

Singapore Management University

Institutional Knowledge at Singapore Management University

Research Collection School Of Computing and Information Systems

School of Computing and Information Systems

10-2016

Online adaptive passive-aggressive methods for non-negative matrix factorization and its applications

Chenghao LIU
Zhejiang University

HOI, Steven C. H.
Singapore Management University, choi@smu.edu.sg

Peilin ZHAO
Ant Financial

Jianling SUN
Zhejiang University

Ee-Peng LIM
Singapore Management University, eplim@smu.edu.sg

Follow this and additional works at: https://ink.library.smu.edu.sg/sis_research



Part of the [Databases and Information Systems Commons](#), and the [Theory and Algorithms Commons](#)

Citation

LIU, Chenghao; HOI, Steven C. H.; ZHAO, Peilin; SUN, Jianling; and LIM, Ee-Peng. Online adaptive passive-aggressive methods for non-negative matrix factorization and its applications. (2016). *CIKM 2016: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management: Indianapolis, October 24-28, 2016*. 1161-1170.

Available at: https://ink.library.smu.edu.sg/sis_research/3450

This Conference Proceeding Article is brought to you for free and open access by the School of Computing and Information Systems at Institutional Knowledge at Singapore Management University. It has been accepted for inclusion in Research Collection School Of Computing and Information Systems by an authorized administrator of Institutional Knowledge at Singapore Management University. For more information, please email cherylids@smu.edu.sg.

Online Adaptive Passive-Aggressive Methods for Non-Negative Matrix Factorization and Its Applications

Chenghao Liu^{1,2}, Steven C.H. Hoi², Peilin Zhao³, Jianling Sun¹, Ee-Peng Lim²

¹ School of Computer Science and Technology, Zhejiang University, China

² School of Information Systems, Singapore Management University, Singapore

³ Artificial intelligence department, Ant Financial, China

twinsken@zju.edu.cn, chhoi@smu.edu.sg, peilin.zpl@antfin.com, sunjl@zju.edu.cn, eplim@smu.edu.sg

ABSTRACT

This paper aims to investigate efficient and scalable machine learning algorithms for resolving Non-negative Matrix Factorization (NMF), which is important for many real-world applications, particularly for collaborative filtering and recommender systems. Unlike traditional batch learning methods, a recently proposed online learning technique named “NN-PA” tackles NMF by applying the popular Passive-Aggressive (PA) online learning, and found promising results. Despite its simplicity and high efficiency, NN-PA falls short in at least two critical limitations: (i) it only exploits the first-order information and thus may converge slowly especially at the beginning of online learning tasks; (ii) it is sensitive to some key parameters which are often difficult to be tuned manually, particularly in a practical online learning system. In this work, we present a novel family of online Adaptive Passive-Aggressive (APA) learning algorithms for NMF, named “NN-APA”, which overcomes two critical limitations of NN-PA by (i) exploiting second-order information to enhance PA in making more informative updates at each iteration; and (ii) achieving the parameter auto-selection by exploring the idea of online learning with expert advice in deciding the optimal combination of the key parameters in NMF. We theoretically analyze the regret bounds of the proposed method and show its advantage over the state-of-the-art NN-PA method, and further validate the efficacy and scalability of the proposed technique through an extensive set of experiments on a variety of large-scale real recommender systems datasets.

CCS Concepts

•Computing methodologies → Machine learning; Artificial intelligence;

Keywords

Non-Negative Matrix Factorization, Online Learning, Adaptive Regularization, Learning with Expert Advice

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16, October 24-28, 2016, Indianapolis, IN, USA

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2983323.2983786>

1. INTRODUCTION

Non-negative Matrix Factorization (NMF) [14] represents an important family of algorithms for matrix completion, which aim to reconstruct a partially observed matrix by factorizing the matrix as the product of two low-rank matrices and imposing non-negativity in the matrix factorization. NMF has achieved great successes in various real-world applications, ranging from recommender systems [24, 2], text mining [25], to feature selection [16]. We focus on the application of NMF techniques for matrix completion tasks in collaborative filtering and recommender systems, where matrix factorization based techniques are often shown as the most successful, according to many competitions, e.g., the Netflix challenge [13].

Traditional approaches usually formulate NMF as a *batch learning* task and solve it by applying different batch optimization techniques [15, 17, 18, 1, 9]. These methods might be good for solving certain tasks in some domain (e.g., image analysis), but often suffer from poor scalability when dealing with online recommender systems where the rating data usually arrives sequentially and thus it may suffer from expensive re-training cost due to their batch learning manner. Recently, online learning methods have been explored to tackle NMF. One state-of-the-art approach is the NN-PA method [2], which formulates NMF as an online learning task and resolves it by applying a popular online Passive-Aggressive (PA) algorithm [6]. Compared with traditional approaches, NN-PA solves NMF in an online learning fashion [11], which is thus more scalable, easier to implement, and does not require tuning the learning rate as often needed in traditional optimization approaches.

Despite the advantages and encouraging results, NN-PA falls short in two major critical limitations. First of all, it only exploits the first-order information in the online update process, and thus may converge slowly especially at the beginning of an online learning task. Second, it is sensitive to some key parameters (e.g., the regularization and matrix rank parameters), which are often difficult or even impossible to be tuned manually in the online learning process, particularly in practical online recommender systems. Our work is motivated to address these two limitations.

To this end, we propose a novel family of online Adaptive Passive-Aggressive learning algorithms for solving NMF, named “NN-APA” for short. Our new method overcomes the two limitations of NN-PA by exploring two ideas. First of all, unlike the first-order learning approach, we exploit second-order information in making more informative updates to enhance the efficacy of PA learning at each iteration. Second, we attempt to achieve the parameter auto-selection by exploring the idea of online learning with expert advice [8, 4] in deciding the optimal combination of the key parameters in NMF. We theoretically analyze the regret bounds of the proposed new method and shows its advantage over the state-of-the-art NN-PA method, and further validates the efficacy and

scalability of the proposed technique through an extensive set of experiments on a variety of large-scale real recommender systems datasets.

The rest of paper is organized as follows. Section 2 gives a formal formulation of NMF and then reviews the key ideas of the important related work of NN-PA for solving online NMF task. Section 3 presents the proposed new family of online NN-APA algorithms. Section 4 gives theoretical analysis of the proposed algorithms, and Section 5 discusses our empirical results. Section 6 briefly reviews related work and their differences to our work, and finally Section 7 concludes this work.

2. PROBLEM FORMULATION

In this section, we first gives a formal problem formulation of a Non-negative Matrix Factorization (NMF) task, and then review the important work of the Non-Negative Passive-Aggressive (NN-PA) method [2] for solving an online NMF task.

2.1 Problem Settings

Given a non-negative data matrix $R \in \mathbb{R}^{m \times n}$, with row index $\{i|i \in 1, \dots, m\}$ and column index $\{j|j \in 1, \dots, n\}$. Let us denote its entries as r_{ij} . The target of Non-negative Matrix Factorization is to find two non-negative matrices $U \in \mathbb{R}^{k \times m}$ and $V \in \mathbb{R}^{k \times n}$ whose product can well approximate the data matrix R , i.e.,

$$\min_{U, V} \frac{1}{2} \|R - U^T V\|_F^2, \quad s.t. U \in \mathbb{R}_+^{k \times m}, V \in \mathbb{R}_+^{k \times n}, \quad (1)$$

where $\|\cdot\|_F$ is the matrix Frobenius norm and k is a rank parameter and is an integer $k \ll \min\{m, n\}$. Although the objective function is convex when U or V is fixed, it is a non-convex optimization problem when considering both of them together. So it is hard to find global minimum. Alternatively, if one can iteratively optimize U and V [5, 12] until convergence to obtain a local minimum.

