The Netflix Problem

Ellen Crombie

1 The rise of Netflix

Reed Hastings and Marc Randolph founded Netflix in 1997, California [1]. Initially customers across America would pay a monthly subscription to rent DVDs and movies, mailed by post. Twenty years later in 2007, Netflix snowballed into the Netflix we know today: an over-the-top video streaming service. At present, there are 220.67 million subscribers globally who pay the monthly fee to gain access to Netflix's catalogue of titles (films and TV series) [2].

2 Why is there a problem?

Today's users want and expect a vast range of choice. Somewhat paradoxically, they don't want to spend time making the correct choice. To continue successfully maintaining and increasing subscription numbers, Netflix must provide long term customer '*joy*'. It is hard to quantify all the factors that contribute to long term user retention. However, there is data to show that 80% of what Netflix users watch comes from the recommendation system [3]. The system recommends content to the user they may like based on the films they have previously rated. In fact, Netflix now employs personalisation in all aspects of the Netflix experience to increase user retention. Two users will most likely see different content thumbnails, home screen layouts and search results.

This report will introduce the Netflix Prize in Section 3, the competition which sparked a revolutionary drive in investment and research into the vast number of algorithms that are combined to form recommender systems. Following this, an in-depth investigation into singular value decomposition (SVD) and its applications in the traditional collaborative filtering approaches which underpinned the original recommender systems takes place in Sections 4 and 5. This includes a mathematical explanation of why SVD produces a low rank approximation of a matrix, and then a proof of the Eckart-Young-Mirsky Theorem.

3 The Netflix Prize

In 2006 Netflix launched a \$1,000,000 open competition to find the best algorithm to recommend content to the user. Contenders were given a training data set of 100,480,507 ratings from 480,189 users for 17,770 moves, while qualifying and quiz sets of data were used for model testing. The prize would be awarded if the algorithm could improve on Netflix's current algorithm, *Cinematch*, reducing the root mean squared error score (RMSE) by a further 10% when evaluated on the test and quiz data sets [4].

The prize was awarded in September 2009 to BellKor's Pragmatic Chaos, a merger of 3 teams of researchers. The solution was predominantly underpinned by collaborative filtering, an approach which requires no information about the users (Netflix consumers) or the items (Netflix titles), instead it recommends items based on users' historical ratings of the items [5]. Within collaborative filtering, algorithms are often categorized as memory-based approaches, for example neighbourhood models, or model-based approaches which are also known as latent-factor methods [1]. Today, Netflix utilises a hybrid recommender system which encompasses both collaborative filtering and content-based approaches, where the latter uses item features to recommend each item: for example, by evaluating the effect of Bill Nighy featuring in a film [6]. This is implemented by deep learning techniques.

3.1 Blending a solution

The structure of most successful competition entries predicted the rating a user would give to an item using a baseline predictor and a term for user-item interaction. This baseline predictor incorporated factors such as the qualities of the user (a critical or generous rater); the overall rating of the title across all users; and the time dependent bias of the title (since popularity changes over time). The user-item interaction term was formed using a blend of collaborative filtering approaches [8].

BellKor's Pragmatic Chaos used a combination of 18 predictors in the 2009 solution. When minimizing the RMSE of this overall model on each data set, the team did not solely have to minimise the RMSE of each individual collaborative filtering predictor. In fact, they had to minimise the RMSE of the blend ('ensemble') of predictors used for optimization. Thus, each predictor's contribution would ideally strike a good balance between having a low RMSE and being uncorrelated with the rest of the ensemble [9]. This ensemble model was trained using machine learning techniques such as stochastic gradient descent. To add another dimension of difficulty to the Netflix Prize, there are also a multitude of possible blending techniques to choose from: these originally focused on regression methods, however methods involving neural networks have now become the norm [5].

4 Singular Value Decomposition

Singular value decomposition (SVD) is a matrix factorisation technique that is fundamental to collaborative filtering algorithms. By definition, for any matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$,

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$$

is a singular value decomposition of $\mathbf{A}$ if $\mathbf{U} \in \mathbb{R}^{m\times m}$ and $\mathbf{V} \in \mathbb{R}^{n\times n}$ are orthogonal, and $\mathbf{\Sigma} \in \mathbb{R}^{m\times n}$ is diagonal with non-negative diagonal entries $\sigma_1, \sigma_2, \ldots, \sigma_p \geq 0$, where $p = \min(m, n)$.[10] Here, the values $\sigma_1, \sigma_2, \ldots, \sigma_p$ are the singular values of $\mathbf{A}$ and are uniquely determined up to permutation, with squares equal to the eigenvalues of the matrix $\mathbf{A}^T\mathbf{A}$ (proof detailed Poole, 2002 [10]). In contrast, the left and right singular vectors (the columns of $\mathbf{U}$, and the columns of $\mathbf{V}$) are not uniquely determined.

