

## Research Article

# Featured Hybrid Recommendation System Using Stochastic Gradient Descent

Si Thin Nguyen<sup>1</sup>, Hyun Young Kwak<sup>1</sup>, Si Young Lee<sup>1</sup>, Gwang Yong Gim<sup>2\*</sup><sup>1</sup>Department of IT Policy and Management, Graduate School, Soongsil University, Seoul, Korea<sup>2</sup>Department of Business Administration, Graduate School, Soongsil University, Seoul, Korea

## ARTICLE INFO

### Article History

Received 09 October 2020

Accepted 18 November 2020

### Keywords

Recommendation system  
stochastic gradient  
descent matrix factorization  
content-based  
collaborative filtering  
incremental learning

## ABSTRACT

Beside cold-start and sparsity, developing incremental algorithms emerge as interesting research to recommendation system in real-data environment. While hybrid system research is insufficient due to the complexity in combining various source of each single such as content-based or collaboration filtering, stochastic gradient descent exposes the limitations regarding optimal process in incremental learning. Stem from these disadvantages, this study adjusts a novel incremental algorithm using in featured hybrid system combining the feature of content-based method and the robustness of matrix factorization in collaboration filtering. To evaluate experiments, the authors simultaneously design an incremental evaluation approach for real data. With the hypothesis results, the study proves that the featured hybrid system is feasible to develop as the future direction research, and the proposed model achieve better results in both learning time and accuracy.

© 2021 The Authors. Published by Atlantis Press B.V.

This is an open access article distributed under the CC BY-NC 4.0 license (<http://creativecommons.org/licenses/by-nc/4.0/>).

## 1. INTRODUCTION

Recommendation system is a branch of information filtering system that aims to predict “preference” or “rating” that a user prefers to give to an item (such as music, movies, or books) or social object (e.g. group or people) that had been unforeseen, using a model created based on the user’s social relationship (collaborative filtering model) or the features of an item (content-based model) [1,2].

Continuous growth of web and its applications has given a significant importance for recommendation systems [3]. However, recommendation system needs to deal with five main research problems: cold-start, data sparsity, accuracy and scalability [4].

Generally, there are three different groups of recommendation system: knowledge-based, Content-based (CB), Collaborative Filtering (CF) [2]. Consequently, these models use different sources of input data, and due to this difference, they all have different weaknesses and strengths. For instance, knowledge-based model can overcome cold-start problems better than CF or CB because they do not use ratings source. Contrarily, they are weaker than CF and CB in using information personalization from historical data [5]. These various personalities exhibited by different recommendation algorithms simultaneously figure out that recommendation model is not a one-size-fits-all model [6]. In another view, the recommendation algorithms themselves are considered as a part of system in giving the recommendations to users [7]. Such algorithms are efficient and simple; however, with the different data, they have limit in performance and it is not easy to further improve the accuracy of the recommendation system [8].

This is the reason why hybrid systems are the solution to explore these problems. However, the studies about hybrid system are still inadequate.

In this study, the authors design a hybrid system that combine CB and Incremental Matrix Factorization for the following reasons. First, CB and CF have different sources of data. The conjunction textual description of item in CB and community ratings in matrix factorization is expected to gain the accuracy in training data process. Second, almost all recommendation systems use numeric rating. Algorithms in incremental matrix factorization are considered in many researches for their benefit. Third, while most of the recent research focuses on algorithms for the basic model, few researches relate with the hybrid system and specific properties of evaluation real data unexplored [2].

In addition, this study aims to overcome the issues related to big data process. As mentioned above, the continuous growth of information, especially through the Internet, has given the issue related to the information overload. Netflix store over 17,000 movies in its data warehouse [9], and Amazon.com contains over 410,000 titles in its Kindle store. However, the speeds of computer is slower than the data sizes. Thus, regarding this issue, the capabilities of statistical machine learning algorithm are limited by the computing time rather than the data size [10].

To deal with this information overload, incremental learning algorithm is the effective solution with Stochastic Gradient Descent (SGD) algorithm [11]. However, the non-flexible feature in learning rate SGD require a better approach [2]. For this is reason, this study adjusts two mechanisms which are  $k$  group and momentum into novel algorithm –  $k$  Momentum SGD (KMSGD).

\*Corresponding author. Email: [gygim@ssu.ac.kr](mailto:gygim@ssu.ac.kr)

The final issue is how to evaluate in real data environment. While offline evaluation is suitable for achieving perspicuity about how different algorithms perform and comparing performance, it is limited in various ways [12,13]. In real data, evaluations need to be directly processed in every data point. This is not easy and comparing various algorithms through online evaluation is costly and difficult to synthetic enough understanding of these relationships [14]. In this study, the authors present a novel approach to evaluation in real-time by experimenting every point data separately.