2.2 The NN-PA Method: Revisited

Passive-Aggressive (PA) [6] algorithms are originally designed for regular online classification and regression problems, which is easy to implement without any hand-tuning of learning rate parameter compared to other online learning algorithms [11]. Recently, PA techniques are extended to solve NMF problems, for which the proposed Non-Negative Passive-Aggressive methods (NN-PA) [2] have demonstrated their scalability when datasets are very large. Specifically, suppose entries of data matrix arrive sequentially and periodically, where entry r_{ij}^t is revealed at round t . We denote the i -th column of U as \mathbf{u}_i^t and j -th column of V as \mathbf{v}_j^t . Then, NN-PA methods alternatively update \mathbf{u}_i^t and \mathbf{v}_j^t while keeping another matrix fixed by solving the following optimization task:

$$\mathbf{u}_i^{t+1} = \arg \min_{\mathbf{u}_i \in \mathbb{R}_+^k} \frac{1}{2} \|\mathbf{u}_i - \mathbf{u}_i^t\|^2 \quad s.t. \quad \ell(\mathbf{u}_i, \mathbf{v}_j^t, r_{ij}^t) = 0, \quad (2)$$

$$\mathbf{v}_j^{t+1} = \arg \min_{\mathbf{v}_j \in \mathbb{R}_+^k} \frac{1}{2} \|\mathbf{v}_j - \mathbf{v}_j^t\|^2 \quad s.t. \quad \ell(\mathbf{u}_i^t, \mathbf{v}_j, r_{ij}^t) = 0, \quad (3)$$

where $\ell(\mathbf{u}, \mathbf{v}, r) = \max(|\mathbf{u}^T \mathbf{v} - r| - \epsilon, 0)$ is the ϵ insensitive loss. Take \mathbf{u}_i^{t+1} into consideration, intuitively speaking, \mathbf{u}_i^{t+1} is assigned to be the projection of previous \mathbf{u}_i^t onto the half-space of vectors which attain a ϵ -intensive loss of zero on the current \mathbf{v}_j^t . An update is passive if ϵ -intensive loss is zero and leave \mathbf{u}_i unchanged in order to retain information learned by previous iterations. In contrast, the algorithm will aggressively force \mathbf{u}_i to satisfy the constraint $\max(|\mathbf{u}_i^T \mathbf{v}_j^t - r_{ij}^t| - \epsilon, 0) = 0$, without any parameter settings about learning rate, due to correctly predicting current r_{ij}^t with a sufficiently high margin.

NN-PA method is similar to PA [6] method except that it has non-negative constraints on the model. It is easy to solve (2) if we

do not consider the non-negative constraints, since the solution of this optimization problem has a simple closed-form solution:

$$\mathbf{u}_i^{t+1} = \mathbf{u}_i^t + \tau_t \text{sgn}(r_{ij}^t - \mathbf{u}_i^t \cdot \mathbf{v}_j^t) \mathbf{v}_j^t, \quad \text{where } \tau_t = \frac{\ell(\mathbf{u}_i^t, \mathbf{v}_j^t, r_{ij}^t)}{\|\mathbf{v}_j^t\|^2}.$$

We can observe that the learning rate depends on the loss of the current model. Unfortunately, with the non-negative constraint, Eq. (2) does not enjoy a general closed-form solution anymore. To overcome this limitation, NN-PA methods provide both exact and approximate solutions. The approximate update solution first computes the closed-form solution without considering the non-negative constraints, and then projects the model into the non-negative subspace. Compared with the exact solution, approximate update is more efficient and performs comparably or even better[2].

Although NN-PA is simple and efficient, it suffers from two major drawbacks: (1) it converges relatively slowly due to the nature of first-order learning, and (2) its performance is sensitive to the settings of some key parameters, such as matrix rank k , and regularization parameter C , which are often difficult to choose or tune in advance prior to the learning tasks.

3. THE PROPOSED NN-APA METHOD

To overcome the limitations of NN-PA for a Non-negative Matrix Factorization (NMF) tasks, we first propose a new scheme of Non-Negative Adaptive Passive-Aggressive learning termed ‘‘NN-APA’’ by exploring the idea of adaptive regularization techniques [7], which attempts to adaptively set the time-varying proximal function for each feature by means of a data-driven way. This method relies on only first order information but has some properties of second order methods, so that it can achieve asymptotically sub-linear regrets. Second, to overcome the challenge of critical parameter selection, we then present a new scheme of Non-Negative Adaptive PA learning with multiple experts denoted as ‘‘NN-APA(m)’’, which combines multiple experts each has a combination of different parameter settings, where the weights of the experts are updated based on the Hedge algorithm [8]. In the following, we discuss the details of the proposed scheme for online NMF tasks.

3.1 Non-Negative Adaptive PA Learning

Our goal is to improve NN-PA by exploring the adaptive regularization [7]. This technique has gained extensive popularity for large-scale optimization problems and especially works well with sparse gradients. It adaptively sets the time-varying proximal function for each feature in a data-driven way to achieve asymptotically small regret. The intuition behind this updating strategy is fairly simple, i.e., rarely occurring features might be more informative and discriminative than those of frequently occurring features. It dynamically incorporates knowledge of the geometry of the data from earlier iterations and pre-emphasizes infrequently occurring features. Therefore, this adaption facilitates the utilization of informative but comparatively rare features and speeds up relative convergence.

Specifically, we incorporate historic geometric property by adjusting the proximal function in Eq. (2) to control the gradient step of the algorithm which keeps the current weight vector staying close to previous weight vector. Instead of original Euclidean distance, we can employ Mahalanobis distance $d_A(\mathbf{u}, \mathbf{u}') = \|\mathbf{u} - \mathbf{u}'\|_{G_t} = \sqrt{(\mathbf{u} - \mathbf{u}')^T G_t (\mathbf{u} - \mathbf{u}')}$ as the proximal function, where $G_t = \sqrt{\sum_{\gamma=0}^t (\mathbf{g}_\gamma^\top \mathbf{g}_\gamma)} + \delta I$ is the covariance matrix of accumulated gradient, \mathbf{g}_γ denotes a subgradient for loss function, and η_0 is a global learning rate shared by all dimensions. Besides the rare features, this accumulation of gradients gradually promotes

weight of proximal function which has the same effect of annealing for reducing the step size over time.

To this end, we propose the Non-Negative Adaptive Passive-Aggressive (NN-APA) method, which extends the NN-PA method in [2] by exploring the proximal function as a squared Mahalanobis norm where the matrix is a covariance matrix of accumulated gradients of online loss functions. Specifically, we formulate the online updating strategy as the following optimization:

$$\mathbf{u}_i^{t+1} = \arg \min_{\mathbf{u}_i \in \mathbb{R}_+^k} \frac{1}{2} \|\mathbf{u}_i - \mathbf{u}_i^t\|_{G_t}^2, \quad (\text{“NN-APA”}) \quad (4)$$

$$s.t. \quad \max(|\mathbf{u}_i^\top \mathbf{v}_j^t - r_{ij}^t| - \epsilon, 0) = 0.$$

Without the non-negative constraint, its Lagrangian is:

$$\mathcal{L}(\mathbf{u}_i, \theta, \mu) = \frac{1}{2} \|\mathbf{u}_i - \mathbf{u}_i^t\|_{G_t}^2 + \theta(-r_{ij}^t + \mathbf{u}_i^\top \mathbf{v}_j^t - \epsilon) + \mu(r_{ij}^t - \mathbf{u}_i^\top \mathbf{v}_j^t - \epsilon), \quad (5)$$

where $\theta \geq 0$ and $\mu \geq 0$ are two Lagrange multipliers. This optimization problem has a convex objective function and feasible affine constraints. These are sufficient conditions for Slater’s condition to hold. Therefore, satisfying the Karush-Kuhn-Tucker (KKT) conditions is a necessary and sufficient condition for finding the problem’s optimum. Let \mathbf{u}_i^* and (θ^*, μ^*) be the primal and dual optimal points. Setting the partial derivatives of $\mathcal{L}(\mathbf{u}_i, \theta, \mu)$ with respect to the elements of \mathbf{u}_i to zero gives:

$$\nabla_{\mathbf{u}_i} \mathcal{L}(\mathbf{u}_i, \theta, \mu) = G_t(\mathbf{u}_i - \mathbf{u}_i^t) + (\theta - \mu)\mathbf{v}_j^t = 0, \quad (6)$$

which can be further simplified as

$$\mathbf{u}_i^* = \mathbf{u}_i^t - (\theta^* - \mu^*)G_t^{-1}\mathbf{v}_j^t. \quad (7)$$

Plugging the above equation into the Eq. (5), taking the derivative of \mathcal{L} with respect to (θ, μ) and setting them to zero, we can get a closed-form solution:

$$\mathbf{u}_i^* = \mathbf{u}_i^t + \tau_t \text{sgn}(r_{ij}^t - \mathbf{u}_i^\top \mathbf{v}_j^t) G_t^{-1} \mathbf{v}_j^t$$

where $\tau_t = \frac{\ell(\mathbf{u}_i^t, \mathbf{v}_j^t, r_{ij}^t)}{\|\mathbf{v}_j^t\|_{G_t^{-1}}^2}$. (8)

After getting the above solution, we can project it into the non-negative subspace using,

$$\mathbf{u}_i^{t+1} = \Pi_{\mathbb{R}_+^k}^{G_t}(\mathbf{u}_i^*) = \arg \min_{\mathbf{u}_i \in \mathbb{R}_+^k} \|\mathbf{u}_i - \mathbf{u}_i^*\|_{G_t}^2. \quad (9)$$

This is a QP problem, which can be efficiently solved using the off-the-shelf tool box. Moreover, if the matrix G_t is diagonal, this optimization enjoys a closed form solution: $\mathbf{u}_i^{t+1} = \max(0, \mathbf{u}_i^*)$.