4.1 Low rank approximation of a matrix

Singular value decomposition is the best known decomposition which gives a low rank approximation of a matrix. In this case, the rank of a matrix can be described most conveniently as the number of nonzero singular values of $\mathbf{A}$, written rank $r = \min(n, m)$ [11]. In particular, the rank of $\mathbf{\Sigma}$ is equal to the number of non-zero diagonal entries. Therefore any column with a zero diagonal entry will be equal to the zero vector which is equivalent to writing that for $\mathbf{A} \in \mathbb{R}^{m\times n}$, $\text{rank}(\mathbf{A}) = n$ if and only if $\sigma_1, \ldots, \sigma_n > 0$[10].

To show why SVD is able to make a low rank approximation of a matrix, it is first noted that matrix $\mathbf{A}$ is of full rank if and only if it does not have a zero singular value. This allows the SVD to be denoted

$$\mathbf{A} = \sum_{i=1}^{r} \sigma_{\mathbf{i}} \mathbf{u}_i \mathbf{v}_i^T, \tag{1}$$

where $\mathbf{u}_i$ are the columns of $\mathbf{U}_{m\times r}$ and $\mathbf{v}_i$ are the columns of $\mathbf{V}_{n\times r}$. Similarly, when matrix $\mathbf{A}$ is not full rank, the Reduced SVD $\mathbf{A}_k$ with $k < r$ can be written:

$$\mathbf{A}_k = \mathbf{U}_{m\times k}\left(\mathbf{\Sigma_k}\right)_{k\times k}\mathbf{V}_{k\times n}^T = \sum_{i=1}^{k} \sigma_{\mathbf{i}} \mathbf{u}_i \mathbf{v}_i^T, \tag{2}$$

where the $r-k$ trailing singular values of $\mathbf{A}$ have been zeroed. At this point it is clear that $\mathbf{A}_k$ is the projection of $\mathbf{A}$ onto the space spanned by the top k singular vectors of $\mathbf{A}$. To show this, it is noted that $\mathbf{\Sigma}_k = \text{diag}(\sigma_1, \dots, \sigma_r)$, allowing the following calculation:

$$\mathbf{A}_k = \mathbf{U}_{m\times k}\mathbf{U}_{k\times m}^T\mathbf{A} = \left(\sum_{i=1}^{k}\mathbf{u}_i\mathbf{u}_i^T\right)\mathbf{A}, \tag{3}$$

$$\text{and} \qquad \mathbf{A}_k = \mathbf{A}\mathbf{V}_{n\times k}V_{k\times n}^T = \mathbf{A}\left(\sum_{i=1}^{k}\mathbf{v}_i\mathbf{v}_i^T\right). \tag{4}$$

[14]

The Eckart-Young-Mirsky Theorem confirms the approximation detailed by equation 2 is the best rank k approximation in the Frobenius and spectral norm [14]. In terms of computation, Reduced SVD is still expensive to compute if matrix $\mathbf{A}$ is large, even though it is a good approximation. Instead, the rank revealing QR factorisation is a useful though less reliable alternative to SVD [15].

4.2 Proof of the Eckart-Young-Mirsky Theorem for the spectral norm

The Eckart-Young-Mirsky Theorem states that:

$$||\mathbf{A} - \mathbf{A}_k|| \leq ||\mathbf{A} - \mathbf{B}|| \quad \text{for all rank } k \text{ matrices } \mathbf{B}. \tag{5}$$

Furthermore,

$$||\mathbf{A} - \mathbf{A}_k|| = \begin{cases} \sigma_{k+1}, & \text{for the } \; ||\cdot||_2 \text{ (Spectral) norm} \\ \left(\sum_{i=k+1}^{r}\sigma_i^2\right)^{\frac{1}{2}}, & \text{for the } \; ||\cdot||_F \text{ (Frobenius) norm.} \end{cases} \tag{6}$$

