

# COMPREHENSIVE PERSONALIZED RANKING USING ONE-BIT COMPARISON DATA

*Aria Ameri\*, Arindam Bose, Mojtaba Soltanalian*

Department of Electrical and Computer Engineering  
University of Illinois at Chicago  
Chicago, USA

## ABSTRACT

The task of a personalization system is to recommend items or a set of items according to the users' taste, and thus predicting their future needs. In this paper, we address such personalized recommendation problems for which one-bit comparison data of user preferences for different items as well as the different user inclinations toward an item are available. We devise a comprehensive personalized ranking (CPR) system by employing a Bayesian treatment. We also provide a connection to the learning method with respect to the CPR optimization criterion to learn the underlying low-rank structure of the rating matrix based on the well-established matrix factorization method. Numerical results are provided to verify the performance of our algorithm.

**Index Terms**— Collaborative filtering, low-rank matrix, matrix factorization, one-bit comparison, recommendation systems.

## 1. INTRODUCTION

Content recommendation is one of the most vital tasks emerging in various online systems [1–4] and mobile applications [5, 6]. Since the advent of the Netflix challenge [7], study of recommendation systems, also known as personalization systems, has been an active research area. The ultimate goal of a recommendation system is to endorse items based on different parameters, such as the preference of a user, the relevance of an item and the quality of its content, contemporary trend, among others [8]. Recommendations can be in the form of predicting the user preferences for an unobserved item or a list of top relevant items. In order to form such recommendations, personalization systems generally rely on explicit preferences of users based on readily available evidence such as ratings, votes or reviews. However, in practice, implicit feedback, such as clicks on a link or views of a video may also contribute to define user preferences. In many cases, the implicit feedback can only be observed in a rather obscure binary manner, e.g. a 'one' can signify that the user has an interest in an item, while a 'zero' may represent a disliked or merely an unobserved item. Note that user ratings can be

broadly subjective and are prone to change over time, and thus, may not reflect an absolute predisposition on the item. Moreover, in a strictly quantized rating system such as a 5-star based system, the choices to show due appreciation for the items are very limited and many products can be identified by an identical rating while their ranking order is somewhat different in the mind of the user.

These central issues associated with recommendation systems can be partly mitigated by using one-on-one comparisons for item pairs. In this way, users can be more expressive in choosing between two items without putting much effort, rather than just providing independent opinions about one item without the presence of any arbitrary frame of reference. In such contexts, implicit feedback can also be utilized to recommend better items. On the other hand, user-user dependency for an item, which may be inferred from the same user's voting pattern, can also provide more information to the recommendation system. For example, it can indicate a stronger inclination of one group of users toward an item than another group, which is very common in practical scenarios.

In the literature, the problem of personalization has been approached from many different angles based on the context of the recommendation process. One conventional method utilized for recommendation systems is k-nearest neighbors (kNN) in order to perform item-item or user-user collaborative filtering employing the correlation between item or user pairs, respectively [9]. Popular methods like Matrix Factorization (MF) has also been considered in several studies [10, 11]. The problem of estimating the rating matrix via the MF method is especially relevant in the low-rank setting where the matrix can be represented as the product of two low-rank matrices that contain the latent features of the corresponding users and items, respectively. In a collaborative filtering context, [11] presents a novel approach to solve the non-convex problem of recovering the underlying low-rank structure of the rating matrix based on one-bit comparisons between pair of items using an alternating optimization algorithm. A novel method of personalized ranking system has been proposed in [8], in which the authors present a probabilistic argument based on a Bayesian analysis using item-item comparison data as a prior.

In this paper, we present a Bayesian framework to address

\* Corresponding author (e-mail: [aameri2@uic.edu](mailto:aameri2@uic.edu)). This work was supported in part by U.S. National Science Foundation Grant CCF-1704401.

the recommendation problem through a Comprehensive Personalized Ranking (CPR) system, which not only utilizes the one-bit item-item preference of a user, but also exploits the implicit inclination of different users towards an item. We provide a stochastic-gradient based approach to learn the system parameters. We also discuss how the aforementioned algorithm can be utilized to estimate the user/item latent features using MF. In fact, MF will be used to jump-start CPR.

