A Quasi-Newton Matrix Factorization-Based Model for Recommendation

Shiyun Shao, University of Montreal, Canada

Yunni Xia, Chongqing University, China

Kaifeng Bai, Shanxi Siji Technology Co., Ltd., China*

Xiaoxin Zhou, University of Montreal, Canada

ABSTRACT

Solving large-scale non-convex optimization problems is the fundamental challenge in the development of matrix factorization (MF)-based recommender systems. Unfortunately, employing conventional first-order optimization approaches proves to be an arduous endeavor since their curves are very complex. The exploration of second-order optimization methods holds great promise. They are more powerful because they consider the curvature of the optimization problem, which is captured by the second-order derivatives of the objective function. However, a significant obstacle arises when directly applying Hessian-based approaches: their computational demands are often prohibitively high. Therefore, the authors propose AdaGO, a novel quasi-Newton method-based optimizer to meet the specific requirements of large-scale non-convex optimization problems. AdaGO can strike a balance between computational efficiency and optimization performance. In the comparative studies with state-of-the-art MF-based models, AdaGO demonstrates its superiority by achieving higher prediction accuracy.

KEYWORDS

Matrix Factorization, Non-Convex Optimization, Quasi-Newton Method, Recommender System

1. INTRODUCTION

In the rapid development of the Internet, information explosion has manifested as a thorny challenge that touches nearly every facet of our digital lives. The sheer volume of data generated, shared, and stored online has reached staggering proportions, making the efficient extraction of meaningful knowledges and insights from this deluge of information an essential endeavor (Adomavicius & Tuzhilin, 2005; Kluver et al., 2018; Pan et al., 2019). This challenge is particularly pressing for large internet platforms including e-commerce, social media, and other online service providers. Within the digital realm, the ability to harness the vast reservoirs of historical data for effective decision-making is a topic of paramount importance. As these platforms strive to meet the needs and expectations of their users, the key to success often lies in comprehending and responding to

DOI: 10.4018/IJWSR.334703

*Corresponding Author

This article published as an Open Access article distributed under the terms of the Creative Commons Attribution License (http://creativecommons.org/licenses/by/4.0/) which permits unrestricted use, distribution, and production in any medium, provided the author of the original work and original publication source are properly credited.

individual preferences. Users' preferences like hidden gems, nestled within their historical records, interactions, and behaviors. Consequently, recommender systems have emerged as transformative tools. By adeptly sifting through, mining and interpreting historical and real data, recommender systems have powerful ability to decipher the intricate tapestry of user preferences, distilling exclusive and complex patterns and trends.

Matrix Factorization (MF)-based models have not only demonstrated their effectiveness but have also changed the way we extract meaningful insights from interconnected data (Lo et al., 2017; Yu et al., 2019). This paradigm has proven instrumental in various domains, from personalized product recommendations in e-commerce to content suggestions in streaming services. The crux of them lies in their ability to distill complex relationships and latent features from raw historical interaction data. This characteristic makes MF-based models as a cornerstone in building collaborative filtering (CF)-based recommender systems.

In the large-scale internet platforms, the number of users and items often reaches astronomical proportions. When dealing with such colossal datasets, the dimensionality of the interaction matrix can skyrocket, potentially including millions of users and items. Paradoxically, while the platform hosts really diverse choices, each individual user typically engages with only a fraction of the available items. It leads to the practical realization that the corresponding interaction matrix in practice is often extremely sparse which poses a substantial challenge in the field of recommender systems.

The current progress of recommender systems research and development has unveiled the remarkable effectiveness and scalability of matrix-factorization (MF)-based models across a multitude of application cases (Adomavicius & Kwon, 2011; Adomavicius & Tuzhilin, 2005; Koren et al., 2021; Luo et al., 2014; Zhang et al., 2006). The fundamental concept of MF-based models revolves around the construction of a low-rank approximation to the original rating matrix. The core idea of MF-based modeling is by mapping both items and users into a shared latent feature space (Luo et al., 2015). Within this space, user and item features are learned through training on existing ratings, enabling the model to generate the predictions for unknown ratings. It heavily depends on the inner products of related user-item feature-vector pairs. One of the outstanding advantages of MF-based models is that they are suitable for extremely sparse situations, a common occurrence in real-world scenarios. Due to this sparsity, the rating matrix typically exhibits a low-rank property, which means that the dimension of the latent feature space can be kept relatively low without compromising prediction accuracy.

The non-negative matrix-factorization (NMF) algorithm is initially proposed (Lee & Seung, 2000; Lee & Seung, 1999) and subsequently applied in the fields of computer vision (Berry et al., 2007; Ding et al., 2008) and collaborative filtering problem (CF) (Gu et al., 2010). Zhang et al. (2006) propose two variations of the MF-based models, namely, Expectation-Maximization (EM) procedure and Weighted Non-negative Matrix Factorization (WNMF). The former assumes that the unknown entries obey a normal distribution and estimates their values based on such an assumption. The latter employs a weight parameter, which is set to one if the current entry is not empty and zero otherwise. Luo *et al.* propose two NMF-based models later. The regularized single-element-based NMF (RSNMF) model proposed in Luo et al. (2014) bases on the idea of single element dependence and the Alternating direction method-based Non-negative Latent Factor analysis (ANLF)-based model proposed in Luo et al. (2015) incorporates the principle of the alternating direction method (ADM) to implement efficient extraction of NLFs for analyzing high-dimensional and sparse data. Liu et al. (2023) design a Bi-regularized Non-negative Matrix Factorization (B-NMF) method, incorporating symmetry and graph regularization, to enhance the availability of learning representation by expanding the latent factor space. NMF even received various application in the clinical (Akçay et al., 2022; Sweeney et al., 2023), genetic (Seo et al., 2022; Wu et al., 2023), autonomous vehicle (Seo et al., 2022; Wu et al., 2023), environmental (Cao et al., 2023; Muñoz-Montoro et al., 2023) and financial (Farzadnia & Vanani, 2023) domain.

However, note that analyzing Sparse Interaction Matrix (SIM) with MF-based model is a non-convex problem (Seo et al., 2022). When addressing non-convex problems, convergence towards a second-order stationary point is more reliable than convergence towards a first-order one. But a glaring reality stands out---most existing Matrix Factorization (MF)-based models, despite their proven effectiveness, have traditionally relied on first-order optimization methods. The popularity of first-order optimization in MF-based models can be attributed to its computational advantages. Techniques like stochastic gradient descent (SGD) have become reliable methods due to their scalability and speed. These methods incrementally update model parameters based on gradient information, making them well-suited for large-scale applications. Unfortunately, their reliance on first-order information leaves them susceptible to slow convergence and the potential to get stuck in local minima, particularly in highly non-convex scenarios. Second-order optimization methods offer a more reliable path to solutions in non-convex problems. Hence, it is vital to prompt a second-order method for SIM representation (Li et al., 2020; Li et al., 2022).

