Active Content Popularity Learning via Query-by-Committee for Edge Caching

Srikanth Bommaraveni*, Thang X. Vu*, Satyanarayana Vuppala+, Symeon Chatzinotas*
and Björn Ottersten*
*Interdisciplinary Centre for Security, Reliability and Trust (SnT), the University of Luxembourg.
+United Technologies Research Centre, Ireland
Email: {srikanth.bommaraveni, thang.vu, symeon.chatzinotas, bjorn.ottersten}@uni.lu,
vuppalsa@utrc.utc.com

***Abstract*—Edge caching has received much attention as an effective solution to face the stringent latency requirements in 5G networks due to the proliferation of handset devices as well as data-hungry applications. One of the challenges in edge caching systems is to optimally cache strategic contents to maximize the percentage of total requests served by the edge caches. To enable the optimal caching strategy, we propose an Active Learning approach (AL) to learn and design an accurate content request prediction algorithm. Specifically, we use an AL based Query-by-committee (QBC) matrix completion algorithm with a strategy of querying the most informative missing entries of the content popularity matrix. The proposed AL framework leverage's the trade-off between exploration and exploitation of the network, and learn the user's preferences by posing queries or recommendations. Later, it exploits the known information to maximize the system performance. The effectiveness of proposed AL based QBC content learning algorithm is demonstrated via numerical results.**

***Index Terms*—Edge caching, Active learning, Matrix completion, Content popularity, 5G cellular network.**

I. Introduction

Global mobile data traffic has experienced tremendous growth in recent years and is predicted to increase sevenfold between 2017 and 2022 [1]. The growth in data traffic is primarily due to the access to video streaming services over cellular networks. On the other hand, the number of mobile devices is expected to reach 8.4 billion by 2022. The demand for data-hungry services like high definition video transmission from such devices is escalating. As a result, the existing cellular networks fail to cope with data demands from the ever increasing number of devices. However, edge caching is recognized as the most effective means to meet the data demands in the 5G networks [2]–[6].

Research in the past years has focused on studying the performance gain in cache-enabled network architectures with various objectives such as minimizing average latency, network congestions, maximizing user's quality of experience (QoE) or energy efficiency. In [2], authors show the effectiveness of caching to reduce network congestion in the back-haul links and maximizing the QoE. An optimization problem for content placement over multiple distributed caches aiming to minimize the latency has been proposed in [3]. The energy efficiency and delivery time of coded and uncoded caching schemes are analysed in [4].

On the contrary, aforementioned works assume that content popularity is known. However, in a practical scenario, the content popularity is often unknown, time-varying and which needs to be estimated. In [5] a content popularity matrix estimation along with leveraging social networks and D2D communications is used to predict the popularity in a proactive manner. A reinforcement learning framework to learn the space-time popularity of requests is proposed in [7]. In [8], the authors propose Poisson factor analysis to capture the correlations of contents and Bayesian learning to estimate content popularity.

A fundamental challenge in the content popularity learning is that only a small subset of requests are observed at the edge cache. This results in the data sparsity and cold start problems. In many real-world applications, it is very difficult, time-consuming, or expensive to collect and label training data to build suitable prediction model. In this context, active learning is an indispensable tool, as it steers the learning process towards achieving the accuracy goal by actively selecting the most useful samples. Sometimes it is also referred as query learning or optimal experimental design [9]. Active learning has been applied in various domains such as fraud detection [10], webpage classification [11] and protein structure prediction [12].

In this paper, we study the content popularity learning in edge caching wireless networks. The main contributions are as follows:

- We use an active learning based Query-by-committee (QBC) matrix completion algorithm to estimate the popularity of contents.

- We propose an active learning online caching framework, with a strategy of querying the most informative missing entries of the matrix.

The rest of the paper is organized as follows: In Section II we describe the system model and problem formulation. In section III, we propose an active learning based content caching algorithm. Section IV presents the simulation results and finally, conclusions are drawn in Section V.

Notation: Lower or upper case letters represent scalars, boldface upper case for matrices, boldface lower case for vectors, $[.]_{a,b}$ represents the element in row a and column b of a matrix, $\|.\|_*$ represents nuclear norm, $\|.\|_F$ represents the frobenius norm, $\odot$ is element-wise product or also called as Hadamard product, $(.)^T$ denotes the transpose operator, $|.|$ represents the cardinality of set and $\mathbf{1}$ represents a vector of all ones.