Finally, the study aims to the following purposes:

- To enhance traditional recommendation system with better prediction and improve accuracy, we design a feature hybrid system combining the textual feature of CB and the robustness of matrix factorization in CF.
- To create a new incremental algorithm KMSGD that overcome the limit of SGD algorithm.
- To evaluate in stream data, we design a simple approach with every point data process.

By comparing with other recommendation algorithms on MovieLens data, it is verified that the traditional model performance can be effectively improved by our model in both accuracy and learning time. The featured hybrid system is appropriate to further develop in future.

The structure of this study is designed as follows. The study shows a broad overview of incremental learning algorithm before showing the detail process of SGD algorithm. Then the ideal to design KMSGD is described typically in incremental learning algorithms section. The development of algorithm in recommender systems from basic model, CF model with Matrix factorization algorithms to hybrid system is figured out in the development of model in recommendation systems section. Proposed model section give a specific description of model. Before discussing about the results in results and discussion section, the authors mention evaluation and data in evaluation issues and data section. The prequential evaluation protocol is also described in this section. Finally, conclusion section provides a brief conclusion about the study as well as the future works.

## 2. INCREMENTAL LEARNING ALGORITHMS

### 2.1. Incremental Learning

Traditional machine learning methods provide typically powerful method to give structural information from original digital data and almost all recent applications are restricted to the batch data. Training is applied for all given data. Consequently, model selection and evaluation measurement can be computed by this full data. The whole training process can understand that the data are static [2]. In contrast, incremental learning is considered to the position of continuous model adaptation which rely on a constantly coming data stream [15]. This process is always present whenever systems operate autonomously such as in autonomous house or driving [16–19]. Moreover, it is necessary to have an online learning in interactive perspective where training data rely on person feedback over time [19]. Finally, albeit static, that are almost digital

data sets, can be so massive that they are de facto to tackle as a data stream, i.e. one incremental pass over the full data [2]. Incremental learning also figures out how to learn in such data streaming. It comes in different definition in the theory, and the utilization of the theory is not always suitable. Consequently, we present a meaningful definition to the appropriate terms of online learning, concept drift, and incremental learning, achieving typical attention to the supervised learning machine [2].

In the literature, there are a few researches on incremental CF. Incremental neighbourhood-based CF algorithm is applied [20,21]. For incremental matrix factorization, one first approach is designed [22], where the authors apply the fold-in method [23] to incrementally update to the factor matrices. An incremental learning algorithm for ranking source that utilizes a selective sampling approach is designed [24]. An incremental algorithm to update user factors is presented [25] by utilizing a simple method of the batch process. Two incremental algorithms using SGD are exposed [26]. This study is different from the above works. Stemming from SGD, we adjust the loop rate and momentum. Then, we design a novel evaluate method to evaluate the evolving accuracy of algorithm.

### 2.2. SGD Algorithm

#### 2.2.1. The mechanism of SGD

Figure 1 present the mechanism of SGD [2]. Given a training dataset including data rows in the form  $\langle \text{user}, \text{item}, \text{rating} \rangle$ , SGD process various passes through the given data - iterations - until some stopping criteria is satisfactory - generally a convergence bound and/or maximum value of iterations [2]. At each iteration, SGD executes all observed ratings  $R_{ui}$  and updates the correlative rows  $W_u$  and  $X_i^T$ , adjusting them in the inverse process of the gradient to the error, using a factor value of  $\eta \leq 1$  - known as learn rate or step size. The correlative error is set as  $\text{err}_{ui} = R_{ui} - \hat{R}_{ui}$  for each observed rating, and the following update process are executed:

$$\begin{aligned} W_u &\leftarrow W_u + \eta(\text{err}_{ui} \cdot X_i - \lambda W_u) \\ X_i &\leftarrow X_i + \eta(\text{err}_{ui} \cdot W_u - \lambda X_i) \end{aligned} \quad (1)$$

#### Algorithm 1. SGD

**Data:**  $D = \{ \langle u, i, r \rangle \}$

**Input:**  $\text{feat}, \lambda, \eta$

**for**  $\langle u, i \rangle \in D$  **do**

{

**if**  $u \notin \text{Rows}(W)$  **then**  $W_u \leftarrow \text{Vector}(\text{size: feat});$

**if**  $i \notin \text{Rows}(X^T)$  **then**  $X_i^T \leftarrow \text{Vector}(\text{size: feat});$

$\text{err}_{ui} \leftarrow R_{ui} - W_u X_i^T$

$W_u \leftarrow W_u + \eta(\text{err}_{ui} X_i - \lambda W_u)$

$X_i \leftarrow X_i + \eta(\text{err}_{ui} W_u - \lambda X_i)$

}

Figure 1 | Algorithm 1-SGD.