The most conventional method for second-order minimization is Newton's method. The essence of Newton's method lies in its feature to refine the optimization process by leveraging not only first-order information, captured through gradients, but also second-order information, encapsulated within the Hessian matrix. The practical application of second-order methods, such as Newton's methods, often relies on their proficiency in manipulating the Hessian matrix (Zhang et al., 2006), which constitutes a critical yet potentially challenging component. In order to minimize a function $f(X)$, a second-order method seeks for the basic Newton step-an increment ΔX as the following linear relationship,

$$H_f(X)\Delta X + \nabla f(X) = 0, \tag{1}$$

Usually, rule 1 will be solved as the following form:

$$\Delta X = -H_f\left(X\right)^{-1}\nabla f\left(X\right), \tag{2}$$

where $\nabla f(X)$ and $H_f(X)$ denote $f(X)$'s gradient and Hessian matrix, respectively. Thus, the computation of full Hessian inverse is required per iteration with the computational cost at $\Theta((|X|\times|X|)^3)$ (Luo et al., 2014) by calculating $H_f(X)^{-1}$ and $\Theta((|X|\times|X|)^2)$, which is unaffordable for practical applications with thousands or millions of variables.

To address the aforementioned issue, we propose a novel model called AdaGo, which belongs to Adaptive Gauss-Newton Diagonal Optimizer. This model aims to improve the optimization of Matrix Factorization (MF) by incorporating a second-order MF-based approach.

The main idea of AdaGo is to leverage the information contained in the Gauss-Newton matrix and utilize its main diagonal along with an exponential diagonal. Then, we can effectively capture the second-order curvature of the MF model and optimize it for better parameter estimation.

The utilization of the main diagonal of the Gauss-Newton matrix allows us to focus on capturing important features that contribute significantly to improving the performance of MF models. Additionally, by incorporating an exponential diagonal, we are able to adaptively adjust our optimization process based on different levels of curvature present in each parameter.

This adaptive nature of AdaGo makes it particularly effective in handling complex datasets where traditional first-order optimization methods may struggle. By considering both first and second-order information simultaneously, our proposed model offers a more comprehensive understanding of data patterns and enables more accurate parameter estimation.

Experimental results on real world datasets indicate that the proposed model not only effectively represents the SIM (Spatial Information Model), but also does so at an affordable cost. The strong ability of this model to capture and represent spatial information has been demonstrated through rigorous testing and analysis. This enables it to provide a comprehensive representation of the underlying SIM, allowing for more accurate analysis and decision-making in various domains such as urban planning, transportation management, environmental monitoring, and disaster response.

The contributions of this study include:

1) A second-order MF-based model. The optimization process for improving Matrix Factorization (MF) models takes an intriguing twist by harnessing the potency of the Gauss-Newton matrix's main diagonal alongside an exponential diagonal. This dynamic approach seeks to encapsulate crucial second-order curvature information, ultimately refining the model's parameters to enhance its performance.
2) Details and analysis of AdaGo algorithm are provided.
3) Experiments are conducted on two distinct datasets stemming from real-world applications.

The remainder of this survey is organized as follow. Section 2 gives the preliminaries. Section 3 presents the more details and analysis of this proposed model. Section 4 states the experiment settings, datasets and analyses the results. Finally, section 5 concludes and summarize the challenges and the method and give a brief conclusion of this paper.

2. PRELIMINARIES

2.1 Sparse Interaction Matrix

The rating about a user's preference on an item is stored in the corresponding position of the source data, e.g., an user-item interaction matrix ***E***. Suppose there is an item set ***I*** and a user set ***J***, *E* is a $|\boldsymbol{I}|\times|\boldsymbol{J}|$ matrix where each element $e_{i,j}$ is a rating and proportional to user *i*'s preference on item *j*. Recall that any user involved in a platform can only grade a limited number of items, hence, ***E*** is a Sparse Interaction Matrix (SIM). Let ***K*** denote the known set of elements in ***E***, we have $|\boldsymbol{K}|<<|\boldsymbol{I}|\times|\boldsymbol{J}|$.

2.2 Matrix Factorization

Given an SIM $\mathbf{E}^{|I|\times|J|}$, it can be analyzed factorizing the it into low dimensional parameter matrices, e.g., $P^{|I|\times R}$ and $Q^{|J|\times R}$, where *R* is much smaller than either $|\boldsymbol{I}|$ or $|\boldsymbol{J}|$. This process is presented as Figure 1. Parameter matrices bring key features of source SIM matrix, and they can work collaboratively to generate a rank-*R* approximation, e.g., $\hat{\mathbf{E}}$, to ***E*** as follows:

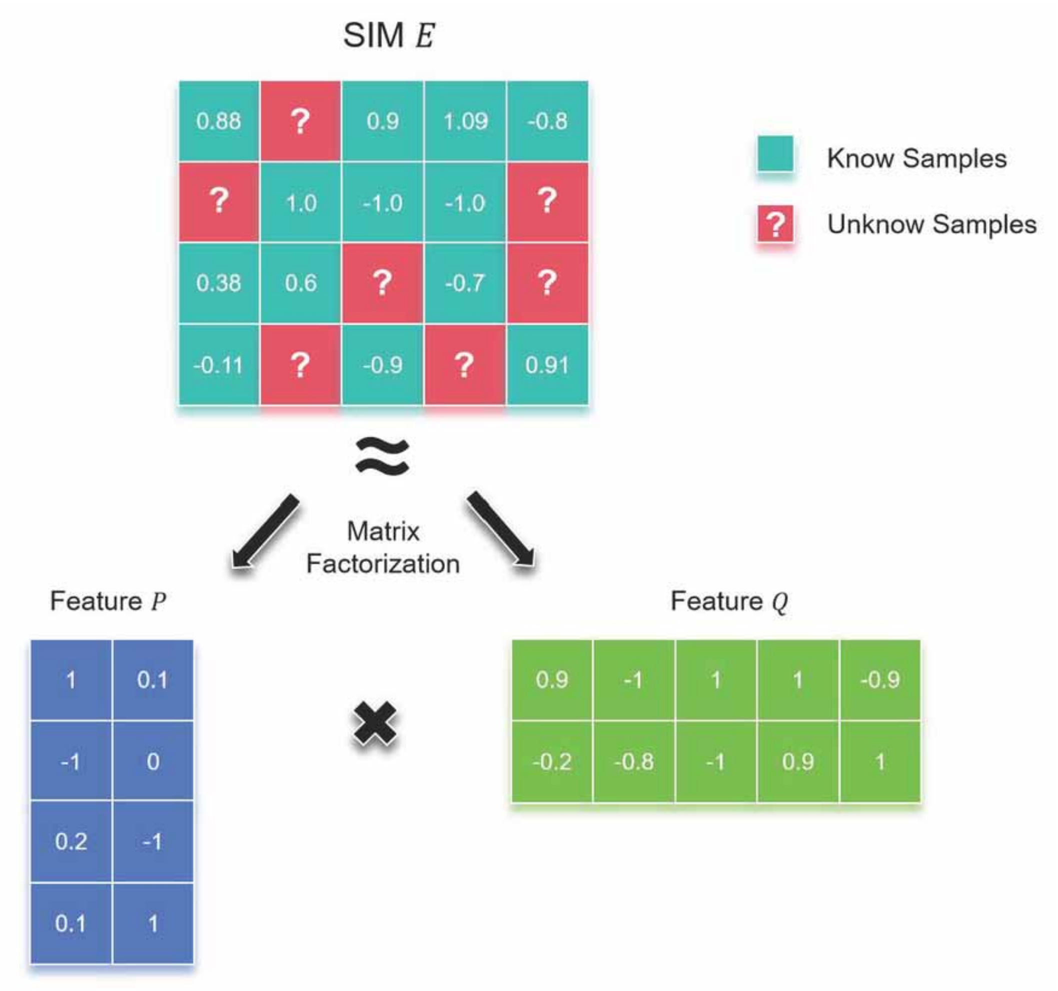
$$\hat{E} = PQ^{T} \rightarrow \hat{e}_{i,j} = \sum_{r=1}^{R} p_{i,r} q_{j,r}, \tag{3}$$