The remainder of the paper is organized as follows. Section 2 presents the problem formulations along with a Bayesian treatment of the data. We also provide an MF based approach to estimate the low-rank rating matrix. Numerical results are presented in Section 3. Finally, Section 4 concludes the paper.

*Notation:* We use bold lowercase letters for vectors and uppercase letters for matrices.  $(\cdot)^T$  denotes the vector/matrix transpose.

## 2. PROBLEM FORMULATION

Let  $U$  denote the set of all users and  $I$  denote the set of all items. We let  $\Omega$  represent the internal system parameter which could be viewed as a user/item latent feature matrix. The users are asked to compare two items based on their best judgment at the time. From such set of queries, we can clearly infer the users' inclination toward a specific item. Such data can also be implicitly collected from user activity associated with an item, e.g. records of clicks on a link, or interest for a video content. The goal of the recommendation system is to provide effective personalized recommendations based on observed user-user and item-item comparison data, denoted by  $>_m \subset U^2$  and  $>_u \subset I^2$ , respectively, where  $>_m$  and  $>_u$  denote the observed user-user comparisons for item  $m$  and the item-item comparisons by user  $u$ , respectively. Note that the operator  $>_m$  has to meet the following three properties [8]:

$$\begin{aligned} \text{totality} : i \neq_m j &\implies i >_m j \vee j >_m i : \forall i, j \in U \\ \text{antisymmetry} : i >_m j \wedge j >_m i &\implies i =_m j : \forall i, j \in U \\ \text{transitivity} : i >_m j \wedge j >_m k &\implies i >_m k : \\ &\forall i, j, k \in U. \end{aligned}$$

Particularly, the transitivity property mentioned above is used to extract additional information from the comparison database. In a similar manner,  $>_u$  has to meet the same three properties as well.

In this paper, as mentioned earlier, instead of using the exact rating values, we use the user-user and item-item one-bit comparison data in order to design an efficient personalized ranking/recommendation system. In the following section, we present a Bayesian treatment of the data to solve the problem of Comprehensive Personalized Ranking (CPR) by using the likelihood function for  $\mathbb{P}\{i >_m j, k >_u l | \Omega\}$  and the prior probability of the system parameters  $\mathbb{P}\{\Omega\}$ .

### 2.1. Analysis of the Likelihood and Posterior Functions

Note that we are interested in the maximization of the following posterior probability  $\mathbb{P}\{\Omega | >_m, >_u\}$  where  $\Omega$  denotes the matrix of system parameters for all  $i, j \in U$  and  $k, l \in I$ :

$$\mathbb{P}\{\Omega | >_m, >_u\} = \alpha \cdot \mathbb{P}\{>_m, >_u | \Omega\} \mathbb{P}\{\Omega\}, \quad (1)$$

where  $\alpha$  is a constant term independent of  $\Omega$ . Here we assume that, when  $\Omega$  is given (equivalent to us having the rating matrix), not only the ordering of each pair of items becomes independent of rest of the orderings, but also no two users can any longer influence the votes of each other. Hence, the likelihood function,  $\mathbb{P}\{i >_m j, k >_u l | \Omega\}$  can not only be represented as a multiplication of two independent probability distributions, i.e.  $\mathbb{P}\{>_m, >_u | \Omega\} = \mathbb{P}\{>_m | \Omega\} \mathbb{P}\{>_u | \Omega\}$ , but also following the totality and antisymmetry property,

$$\begin{aligned} \mathbb{P}\{>_m | \Omega\} &= \prod_{(i,j) \in D_m} \mathbb{P}\{i >_m j | \Omega\}, \\ \mathbb{P}\{>_u | \Omega\} &= \prod_{(k,l) \in D_u} \mathbb{P}\{k >_u l | \Omega\}, \end{aligned} \quad (2)$$

hold true, for each independent item/user,  $m \in I$  and  $u \in U$ , where  $D_m$  and  $D_u$  are the set of users who rated item  $m$ , and the set of observed or rated items for user  $u$ , respectively.