[14]

The claim here is that the best rank k approximation for $\mathbf{A}$ in the spectral norm is

$$\mathbf{A}_k = \sum_{i=1}^{k}\sigma_i\mathbf{u}_i\mathbf{v}_i^T, \tag{7}$$

where σ_i are the singular values of $\mathbf{A}$ such that $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n$ and $r = \text{rank}(\mathbf{A})$. It is also noted that the spectral norm is defined as the largest singular value of $\mathbf{A}$, σ_{max}, and the SVD of $\mathbf{A} = \mathbf{U\Sigma V}^T$ as previously detailed in section 4. To begin the proof, these definitions are used to simplify the following expression:

$$||\mathbf{A} - \mathbf{A}_k||_2 = \left\|\sum_{i=1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T - \sum_{i=1}^{k}\sigma_i\mathbf{u}_i\mathbf{v}_i^T{}_2\right\| = \left\|\sum_{i=k+1}^{r}\sigma_i\mathbf{u}_i\mathbf{v}_i^T\right\|_2 = \sigma_{k+1}. \tag{8}$$

Next, it must be shown that

$$||\mathbf{A} - \mathbf{B}||_2 \geq ||\mathbf{A} - \mathbf{A}_k||_2 = \sigma_{k+1}, \tag{9}$$

where $\mathbf{B} \in \mathbb{R}^{n\times p}$ has rank k. By the rank nullity theorem, the null space of $\mathbf{B}$, $N(\mathbf{B})$, must be of dimension $p-k$ [10]. Now, consider the matrix $\mathbf{V}_{k+1} \in \mathbb{R}^{p\times k+1}$ of the first $k+1$ right singular vectors of $\mathbf{V}$, with rank $k+1$. This matrix must have column space, $C(\mathbf{V}_{k+1})$, with dimension $k+1$. Further, $N(\mathbf{B})$ and $C(\mathbf{V}_{k+1})$ cannot be disjoint since they are both subsets of $\mathbb{R}^p$. Calculating

$$\text{dim}N(\mathbf{B}) + \text{dim}C(\mathbf{V}_{k+1}) = p - k + k + 1 = p + 1,$$

it becomes clear there must be some vector $\mathbf{w} \in N(\mathbf{B}) \cap C(\mathbf{V}_{k+1})$. Without loss of generality, this vector can be scaled such that $||\mathbf{w}||_2 = 1$. Since $\mathbf{w} \in C(\mathbf{V}_{k+1})$, it can also be defined $\mathbf{w} = \sum_{i=1}^{k+1} w_i \mathbf{v}_i$ with $\sum_{i=1}^{k+1} w_i^2 = 1$. Now returning to the inequality,

$$\begin{aligned} ||\mathbf{A} - \mathbf{B}||_2^2 &\geq ||(\mathbf{A} - \mathbf{B})\mathbf{w}||_2^2, \\ &= ||\mathbf{A}\mathbf{w}||_2^2 \quad \text{since } \mathbf{w} \in N(\mathbf{B}), \\ &= \mathbf{w}^T \mathbf{V} \mathbf{\Sigma}^2 \mathbf{V}^T \mathbf{w}, \end{aligned}$$

where the last step is derived by noting that multiplying by an orthogonal matrix does not change the length. To explain in more detail, by definition for an orthogonal $\mathbf{U}$ the product $\mathbf{U}^T\mathbf{U} = 1$, therefore one can manipulate $||U\mathbf{w}||^2 = \mathbf{w}^T\mathbf{U}^T\mathbf{U}\mathbf{w} = \mathbf{w}^T\mathbf{w}$ [17]. Similarly, using the SVD of $\mathbf{A}$ and expanding reproduces the following result:

$$||\mathbf{A}\mathbf{w}||_2^2 = ||\mathbf{U}\mathbf{\Sigma}\mathbf{V}^T\mathbf{w}||_2^2 = ||\mathbf{\Sigma}\mathbf{V}^T\mathbf{w}||_2^2 = (\mathbf{\Sigma}\mathbf{V}^T\mathbf{w})^T(\mathbf{\Sigma}\mathbf{V}^T\mathbf{w}) = \mathbf{w}^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\mathbf{w}.$$