where $\hat{e}_{i,j}$ denotes the generated approximation of $e_{i,j}$ in $\hat{\boldsymbol{E}}$. $p_{i,r}$ and $q_{j,r}$ are specified entries in ***P*** and ***Q***, respectively.

The key to obtain an accurate approximation is to minimize the difference, *e.g.*, $L(\boldsymbol{P},\boldsymbol{Q})$, between $\hat{\boldsymbol{E}}$ and ***E***. Therefore, the optimum of each $p_{i,r}\in\boldsymbol{P}$ and $q_{j,r}\in\boldsymbol{Q}$ is required. Least-squared error is a good choice to measure the difference:

$$\underset{P,Q}{\arg\min}\, L\left(P,Q\right) = \frac{1}{2}\sum_{e_{i,j}\in K}\left(e_{i,j} - \sum_{r=1}^{R} p_{i,r} q_{j,r}\right)^{2} \tag{4}$$

Figure 1. The illustration that sparse interaction matrix (SIM) *E* can be analyzed factorizing the it into low dimensional parameter matrices *P* and *Q*



0.88	?	0.9	1.09	-0.8
?	1.0	-1.0	-1.0	?
0.38	0.6	?	-0.7	?
-0.11	?	-0.9	?	0.91

1	0.1
-1	0
0.2	-1
0.1	1

0.9	-1	1	1	-0.9
-0.2	-0.8	-1	0.9	1

Moreover, as did in Li et al. (2020, 2022), ***P*** and ***Q*** can be factorized to vectors by row and stored in a uniform vector, *e.g.*, ***y***, for inferencing *L*'s Hessian matrix more concisely. Hence, we have following definition:

Given two matrices, e.g., $\boldsymbol{P}^{|\boldsymbol{I}|\times R}$ and $Q^{|\boldsymbol{J}|\times R}$, ***Y*** is a ((|***J***|+|***I***|)×*R*)-dimension vector consists of (|***J***|+|***I***|) *R*-dimensional sub-vectors. A sub-vector $\boldsymbol{y}_{(i)}$ in ***y*** denotes ***P***'s *i*-th row vector for ∀*i*∈***I*** and $\boldsymbol{y}_{(j)}$ are ***Q***'s *j*-th row vector for ∀*j*∈***J***. Thus, $\hat{e}_{i,j}$ can be obtained as:

$$\hat{e}_{i,j}=\sum_{r=1}^{R} y_{i,r} y_{j,r} \tag{5}$$

By combining (3) and (2), we have:

$$\arg\min_{\boldsymbol{y}} L\left(\boldsymbol{x}\right) = \frac{1}{2}\sum_{e_{i,j}\in \boldsymbol{K}}\left(e_{i,j} - \sum_{r=1}^{R} y_{i,r}y_{j,r}\right)^2 \tag{6}$$

2.3 AdaHessian

Adahessian belongs to a kind of Quasi-Newton Methods. It is proposed by Yao in 2021 (Yao et al., 2021). Adahessian Using random linear algebra to estimate the Hessian diagonal, and embedding the estimated Hessian matrix main diagonal vector into the Adam framework for gradient preprocessing, as shown in (7):

$$\Delta \boldsymbol{y}^t = -\operatorname{diag}\left(\mathbf{H}_L\left(\boldsymbol{y}^t\right)\right)^{-1}\nabla L\left(\boldsymbol{y}^t\right) \tag{7}$$

where *diag(**A**)* is the main diagonal vector of matrix ***A***. The update rule of AdaHessian is as follows:

$$\begin{cases} \boldsymbol{d} = \boldsymbol{v}^{\mathrm{T}}\mathbf{H}_L\left(\boldsymbol{y}^t\right)\boldsymbol{v} \\ \boldsymbol{m}^{t+1} = \beta\boldsymbol{m}^t + \left(1-\beta\right)\boldsymbol{g}_L\left(\boldsymbol{y}^t\right) \\ \boldsymbol{s}^{t+1} = \omega\boldsymbol{s}^t + \left(1-\omega\right)\boldsymbol{d}\odot\boldsymbol{d} \\ \hat{\boldsymbol{m}}^{t+1} = \dfrac{\boldsymbol{m}^{t+1}}{1-\beta^{t+1}} \\ \hat{\boldsymbol{s}}^{t+1} = \dfrac{\boldsymbol{s}^{t+1}}{1-\omega^{t+1}} \\ \boldsymbol{y}^{t+1} = \boldsymbol{y}^t - \left(\sqrt{\boldsymbol{s}^{t+1}}\right)^{-1}\boldsymbol{m}^{t+1} \end{cases} \tag{8}$$

In the update rule (8), vector ***d*** is the estimated Hessian matrix diagonal vector, ***m*** is the first-order momentum vector, ***s*** is the second-order momentum vector, $\hat{\boldsymbol{m}}$ is the first-order momentum after exponential average shift, and $\hat{\boldsymbol{s}}$ is the second-order momentum after exponential average shift, *β* is the first-order momentum decay rate hyper-parameter, *ω* is a second order momentum decay rate hyper-parameter. AdaHessian offers a significant advantage by providing partial second-order information, which allows for the preprocessing of gradients to determine better descent directions. This approach eliminates the need for calculating and storing Hessian matrices, making it computationally efficient and memory-friendly.