Or with  $\phi = (W; X)$ , SGD contrarily process a parameter update for every training example  $x^{(i)}$  and label  $y^{(i)}$ :

$$\phi = \phi - \eta \cdot \nabla \mathcal{L}(\phi; x^{(i)}; y^{(i)}) \quad (2)$$

In the training data, one major benefit of SGD is that complexity upgrades linearly with the number of observed ratings by getting benefit from the high sparsity of  $R$  [2].

Notably, after selecting a random training data point  $(i, j) \in D$ , SGD update  $W_i$  and  $X_j$  and do not update factors for other training data point. This process savings follow process from representation of the global loss as a sum of local losses [27].

Thus, SGD refer to online learning or incremental gradient descent. In batched method, multiple local losses are averaged, and are also appropriate but usually have lower performance in experiment [2].

### 2.2.2. The limitation of SGD

In SGD, value of the learning rate (or step size) is very small  $\alpha = 0.005$ . Consequently, a more suitable method to avoid local minimum and speed up convergence is to use the bold driver algorithm [28,29] to identify  $\alpha$  in each iteration. Theoretically, it is feasible to use different step size for different factors [30].

An impressive method of these models is that executing until convergence for all iterations can lead to small worsening of the resolution quality of the unobserved value. Finally, it is usually reasonable not to set the convergence criteria too strictly [2].

Another key problem with SGD is about initialization. For example, it can begin the factor matrices to smaller values in  $(-1, 1)$ . However, the choice of initialization, can alter the eventual solution quality.

## 2.3. Proposed Algorithm – KMSGD

### 2.3.1. Group in mini batch

Mini-batch gradient descent is an adjustment of the SGD that divides the training data into smaller groups that are utilized to identify model error and update model coefficients. Process may choose to compute the gradient over the mini-batch or take the average of the gradient which further decrease the variance of the gradient [2].

The ideal of method is illustrated in Figure 2. Mini-batch gradient descent aim to set a balance between the robustness of SGD and the speed-up of batch gradient descent. It is the most important process of gradient descent applied in machine learning.

Figure 3 explains the algorithm 2 we propose [2]. According to it, this algorithm has two different things from algorithm 1-SGD. First, the learning process set a single pass over the observed data. Notable in algorithm 2, at each data point  $u, i$ , the alteration to factor matrices  $W$  and  $X$  are set in a single step. One other possible method is to perform several iterations over each new data point, with increasing accuracy, at the cost of the additional time required to re-iterate [2]. Second, there are no data shuffling or other pre-processing is processed. Thus, we set the error:  $\text{err}_{ui} = R_{ui} - \hat{R}_{ui}$ ,

#### Algorithm 3. KMSGD

**Data:**  $D = \{ \langle u, i, r \rangle \}$

**Input:**  $\text{feat}, \lambda, \eta, k$

**for count**  $\leftarrow 1$  to  $k$

```
{
  for  $\langle u, i \rangle \in D$  do
    {
      if  $u \notin \text{Rows}(W)$  then  $W_u \leftarrow \text{Vector}(\text{size: feat})$ ;
      if  $i \notin \text{Rows}(X^T)$  then  $X_i^T \leftarrow \text{Vector}(\text{size: feat})$ ;
       $\text{err}_{ui} \leftarrow 1 - W_u X_i^T - v_i$ 
       $W_u \leftarrow W_u + \eta(\text{err}_{ui} X_i - \lambda W_u) - v_i$ 
       $X_i \leftarrow X_i + \eta(\text{err}_{ui} W_u - \lambda X_i) - v_i$ 
    }
}
```

Figure 2 | Algorithm for KMSGD.

#### Algorithm 2. K-SGD

**Data:**  $D = \{ \langle u, i, r \rangle \}$

**Input:**  $\text{feat}, \lambda, \eta, k$

**for count**  $\leftarrow 1$  to  $k$

```
{
  for  $\langle u, i \rangle \in D$  do
    {
      if  $u \notin \text{Rows}(W)$  then  $W_u \leftarrow \text{Vector}(\text{size: feat})$ ;
      if  $i \notin \text{Rows}(X^T)$  then  $X_i^T \leftarrow \text{Vector}(\text{size: feat})$ ;
       $\text{err}_{ui} \leftarrow 1 - W_u X_i^T$ 
       $W_u \leftarrow W_u + \eta(\text{err}_{ui} X_i - \lambda W_u)$ 
       $X_i \leftarrow X_i + \eta(\text{err}_{ui} W_u - \lambda X_i)$ 
    }
}
```

Figure 3 | Algorithm 2 K-SGD.

and upgrade the rows in  $W$  and  $X^T$  by using the upgrade process in (3).

$$\phi = \phi - \eta \cdot \nabla \mathcal{L}(\phi; x^{(ii+k)}; y^{(ii+k)}) \quad (3)$$

There are two advantages in algorithm 2:

- To decrease the time of the parameter upgrade, which can aim to more suitable convergence.
- To make using of matrix optimizations machine learning that compute the gradient  $W.X.R.$  very effective.