Returning to the proof, the inequality can be further manipulated:

$$\begin{aligned} ||\mathbf{A} - \mathbf{B}||_2^2 &\geq \mathbf{w}^T\mathbf{V}\mathbf{\Sigma}^2\mathbf{V}^T\mathbf{w}, \\ &= \sum_{i=1}^{k+1} w_i^2 \sigma_i^2 \qquad \text{by substituting } \mathbf{w} = \sum_{i=1}^{k+1} w_i \mathbf{v}_i, \\ &\geq \sigma_{k+1}^2 \sum_{i=1}^{k+1} w_i^2 \quad \text{since by definition } \sigma_{k+1} \text{ is the smallest singular value}, \\ &= \sigma_{k+1}^2 \qquad \text{using } \sum_{i=1}^{k+1} w_i^2 = 1, \\ &= ||\mathbf{A} - \mathbf{A}_k||_2^2. \end{aligned}$$

Taking the square root of each side of the equation produces the desired result and proves the Eckart-Young-Mirsky Theorem for the spectral norm [16].

5 Applications of SVD within collaborative filtering

SVD can be applied in model-based collaborative filtering to predict ratings from items (titles) using explicit feedback from the user (and other users). The Netflix Prize data sets contain user ratings for titles on a scale of 1 to 5 stars. Table 1 presents a fictitious subset of this data, where some entries are incomplete to replicate the real world data where it is unlikely that any user has rated every title in the matrix [9].

To begin, the data is first compiled in matrix $\mathbf{R} \in \mathbb{R}^{m \times n}$, where each term r_{ui} represents the user u's real rating of title i. In an ideal world where every user has rated every item in the subset, the next step would be to perform SVD to obtain the factorisation:

$$\mathbf{R} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \mathbf{U}\sqrt{\mathbf{\Sigma}}\sqrt{\mathbf{\Sigma}}\mathbf{V}^T, \tag{10}$$

where $\mathbf{p}_u = \mathbf{U}\sqrt{\mathbf{\Sigma}}$ is a user-factor vector and $\mathbf{q}_i = \sqrt{\mathbf{\Sigma}}\mathbf{V}^T$ is the item-factor vector.[8] The value of separating matrix $\mathbf{R}$ into component parts is to understand the user and item factors that contribute to the final rating. Fundamentally, these vectors represent the features which distinguish each title from

Subset of Netflix Prize Dataset				
User or Title	About Time	Good Will Hunting	Inception	Forrest Gump
Hannah	4	4	3	4
Zoe	5	?	2	3
Finn	3	3	5	3
Constance	3	4	?	4
Matthew	4	5	5	3

Table 1: Subset of a fictitious Netflix Prize Dataset showing user ratings for 4 titles.

the next, and also the features relating to each type of user in the scenario [6]. Fortunately, it does not matter that these features are unknown.

However, this calculation fails immediately because standard SVD decomposition cannot be carried out when matrix $\mathbf{R}$ is not complete since the eigenvectors and eigenvalues need to be computed. This is essentially the crux of the problem recommender systems aim to solve on a broader scale: how to accurately fill in the missing ratings.

5.1 SVD technique for incomplete matrices

One matrix factorisation technique is to approximate the user-item rating matrix $\mathbf{R}$ using two low dimensional matrices, under the assumption that $\mathbf{R}$ is low rank. The approximation selects the f largest singular values as these have the greatest effect on the user and item latent features which contribute to the final rating. This is equivalent to choosing $f = k < r$ to form a Reduced SVD and allows a prediction of the rating of user u for item i:

$$\widehat{r_{ui}} = \mathbf{p}_u^T \mathbf{q}_i, \tag{11}$$

with $\mathbf{p}_u \in \mathbb{R}^f$ and $\mathbf{q}_i \in \mathbb{R}^f$.[8] Relating this to the fictitious data set, the dot product $\widehat{r_{22}} = \mathbf{p}_2^T \mathbf{q}_2$ represents Zoe's overall interest in Good Will Huntings' characteristics. The elements of vector $\mathbf{q}_2$ will measure the extent to which Good Will Hunting possesses latent (hidden) features, which may be positive or negative. Likewise, the elements of $\mathbf{p}_2$ measures the extent of the interest that Zoe has in titles that are high on the corresponding factors, positive or negative.