II. System Model and Problem Formulation

A. System Model

We consider the downlink edge network scenario with small base stations (SBSs) connected to the mobile core network over the reliable back-haul links. To leverage the back-haul traffic offloading and satisfy the latency requirements of user terminals (UTs), each SBS is equipped with mobile edge computing server to process and cache the contents in the finite storage memory as depicted in figure 1. Further, coverage areas of the SBS are assumed to be disjoint, thus a UT can only be connected to the closest SBS at a time. Besides, it is assumed that SBS serves the UT request immediately if the content is cached locally at SBS else the content is fetched from the content servers. In the sequel, a model for requests from UT's to contents is defined.

B. Demand matrix

Denote $\mathcal{U} = \{ut_1, \ldots, ut_K\}$ as a set of the UTs connected to the SBS of interest, which has a full access to a library $\mathcal{F} = \{f_1, \ldots, f_F\}$ of F contents at the content servers via the back-haul link. Without loss of generality, all the contents are assumed to be of the same size which is trivially satisfied by breaking any content into smaller blocks of the same size [13], [14]. Each SBS can store up to D contents, with $D < F$. During the T time slots, R requests are made by the UTs to the SBS.

The correlations between UTs requests and contents are defined by $\mathbf{L} \in \mathbb{R}^{K \times F}$ which is referred as demand matrix at the SBS. The rows and columns correspond to anonymous UT profiles and the contents, respectively. Each element represents the probability of a request made by a UT to a content. Let $[L]_{k,f} \in \mathbb{Z}^+$ be the number of requests for content f from the UT k. In reality, each UT only requests for a small subset of contents, therefore the demand matrix is large rectangular and sparse. To know which elements are available and which are missing, a binary matrix $\mathbf{\Omega} \in \{0,1\}^{K \times F}$ is defined corresponding to $\mathbf{L}$ such that if the user requests the a content then the entry is 1 and 0 otherwise:

$$[\Omega]_{k,f} = \begin{cases} 1, & [L]_{k,f} \neq 0 \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

In order to maximize the total requests served by the cache at SBS, the demands of content are estimated by predicting the missing entries of the demand matrix. In the next section, we propose an active learning based matrix completion for estimation of missing entries of demand matrix.

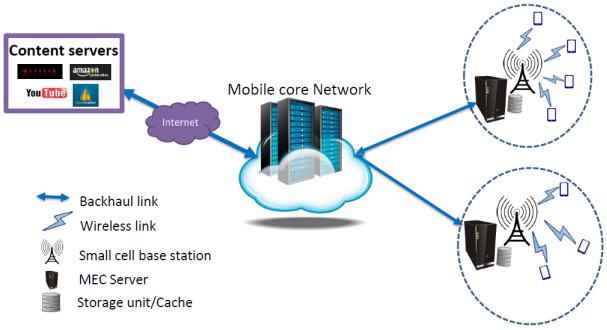

Fig. 1: The network model

III. Active popularity learning and caching algorithm

Active learning is a special case of semi-supervised learning in which a learning algorithm is able to interactively query the oracle to acquire high-quality data [15]. The missing entries of the demand matrix at the SBS are estimated using active learning based Query-by-committee [16] matrix completion algorithm as described in Algorithm 1. The intuition of QBC is that a group of passive matrix completion models/algorithms form as a committee aiming to minimize the version space, which is the subset of all hypothesis space[1] that are consistent with the known entries. And the choice of models in a committee can be constructed in many ways [15], for example using simple sampling [16]. To constrain the size of version space as small as possible, QBC uses the uncertainty of prediction for each missing entry. This is used to find the most informative missing entries, that will enable to accelerate the learning rate of the system under query budget. We define query budget as the number of uncertain contents to be stored in the cache. In the follow-up, QBC with three passive matrix completion algorithms and active learning caching strategy is presented.

[1]set of possible approximations of true function that the algorithm can create.

A. QBC Matrix completion

Let N be the total number of QBC members. Then, the predicted matrix by the n-th committee member is denoted by $\mathbf{M}_n$ for $n \in \{1, \ldots, N\}$. We have used the following three low-rank approximation variants of passive matrix completion algorithms as the members of the committee.

