14, 2022	7	Third draft of paper due
Mar 21, 2022	8	Finalize paper and software
Mar 28, 2022	9	Finalize paper and software
Apr 4, 2022	10	Make poster and presentation
Apr 11, 2022	11	Finalize paper, software, and poster
Apr 18, 2022	12	Final due paper, software, and poster
Apr 25, 2022	13	
May 3, 2022	14	
May 9, 2022	15	

REFERENCES

[1] Zahra Ahmad. 2021. Recommender Systems: Explicit Feedback, Implicit Feedback and Hybrid Feedback. Retrieved Sep 20, 2021 from https://medium.com/analytics-vidhya/recommender-systems-explicit-feedback-implicit-feedback-and-hybrid-feedback-ddd1b2cdb3b

[2] Mazhar Javed Awan, Rafia Asad Khan, Haitham Nobanee, Awais Yasin, Syed Muhammad Anwar, Usman Naseem, and Vishwa Pratap Singh. 2021. A Recommendation Engine for Predicting Movie Ratings Using a Big Data Approach. *Electronics* 10, 10 (2021), 1215.

[3] Matthew Brand. 2002. Incremental singular value decomposition of uncertain data with missing values. In *European Conference on Computer Vision*. Springer, 707–720.

[4] Chentung Chen and Weishen Tai. 2004. A User Preference Classification Method in Information Recommendation System.. In *ICEB*. 1091–1096.

[5] Denise Chen. 2020. Recommender System — Matrix Factorization. Retrieved Sep 14, 2021 from https://towardsdatascience.com/recommendation-system-matrix-factorization-d61978660b4b

[6] Mohammad Daoud, SK Naqvi, and Asad Ahmad. 2014. Opinion Observer: Recommendation System on ECommerce Website. *International Journal of Computer Applications* 105, 14 (2014), 37–42.

[7] Suzy Davidkhanian. 2021. US Ecommerce Forecast 2021. Retrieved Sep 14, 2021 from https://www.emarketer.com/content/us-ecommerce-forecast-2021

[8] GroupLens Kaggle. 2018. MovieLens 20M Dataset. Retrieved Sep 2, 2021 from https://grouplens.org/datasets/movielens/20m/

[9] Julian McAuley. 2018. Amazon product data. Retrieved Sep 21, 2021 from http://jmcauley.ucsd.edu/data/amazon/

[10] Julian McAuley. 2019. A simple way to explain the Recommendation Engine in AI. Retrieved Sep 21, 2021 from https://medium.com/voice-tech-podcast/a-simple-way-to-explain-the-recommendation-engine-in-ai-d1a609f59d97

[11] Douglas W Oard, Jinmook Kim, et al. 1998. Implicit feedback for recommender systems. In *Proceedings of the AAAI workshop on recommender systems*, Vol. 83. WoUongong, 81–83.

[12] Fernando Ortega, Antonio Hernando, Jesus Bobadilla, and Jeon Hyung Kang. 2016. Recommending items to group of users using matrix factorization based collaborative filtering. *Information Sciences* 345 (2016), 313–324.

[13] Manos Papagelis, Ioannis Rousidis, Dimitris Plexousakis, and Elias Theoharopoulos. 2005. Incremental collaborative filtering for highly-scalable recommendation algorithms. In *International Symposium on Methodologies for Intelligent Systems*. Springer, 553–561.

[14] Sarah Perez. 2020. COVID-19 pandemic accelerated shift to e-commerce by 5 years, new report says. Retrieved Sep 14, 2021 from https://techcrunch.com/2020/08/24/covid-19-pandemic-accelerated-shift-to-e-commerce-by-5-years-new-report-says/

[15] Kaja Polachowska. 2019. Is It Worth It? ROI of Recommender Systems. Retrieved Sep 2, 2021 from https://dzone.com/articles/is-it-worth-it-roi-of-recommender-systems

[16] Abhijit Roy. 2020. Introduction To Recommender Systems- 1: Content-Based Filtering And Collaborative Filtering. Retrieved Sep 14, 2021 from https://towardsdatascience.com/introduction-to-recommender-systems-1-971bd274f421

[17] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2002. Incremental singular value decomposition algorithms for highly scalable recommender systems. In *Fifth international conference on computer and information science*, Vol. 1. Citeseer, 27–8.

[18] Rodrigo Schiavini. 2021. What is an E-commerce Recommendation Engine? Retrieved Sep 14, 2021 from https://www.smarthint.co/en/5-reasons-why-ecommerce-store-needs-recommendation-engine/

[19] Brent Smith and Greg Linden. 2017. Two decades of recommender systems at Amazon. com. *Ieee internet computing* 21, 3 (2017), 12–18.

[20] Xun Zhou, Jing He, Guangyan Huang, and Yanchun Zhang. 2012. A personalized recommendation algorithm based on approximating the singular value decomposition (ApproSVD). In *2012 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology*, Vol. 2. IEEE, 458–464.

[21] Xun Zhou, Jing He, Guangyan Huang, and Yanchun Zhang. 2015. SVD-based incremental approaches for recommender systems. *J. Comput. System Sci.* 81, 4 (2015), 717–733.