The prediction can be formed by estimating factors $\mathbf{q}_i$ and $\mathbf{p}_u$ by minimising the difference between the true rating r_{ui} and the predicted rating $\widehat{r_{ui}}$ for a known set L of (u, i) pairs from the training set. This is equivalent to finding the lowest rank approximation of $\mathbf{R}$ and takes the form of the Least Squares Problem:

$$\min_{p,q} \sum_{(u,i)\in L} (r_{ui} - \mathbf{q}_i^T \mathbf{p}_u)^2 + \lambda \left(||\mathbf{q}_i||^2 + ||\mathbf{p}_u||^2\right), \tag{12}$$

which is will be solved via stochastic gradient descent [7]. This is a method of training machine learning models for optimisation, and is commonly used when dealing with sparse data. The aim here is to avoid over-fitting the observed data both by normalisation and finding good parameters for the hyper-parameter λ when L is large.

5.2 Further SVD applications: NSVD

The SVD technique above can be modified to incorporate further aspects of the data set and improve the reliability of the predicted rating. This is necessary, since large SVD models only achieved a RMSE of

0.905 on the probe set. The BigChaos solution describes how a variety of techniques such as SVD-Time, Symmetric-View-SVD++, and SVD with Adaptive User Factors are blended to incorporate temporal effects and user biases into the model [9]. This section will focus on just one further application of SVD: Neighbourhood Singular Value Decomposition (NSVD).

NSVD1 and NSVD2 (Neighbourhood SVD 1 and 2) are generally deemed the most important approaches in reducing the RMSE in the BigChaos solution. They utilise implicit feedback to describe user preferences, regardless of whether the user has explicitly rated a particular item [19]. When performing SVD, each user is explicitly parameterised, assuming no item or user biases. However in NSVD1, the user is modelled based on items they have rated. To include this term, item i is associated with both vector $\mathbf{p}_i \in \mathbb{R}^f$ which represents a new set of item features, and also the original vector $\mathbf{q}_i$. In addition, the introduction of a Boolean term $N(u)$ accounts for the set of terms for which the user u implicitly prefers, either interest or no interest. The following prediction is obtained:

$$\widehat{r_{ui}} = \mu_i + \mu_u + \mathbf{p}_i^T \left(\frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} \mathbf{q}_j \right). \tag{13}$$

Comparatively, NSVD2 chooses to use the same item features for $\mathbf{p}_i$ and $\mathbf{q}_i$, and is modelled by the equation:

$$\widehat{r_{ui}} = \mu_i + \mu_u + \mathbf{p}_i^T \left(\frac{1}{\sqrt{|N(u)|}} \sum_{j \in N(u)} \mathbf{p}_j \right). \tag{14}$$

[9]

The benefits of these two models directly relates to the way Netflix operates, as described in the introduction. Firstly, there are 220.67 million subscribers (users) yet a relatively small number of titles, thus utilising item-parameters instead of user parameters reduces the complexity of the model. Furthermore, the models can predict titles for new users as soon as they have provided ratings to the system, without needing to re-train the model in order to estimate new parameters. Additional benefits also include improved accuracy of predictions since implicit feedback is incorporated efficiently, becoming more dominant as $|N(u)|$ increases, and also reaping the benefits of explaining predictions by users' past feedback in the same manner as neighborhood models [19].

5.3 Matrix factorisation alternative: Restricted Boltzmann machines

Deep learning methods are now prevalent in modern day recommender systems. Typically, deep learning models use a variant of stochastic gradient descent to optimize an objective function, and evaluate multiple levels of features of the data [21]. The research literature surrounding the topic is vast and extensive so this section will not mathematically describe any of the methods, but will instead draw attention to Restricted Boltzmann machines (RBM), which are often employed instead of or in conjunction with SVD techniques.

The deep learning model will learn the user and item latent feature vectors $\mathbf{p}_u$ and $\mathbf{q}_i$ based on the user and item interactions using a neural network instead of performing matrix factorization. This is particularly useful when the training set is large or highly complex, as the deep neural networks enable automatic feature learning from raw data, in a supervised or unsupervised method. RBM in particular can be described as a two-layer neural neural network encompassing a visible and a hidden layer. This approach is scalable to large data sets whilst producing high-quality recommendations of items for each particular user [20].