Algorithm 1: QBC Matrix completion

Input: partially observed matrix $\mathbf{L} \in \mathbb{R}^{K \times F}$, binary matrix $\mathbf{\Omega} \in \{0,1\}^{K \times F}$, $N = 3$.
Output: Estimation of missing entries matrix: $\mathbf{P}$, Uncertainty matrix: $\mathbf{U}$

1: Estimate the predicted matrices $\mathbf{M}_n$ using the N matrix completion methods,
2: $\mathbf{P} = \frac{1}{N} \sum_{n=1}^{N} \mathbf{M}_n$
3: Calculate the entries of $\mathbf{U}$ using (2)

1) Singular Value Thresholding [17]: Singular value thresholding (SVT) is a standard low rank approximation matrix completion algorithm. In this method the nuclear norm of the matrix is minimized subject to certain constraints. The minimization problem is defined as,

$$\begin{aligned} &\underset{\mathbf{M}_n}{\text{minimize}} \quad \|\mathbf{M}_n\|_* \\ &\text{subject to} \quad [M_n]_{k,f} = [L]_{k,f}, \forall (k,f) : [\Omega]_{k,f} = 1. \end{aligned}$$

2) Unconstrained nuclear norm minimization [18]: The unconstrained nuclear norm minimization of a matrix is given as,

$$\underset{\mathbf{M}_n}{\text{minimize}} \quad \mu \|\mathbf{M}_n\|_* + \|\mathbf{\Omega} \odot (\mathbf{L} - \mathbf{M}_n)\|_F^2,$$

where μ is the regularization constant to avoid overfitting and is determined by cross-validation. It is solved by using fixed point and iterative algorithm proposed in [18]. The algorithm is very fast, robust compared to SVT and uses the approximate of singular value decomposition.

3) Matrix Factorization [19]: Matrix factorization is a way of reducing a matrix into its latent factor space. The minimization of regularized squared error on the set of observed entries is given as,

$$\underset{\mathbf{X},\mathbf{Y}}{\text{minimize}} \quad \|\mathbf{\Omega} \odot (\mathbf{L} - \mathbf{X}\mathbf{Y}^T)\|_F^2 + \lambda(\|\mathbf{X}\|_F^2 + \|\mathbf{Y}\|_F^2),$$

where $\mathbf{X} \in \mathbb{R}^{K \times r}$ and $\mathbf{Y} \in \mathbb{R}^{F \times r}$ are two latent factor matrices such that $\mathbf{M}_n = \mathbf{X}\mathbf{Y}^T$, r is the rank and λ is the regularization constant.

The final estimated demand matrix, say $\mathbf{P}$, is calculated as the average of predicted matrices by the committee members as described in step 2 of Algorithm 1. Now, the task is to find the most informative missing entries that increase the learning accuracy of the system based on the predicted matrices. For this, we define an uncertainty matrix $\mathbf{U}$ corresponding to each entry of estimated demand matrix $\mathbf{P}$. The entry $[U]_{k,f}$ of the uncertainty matrix is calculated as,

$$[U]_{k,f} = \frac{1}{N} \sum_{n=1}^{N} \Big[[M_n]_{k,f} - [P]_{k,f} \Big]^2 \tag{2}$$

Note, (2) calculates the variance of the missing entry predictions and is equal to zero for the known entries. In the sequel, we propose an online active learning cache and query strategy based on the estimated demand and uncertainty matrices.

B. Cache and Query strategy

In this sub-section, an active learning based online caching framework is proposed to store the contents at the cache. The storage at the SBS is divided into two parts, one part stores the most demand contents obtained from the estimated demand matrix and second part stores the uncertain contents obtained from the uncertainty matrix. The allows the system to leverage the trade-off between exploration and exploitation, that is, it finds more information about the UT's preferences by caching the uncertain contents and then exploits the known information to maximize the system performance (cache hit ratio). Let the demand values of the contents are represented by a vector $\mathbf{d} = [d_1, \ldots, d_F]$, where the demand value of content f is calculated as,

$$d_f = \sum_{k=1}^{K} [P]_{k,f} \tag{3}$$

Also, the uncertainty values of the contents are represented by a vector $\mathbf{u} = [u_1, \ldots, u_F]$, where the uncertain value of f content is calculated as,

$$u_f = \sum_{k=1}^{K} [U]_{k,f} \tag{4}$$

The number of uncertain contents to be cached alongside with the demand contents in the cache is given by query budget Q. The selection set and cache placement vector is defined by $\mathcal{C}$ and binary vector $\mathbf{x} \in \{0,1\}^{1 \times F}$ respectively. So the top $D-Q$ contents of $\mathbf{d}$ and the top Q contents of $\mathbf{u}$ are selected to cache placement $\mathbf{x}$ as described in Algorithm 2. Note that, before storing the files in storage units, computing server check and skip the files if it is already stored. After the cache placement, the queries are generated for the uncertain contents in cache based on the uncertain matrix.