Now, in order to form personalization based on users' own choices, we define the individual probability functions for the case where user  $u$  likes item  $k$  over item  $l$ , and in a similar fashion, user  $i$  has a stronger inclination toward item  $m$  than user  $j$ , based on the system parameters:

$$\begin{aligned} \mathbb{P}\{i >_m j | \Omega\} &\triangleq f(c_m, \hat{x}_{ijm}(\Omega)), \\ \mathbb{P}\{k >_u l | \Omega\} &\triangleq f(c_u, \hat{x}_{ukl}(\Omega)), \end{aligned} \quad (3)$$

where we define the estimators  $\hat{x}_{ijm}(\Omega)$  and  $\hat{x}_{ukl}(\Omega)$  as the difference between two individual estimators of the actual ratings:

$$\begin{aligned} \hat{x}_{ijm}(\Omega) &\triangleq \hat{x}_{im}(\Omega) - \hat{x}_{jm}(\Omega), \\ \hat{x}_{ukl}(\Omega) &\triangleq \hat{x}_{uk}(\Omega) - \hat{x}_{ul}(\Omega), \end{aligned} \quad (4)$$

and where  $\hat{x}_{pq}(\Omega)$ , for  $p \in U$  and  $q \in I$ , represents the estimate of the actual rating of user  $p$  for item  $q$ . According to item response theory [14], the function  $f(\cdot)$  is a mapping from the (comparative) rating space to a probability, defined as  $f(c, x) \triangleq \frac{1}{2} + \frac{1}{2} \tanh(cx)$ .

*Remark:* The constants  $c_m$  and  $c_u$  in (3) are the model specific parameters which determine the relevance of the query about item  $m$  and user  $u$ , respectively, and their values can be found empirically. The bigger the constant  $c$ , the steeper the function  $f(c, x)$  becomes around the origin, and thus, the more reliability we put on the estimated ratings. ■

In the following, we discuss how these user/item entity specific functions can be estimated for a given  $\Omega$  using well established methods such as matrix factorization (MF).

### 2.1.1. Derivation of the user/item entity specific functions $\hat{x}_{ijm}(\Omega)$ and $\hat{x}_{ukl}(\Omega)$ using MF

Note that the problem of estimating  $\hat{x}_{im}(\Omega)$  in (4), for all  $i \in U$  and all  $m \in I$ , is equivalent to estimating the rating matrix  $X \in \mathbb{R}^{|U| \times |I|}$ . This task has been successfully handled before by using the MF method, where the matrix  $X$  can be approximated by the product of two low-dimensional matrices  $P \in \mathbb{R}^{|U| \times r}$  and  $Q \in \mathbb{R}^{|I| \times r}$  as  $\hat{X} = PQ^T$ , and where  $r$  is the number of features that characterizes items and users, or in other words, the rank of the approximation [11]. Note that the corresponding rows of  $P$  and  $Q$ , denoted by  $\{\mathbf{p}_u^T\}_{u=1}^{|U|}$  and  $\{\mathbf{q}_m^T\}_{m=1}^{|I|}$ , can be used to constitute a modified system parameter matrix  $\Omega = [P^T|Q^T]$ , where the rating estimates are:

$$\hat{x}_{im} \triangleq \langle \mathbf{p}_i, \mathbf{q}_m \rangle = \mathbf{p}_i^T \mathbf{q}_m = \sum_{t=1}^r p_{it} q_{tm},$$

$$\hat{x}_{uk} \triangleq \langle \mathbf{p}_u, \mathbf{q}_k \rangle = \mathbf{p}_u^T \mathbf{q}_k = \sum_{t=1}^r p_{ut} q_{tk}.$$

It is now straightforward to verify that  $\hat{x}_{ijm} = (\mathbf{p}_i - \mathbf{p}_j)^T \mathbf{q}_m$  and  $\hat{x}_{ukl} = \mathbf{p}_u^T (\mathbf{q}_k - \mathbf{q}_l)$ . Another method to estimate these entity specific functions using a non-convex optimization method based on MF can be found in [11].

### 2.2. Analysis of the Prior Function

In this section, we introduce the general prior density function  $\mathbb{P}\{\Omega\}$  of the system parameters  $\Omega = [P^T|Q^T] = [\omega_1 \cdots \omega_N]$  as independent normalized zero-mean multivariate normal random variables with known covariance matrices  $\{\Sigma_n\}_{n=1}^N$  where  $N$  is the number of parameter vectors in  $\Omega$ . For  $\Omega$ , as described in subsection 2.1.1,  $N$  would be equal to  $|U| + |I|$ . Given these assumptions,  $\mathbb{P}\{\Omega\}$  can be formulated as,

$$\mathbb{P}\{\Omega\} = \frac{1}{(2\pi)^{\frac{N}{2}} \prod_n |\Sigma_n|^{\frac{1}{2}}} \exp \left\{ -\frac{1}{2} \sum_n \omega_n^T \Sigma_n^{-1} \omega_n \right\}. \quad (5)$$