2.4 Notations

In this paper, we denote the required symbols for conducting MF w.r.t the SIM, as described in Table 1.

3. PROPOSED METHODS

Note that, second-order optimization usually solves the following Linear system:

$$\mathbf{H}_L(\boldsymbol{y}^t)\Delta\boldsymbol{y}^t + \nabla L(\boldsymbol{y}^t) = 0 \tag{9}$$

Table 1. Adopted symbols and their descriptions

Symbol	Description
$\mathbf{E}^{\lvert \boldsymbol{I}\rvert \times \lvert \boldsymbol{J}\rvert}$	Target three dimensional tensors
$\hat{\mathbf{E}}^{\lvert \boldsymbol{I}\rvert \times \lvert \boldsymbol{J}\rvert}$	Approximation tensor to $\boldsymbol{E}$
$\boldsymbol{I}, \boldsymbol{J}$	Sets of horizontal and vertical coordinates and event frames
$e_{ijn}, \hat{e}_{ijn}$	Single elements in $\boldsymbol{E}$ and $\hat{\boldsymbol{E}}$, respectively
$\boldsymbol{K}$	Known set of $\boldsymbol{E}$
L	Difference between $\boldsymbol{E}$ and $\hat{\boldsymbol{E}}$
R	Rank of $\hat{\boldsymbol{E}}$; also denotes the dimension of the parameter space
$\boldsymbol{Y}$	parameter matrices of objective matrix
$\boldsymbol{y}_i, \boldsymbol{y}_j$	Vectors in $\boldsymbol{Y}$
$\boldsymbol{y}_{i,r}, \boldsymbol{y}_{j,r}$	Single elements in $\boldsymbol{Y}_{Ii}$, $\boldsymbol{Y}_{Jj}$, respectively
$\mathbf{G}_L(y)$	Gauss-Newton matrix of $L(y)$
$\mathbf{H}_L(y)$	Hessian matrix of $L(y)$
$\mathbf{J}_L(y)$	Jaocibian matrix of $L(y)$
$\nabla L(y)$	Gradient of $L(y)$
$\mathbb{R}$	Real number domain
Γ	Unknown set of $\boldsymbol{E}$
$\lvert\cdot\rvert_{\text{abs}}$	Absolute value of an input

where $\boldsymbol{y}\in\mathbb{R}^{(\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R}$ is decision parameter vector. $\mathbf{H}_L(\boldsymbol{y})\in\mathbb{R}^{((\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R)\times(\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R)}$ and $\nabla L(y)\in\mathbb{R}^{(\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R}$ are Hessian matrix and first-order gradient of the objective function L respectively. With the Newton approach, $\Delta\boldsymbol{y}$ is formulated as

$$\Delta \boldsymbol{y}^t = -\mathbf{H}_L(\boldsymbol{y}^t)^{-1}\nabla L(\boldsymbol{y}^t) \tag{10}$$

where $\Delta\boldsymbol{y}\in\mathbb{R}^{(\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R}$ denotes updated increment of decision vector $\boldsymbol{y}$ by employing Newton method in each turn to inverse the operation of computing Hessian matrix.

When employing second-order optimization, $O(((\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R)^2)$ and $O(((\lvert \boldsymbol{I}\rvert+\lvert \boldsymbol{J}\rvert)\times R)^3)$ space and computation overhead are required. Note that the objective function L is non-convex. Hence its Hessian matrix may be irreversible. Thus, we adopt Gauss-Newton matrix, a semi-positive matrix to approximate to Hessian matrix:

$$\begin{aligned}\mathbf{H}_L(\boldsymbol{y}) &\approx \mathbf{G}_L(\boldsymbol{y}) \\ &= \mathbf{J}_L\left(\boldsymbol{y}\right)^{\mathrm{T}}\mathbf{J}_L\left(\boldsymbol{y}\right)\end{aligned} \tag{11}$$

where $\mathbf{G}_L(\boldsymbol{y})$ is Gauss-Newton matrix and $\mathbf{J}_L(\boldsymbol{y})$ is the Jaocibian matrix of L. However, we cannot fully believe in the approximation $\mathbf{G}_L(\boldsymbol{y})$ to $\mathbf{H}_L(\boldsymbol{y})$. Adding damping term, α, to $\boldsymbol{A}$ is a feasible method:

$$\begin{aligned}\mathbf{H}_L\left(\boldsymbol{y}\right) &\approx \left(\mathbf{G}_L\left(\boldsymbol{y}\right)+\alpha\mathbf{I}\right)\\ &=\left(\mathbf{J}_L\left(\boldsymbol{y}\right)\mathbf{J}_L\left(\boldsymbol{y}\right)+\alpha\mathbf{I}\right)\end{aligned} \tag{12}$$

Thus, (10) and (12) are reformulated into:

$$\begin{aligned}&\left(\mathbf{G}_L\left(\boldsymbol{y}^t\right)+\alpha\mathbf{I}\right)\Delta\boldsymbol{y}^t+\nabla L\left(\boldsymbol{y}^t\right)\\ &=\left(\mathbf{J}_L\left(\boldsymbol{y}^t\right)^{\mathrm{T}}\mathbf{J}_L\left(\boldsymbol{y}^t\right)+\alpha\mathbf{I}\right)\Delta\boldsymbol{y}^t+\nabla L\left(\boldsymbol{y}^t\right)=0\\ &\Rightarrow\Delta\boldsymbol{y}^t=-\left(\mathbf{J}_L\left(\boldsymbol{y}^t\right)^{\mathrm{T}}\mathbf{J}_L\left(\boldsymbol{y}^t\right)+\alpha\mathbf{I}\right)^{-1}\nabla L\left(\boldsymbol{y}^t\right)\end{aligned} \tag{13}$$

From (13), we notice that an Hessian matrix based model updates decision vector with the increment of storage cost and computational complexity. AdaHessian model embed Hessian main diagonal into Adam framework to preprocess gradient to reduce the cost and update gradient descent direction. However, objective function L is a high-dimension non-convex function with irreversible Hessian matrix. Hence, to obtain second curvature information of L, we further propose AdaGO which exploits main diagonal of Gauss-Newton and Jaocibian matrix. The update framework is defined as follows:

$$\begin{cases}\boldsymbol{d}=\operatorname{diag}\left(\mathbf{J}_L\left(\boldsymbol{y}\right)^{\mathrm{T}}\mathbf{J}_L\left(\boldsymbol{y}\right)+\alpha\mathbf{I}\right)\\ \boldsymbol{m}^{t+1}=\beta\boldsymbol{m}^t+\left(1-\beta\right)\nabla L\left(\boldsymbol{y}^t\right)\\ \boldsymbol{s}^{t+1}=\omega\boldsymbol{s}^t+\left(1-\omega\right)\boldsymbol{d}\odot\boldsymbol{d}\\ \hat{\boldsymbol{m}}^{t+1}=\dfrac{\boldsymbol{m}^{t+1}}{1-\beta^{t+1}}\\ \hat{\boldsymbol{s}}^{t+1}=\dfrac{\boldsymbol{s}^{t+1}}{1-\omega^{t+1}}\\ \boldsymbol{y}^{t+1}=\boldsymbol{y}^t-\left(\sqrt{\boldsymbol{s}^{t+1}}\right)^{-1}\boldsymbol{m}^{t+1}\end{cases} \tag{14}$$

where $diag(\boldsymbol{J}_L(y)^T\boldsymbol{J}_L(y)+\alpha I)$ is the Gauss-Newton diagonal, which we denote as *d*. *m*, *s* are the first and second order moments, and β, ω denote the first and second moment hyperparameters which are also used in Adam.