IV. Numerical Results

In this section, we demonstrate the effectiveness of proposed active learning approach for content caching through numerical results.

Following are the system set-up and simulation parameters that are considered in numerical results. The SBS is

Algorithm 2: Content caching with active matrix completion

1: Initialize: $\mathbf{L}, \mathbf{\Omega}, N, D, Q$
2: repeat
3: $[\mathbf{P}, \mathbf{U}] = \text{QBC}(\mathbf{L}, \mathbf{\Omega}, N)$,
4: calculate $\mathbf{d}$ and $\mathbf{u}$ as given in (3), and (4) respectively
5: Sorting and Indexing :
$[\mathbf{d}_{value}, \mathbf{d}_{index}] = \text{sort}(\mathbf{d}, \text{“descend”})$
$[\mathbf{u}_{value}, \mathbf{u}_{index}] = \text{sort}(\mathbf{u}, \text{“descend”})$
6: Selection set : $\mathcal{C}$
$\mathcal{C} = \{\mathcal{F}_i \cup \mathcal{F}_{i^\dagger} \mid i = \mathbf{d}_{index}[1 : D - Q], i^\dagger = \mathbf{u}_{index}[1 : Q]\}$
7: Placement vector : $\mathbf{x}$

$$[x]_f = \begin{cases} 1, & f \in \mathcal{C} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

8: Query generation :
Get the uncertain entries of the Q placed contents to query from $\mathbf{U}$
9: until: stopping criteria

equipped with a storage memory of 30 MB. The number of contents in the servers (F) is 100 and all the contents have equal size of 1 MB. Unless otherwise stated, the query budget Q is considered to be 2 i.e., $Q = 2$. The number of UTs (K) is 30 and requests from UTs for the contents follow a zipf-like distribution denoted by

$$P_k(f) = \omega / f^\alpha, \quad (6)$$

where $\omega = \left(\sum_{f=1}^{F} 1/f^\alpha\right)^{-1}$ and $\alpha = 0.8$ is the Zipf skewness factor.

To evaluate the performance, a reference demand matrix ($\mathbf{L}_{true}$) is generated where element in row k column f represent a request of user k for content f. Similarly, $\mathbf{d}_{true}$ is the true content demands calculated from $\mathbf{L}_{true}$ as given in Section III. Further, demand matrix $\mathbf{L}$ is generated from $\mathbf{L}_{true}$ by deleting 98% of its entries randomly.

The performance of the proposed framework is evaluated via three metrics: root mean square error (RMSE), cache hit ratio (CHR), and the backhaul load for active learning. With the help of notations defined, RMSE and CHR are computed as

$$RMSE = \frac{1}{\|\mathbf{L}_{true}\|_F} \|\mathbf{L}_{true} - \mathbf{L}\|_F, \quad (7)$$

$$CHR = \frac{\mathbf{d}_{true}\mathbf{x}^T}{\mathbf{d}_{true}\mathbf{1}^T}. \quad (8)$$

$$\text{Back-haul load} = |\{\mathcal{C}\}_{new} \setminus \{\mathcal{C}\}_{old}| \text{ (MB)} \quad (9)$$