### 2.3. Comprehensive Personalized Ranking (CPR)

With the above discussion in mind, we can rewrite our likelihood function in (2.1) as,

$$\mathbb{P}\{\>_m, >_u | \Omega\} = \prod_{m=1}^{|I|} \prod_{(i,j) \in D_m} f(c_m, \hat{x}_{ijm}(\Omega)) \times \prod_{u=1}^{|U|} \prod_{(k,l) \in D_u} f(c_u, \hat{x}_{ukl}(\Omega)).$$

Furthermore, we finally formulate our personalization problem in terms of maximizing the below criterion with respect to (w.r.t.)  $\Omega$ :

$$\begin{aligned} \text{CPR} &\triangleq \ln \mathbb{P}\{\Omega | >_m, >_u\} \\ &\simeq \ln \mathbb{P}\{\>_m, >_u | \Omega\} \mathbb{P}\{\Omega\} \\ &\simeq \sum_m \sum_{(i,j) \in D_m} \ln f(c_m, \hat{x}_{ijm}(\Omega)) \\ &\quad + \sum_u \sum_{(k,l) \in D_u} \ln f(c_u, \hat{x}_{ukl}(\Omega)) \\ &\quad - \frac{1}{2} \sum_n \omega_n^T \Sigma_n^{-1} \omega_n \end{aligned} \quad (6)$$

### 2.4. Learning the CPR

We begin by calculating the gradient of CPR w.r.t. the model parameter  $\Omega$ , denoted as  $\frac{\partial}{\partial \Omega} \text{CPR}$ , through considering each term at the right hand-side of (6) individually. It can be easily verified that,

$$\frac{\partial}{\partial \Omega} \ln f(c, \hat{x}) = c(1 - \tanh(c\hat{x})) \frac{\partial}{\partial \Omega} \hat{x}. \quad (7)$$

As one can observe from (6), we also need to know the gradients of  $\hat{x}_{ijm}(\Omega)$  and  $\hat{x}_{ukl}(\Omega)$ , w.r.t. each system parameter  $\omega = \{\omega_t\}_{t=1}^r$ , and can be given for (4) as,

$$\frac{\partial}{\partial \Omega} \hat{x}_{ijm} = \begin{cases} (p_{it} - p_{jt}), & \omega_t = q_{tm}, \\ q_{tm}, & \omega_t = p_{it}, \\ -q_{tm}, & \omega_t = p_{jt}, \\ 0, & \text{otherwise,} \end{cases} \quad (8)$$

and  $\frac{\partial}{\partial \Omega} \hat{x}_{ukl}$  can be derived in a similar fashion. The gradient of the last term in (6) can be derived as,

$$\frac{\partial}{\partial \Omega} \frac{1}{2} \sum_n \omega_n^T \Sigma_n^{-1} \omega_n = [\Sigma_1^{-1} \omega_1 \cdots \Sigma_N^{-1} \omega_N]. \quad (9)$$

Note that in a case where  $\{\Sigma_n\}_{n=1}^N = \Sigma$ , the latter expression can be simplified as  $\Sigma^{-1} \Omega$ .

In light of the above, the system parameter  $\Omega$  can be updated by employing a (stochastic) gradient descent method:

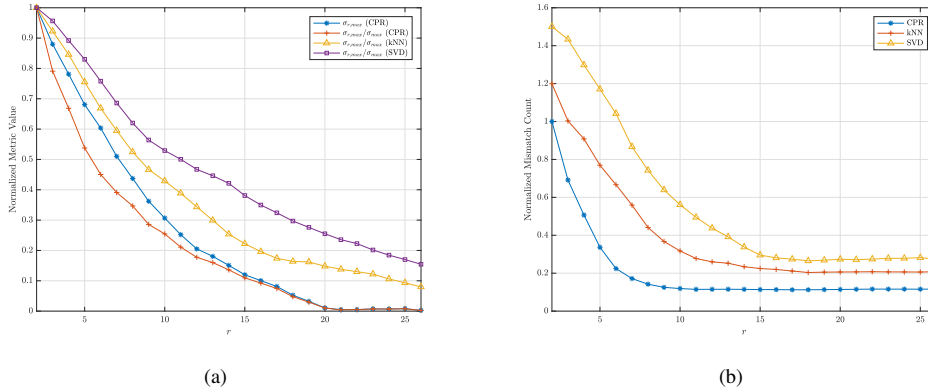
$$\Omega_{new} \leftarrow \Omega - \mu \frac{\partial}{\partial \Omega} \text{CPR}, \quad (10)$$

where  $\mu$  represents a proper positive learning rate.