As discussed above, we can also introduce slacks variable ξ for NN-APA to avoid overfitting when dealing with noisy observations. Analogous to the PA variants, we propose NN-APA-I and NN-APA-II respectively with two types of loss as follows:

$$\mathbf{u}_i^{t+1} = \arg \min_{\mathbf{u}_i \in \mathbb{R}_+^k} \frac{1}{2} \|\mathbf{u}_i - \mathbf{u}_i^t\|_{G_t}^2 + C\xi, \quad (\text{“NN-APA-I”})$$

$$s.t. \quad \max(|\mathbf{u}_i^\top \mathbf{v}_j^t - r_{ij}^t| - \epsilon, 0) \leq \xi \quad \text{and} \quad \xi \geq 0,$$

$$\mathbf{u}_i^{t+1} = \arg \min_{\mathbf{u}_i \in \mathbb{R}_+^k} \frac{1}{2} \|\mathbf{u}_i - \mathbf{u}_i^t\|_{G_t}^2 + C\xi^2, \quad (\text{“NN-APA-II”})$$

$$s.t. \quad \max(|\mathbf{u}_i^\top \mathbf{v}_j^t - r_{ij}^t| - \epsilon, 0) \leq \xi,$$

Without considering the non-negative constraints, we can derive the closed-form solutions for the two algorithms as

$$\mathbf{u}_i^* = \max(\mathbf{u}_i^t + \tau_t \text{sgn}(r_{ij}^t - \mathbf{u}_i^\top \mathbf{v}_j^t) G_t^{-1} \mathbf{v}_j^t, 0) \quad (10)$$

where

$$\tau_t = \begin{cases} \min\left(C, \frac{\ell(\mathbf{u}_i^t, \mathbf{v}_j^t, r_{ij}^t)}{\|\mathbf{v}_j^t\|_{G_t^{-1}}^2}\right) & (\text{NN-APA-I}) \\ \frac{\ell(\mathbf{u}_i^t, \mathbf{v}_j^t, r_{ij}^t)}{\|\mathbf{v}_j^t\|_{G_t^{-1}}^2 + \frac{1}{2C}} & (\text{NN-APA-II}) \end{cases} \quad (11)$$

Finally, we project the above solutions onto the non-negative domain. Algorithm 1 summarizes the proposed NN-APA method.

Algorithm 1 Non-Negative Adaptive PA Algorithm “NN-APA”

Input: current iteration variable \mathbf{u}_i^t , input data \mathbf{v}_j^t , C for slack variable, ϵ for loss function, $\delta \geq 0$ for adaptive regularization
 Compute prediction $p_{ij}^t = \mathbf{u}_i^t \cdot \mathbf{v}_j^t$
 Receive an incoming rating instance r_{ij}^t
 Suffer a loss $\ell_t = \max(|p_{ij}^t - r_{ij}^t| - \epsilon, 0)$
if $\ell_t = 0$ **then**
 $\mathbf{u}_i^{t+1} = \mathbf{u}_i^t$
else
 Compute subgradient $\mathbf{g}_{\mathbf{u}_i} \in \partial_{\mathbf{u}_i} \ell(\mathbf{u}_i, \mathbf{v}_j, r_{ij})$
 $H_{\mathbf{u}_i} = H_{\mathbf{u}_i} + \mathbf{g}_{\mathbf{u}_i} \mathbf{g}_{\mathbf{u}_i}^\top$ (full matrix adaption)
 $H_{\mathbf{u}_i} = H_{\mathbf{u}_i} + \text{diag}(\mathbf{g}_{\mathbf{u}_i} \mathbf{g}_{\mathbf{u}_i}^\top)$ (diagonal matrix adaption)
 Compute $G_{\mathbf{u}_i}^t = \sqrt{\delta I + H_{\mathbf{u}_i}}$
 Compute τ_t by
 Eq. (8) for NN-APA
 Eq. (11) for NN-APA-I and NN-APA-II
 $\mathbf{u}_i^{t+1} = \Pi_{\mathbb{R}_+^k}^{G_t}[\mathbf{u}_i^t + \tau_t \text{sgn}(r_{ij}^t - \mathbf{u}_i^t \cdot \mathbf{v}_j^t) G_t^{-1} \mathbf{v}_j^t]$
end if
Output: \mathbf{u}_i^{t+1}

Remark. Computing the full matrix G_t for the adaptive regularization function could be computationally expensive, particularly when the rank k is large. To reduce computational cost, a common way is to explore diagonal matrix adaption, which only considers diagonal matrix of outer products of the gradients that have been observed to update the parameters. This strategy is significantly more efficient than the full matrix adaption, and enjoys the same worst-case time complexity as the existing NN-PA method.

3.2 NN-APA Learning with Multiple Experts

The goal of NMF is to seek a low-rank matrix that can be factored into two non-negative matrices $U \in \mathbb{R}_+^{k \times m}$, $V \in \mathbb{R}_+^{k \times n}$ with rank of at most k . NMF is sensitive to the setting of the rank parameter k . Specifically, if k is too small, the feature space is too restricted to represent the original matrix accurately, and thus suffers from underfitting. On the other hand, if k is too large, it may be over complex and thus suffers from overfitting. In addition, the proposed NN-APA method also could be sensitive to the setting of parameter C , which acts as a form of regularization and balances the trade-off between aggressiveness at each iteration. The parameter sensitivity of both k and C will be analyzed in our experiments.

Typically, in a batch NMF task, one can choose parameters such as k manually or tuned by grid search on training data via cross validation. Unfortunately, such an approach has some critical drawbacks for online learning tasks. First of all, data arrives sequentially in online learning, making it difficult to choose a good parameter prior to the learning task. Second, even if one manages some way to fix the parameter manually prior to the online learning tasks, the optimal values of parameters may change over time in the online learning process, which is common for many real-world online recommender systems where user preferences may change over time.

As a result, such kind of traditional approaches would suffer from sub-optimal performance for online applications.

To tackle the above challenge, we propose a new method of NN-APA learning with multiple experts termed “NN-APA(m)” for short. We create a set of experts \mathcal{S} with diverse combinations of parameter settings (e.g., rank parameter k and regularization parameter C in our experiments), and define it as $\mathcal{S} = \{(k_s, C_s), s = 1, \dots, |\mathcal{S}|\}$. The key idea of our approach is to approximate the original matrix R by learning a weighted combination of multiple experts without requiring manual parameter selection, i.e.,

$$\hat{R} = \sum_{s=1}^{|\mathcal{S}|} w_s^t (U_s^t)^\top V_s^t \quad (12)$$

where $w_s^t \geq 0$ denotes the importance weight of the expert s at the t -th iteration. The rest challenge then is how to learn w_s^t, U_s^t, V_s^t sequentially by an effective online learning scheme.

To tackle this problem, we apply the Hedge algorithm [8], a well-known technique for prediction with expert advice in decision-theoretic online learning [4, 21]. The intuition of our approach is to dynamically update the weight of each expert according to their online predictive performance in the online learning process.

Specifically, at the beginning, the weights for the experts w^0 are initialized as a uniform distribution, i.e., $w_s^0 = 1/|\mathcal{S}|, s = 1, \dots, |\mathcal{S}|$. At the end of each iteration, the weights are updated according to the instantaneous loss of each expert, i.e.,

$$\hat{w}_s^{t+1} = w_s^t \beta^{\ell_s^t}, s = 1, \dots, |\mathcal{S}| \quad (13)$$

where $\beta \in (0, 1)$ is a decaying learning rate to decrease importance with respect to the loss ℓ_s^t suffered at current iteration t , i.e.,

$$\ell_s^t = \ell(\mathbf{u}_{s,i}^t, \mathbf{v}_{s,j}^t, r_{ij}) = \max(|\mathbf{u}_{s,i}^t \top \mathbf{v}_{s,j}^t - r_{ij}| - \epsilon, 0) \quad (14)$$

In addition, to ensure the weight vector is a distribution, the weight vector is normalized at the end of each iteration as follows:

$$w_s^{t+1} = \hat{w}_s^{t+1} / \sum_{s=1}^{|\mathcal{S}|} \hat{w}_s^{t+1}. \quad (15)$$

Although the above scheme can achieve auto-selection of parameters in online learning, a drawback with this scheme is the extra computational cost as every expert must be updated at each iteration, which is expensive particularly when the number of experts $|\mathcal{S}|$ is large. To address this issue, we explore a stochastic updating scheme to reduce the computational cost. Specifically, we define a sampling probability denoted by \hat{q}_s^t , which determines the probability of an expert being selected for update:

$$\hat{q}_s^t = \frac{w_s^t}{\max_{1 \leq s \leq |\mathcal{S}|} w_s^t}. \quad (16)$$

This sampling probability implies that the higher the combination weight, the higher the probability of being selected for update at each iteration. To avoid potentially good experts with low weights at the beginning from completely losing out, we introduce a smoothing term $\rho \in (0, 1)$, so that the new probability of a matrices pair being selected for update becomes:

$$q_s^t = (1 - \rho) \frac{w_s^t}{\max_{1 \leq s \leq |\mathcal{S}|} w_s^t} + \rho, \quad (17)$$

which balances the trade-off between exploration and exploitation. Algorithm 2 summarizes the proposed “NN-APA(m)” algorithm.