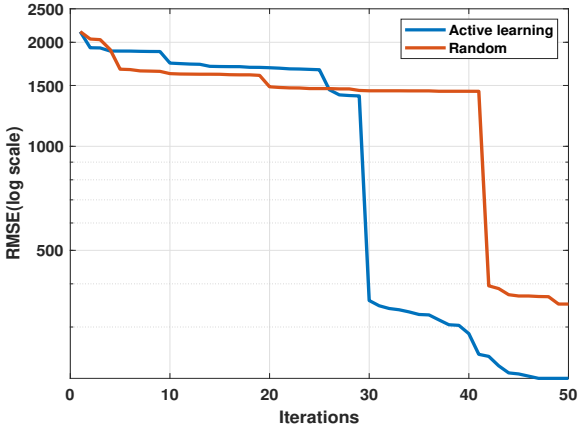

Fig. 2: RMSE as a function of iterations

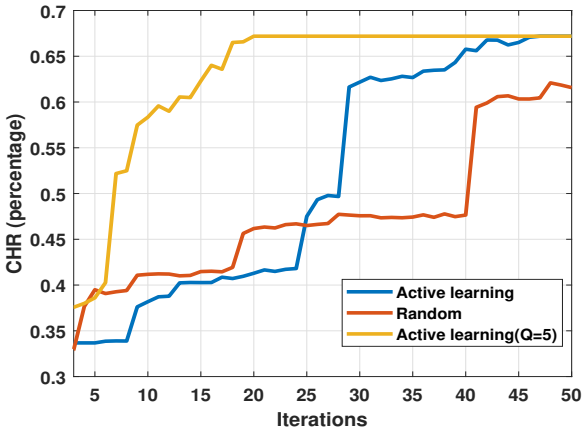

Fig. 3: Impact of CHR

The back-haul load for active learning is defined as the number of contents that are newly fetched from the content server. These newly fetched contents are replaced by the old contents in the storage unit.

Further, the performance obtained by a random querying strategy is used as a benchmark for the comparison of the results. In figure 2, the performance i.e., RMSE of AL query strategy is compared with random query strategy as a function of iterations. The goal is to study the rate of reduction of the RMSE, as more and more contents are explored. Initially, since 98% of entries are missing, the number of known entries are insufficient to estimate the missing entries and as a result AL performs poor. This behaviour can be observed till iteration 25 in figure 2. This effect is explained by the concept called incoherence property introduced by the authors in [20]. It is evident that the rate of decrease in RMSE with AL compared to querying at random is much faster after 25 iterations. This is due to the fact that, using AL the contents to query are selected in a way that yields a finite information gain.

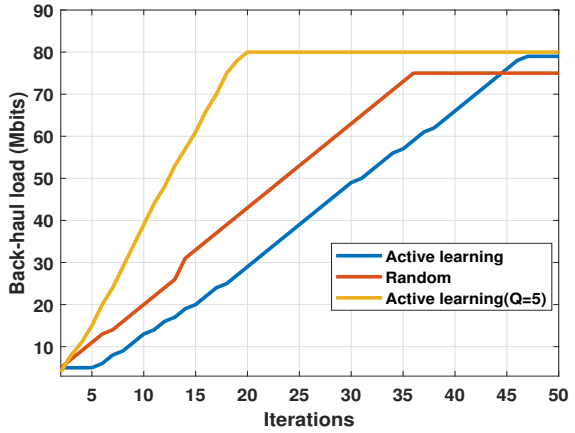

Fig. 4: Impact of Back-haul load

In figure 3, CHR is illustrated as a function of iterations. Similar to figure 2, due to the incoherency the proposed AL approach performance poorer than random strategy which can be observed till iteration 25. However, the CHR of improves drastically compared to random queries beyond iteration 25 for the same reason mentioned in figure 2. Moreover, the performance of AL method can be further improved by increasing query budget. This is illustrated with $Q = 5$ in figure 3. However, this induces higher back-haul load (which is defined shortly) as shown in figure 4. The performance of AL with $Q = 5$ can be achieved without increasing back-haul load by AL with $Q = 2$ with the number of iterations. This performance convergence of AL with $Q = 2$ to AL with $Q = 5$ can be observed around the iterations 45-50 in figure 3.

The back-haul load is illustrated as a function of iterations as shown in figure 4. Active learning requires less back-haul load compared to random query strategy till iteration 45. However, AL methods impose slightly higher back-haul load compared to random query strategy beyond iteration 45. This slight increase in back-haul load results in higher CHR by AL methods as explained in figure 3. Moreover, the increase in back-haul load with query budget Q during the initial iterations can also be observed in figure 4.

V. Conclusion

In this paper, we proposed a novel active learning based approach for proactive content caching algorithm. The key idea is to leverage the intelligence of UT's to actively complete the partially observed matrices and thereby to estimate the popularity of contents. The interactive learning between the system and UT's helps the UT's become more self-aware of their own likes/dislikes while at the same time providing new information to the system which helps in better estimation of the popularity of contents. Since our proposed algorithm is model-free, it can be used either for fixed or time-varying popularity learning situations. Our simulations results show that the advantages of AL-based matrix completion and also the efficacy of the algorithms in appropriately identifying a set of missing entries over random sampling in reducing the reconstruction error.