## 3. NUMERICAL RESULTS

In this section, we show that the proposed method can efficiently recover the original low-rank rating matrix in terms of the comparison relationships through learning the underlying low-rank structure. As a byproduct, it offers the missing information in the rating matrix, which is the desired objective of a good recommendation system: recommending new items to users based on their own interests and the interests of other like-minded users. [15].

For simulations, we use a portion of the publicly available MovieLens datasets containing movie ratings—about 600 ratings given by 40 users judging 60 movies on a scale between



**Fig. 1.** The results for different algorithms: (a) normalized values of various metrics on the recovered rating matrices versus the expected rank  $r$ , (b) the normalized number of mismatches between the original comparison data and the comparisons made from the recovered data for *CPR*, *kNN* and *SVD*.

1 and 5 [16]. We start by converting the rating matrix to comparison data and these data are stored in a memory-efficient way. In order to train the system based on a large amount of observed data, we resort to the stochastic gradient descent method and mini-batch learning [17], which is proven to be robust enough to provide near optimal results asymptotically [18]. Accordingly, we utilize the stochastic gradient method with a batch size of 32 for simulation purposes.

According to the analysis in Section 2, it is clear that the proposed method relies on processing the data in an  $r$ -dimensional space. Additionally, as many users tend to show shared interest in only specific subsets of items, the rating matrix is, naturally, low-rank. In order to concur with this behavior of the rating matrix and to use the many advantages of low-rank matrices such as decrease in the computation and storage burdens, we resort to finding the best low-rank matrix approximation for the given data. A natural metric for determining the rank of the original rating matrix, denoted by  $r_X$ , can be to look at its  $r$  largest singular values of the recovered matrix. When the expected rank  $r$  is smaller than  $r_X$ , the algorithm cannot allocate all the information in an  $r$ -dimensional space. As a result, the updated  $(r + 1)$ -th dimension will have a considerable amount of information that cannot be ignored. Alternatively, when  $r$  is larger than  $r_X$ , the algorithm transcribes most of the recovered information in an  $r_X$ -dimensional space and places little to no information in the remaining  $r - r_X$  dimensions. Consequently, one can use the ratio of the  $r$ -th largest to the largest singular value of the recovered matrix as a metric to determine the true rank. According to the above analysis, the mentioned ratio should drop drastically as soon as  $r$  gets greater than  $r_X$ , enabling us to determine  $r_X$ .

In Fig. 1 (a), the average performance of the proposed method, denoted by *CPR*, is quantified w.r.t. (i) the  $r$ -th largest singular value of the recovered rating matrix, denoted by  $\sigma_{r,max}$ , and (ii) the ratio of  $r$ -th largest to the largest singular value of the recovered rating matrix, denoted by

$\sigma_{r,max}/\sigma_{max}$ , where all traces are normalized by their first elements. It can be seen that the knee point occurs at  $r = 20$ , which means that the best rank for the recovery is 20. Similar phenomenon can be noticed for  $\sigma_{r,max}/\sigma_{max}$ , as the added dimensions do not bear as much information as before, verifying the correctness our intuition. It can also be seen that the performance of *CPR*, in terms of finding the best low-rank matrix by allocating more information in smaller dimensions, is better than the other state of the art methods such as *kNN* and *SVD* [4, 9]. Fig. 1 (a) depicts that *CPR* stores the information in a rank-20 matrix while other methods still have information to be stored in the matrix of rank above 20.

Furthermore, in Fig. 1 (b) we compare the number of mismatches between the original comparison data and the comparisons made from the recovered data for *CPR*, *kNN*, and *SVD*, all normalized to the value of mismatches in rank-2 recovered matrix by *CPR*. It should be noted that the number of mismatches for the proposed method is less than that of other methods and remains virtually the same for  $r \geq 10$ . This further verifies superior performance of *CPR* as compared to other methods. Thus, it achieves the minimum number of mismatches among others while finding the best low-rank matrix approximation.