Algorithm 2 The Multi-expert NN-APA algorithm “NN-APA(m)”

Input: update parameter β , parameter ρ , a set of $|\mathcal{S}|$ experts with different values of k and C

Initialization: $w_0 = \frac{1}{|\mathcal{S}|} \mathbf{1}$, initialize each (U_s, V_s) randomly

for $t = 1, \dots, T$ **do**

Receive an incoming rating instance r_{ij} ;

Compute prediction \hat{r}_{ij} by weighted combination in Eq. (12);

Compute sampling probability for each expert q_e^t by Eq. (17);

for $s = 1, \dots, \mathcal{S}$ **do**

Draw a bernoulli sampling $b_s \sim \text{Bernoulli}(q_s^t)$;

if $b_s == 1$ **then**

Compute loss $\ell(\mathbf{u}_{s,i}^t, \mathbf{v}_{s,j}^t, r_{ij})$ by (14);

Update matrix pair U_s and V_s as Algorithm 1;

end if

end for

Update w_{t+1} based on (13) and (15)

end for

4. REGRET ANALYSIS

We now analyze the theoretical performance of the proposed method in terms of online regret bound analysis. To ease our discussion, we simplify some notations in our analysis as follows:

$$\begin{aligned} \mathbf{x}_t &= \mathbf{v}_{j_t}^t \quad \text{or} \quad \mathbf{u}_{i_t}^t, \quad (\text{“input”}) \\ y_t &= r_{i_t, j_t} \quad (\text{“target”}) \\ \mathbf{w} &= \mathbf{u} \quad \text{or} \quad \mathbf{v} \quad (\text{“variable”}) \\ \mathbf{w}_{t+1} &= \mathbf{u}_{i_t}^{t+1} \quad \text{or} \quad \mathbf{v}_{j_t}^{t+1} \quad (\text{“solution”}) \\ \mathbf{w}_t &= \mathbf{u}_{i_t}^t \quad \text{or} \quad \mathbf{v}_{j_t}^t \quad (\text{“current status”}) \\ G_t &= G_{\mathbf{u}_{i_t}^t}^t \quad \text{or} \quad G_{\mathbf{v}_{j_t}^t}^t \quad (\text{“current status”}) \end{aligned}$$

Then, we can prove the following theorem to facilitate later proofs.

THEOREM 1. *Let $(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_T, y_T)$ be a sequence of input-target pairs, where $\mathbf{x}_t \in \mathbb{R}_+^k$ and $y_t \in \mathbb{R}_+$. Let $\mathbf{w}_1, \dots, \mathbf{w}_T$ be a sequence of vectors obtained by the proposed NN-APA-I algorithm, then the following inequality holds for any $\mathbf{w} \in \mathbb{R}_+^k$:*

$$\sum_{t=1}^T [\ell(\mathbf{w}_t, \mathbf{x}_t, y_t) - \ell(\mathbf{w}, \mathbf{x}_t, y_t)] \leq \frac{1}{2C} D^2 \text{tr}(G_T) + C \text{tr}(G_T), \quad (18)$$

where $D = \max_t \|\mathbf{w}_t - \mathbf{w}\|$. Moreover, taking $C = \frac{D}{\sqrt{2}}$, we have

$$\sum_{t=1}^T [\ell(\mathbf{w}_t, \mathbf{x}_t, y_t) - \ell(\mathbf{w}, \mathbf{x}_t, y_t)] \leq \sqrt{2} D \text{tr}(G_T). \quad (19)$$

The detailed proof can be found in Appendix A.

Remark. We now try to compare the regret bound of our NN-APA directly with the regret bound of NN-PA in [2]. To do so, we rewrite the regret bound of NN-PA as follows:

$$\sum_{t=1}^T [\ell(\mathbf{w}_t, \mathbf{x}_t, y_t) - \ell(\mathbf{w}, \mathbf{x}_t, y_t)] \leq \frac{\|\mathbf{w}\|_A^2}{2C} + \frac{C \sum_{t=1}^T \|\mathbf{g}_t\|_{A^{-1}}^2}{2},$$

where $A = I$, $\mathbf{g}_t = L_t \mathbf{x}_t$ and $L_t = \mathbb{I}(\ell(\mathbf{w}_t, \mathbf{x}_t, y_t) > 0)$. To compare this with our bound, we allow $A \succeq 0$, $\text{tr}(A) \leq 1$, and try to optimize the leading factor in the upper bound of NN-PA, i.e.,

$$\min_A \sum_{t=1}^T \|\mathbf{g}_t\|_{A^{-1}}^2, \text{ s.t. } A \succeq 0, \text{tr}(A) \leq 1,$$

As a result, we can find the optimal choice of $A = G_T/\text{tr}(G_T)$ and the best possible bound achieved by NN-PA as follows:

$$\begin{aligned} \sum_{t=1}^T [\ell(\mathbf{w}_t, \mathbf{x}_t, y_t) - \ell(\mathbf{w}, \mathbf{x}_t, y_t)] &\leq \frac{\|\mathbf{w}\|_{G_T/\text{tr}(G_T)}^2}{2C} + \frac{C[\text{tr}(G_T)^2]}{2} \\ &\leq \frac{\|\mathbf{w}\|^2}{2C} + \frac{C[\text{tr}(G_T)^2]}{2} \leq \|\mathbf{w}\| \text{tr}(G_T). \end{aligned}$$

It is important to note that this bound is the ‘‘ideally’’ optimal bound for NN-PA, but practically is not achievable because the parameter $A = I$ is used by default for NN-PA. Even if we allow one to use any PSD matrix, it is also impossible to choose the optimal A prior to the learning task since G_T is not known in advance. However, the proposed NN-APA enjoys $O(D\text{tr}(G_T))$, but does not require knowing G_T in advance. This implies that NN-APA is theoretically better than NN-PA, since it achieves the optimal regret bound (without considering the constant factor) while NN-PA does not.

According to the theory of the Hedge algorithm [8], we can show that our multi-expert method can achieve an optimal upper bound of regret by $\sqrt{T \ln |\mathcal{S}|/2}$ with $|\mathcal{S}|$ experts after T iterations. This implies that NN-APA(m) can asymptotically approach the best expert (i.e., the NN-APA with best parameter setting) and ensure the per-round regret vanishes over time in a sub-linear rate. The theoretical analysis can be followed directly from the result in [8]. We omit the detailed proof due to space limitation.

5. EXPERIMENTS

In this section, we evaluate the empirical performance of the proposed NN-APA method on real-world datasets. To examine every aspect of the proposed method, we have implemented several variants of the NN-APA algorithms, and compare with the state-of-the-art algorithms for online NMF tasks on a wide range of datasets collected from real-world recommender systems.

5.1 Experimental Testbed and Setup

5.1.1 Datasets

To comprehensively examine the empirical performance, we conduct the experiments of online NMF techniques for collaborative filtering on a variety of publicly available recommender systems datasets, ranging from small-scale datasets including MovieLens¹ and HetRec2011², to medium-scale datasets including Flixster³ and large-scale datasets with over one hundred million rating samples, including the popular Netflix⁴ and Yahoo-music⁵ datasets. Table 1 gives a summary of the dataset statistics used in our experiments.

Dataset	#Ratings	#Items	#Users	#density
HetRec 2011	855,598	10109	2113	4.0%
MovieLens 100k	100,000	1682	943	6.3%
MovieLens 1M	1,000,209	3900	6040	4.2%
Flixster	8,196,077	147,612	48,794	0.12%
MovieLens 10M	10,000,054	10,681	71,567	1.3%
Netflix	100,480,507	17770	480,189	1.18%
Yahoo-Music	115,579,440	98,213	1,948,882	0.06%

Table 1: Summary of datasets used in our experiments.

¹<http://grouplens.org/datasets/movielens/>

²<http://grouplens.org/datasets/hetrec-2011/>

³<http://www2.cs.sfu.ca/~sja25/personal/datasets/>

⁴<http://www.netflixprize.com/>

⁵<http://webscope.sandbox.yahoo.com/catalog.php?datatype=r>

5.1.2 Compared Methods and Performance Metrics

We compare the proposed algorithms and their variants with the state-of-the-art algorithms for NMF tasks as follows:

- **SGD**: Stochastic Gradient Descent (SGD), or equivalently Online GD, widely used in solving MF tasks for collaborative filtering [13]. We adapt SGD for solving NMF tasks by enforcing the NN constraint at the end of each iteration;
- **NN-PA**: the state-of-the-art Non-negative Passive Aggressive (NN-PA) learning algorithm proposed in [2];
- **OMTCF**: the Online Multi-Task Collaborate Filtering algorithm in [23];
- **NN-APA_{diag}**: the proposed NN-APA algorithm by only exploiting diagonal matrix adaptation;
- **NN-APA_{full}**: the proposed NN-APA algorithm by exploiting full matrix adaptation;
- **NN-APA(u)**: the proposed multi-expert NN-APA algorithm using a naive uniform combination of multiple experts, which allows us to examine the efficacy of the proposed online learning with expert advice;
- **NN-APA(m)**: the proposed multi-expert NN-APA algorithm using the Hedge algorithm for learning with expert advice.