Recall that $\boldsymbol{E}$ is an SIM, hence, Hessian matrix of L which constructed based on the known event data is a high-dimensional sparse matrix. Otherwise, the main diagonal elements are significantly larger than the non-diagonal elements of Hessian matrix in L. Thus, the main diagonal of Gauss-Newton matrix with damping terms approximate to Hessian matrix accurately and make it invertible. With (14), we preprocess gradient through the partial curvature information of L at a certain point can update gradient direction better.

4. EXPERIMENT

4.1 Experiment Settings

Datasets: Two datasets are employed in our experiments to prove the performance of the proposed method AdaGo.

1) MovieLens 1M (D1) (Harper & Konstan, 2015): It is a widely used and well-known dataset in the field of recommender systems. This dataset contains one million ratings given by users to movies, along with movie titles, genres, and user demographics like age, gender, occupation and zip-code. |I| means UserIDs range from 1 to 6040. |J| is the number of movieID's range between 1 and 3952.
2) Douban monti(D2) (Han et al., 2021): It provides valuable insights into user preferences, interests and behaviors. It preprocessed and provided by Monti. This dataset offers a wealth of information about users' tastes, inclinations, and actions on the Douban platform.

Their details are summarized in Table 2.

Evaluation Metrics: The main task of this experiment is to complete the unknown entries in Γ based on the known set ***K***. For measuring the accuracy with the model AdaGo, we adopt root mean square error (RMSE) as the metrics to evaluate the prediction accuracy between the ground truth e_{ij} and the predicted rating $\hat{e}_{ij}$:

$$RMSE = \sqrt{\left(\sum_{e_{ij}\in\Gamma}\left(e_{ij} - \hat{e}_{ij}\right)^2\right)\Big/|\,\Gamma\,|},\quad MAE = \left(\sum_{e_{ij}\in\Gamma}\left|e_{ij} - \hat{e}_{ij}\right|_{abs}\right)\Big/|\,\Gamma\,|; \tag{15}$$

where Γ denotes the validation dataset and $|\cdot|_{abs}$ the absolute value of a given input.

Implementation Settings: The choice of using a machine with an Intel-i5 2.5 GHz CPU and 32GB RAM for all experiments was made to ensure sufficient computational power and memory capacity. This configuration allows for efficient execution of the involved models, enabling accurate analysis and reliable results. The decision to use JAVA SE 7U60 as the programming language for implementing all models was based on its compatibility, stability, and widespread usage in the field of software development.

4.2 Experiment Results

Table 3 lists three methods involved in the stage of verification. All adopted benchmark methods are the most representative method for recommendation systems in recent years.

Table 2. Adopted symbols and their descriptions

Dataset	|*I*|	|*J*|	|*K*|
D1	6,040	3,952	1,000,290
D2	3,000	3,000	136,891

Table 3. Involved methods in our experiments

No.	Method	Description
M1	AdaGo	An optimization algorithm based on the Quasi-Newton method
M2	Adam (Kingma & Ba, 2014)	AN Adam-based LF model (Kingma & Ba, 2014)
M3	Extended SGD Model (Luo et al., 2019)	A SGD-based LF model (Kluver et al., 2018)

1) Adam (Kingma & Ba, 2014): Adam is a gradient-based optimization method that leverages adaptive estimates of lower-order moments to optimize stochastic objective functions. It offers straightforward implementation, computational efficiency, low memory requirements, and robustness against diagonal rescaling of gradients. Adam excels in tackling large-scale problems involving extensive data and/or parameters, as well as non-stationary objectives with noisy or sparse gradients. Moreover, its hyper-parameters are intuitive only with little tuning.
2) Extended SGD Model (Luo et al., 2019): Eight extended LF models realize higher accuracy for unknown data and faster convergence rate by sensitively extracting the hidden insights of users' potential preferences and community tendency.

The evaluation results are listed in Table 4 and 5. Based on them, we conclude that M1 outperforms both M2 and M3 in terms of prediction accuracies. It is worth noting that M2 and M3 are based on first-order solvers, while our method M1 AdaGo utilizes a different approach.

Specifically, when considering the performance on dataset D1, the minimum Mean Absolute Error (MAE) achieved by M1 is 0.6730. This value is slightly lower than that of M2 by 0.1% and even lower than that of M3 by 0.01%. These differences may seem small but indicate the superior accuracy of M1 compared to its counterparts.

Moving on to dataset D2, we observe a similar trend where the MAE obtained by M1 is consistently lower than those achieved by both M2 and M3. The margin between their performances becomes more pronounced with an improvement of approximately 0.32% for each metric.

In summary, based on the analysis of Fig. 2 to Fig. 7, our comprehensive evaluation demonstrates that M1 achieves higher prediction accuracies compared to model M2 and M3 which rely on first-order solvers. These improvements are evident across multiple datasets as indicated by both MAE and RMSE metrics.