Acknowledgment

This work has been supported by the National Research Fund, Luxembourg project AGNOSTIC (742648), the FNR bilateral project LARGOS (12173206), and the FNR CORE ProCAST (C17/IS/11691338).

References

[1] Cisco, "Cisco visual networking index: Forecast and trends, 2017 -2022," *White paper*, Feb. 2019.

[2] E. Batu, J. Gungo, and M. Debbah, "Proactive small cell networks," in *ICT 2013*, May. 2013, pp. 1–5.

[3] K. Shanmugam, N. Golrezaei, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless content delivery through distributed caching helpers," *IEEE Transactions on Information Theory*, vol. 59, no. 12, pp. 8402–8413, Dec. 2013.

[4] T. X. Vu, S. Chatzinotas, and B. Ottersten, "Edge-caching wireless networks: Performance analysis and optimization," *IEEE Transactions on Wireless Communications*, vol. 17, no. 4, pp. 2827–2839, Apr. 2018.

[5] E. Bastug, M. Bennis, and M. Debbah, "Living on the edge: The role of proactive caching in 5g wireless networks," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 82–89, Aug. 2014.

[6] G. S. Paschos, G. Iosifidis, M. Tao, D. Towsley, and G. Caire, "The role of caching in future communication systems and networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 6, pp. 1111–1125, Jun. 2018.

[7] A. Sadeghi, F. Sheikholeslami, and G. B. Giannakis, "Optimal and scalable caching for 5g using reinforcement learning of space-time popularities," *IEEE Journal of Selected Topics in Signal Processing*, vol. 12, no. 1, pp. 180–190, Feb. 2018.

[8] S. Mehrizi, A. Tsakmalis, S. Chatzinotas, and B. Ottersten, "Content popularity estimation in edge-caching networks from bayesian inference perspective," in *2019 16th IEEE Annual Consumer Communications Networking Conference (CCNC)*, Jan. 2019, pp. 1–6.

[9] B. Settles, "Active learning literature survey," University of Wisconsin–Madison, Computer Sciences Technical Report 1648, 2009.

[10] S. Stolfo, D. W. Fan, W. Lee, A. Prodromidis, and P. Chan, "Credit card fraud detection using meta-learning: Issues and initial results," in *AAAI-97 Workshop on Fraud Detection and Risk Management*, 1997.

[11] S. Sun and D. R. Hardoon, "Active learning with extremely sparse labeled examples," *Neurocomputing*, vol. 73, no. 16-18, pp. 2980–2988, 2010.

[12] M.-W. Chang, L.-A. Ratinov, N. Rizzolo, and D. Roth, "Learning and inference with constraints." in *AAAI*, 2008, pp. 1513–1518.

[13] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *2012 Proceedings IEEE INFOCOM*, Mar. 2012, pp. 1107–1115.

[14] K. Poularakis, G. Iosifidis, and L. Tassiulas, "Approximation algorithms for mobile data caching in small cell networks," *IEEE Transactions on Communications*, vol. 62, no. 10, pp. 3665–3677, Oct. 2014.

[15] B. Settles, *Active Learning*. Morgan & Claypool Publishers, 2012.

[16] H. S. Seung, M. Opper, and H. Sompolinsky, "Query by committee," in *Proceedings of the Fifth Annual Workshop on Computational Learning Theory*. ACM, 1992, pp. 287–294.

[17] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM J. on Optimization*, vol. 20, no. 4, pp. 1956–1982, Mar. 2010.

[18] S. Ma, D. Goldfarb, and L. Chen, "Fixed point and bregman iterative methods for matrix rank minimization," *Mathematical Programming*, vol. 128, no. 1, pp. 321–353, Jun. 2011.

[19] Y. Koren, R. Bell, and C. Volinsky, "Matrix factorization techniques for recommender systems," *Computer*, vol. 42, no. 8, pp. 30–37, Aug. 2009.

[20] E. J. Candes and B. Recht, "Exact low-rank matrix completion via convex optimization," in *2008 46th Annual Allerton Conference on Communication, Control, and Computing*, Sep. 2008, pp. 806–812.