Note that PA algorithms has three variants [6]. In our experiments, due to space limitation, we only evaluate the series of PA-II variants (i.e., NN-PA-II and NN-APA-II) which are highly comparable to the series of PA-I variants, but often better than the PA algorithm without soft margin. For performance metrics, we adopt the widely used ‘‘Mean Absolute Error (MAE)’’ and also measure time cost of each algorithm.

5.1.3 Experimental Setup and Parameter Settings

For experimental setup, each dataset is randomly divided into two parts: 80% for training and 20% for test. We repeat such a random permutation 10 times for each dataset and compute the average results of each algorithm over the 10 runs. For parameter settings, we adopt the same parameter tuning schemes for all the compared algorithm to enable fair comparisons. We perform grid search to choose the best parameters for each algorithm on the training set. Specifically, we search the ranges of values for parameter C in $[10^{-2}, 10]$, and for parameter δ in $[10^{-2}, 1]$. For the setup of the proposed NN-APA(m) algorithm with multiple experts, we create a set of 15 experts specifying by different combinations of parameters C and K and adopt the NN-APA_{diag} algorithm for each expert learning. More specifically, we create experts by setting $C \in \{0.01, 0.05, 0.4, 0.8, 5\}$ and $K \in \{5, 10, 15\}$ for small-scale datasets, and $C \in \{0.02, 0.1, 0.8, 1.6, 10\}$ and $K \in \{10, 15, 20\}$ for medium-scale datasets.

5.2 Experimental Results

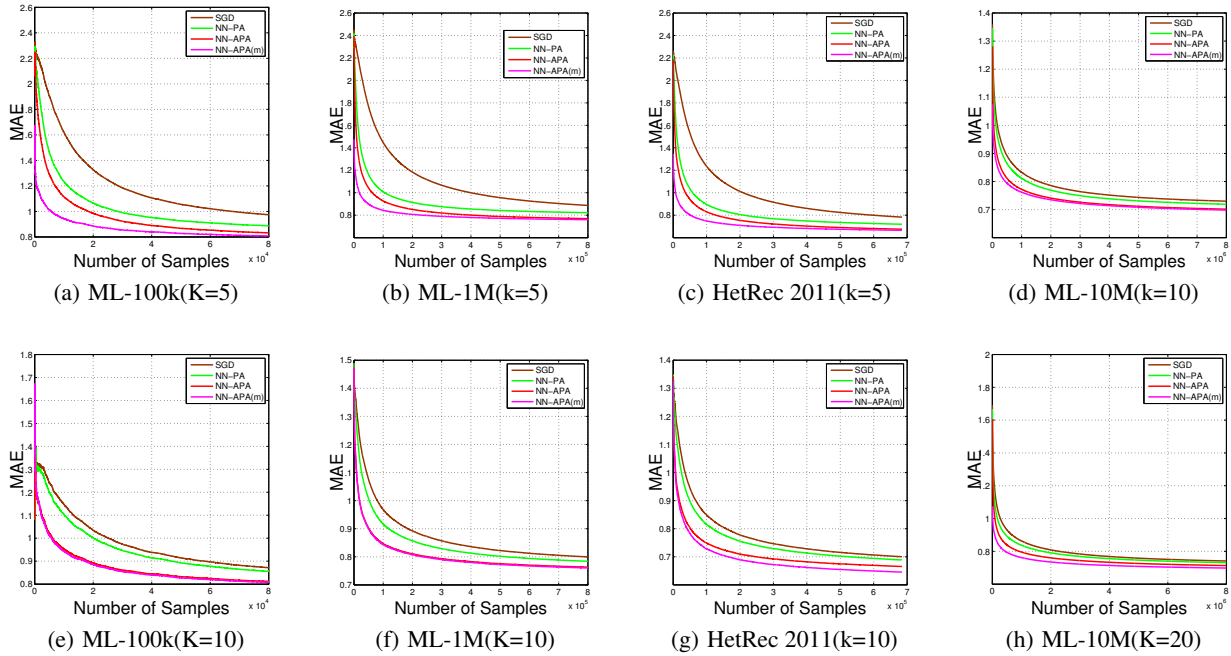
5.2.1 Evaluation of Recommendation Errors

Table 2 and Table 3 summarize the average MAE results of different algorithms for recommendation tasks on small-scale datasets and medium-scale datasets, respectively. From the results, we can draw several observations as follows.

First of all, by examining the MAE results, it is clear to see that the proposed NN-APA algorithms, NN-APA_{diag} and NN-APA_{full}, outperform both SGD, NN-PA and OMTCF significantly for all cases. This encouraging results validate the efficacy of the proposed adaptive learning technique in exploiting second-order information. By examining the time costs, NN-APA_{diag} runs slightly

Table 2: Evaluation of predictive errors of recommendation on small-scale datasets

ML100K	Training MAE			Test MAE			Time (s)		
	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15
SGD	0.9743 ± 0.0016	0.8891 ± 0.0021	0.8720 ± 0.0018	0.8348 ± 0.0026	0.7795 ± 0.0019	0.7961 ± 0.0021	4.44	4.56	4.53
NN-PA	0.8877 ± 0.0010	0.8440 ± 0.0010	0.8386 ± 0.0016	0.8199 ± 0.0011	0.7837 ± 0.0010	0.7961 ± 0.0012	6.19	6.64	6.88
OMTCF	1.0700 ± 0.0012	0.8721 ± 0.0008	0.8759 ± 0.0010	0.7954 ± 0.0007	0.7932 ± 0.0013	0.7960 ± 0.0008	5.71	5.82	6.02
NN-APA _{diag}	0.8319 ± 0.0006	0.8090 ± 0.0006	0.8065 ± 0.0009	0.7663 ± 0.0005	0.7706 ± 0.0004	0.7685 ± 0.0006	8.90	8.97	8.91
NN-APA _{full}	0.8505 ± 0.0009	0.8169 ± 0.0010	0.8173 ± 0.0006	0.7692 ± 0.0007	0.7689 ± 0.0008	0.7727 ± 0.0006	54.24	70.45	94.23
NN-APA(u)	0.8842 ± 0.0006			0.7779 ± 0.0010			64.40		
NN-APA(m)	0.8028 ± 0.0004			0.7681 ± 0.0008			16.39		
HetRec 2011	Training MAE			Test MAE			Time (s)		
	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15
SGD	0.7622 ± 0.0276	0.7176 ± 0.0245	0.6990 ± 0.0228	0.6872 ± 0.0226	0.6896 ± 0.0227	0.6627 ± 0.0201	35.70	37.46	38.99
NN-PA	0.7333 ± 0.0077	0.6877 ± 0.0083	0.6889 ± 0.0092	0.6490 ± 0.0084	0.6482 ± 0.0091	0.6509 ± 0.0079	45.13	50.31	53.99
OMTCF	0.7825 ± 0.0067	0.6990 ± 0.0062	0.6912 ± 0.0040	0.6619 ± 0.0072	0.6530 ± 0.0067	0.6539 ± 0.0038	40.71	43.82	48.02
NN-APA _{diag}	0.6781 ± 0.0051	0.6651 ± 0.0037	0.6710 ± 0.0032	0.6367 ± 0.0057	0.6370 ± 0.0046	0.6447 ± 0.0039	73.72	78.02	83.91
NN-APA _{full}	0.6769 ± 0.0044	0.6617 ± 0.0045	0.6606 ± 0.0022	0.6262 ± 0.0036	0.6259 ± 0.0046	0.6191 ± 0.0019	475.66	570.57	643.71
NN-APA(u)	0.6921 ± 0.0006			0.6363 ± 0.0008			555.69		
NN-APA(m)	0.6581 ± 0.0010			0.6324 ± 0.0007			170.94		
ML1M	Training MAE			Test MAE			Time (s)		
	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15	k = 5	k = 10	k = 15
SGD	0.8891 ± 0.0021	0.7997 ± 0.0022	0.8585 ± 0.0028	0.7610 ± 0.0021	0.7575 ± 0.0017	0.8040 ± 0.0021	41.85	43.07	45.13
NN-PA	0.8228 ± 0.0011	0.7841 ± 0.0011	0.8382 ± 0.0018	0.7867 ± 0.0019	0.7490 ± 0.0027	0.7925 ± 0.0022	58.82	60.07	68.31
OMTCF	0.8821 ± 0.0018	0.7987 ± 0.0014	0.7898 ± 0.0014	0.7641 ± 0.0017	0.7544 ± 0.0021	0.7569 ± 0.0018	48.97	53.02	60.91
NN-APA _{diag}	0.7677 ± 0.0013	0.7623 ± 0.008	0.7706 ± 0.0011	0.7347 ± 0.0023	0.7410 ± 0.0017	0.7480 ± 0.0012	91.32	97.62	103.31
NN-APA _{full}	0.7790 ± 0.0006	0.7694 ± 0.011	0.7661 ± 0.0011	0.7346 ± 0.0008	0.7351 ± 0.0005	0.7339 ± 0.0008	518.33	590.65	653.14
NN-APA(u)	0.7935 ± 0.0006			0.7406 ± 0.0010			691.01		
NN-APA(m)	0.7572 ± 0.0012			0.7339 ± 0.0008			191.91		