5. CONCLUSION

The exploration for solving large-scale non-convex optimization problems is the core of advancing Matrix Factorization (MF)-based recommender systems, which have become indispensable tools in our digitally interconnected world. While the potential for leveraging second-order approaches to

Table 4. The average RMSE and time cost of involved methods

		D1			D2	
	RMSE	Epoch	Time	RMSE	Epoch	Time
M1	**0.8549±0.0015**	476±34	425±23	**0.7420±0.0067**	435±4	59±5
M2	0.8553±0.0012	**444±3**	939±2	0.7431±0.0067	279±61	84±18
M3	0.8551±0.0014	461±3	**45±4**	0.7434±0.0062	**267±66**	**4±0**

Table 5. The average MAE and time cost involved methods

	MAE	D1		MAE	D2	
		Epoch	Time		Epoch	Time
M1	**0.6730±0.0006**	458±37	414±43	**0.5811±0.0053**	443±8	**63±6**
M2	0.6737±0.0007	400±1	826±15	0.5830±0.0053	255±62	**75±16**
M3	0.6731±0.0007	**334±2**	**33±4**	0.5830±0.0054	**218±28**	**4±1**

Figure 2. The average RMSE of involved methods

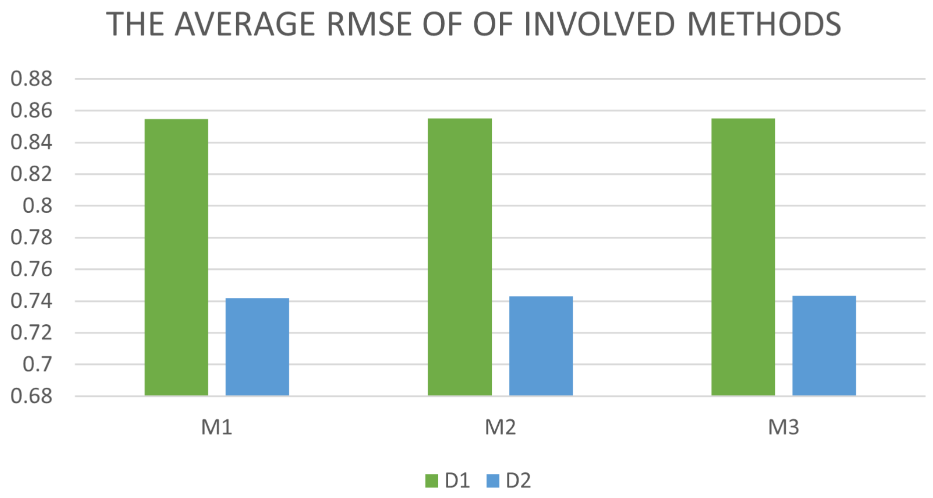
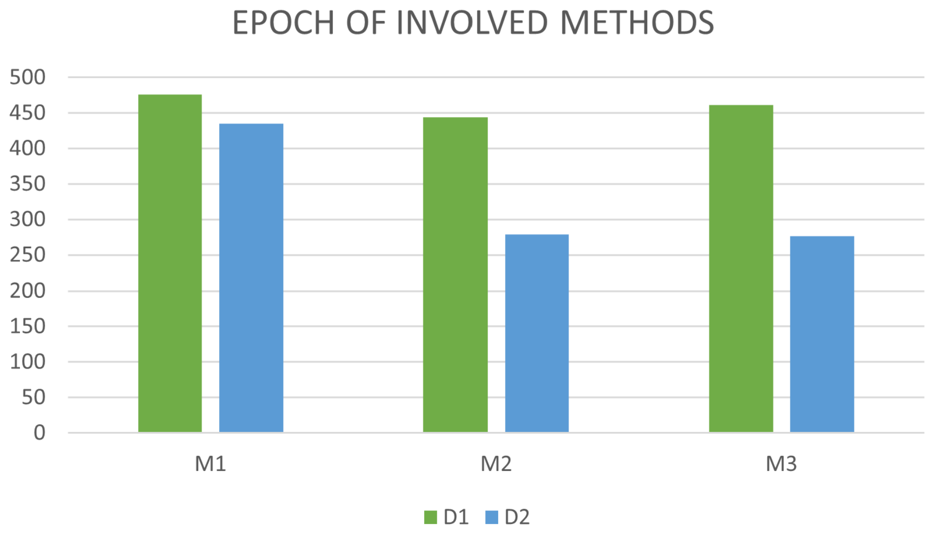


Figure 3. Epoch used in RMSE of involved methods



navigate the intricate landscapes of these optimization problems has generated great anticipation, the practicality of directly employing Hessian-based methods often remains an elusive goal due to their formidable computational demands.

In response to this challenge, we introduce AdaGO, a Quasi-Newton method-based optimizer that makes a harmonious balance between computational efficiency and optimization performance. Compared to state-of-the-art MF-based models, AdaGO has been proved its remarkable ability to enhance prediction accuracies. With the help of AdaGO, recommender systems can make more precise and context-aware recommendations, further cementing their role as essential components of e-commerce, content streaming, social media, and countless other digital platforms.

Figure 4. Time used in RMSE of involved methods

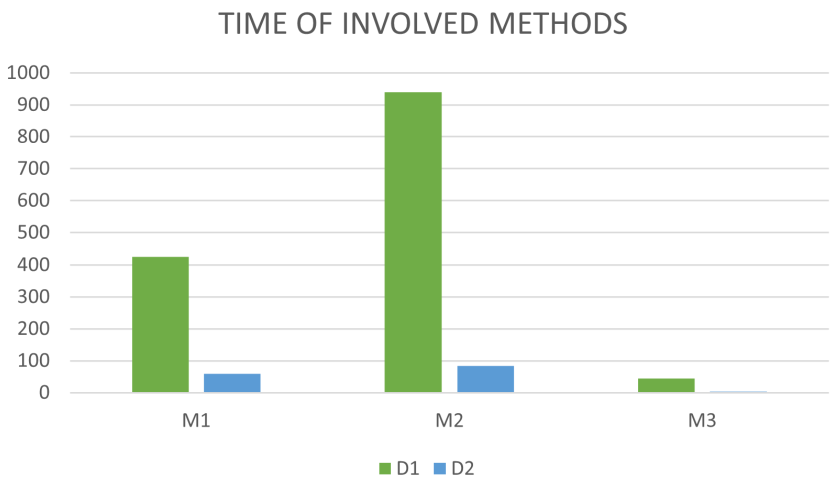
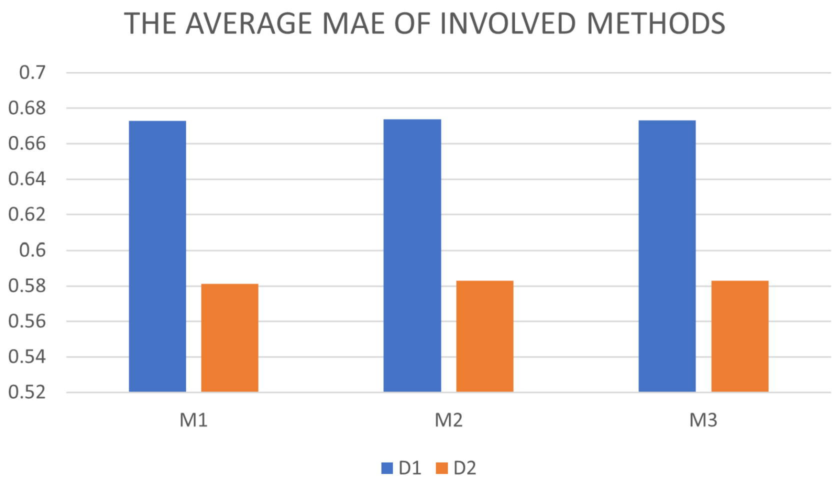


Figure 5. The average MAE of involved methods



The continued refinement of optimization techniques promises to unlock new frontiers in recommendation technology. These advancements not only empower users with more relevant and engaging content but also offer online service providers a competitive edge in the field of data-driven decision-making. The synergy between optimization innovation and the ever-expanding universe of recommender systems holds the potential to reshape how we discover, engage with, and benefit from the wealth of information and services available in our digital age.