#### 4. CONCLUSION AND FUTURE WORK

In this paper, we studied a new optimization framework based on one-bit preference comparison data to develop a Comprehensive Personalized Ranking (CPR) system. The algorithm relies on a Bayesian treatment of the data, and maximizes the posterior probability of the system parameters. A learning model w.r.t. the aforementioned optimization problem is provided. Several numerical results were provided to show the effectiveness of the algorithm. The study of the impact of the rating matrix size on the projected rank would be an interesting future research avenue as the projected rank of a matrix significantly controls the storage and computational efficiency of the algorithm.

## 5. REFERENCES

- [1] A. S. Das, M. Datar, A. Garg, and S. Rajaram, "Google news personalization: Scalable online collaborative filtering," in *Proceedings of the 16th International Conference on World Wide Web*, New York, NY, USA, 2007, pp. 271–280, ACM.
- [2] G. Linden, B. Smith, and J. York, "Amazon.com recommendations: item-to-item collaborative filtering," *IEEE Internet Computing*, vol. 7, no. 1, pp. 76–80, Jan 2003.
- [3] A. Sharma and B. Yan, "Pairwise learning in recommendation: Experiments with community recommendation on linkedin," in *Proceedings of the 7th ACM Conference on Recommender Systems*, New York, NY, USA, 2013, pp. 193–200, ACM.
- [4] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions," *IEEE Transactions on Knowledge & Data Engineering*, , no. 6, pp. 734–749, 2005.
- [5] Q. Liu, H. Ma, E. Chen, and H. Xiong, "A survey of context-aware mobile recommendations," *International Journal of Information Technology & Decision Making*, vol. 12, no. 01, pp. 139–172, 2013.
- [6] W. Pan, H. Zhong, C. Xu, and Z. Ming, "Adaptive Bayesian personalized ranking for heterogeneous implicit feedbacks," *Knowledge-Based Systems*, vol. 73, pp. 173 – 180, 2015.
- [7] J. Bennett and S. Lanning, "The Netflix Prize," in *Proceedings of KDD Cup and Workshop 2007*, San Jose, CA, USA, Aug 2007.
- [8] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme, "BPR: Bayesian Personalized Ranking from implicit feedback," in *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, Arlington, Virginia, United States, 2009, UAI '09, pp. 452–461, AUAI Press.
- [9] M. Deshpande and G. Karypis, "Item-based top-N recommendation algorithms," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 143–177, Jan. 2004.
- [10] L. Schmidt-Thieme, "Compound classification models for recommender systems," in *Fifth IEEE International Conference on Data Mining (ICDM'05)*, Nov 2005.
- [11] A. Bose, A. Ameri, M. Klug, and M. Soltanalian, "Low-rank matrix recovery from one-bit comparison information," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4734–4738.
- [12] S. Khobahi and M. Soltanalian, "Signal recovery from 1-bit quantized noisy samples via adaptive thresholding," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018.
- [13] N. Naimipour and M. Soltanalian, "Graph clustering using one-bit comparison data," in *2018 52nd Asilomar Conference on Signals, Systems, and Computers*, Oct 2018.
- [14] S. E. Embretson and S. P. Reise, *Item response theory*, Psychology Press, 2013.
- [15] M. Deshpande and G. Karypis, "Item-based top-n recommendation algorithms," *ACM Transactions on Information Systems (TOIS)*, vol. 22, no. 1, pp. 143–177, 2004.
- [16] F. Maxwell Harper and Joseph A. Konstan, "The movielens datasets: History and context," *ACM Trans. Interact. Intell. Syst.*, vol. 5, no. 4, pp. 19:1–19:19, Dec. 2015.
- [17] G. Hinton, N. Srivastava, and K. Swersky, "Neural networks for machine learning lecture 6a overview of mini-batch gradient descent," *Cited on*, p. 14, 2012.
- [18] O. Shamir and T. Zhang, "Stochastic gradient descent for non-smooth optimization: Convergence results and optimal averaging schemes," in *International Conference on Machine Learning*, 2013, pp. 71–79.