Figure 1: Evaluation of online cumulative MAE performance of different algorithms in the online learning process.

slowly than NN-PA but it is highly competitive to NN-PA with the same time complexity. However, the NN-APA_{full} using full matrix adaptation is much slower than the other algorithms, although it achieves the best MAE results for most cases. The high computation cost is because the update of the full matrix has to deal with

$O(K^2)$ number of parameters and the time cost of inverting the full matrix is very expensive, particularly for a large value of K .

Furthermore, by examining the two variants of the proposed NN-APA with multiple experts, we found that the NN-APA(m) using the Hedge algorithm significantly outperforms the NN-APA(u) us-

Table 3: Evaluation of predictive errors of recommendation on medium-scale datasets

ML10M	Training MAE		Test MAE		Time (s)	
	k = 10	k = 20	k = 10	k = 20	k = 10	k = 20
SGD	0.7300 ± 0.0011	0.7411 ± 0.0012	0.7138 ± 0.0013	0.7199 ± 0.0021	513.4880	601.5571
NN-PA	0.7191 ± 0.0005	0.7309 ± 0.0002	0.6967 ± 0.0003	0.7005 ± 0.0002	659.0901	722.4095
OMCTF	0.7283 ± 0.0014	0.7312 ± 0.0011	0.7179 ± 0.0010	0.7222 ± 0.0016	513.4880	601.5571
NN-APA _{diag}	0.7015 ± 0.0003	0.7133 ± 0.0003	0.6820 ± 0.0006	0.6886 ± 0.0001	967.6965	987.1562
NN-APA _{full}	0.6975 ± 0.0004	0.7076 ± 0.0002	0.6593 ± 0.0001	0.6502 ± 0.0008	7033.1093	10254.1272
NN-APA(u)	0.7214 ± 0.0006		0.6841 ± 0.0005		6623.92	
NN-APA(m)	0.6953 ± 0.0008		0.6803 ± 0.0008		2556.81	

Flixster	Training MAE		Test MAE		Time (s)	
	k = 10	k = 20	k = 10	k = 20	k = 10	k = 20
SGD	0.7526 ± 0.0046	0.7499 ± 0.0031	0.7180 ± 0.0036	0.7140 ± 0.0029	426.6991	440.8333
NN-PA	0.7458 ± 0.0021	0.7485 ± 0.0021	0.7112 ± 0.0018	0.7104 ± 0.019	460.3416	464.5119
OMCTF	0.7482 ± 0.0037	0.7467 ± 0.0042	0.7122 ± 0.0028	0.7081 ± 0.0031	426.6991	440.8333
NN-APA _{diag}	0.7255 ± 0.0011	0.7314 ± 0.0006	0.7041 ± 0.0015	0.7106 ± 0.0014	728.9019	788.9788
NN-APA _{full}	0.7268 ± 0.0009	0.7173 ± 0.0010	0.6882 ± 0.0005	0.6686 ± 0.0011	5691.3419	8621.1302
NN-APA(u)	0.7349 ± 0.0016		0.6978 ± 0.0007		5246.07	
NN-APA(m)	0.7055 ± 0.0011		0.6867 ± 0.0008		2373.21	

ing the naive uniform combination in terms of both MAE results and computational efficiency. The gain in MAE by NN-APA(m) is because it automatically gives more weights to good experts while NN-APA(u) treats all the experts the same and thus could be harmed by the poor experts. The gain in time cost is because NN-APA(m) does not necessarily update each expert during the online learning process due to its stochastic sampling strategy, while NN-APA(u) treats all experts equally and has to update each expert whenever the loss is nonzero. This encouraging result validates the efficacy of the Hedge algorithm for learning with expert advice and the importance of focusing the updates on good experts to save computational costs. By further comparing NN-APA(m) with the other algorithms without multiple experts (but their parameters were tuned by grid search), we found that the proposed NN-APA(m) is able to achieve highly competitive or even better results than the existing single-expert algorithms.

Finally, it is very important to note that NN-APA(m) does not require tuning the parameters manually, and is thus particularly suitable for online learning tasks.

5.2.2 Evaluation of Online Performance

To further examine the online learning performance of different algorithms in detail, Figure 1 shows some examples of online cumulative MAE performance of different algorithms in the online learning process.

From the results, we can see that the proposed NN-APA algorithms significantly outperform the other existing algorithms. This confirms our theoretical results of online regret analysis in that NN-APA can effectively exploit the underlying geometry of the data observed so far to achieve a more informative and effective update for online learning tasks. Last but not least, we found the NN-APA(m) with multiple experts outperforms the single-expert NN-APA for most cases. We conjecture that the reason is not only because NN-APA(m) is able to automatically identify the good expert with best parameters and but also has ensemble effect for boosting the performance using multiple good experts.

5.2.3 Evaluation of Training Efficiency

The previous experimental results in Table 2 and Table 3 indicate two facts: (i) the proposed NN-APA algorithm outperforms the existing SGD and NN-PA algorithms in terms of MAE after

processing a single pass of all rating samples; but (ii) NN-APA is slightly slow in terms of total time cost for a single pass. This raises a question, that is, if each algorithm is given the same amount of time for training, which algorithm is able to achieve better learning performance in terms of MAE results.

To answer the above question, Figure 2(e) shows our experimental results by comparing three different algorithms given the same training time on two large-scale datasets under different settings of K . From the experimental results, it is clear to see that the proposed NN-APA algorithm using diagonal matrix adaptation achieves the best results for all the cases. Besides, it is interesting to observe that the gain of our algorithm over the others becomes more significant when K is large. We think the reason is primarily because when using a large K , the existing first-order algorithms may be more risky and unreliable in finding the right direction for online updates. However, the proposed NN-APA algorithm can take advantage of exploiting the second order information in making a more precise and reliable update. This result is also consistent to our subsequent experiments of parameter sensitivity in that our NN-APA algorithm is much more robust in terms of parameter settings.

5.2.4 Evaluation of Parameter Sensitivity

For the proposed NN-APA algorithm, there are two key parameters: the rank parameter K and the regularization parameter C . It will be interesting to examine how the algorithm may be sensitive to the parameter settings. Figure 4 and Figure 3 show the results of parameter sensitive evaluations using different values of K and C .

First of all, by examining the influence of rank parameter K , we found all the algorithms are sensitive to the setting of parameter K , where K cannot be too small ("underfitting") or too large ("overfitting"). This result further confirms the importance and challenge of setting a proper value of parameter K in an online collaborative filtering task.

Second, by examining the influence of regularization parameter C , we found that the proposed NN-APA algorithm is relatively less sensitive to the setting of parameter C as compared to the other algorithms. This is primarily because of imposing the non-negative constraint in our formulation. To further validate this importance, we examine the impact of imposing non-negative constraints in the next experiment.