In the future, prioritizing knowledge introduction, we will focus on testing and deployment of algorithms not only in recommendation systems but also across various domains. One such domain is environmental conservation, where algorithms can play a crucial role in analyzing large datasets to identify patterns and trends related to climate change, deforestation, or pollution. These insights can then be used to develop effective strategies for sustainable resource management and conservation efforts.

Figure 6. Epoch used in MAE of involved methods

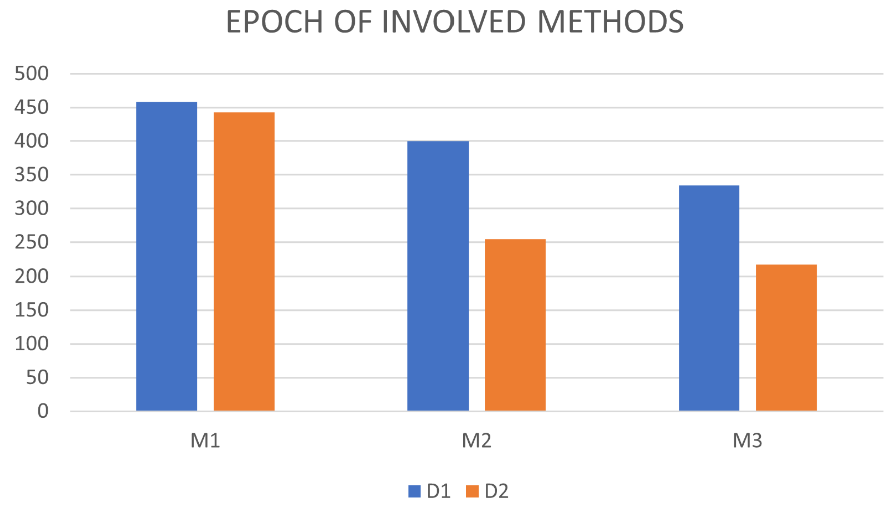
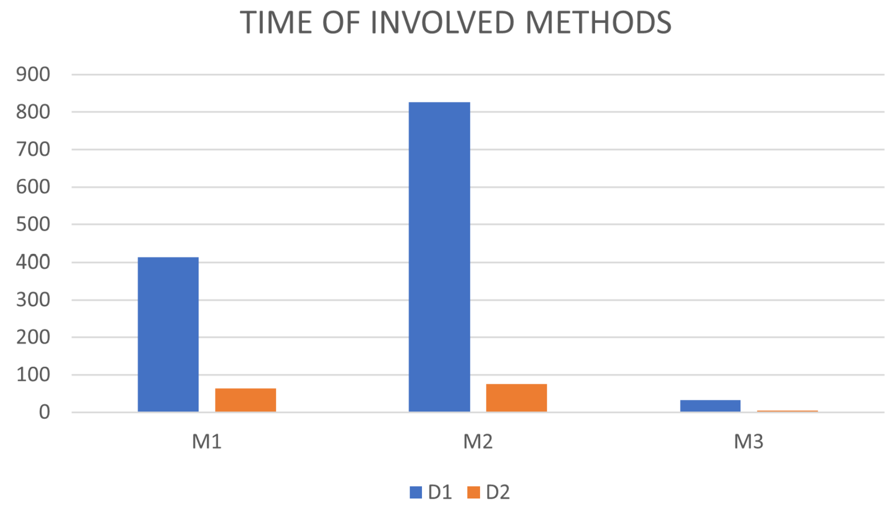


Figure 7. Time used in MAE of involved methods



Furthermore, algorithms will also find applications in smart citizen healthcare. With the advancement of technology and the increasing availability of health-related data from wearable devices and electronic medical records, algorithms can assist in analyzing this vast amount of information to provide personalized healthcare recommendations. This could include early detection of diseases through predictive analytics or suggesting tailored treatment plans based on individual patient characteristics. The integration of algorithms into these diverse domains signifies a shift towards data-driven decision-making processes. By harnessing the power of artificial intelligence and machine learning techniques, we can expect significant advancements in fields like environmental science and healthcare that will ultimately benefit society as a whole.

REFERENCES

Adomavicius, G., & Kwon, Y. (2011). Improving aggregate recommendation diversity using ranking-based techniques. *IEEE Transactions on Knowledge and Data Engineering*, *24*(5), 896–911. doi:10.1109/TKDE.2011.15

Adomavicius, G., & Tuzhilin, A. (2005). Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. *IEEE Transactions on Knowledge and Data Engineering*, *17*(6), 734–749. doi:10.1109/TKDE.2005.99

Akçay, S., Güven, E., Afzal, M., & Kazmi, I. (2022). Non-negative matrix factorization and differential expression analyses identify hub genes linked to progression and prognosis of glioblastoma multiforme. *Gene*, *824*, 146395. doi:10.1016/j.gene.2022.146395 PMID:35283227

Berry, M. W., Browne, M., Langville, A. N., Pauca, V. P., & Plemmons, R. J. (2007). Algorithms and applications for approximate nonnegative matrix factorization. *Computational Statistics & Data Analysis*, *52*(1), 155–173. doi:10.1016/j.csda.2006.11.006

Cao, J., Yang, H., Lv, J., Wu, Q., & Zhang, B. (2023). Estimating Soil Salinity with Different Levels of Vegetation Cover by Using Hyperspectral and Non-Negative Matrix Factorization Algorithm. *International Journal of Environmental Research and Public Health*, *20*(4), 2853. doi:10.3390/ijerph20042853 PMID:36833548

Ding, C. H., Li, T., & Jordan, M. I. (2008). Convex and semi-nonnegative matrix factorizations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *32*(1), 45–55. doi:10.1109/TPAMI.2008.277 PMID:19926898

Farzadnia, S., & Vanani, I. R. (2023, May). Using Non-Negative Matrix Factorization to Identify the Factors Affecting the Performance of Bank Employees in Australia. In *2023 9th International Conference on Web Research (ICWR)* (pp. 74-80). IEEE. doi:10.1109/ICWR57742.2023.10139075

Gu, Q., Zhou, J., & Ding, C. (2010, April). Collaborative filtering: Weighted nonnegative matrix factorization incorporating user and item graphs. In *Proceedings of the 2010 SIAM international conference on data mining* (pp. 199-210). Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611972801.18