6 Conclusion

The Netflix Prize and accompanying data sets prompted a crucial research effort into recommender systems and the algorithms and machine learning methods which underpin them. This article has investigated SVD and its applications within collaborative filtering. Much has changed since the competition was launched, but Netflix still utilises many of the approaches discussed: in 2013 Netflix even won an Emmy award for "Personalized Recommendation Engines For Video Discovery" [1]. Remarkably, Netflix has been able to garner the knowledge of user preference to predict user interest in new series and invest accordingly. Areas for recommendation outside of this article include a deep dive into Restricted Boltzman Machines, and how they are implemented alongside SVD approaches to improve recommender systems.

References

[1] Wikipedia Contributers: "Netflix", `https://en.wikipedia.org/wiki/Netflix`
[Accessed 11/10/2020.]

[2] Statista: Number of Netflix paid subscribers worldwide, `https://www.statista.com/statistics/250934/quarterly-number-of-netflix-streaming-subscribers-worldwide/`
[Accessed 11/10/2022.]

[3] Netflix Research Area: Recommendations, `https://research.netflix.com/research-area/recommendations`
[Accessed 11/10/2022.]

[4] J Bennett and S. Lanning, *The Netflix Prize*, 2007, `https://www.cs.uic.edu/~liub/KDD-cup-2007/proceedings/The-Netflix-Prize-Bennett.pdf` [Accessed 11/10/2022.]

[5] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. *Deep Learning Based Recommender System: A Survey and New Perspectives*, 2019. ACM Comput. Surv. 52, 1, Article 5 (February 2019), 38 pages, `https://doi.org/10.1145/3285029` [Accessed 11/10/2022.]

[6] UCL: How Netflix Uses Recommender Systems, `https://reflect.ucl.ac.uk/nsci0010-2021-class-blog/2021/02/24/how-netflix-uses-recommender-systems/`
[Accessed 11/10/2022.]

[7] Yehuda Koren, Robert Bell and Chris Volinsky, *Matrix Factorization Techniques for Recommender Systems*, 2009 `https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf` [Accessed 11/10/2020.]

[8] Yehuda Koren, *Factorization meets the neighborhood: a multifaceted collaborative filtering model*, 2008, Association for Computing Machinery, New York, NY, USA, 426–434. `https://doi.org/10.1145/1401890.1401944` [Accessed 11/10/2020.]

[9] Töscher, Andreas and Jahrer, Michael, *The BigChaos Solution to the Netflix Grand Prize*, 2009

[10] D. Poole. *Linear Algebra: A modern introduction*, Cengage Learning 2002, Sections 7.1 - 7.5

[11] Wikipedia Contributers: "Linear Algebra" `https://en.wikipedia.org/wiki/Rank_(linear_algebra)`
[Accessed 11/10/2020.]

[12] V. Klema and A. Laub, *The singular value decomposition: Its computation and some applications*, in IEEE Transactions on Automatic Control, vol. 25, no. 2, pp. 164-176, April 1980.

[13] Golub, Gene H., Virginia C. Klema and G. W. Stewart. "Rank degeneracy and least squares problems." (1976).

[14] N. Kishore Kumar and J. Schneider, *Literature survey on low rank approximation of matrices*, Linear and Multilinear Algebra, 2017

[15] Y. P. Hong and C.-T. Pan, *Mathematics of Computation* Vol. 58, No. 197 (Jan., 1992), pp. 213-232 (20 pages) Published By: American Mathematical Society

[16] Wikipedia Contributors: 'Low Rank Approximation' `https://en.wikipedia.org/w/index.php?title=Low-rank_approximation&oldid=1109696138`
[Accessed 20/10/2020.]

[17] Gilbert Strang, The Singual Value Decomposition of a Matrix Lecture Notes, Massachusetts Institute of Technology, 2016.

[18] M. Linkmann, Entrepreneurship in the Mathematical Sciences Lecture Notes, University of Edinburgh, 2021.

[19] R. Bell and Y. Koren. *Scalable collaborative filtering with jointly derived neighborhood interpolation weights,* In IEEE International Conference on Data Mining. KDD-Cup07, 2007.

[20] Ruslan Salakhutdinov, Andriy Mnih and Geoffrey Hinton *Restricted Boltzmann Machines for Collaborative Filtering*, Appearing in Proceedings of the 24 th International Conference on Machine Learning, Corvallis, OR, 2007.

[21] IMB: Neural Networks, `https://www.ibm.com/cloud/learn/neural-networks`
[Accessed 11/10/2020.]

[Word count: 2422]