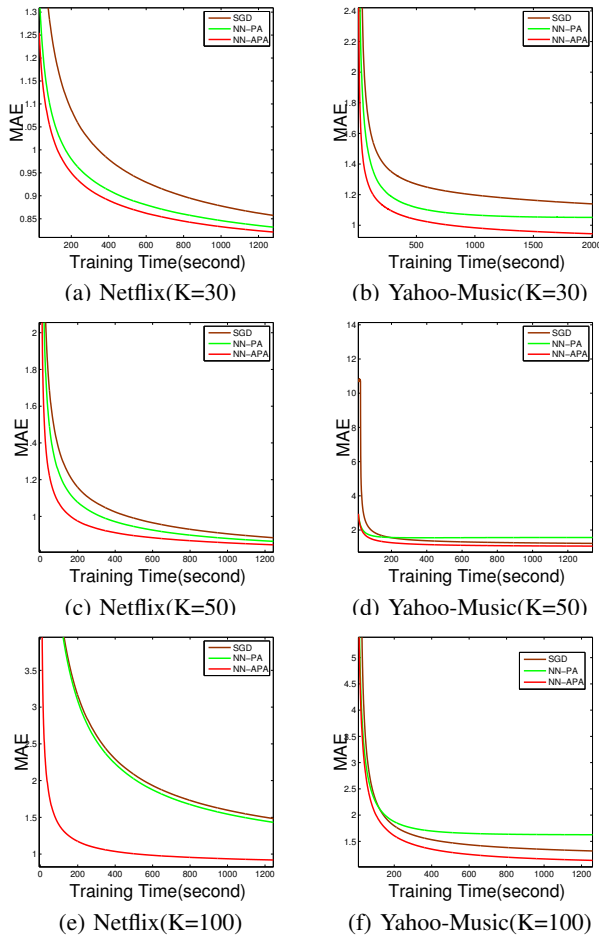


Figure 2: Cumulative MAE over time on large-scale datasets

5.2.5 Evaluation of Non-Negative Constraint Impact

To examine the importance of imposing the non-negative constraint in the proposed NN-APA algorithm, we implement a variant of Adaptive PA learning (APA) without imposing the non-negative constraint. As seen from the results in Figure 3, the NN-APA with non-negative constraint is much more robust than the APA without non-negative constraint, which validates the importance of non-negative matrix factorization techniques.

5.2.6 Interpretation of Qualitative NMF Results

Another important advantage of NMF is that it tends to yield more interpretable solutions. The recommendation system data contains tag information for each item and the coefficients in a basis of NMF results could be regarded as the relative importance. Thus, in this experiment, we attempt to examine the interpretation quality of our final NMF results with the tag of each item. We thus compute the matrix decomposition of the ML10M dataset and illustrate the top 5 coefficients of each basis with the tag information of each movie. Table 4 indicates that the related movies tend to be clustered into the same topic while some topics may contain irrelevant movies. For example, topic 1 contains mostly about “comedy”, topic 3 contains mostly about “drama”, but topic 5 is relatively less coherent as compared to the other topics. Therefore, in addition to obtaining better quantitative MAE results, NN-APA potentially

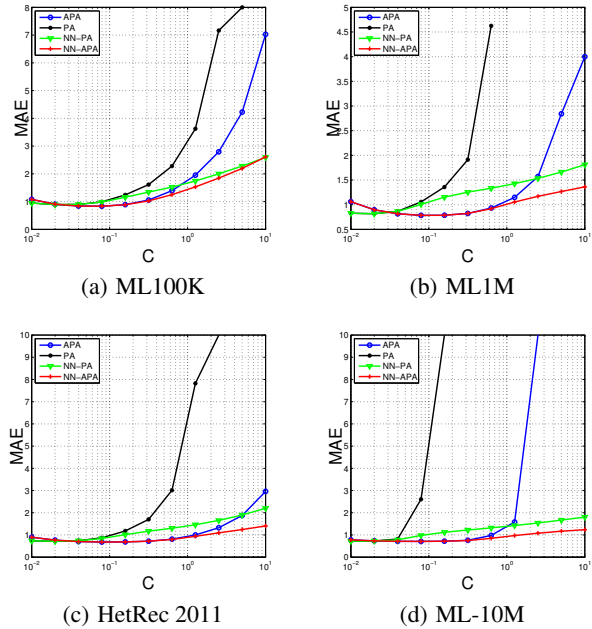


Figure 3: Evaluation of C and non-negative constraint impact

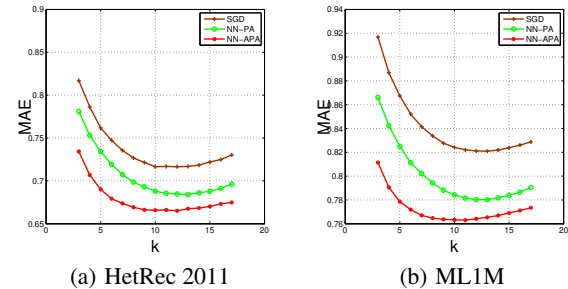


Figure 4: Prediction error when using different values of K

also yields interpretable qualitative results, which may gain important insights when being deployed in real-world applications.

6. RELATED WORK

Non-negative matrix factorization (NMF) has been a challenging open research problem in multivariate analysis and linear algebra as well as numerical optimization. NMF found many applications in different fields, such as computer vision, document clustering, audio signal processing, web search, recommender systems, and beyond. To resolve the optimization challenge of NMF problems, many optimization techniques have been extensively proposed over the past decade, including multiplicative methods [15, 17], projected gradient descent [18], active set methods [1], and MahNMF [9]. However, most of them are offline or batch learning methods, which suffer from a number of limitations when dealing with large-scale online applications where data may often arrive sequentially. The batch learning approaches are usually computationally intensive, of very high space complexity, and extremely expensive cost for re-training the models with new training data.

Recently, online learning methods have also been actively studied for solving collaborative filtering and NMF tasks [3, 20, 22, 23, 19], which enjoy high efficiency and scalability over batch learning methods. However, they often assume data is fully observed and

Table 4: Interpretation of the qualitative results. We extracted 6 components of item matrix from ML10M dataset and show the top 5 movies of each component. For comparison, we also show annotated tags of each movie.

Topic 1	Topic 2	Topic 3
Martin Lawrence Live: Runteldat (Comedy, Documentary)	Cinderella (Animation, Children)	Roller Boogie (Drama)
Clifford (Comedy)	Jonah: A VeggieTales Movie (Animation, Children, Musical)	Girl Who Leapt Through Time (Animation, Drama)
Even Cowgirls Get the Blues (Comedy, Romance)	Digimon: The Movie (Adventure, Animation, Children)	Cool as Ice (Drama)
Bratz (Comedy)	Spirited Away (Adventure, Animation, Children, Fantasy)	I Never Promised You a Rose Garden (Drama)
Who Pulled the Plug (Comedy)	Material Girls (Children, Comedy, Drama)	Nazar (Drama)
Topic 4	Topic 5	Topic 6
Slaughterhouse 2 (Horror)	Ape (Horror, Sci-Fi)	Cloverfield (Action, Mystery, Sci-Fi, Thriller)
Amityville Curse (Horror)	Carnosaur (Horror, Sci-Fi)	Omega Code (Action)
Howling III: The Marsupials (Comedy, Horror)	Night of the Comet (Comedy, Horror, Sci-Fi)	Cannonball Run III (Action, Comedy)
Baby (Horror)	Spirited Away (Adventure, Animation, Children, Fantasy)	No Holds Barred (Action)
Hollow Man II (Action, Horror, Sci-Fi, Thriller)	Material Girls (Children, Comedy, Drama)	Cobra (Action, Crime)

need to restart on the arrival of new data. One solution is to directly utilize state-of-the-art optimization methods in the recommendation system like SGD [13] with an additional non-negative constraint. The other one is to treat NMF as a stochastic optimization problem and updating the matrices in an incremental manner. For example, OR-NMF [10] utilizes the robust stochastic approximation, but is not directly applicable to solve online collaborative filtering tasks. Our work is closest to the state-of-the-art online NMF method "NN-PA" [2], which uses the popular PA online learning without requiring to hand-tune the learning rate, and can achieve $O(\sqrt{T})$ regret bound. Unlike the first-order NN-PA method, our NN-APA method improves the efficacy of NN-PA via a second-order online learning approach.

7. CONCLUSIONS

This paper presented NN-APA — a novel family of online learning algorithms for Non-negative Matrix Factorization (NMF) tasks, and explored the application of the proposed technique for resolving online collaborative filtering tasks from rating data arriving sequentially in a recommender system. The proposed NN-APA technique is able to overcome two critical limitations of the state-of-the-art NN-PA method by two ideas: (i) exploring the second-order information of underlying data in improving the slow convergence particularly at the beginning of online learning task, and (ii) exploring online learning with expert advice in avoiding tedious parameter selection and tuning during the online learning processes. We also analyzed the regret bound of the proposed method and showed that it is theoretically better than that of the NN-PA algorithm. Finally, our encouraging results from extensive experiments validated the efficacy of the proposed new technique towards real-world large-scale collaborative filtering and recommender systems.

Acknowledgments

This work was done when the first author visited Prof. Hoi's research group at Singapore Management University. This work was supported by Singapore Ministry of Education Academic Research Fund Tier 1 Grant (14-C220-SMU-016) and partially supported by

China Knowledge Centre for Engineering Sciences and Technology (No. CKCEST-2014-1-5).