Han, S., Cui, Q., Zheng, R., Li, S., Zhou, B., Fang, K., Sheng, W., Wen, B., Liu, L., Wei, Y., Chen, H., Chen, Y., Cheng, J., & Zhang, Y. (2023). Parsing altered gray matter morphology of depression using a framework integrating the normative model and non-negative matrix factorization. *Nature Communications*, *14*(1), 4053. doi:10.1038/s41467-023-39861-z PMID:37422463

Han, S. C., Lim, T., Long, S., Burgstaller, B., & Poon, J. (2021, October). GLocal-K: Global and local kernels for recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management* (pp. 3063-3067). doi:10.1145/3459637.3482112

Harper, F. M., & Konstan, J. A. (2015). The movielens datasets: History and context. *ACM Transactions on Interactive Intelligent Systems*, *5*(4), 1–19. doi:10.1145/2827872

He, J., Liu, Y., Li, S., Zhou, P., & Zhang, Y. (2023). Enhanced Dynamic Surface EMG Decomposition Using the Non-Negative Matrix Factorization and Three-Dimensional Motor Unit Localization. *IEEE Transactions on Biomedical Engineering*, 1–12. doi:10.1109/TBME.2023.3309969 PMID:37656646

Kingma, D. P., & Ba, J. (2014). *Adam: A method for stochastic optimization*. arXiv preprint arXiv:1412.6980.

Kluver, D., Ekstrand, M. D., & Konstan, J. A. (2018). Rating-based collaborative filtering: algorithms and evaluation. *Social information access: Systems and technologies*, 344-390.

Koren, Y., Rendle, S., & Bell, R. (2021). Advances in collaborative filtering. *Recommender systems handbook*, 91-142.

Lee, D., & Seung, H. S. (2000). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems*, 13.

Lee, D. D., & Seung, H. S. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*(6755), 788–791. doi:10.1038/44565 PMID:10548103

Li, W., He, Q., Luo, X., & Wang, Z. (2020). Assimilating second-order information for building non-negative latent factor analysis-based recommenders. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, *52*(1), 485–497. doi:10.1109/TSMC.2020.3002762

Li, W., Luo, X., Yuan, H., & Zhou, M. (2022). A momentum-accelerated Hessian-vector-based latent factor analysis model. *IEEE Transactions on Services Computing*, *16*(2), 830–844. doi:10.1109/TSC.2022.3177316

Liu, Z., Luo, X., & Zhou, M. (2023). Symmetry and graph bi-regularized non-negative matrix factorization for precise community detection. *IEEE Transactions on Automation Science and Engineering*, 1–15. doi:10.1109/TASE.2023.3240335

Lo, Y. Y., Liao, W., Chang, C. S., & Lee, Y. C. (2017). Temporal matrix factorization for tracking concept drift in individual user preferences. *IEEE Transactions on Computational Social Systems*, *5*(1), 156–168. doi:10.1109/TCSS.2017.2772295

Luo, X., Wang, D., Zhou, M., & Yuan, H. (2019). Latent factor-based recommenders relying on extended stochastic gradient descent algorithms. *IEEE Transactions on Systems, Man, and Cybernetics. Systems*, *51*(2), 916–926. doi:10.1109/TSMC.2018.2884191

Luo, X., Zhou, M., Li, S., You, Z., Xia, Y., & Zhu, Q. (2015). A nonnegative latent factor model for large-scale sparse matrices in recommender systems via alternating direction method. *IEEE Transactions on Neural Networks and Learning Systems*, *27*(3), 579–592. doi:10.1109/TNNLS.2015.2415257 PMID:26011893

Luo, X., Zhou, M., Xia, Y., & Zhu, Q. (2014). An efficient non-negative matrix-factorization-based approach to collaborative filtering for recommender systems. *IEEE Transactions on Industrial Informatics*, *10*(2), 1273–1284. doi:10.1109/TII.2014.2308433

Mukkamala, M. C., & Ochs, P. (2019). Beyond alternating updates for matrix factorization with inertial Bregman proximal gradient algorithms. *Advances in Neural Information Processing Systems*, 32.

Muñoz-Montoro, A. J., Revuelta-Sanz, P., Martínez-Muñoz, D., Torre-Cruz, J., & Ranilla, J. (2023). An ambient denoising method based on multi-channel non-negative matrix factorization for wheezing detection. *The Journal of Supercomputing*, *79*(2), 1571–1591. doi:10.1007/s11227-022-04706-x

Pan, W., Yang, Q., Cai, W., Chen, Y., Zhang, Q., Peng, X., & Ming, Z. (2019). Transfer to rank for heterogeneous one-class collaborative filtering. *ACM Transactions on Information Systems*, *37*(1), 1–20. doi:10.1145/3243652

Seo, H., Shin, J., Kim, K. H., Lim, C., & Bae, J. (2022). Driving Risk Assessment Using Non-Negative Matrix Factorization With Driving Behavior Records. *IEEE Transactions on Intelligent Transportation Systems*, *23*(11), 20398–20412. doi:10.1109/TITS.2022.3193125

Sweeney, M. D., Torre-Healy, L. A., Ma, V. L., Hall, M. A., Chrastecka, L., Yurovsky, A., & Moffitt, R. A. (2023). FaStaNMF: A Fast and Stable Non-negative Matrix Factorization for Gene Expression. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 1–12. doi:10.1109/TCBB.2023.3296979 PMID:37467096

Tang, X., Cai, L., Meng, Y., Xu, J., Lu, C., & Yang, J. (2021). Indicator regularized non-negative matrix factorization method-based drug repurposing for COVID-19. *Frontiers in Immunology*, *11*, 603615. doi:10.3389/fimmu.2020.603615 PMID:33584672

Wu, P., Gao, F., Tang, X., & Li, K. (2023). An Integrated Decision and Motion Planning Framework for Automated Driving on Highway. *IEEE Transactions on Vehicular Technology*, 1–11. doi:10.1109/TVT.2023.3293833

Yao, Z., Gholami, A., Shen, S., Mustafa, M., Keutzer, K., & Mahoney, M. (2021, May). Adahessian: An adaptive second order optimizer for machine learning. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 35, No. 12, pp. 10665-10673). AAAI.

Yu, X., Jiang, F., Du, J., & Gong, D. (2019). A cross-domain collaborative filtering algorithm with expanding user and item features via the latent factor space of auxiliary domains. *Pattern Recognition*, *94*, 96–109. doi:10.1016/j.patcog.2019.05.030

Zhang, S., Wang, W., Ford, J., & Makedon, F. (2006, April). Learning from incomplete ratings using non-negative matrix factorization. In *Proceedings of the 2006 SIAM international conference on data mining* (pp. 549-553). Society for Industrial and Applied Mathematics. doi:10.1137/1.9781611972764.58