Appendix: Proof of Theorem 1

PROOF. We simplify the notation by denoting $\ell_t(\mathbf{w}) = \ell_t(\mathbf{w}, \mathbf{x}_t, y_t)$. Then, we can show that the update of NN-APA-I is the same as

$$\mathbf{w}_{t+1} = \arg \min_{\mathbf{w}} P(\mathbf{w}) = \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|_{G_t}^2 + C\ell_t(\mathbf{w}),$$

where $\ell_t(\mathbf{w}) = \ell(\mathbf{w}, \mathbf{x}_t, y_t)$. Since \mathbf{w}_{t+1} minimizes $P(\mathbf{w})$, and $P(\mathbf{w})$ is strongly convex with respect to $\|\cdot\|_{G_t}$, we have

$$P(\mathbf{w}) \geq P(\mathbf{w}_{t+1}) + \nabla P(\mathbf{w}_{t+1})^\top (\mathbf{w} - \mathbf{w}_{t+1}) + \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t+1}\|_{G_t}^2.$$

Because $\nabla P(\mathbf{w}_{t+1}) = 0$, we have

$$\begin{aligned} & \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|_{G_t}^2 + C\ell_t(\mathbf{w}) \\ & \geq \frac{1}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_{G_t}^2 + C\ell_t(\mathbf{w}_{t+1}) + \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t+1}\|_{G_t}^2. \end{aligned}$$

Re-arranging the above inequality gives

$$\begin{aligned} & C\ell_t(\mathbf{w}_{t+1}) - C\ell_t(\mathbf{w}) \\ & \leq \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|_{G_t}^2 - \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t+1}\|_{G_t}^2 - \frac{1}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_{G_t}^2. \end{aligned}$$

In addition, since ℓ_t is $\|\mathbf{x}_t\|_{G_t^{-1}}$ -Lipschitz w.r.t. $\|\cdot\|_{G_t}$, we have

$$\begin{aligned} \ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w}_{t+1}) & \leq \|L_t \mathbf{x}_t\|_{G_t^{-1}} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{G_t} \\ & = \|\mathbf{g}_t\|_{G_t^{-1}} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{G_t}, \end{aligned}$$

where $L_t = \mathbb{I}(\ell_t(\mathbf{w}_t) > 0)$, and $\mathbf{g}_t = \nabla_{\mathbf{w}} \ell_t(\mathbf{w}_t)$. Combining the above two inequalities, we get

$$\begin{aligned} C\ell_t(\mathbf{w}_t) - C\ell_t(\mathbf{w}) & \leq \frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|_{G_t}^2 - \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t+1}\|_{G_t}^2 \\ & \quad - \frac{1}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_{G_t}^2 + C\|\mathbf{g}_t\|_{G_t^{-1}} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{G_t}. \end{aligned}$$

Summing the above inequality over $t = 1, \dots, T$, we have

$$\sum_{t=1}^T C[\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})] \leq \sum_{t=1}^T \left[\frac{1}{2} \|\mathbf{w} - \mathbf{w}_t\|_{G_t}^2 - \frac{1}{2} \|\mathbf{w} - \mathbf{w}_{t+1}\|_{G_t}^2 - \frac{1}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_{G_t}^2 + C \|\mathbf{g}_t\|_{G_t^{-1}} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{G_t} \right].$$

We can bound each of the terms respectively as follows

$$\begin{aligned} & \sum_{t=1}^T [\|\mathbf{w}_t - \mathbf{w}\|_{G_t}^2 - \|\mathbf{w}_{t+1} - \mathbf{w}\|_{G_t}^2] \\ & \leq \|\mathbf{w}_1 - \mathbf{w}\|_{G_1}^2 + \sum_{t=2}^T [\|\mathbf{w}_t - \mathbf{w}\|_{G_t}^2 - \|\mathbf{w}_t - \mathbf{w}\|_{G_{t-1}}^2] \\ & = \|\mathbf{w}_1 - \mathbf{w}\|_{G_1}^2 + \sum_{t=2}^T [\|\mathbf{w}_t - \mathbf{w}\|_{(G_t - G_{t-1})}^2] \\ & \leq \|\mathbf{w}_1 - \mathbf{w}\|^2 \text{tr}(G_1) + \sum_{t=1}^T \text{tr}(G_t - G_{t-1}) \|\mathbf{w}_t - \mathbf{w}\|^2 \\ & \leq D^2 \text{tr}(G_T), \quad \text{and} \\ & \sum_{t=1}^T \left[-\frac{1}{2} \|\mathbf{w}_{t+1} - \mathbf{w}_t\|_{G_t}^2 + C \|\mathbf{g}_t\|_{G_t^{-1}} \|\mathbf{w}_t - \mathbf{w}_{t+1}\|_{G_t} \right] \\ & \leq \frac{C^2}{2} \sum_{t=1}^T \|\mathbf{g}_t\|_{G_t^{-1}}^2 \leq C^2 \text{tr}(G_T), \end{aligned}$$

where we used $-a^2/2 + ab \leq b^2/2$ for the first inequality and Lemma 10 of the paper [7]. In summary, we have

$$\sum_{t=1}^T C[\ell_t(\mathbf{w}_t) - \ell_t(\mathbf{w})] \leq \frac{1}{2} D^2 \text{tr}(G_T) + C^2 \text{tr}(G_T)$$

Re-arranging the above inequality concludes the proof. \square

8. REFERENCES

- [1] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons. Algorithms and applications for approximate nonnegative matrix factorization. *Computational statistics & data analysis*, 52(1):155–173, 2007.
- [2] M. Blondel, Y. Kubo, and N. Ueda. Online passive-aggressive algorithms for non-negative matrix factorization and completion. In *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, pages 96–104, 2014.
- [3] B. Cao, D. Shen, J.-T. Sun, X. Wang, Q. Yang, and Z. Chen. Detect and track latent factors with online nonnegative matrix factorization. In *IJCAI*, volume 7, pages 2689–2694, 2007.
- [4] N. Cesa-Bianchi and G. Lugosi. *Prediction, learning, and games*. Cambridge University Press, 2006.
- [5] A. Cichocki and P. Anh-Huy. Fast local algorithms for large scale nonnegative matrix and tensor factorizations. *IEICE transactions on fundamentals of electronics, communications and computer sciences*, 92(3):708–721, 2009.
- [6] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. *The Journal of Machine Learning Research*, 7:551–585, 2006.
- [7] J. Duchi, E. Hazan, and Y. Singer. Adaptive subgradient methods for online learning and stochastic optimization. *The Journal of Machine Learning Research*, 12:2121–2159, 2011.
- [8] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. In *Computational learning theory*, pages 23–37. Springer, 1995.
- [9] N. Guan, D. Tao, Z. Luo, and J. Shave-Taylor. Mahnmf: Manhattan non-negative matrix factorization. *arXiv preprint arXiv:1207.3438*, 2012.
- [10] N. Guan, D. Tao, Z. Luo, and B. Yuan. Online nonnegative matrix factorization with robust stochastic approximation. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(7):1087–1099, 2012.
- [11] S. C. Hoi, J. Wang, and P. Zhao. Libol: A library for online learning algorithms. *The Journal of Machine Learning Research*, 15(1):495–499, 2014.
- [12] C.-J. Hsieh and I. S. Dhillon. Fast coordinate descent methods with variable selection for non-negative matrix factorization. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1064–1072. ACM, 2011.
- [13] Y. Koren, R. Bell, and C. Volinsky. Matrix factorization techniques for recommender systems. *Computer*, (8):30–37, 2009.
- [14] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.
- [15] D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *Advances in neural information processing systems*, pages 556–562, 2001.
- [16] J. Li, X. Hu, L. Wu, and H. Liu. Robust unsupervised feature selection on networked data. *SDM*, 2016.
- [17] C.-J. Lin. On the convergence of multiplicative update algorithms for nonnegative matrix factorization. *Neural Networks, IEEE Transactions on*, 18(6):1589–1596, 2007.
- [18] C.-J. Lin. Projected gradient methods for nonnegative matrix factorization. *Neural computation*, 19(10):2756–2779, 2007.
- [19] J. Lu, S. C. Hoi, J. Wang, and P. Zhao. Second order online collaborative filtering. *JMLR Workshop and Conference Proceedings*, 29:325–340, 2013.
- [20] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online learning for matrix factorization and sparse coding. *The Journal of Machine Learning Research*, 11:19–60, 2010.
- [21] V. G. Vovk. A game of prediction with expert advice. In *Proceedings of the eighth annual conference on Computational learning theory*, pages 51–60. ACM, 1995.
- [22] F. Wang, P. Li, and A. C. König. Efficient document clustering via online nonnegative matrix factorizations. In *SDM*, volume 11, pages 908–919. SIAM, 2011.
- [23] J. Wang, S. C. Hoi, P. Zhao, and Z.-Y. Liu. Online multi-task collaborative filtering for on-the-fly recommender systems. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 237–244. ACM, 2013.
- [24] X. Wang, R. Donaldson, C. Nell, P. Gorniak, M. Ester, and J. Bu. Recommending groups to users using user-group engagement and time-dependent matrix factorization. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.
- [25] X. Yan, J. Guo, S. Liu, X. Cheng, and Y. Wang. Learning topics in short texts by non-negative matrix factorization on term correlation matrix. In *Proceedings of the SIAM International Conference on Data Mining*, 2013.