### 2.3.2. Momentum

The name momentum stems from momentum in physics, wherein the weighted vector  $w$ , considered through a slight moving through parameter space, encounter acceleration from the gradient of

the loss function [2]. Contrarily, traditional SGD is expected to keep moving in the same situation, preventing oscillations. Thus, momentum has been utilized effectively in studying artificial neural networks in recent years.

Momentum is a value that accelerate SGD in relevant direction and dampens oscillations [2]. This method utilizes a fraction of the vector  $v_t$  of the previous-time step to the recent update vector:

$$\begin{aligned} v_t &= \gamma \cdot v_{t-1} + \eta \cdot \nabla \phi L(\phi) \\ \phi &= \phi - v_t \end{aligned} \quad (4)$$

The momentum term  $\gamma$  is set to 0.9 value. Basically, when momentum is utilized the process gains faster convergence and reduces oscillation. The mechanism of algorithm is illustrated in Figure 2 by adding  $v_t$  in each iteration [2].

### 3. THE DEVELOPMENT OF MODEL IN RECOMMENDATION SYSTEMS

#### 3.1. Single Model

In single model, algorithms play an important role and directly affect the performance of the recommendation system. The popular algorithms in recommendation can be divided into two main groups: CB model [31] and CF model [32].

##### 3.1.1. Content-based

Content-based model utilizes features of users and items through the analysis of textual information, such as the features of items or user demography and document content to make recommendations. However, this feature extraction is difficult to gather or even fake; thus, this model has considerable limitations.

Figure 4 shows that the extraction the feature vector of item is the most important phase. Therefore, TF-IDF text mining algorithm is utilized in this study:

$$tf(t, d) = 0.5 + 0.5 \frac{f_{t,d}}{\max\{f_{t',d} : t' \in d\}} \quad (5)$$

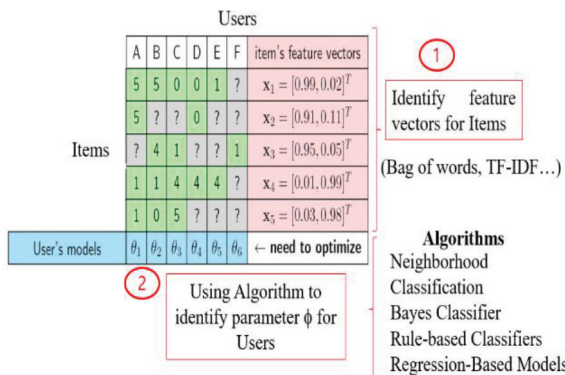


Figure 4 | Content-based.

The loss function of content-base is presented as below:

$$\tau_n(W_n, b_n) = \frac{1}{2s_n} \|\hat{X}_n^T W_n + b_n e_n - \hat{y}_n\|_2^2 + \frac{\lambda}{2s_n} \|W_n\|_2^2 \quad (6)$$

In content-based model, the important step is to identify the typical feature of each item, which is a numeric vector record indicating the key features of item. For example, the featured vector includes the typical feature of the item that are easily identify. Such features include the characteristics of a movie that is relevant to system [2].

##### 3.1.2. Matrix factorization

In Matrix factorization, the system that includes  $n$  users and  $m$  items, we correspondingly establish a matrix  $R$  with size  $(n \times m)$ , in which the  $(u, i)$ th entry is the  $r_{ui}$  value - the rating value of user  $u$  to item  $i$  [2].

In fact,  $R$  matrix with many empty cells-sparsity problem, is considered to overcome in recommendation system. Estimating unobserved ratings is also a necessary requirement [2].

The loss function of Matrix factorization is presented as below:

$$\begin{aligned} \tau_n(X, W, b, d) &= \frac{1}{2s} \sum_{n=1}^N \sum_{m:r_{nm}=1} (X_m^T W_n + b_m + d_n - y_{nm}) \\ &+ \frac{\lambda}{2} (\|W\|_F^2 + \|b\|_F^2) \end{aligned} \quad (7)$$

#### 3.2. Hybrid System

Some recommendation systems combine various source aspects to create hybrid systems. Thus, hybrid systems can combine the strengths of different single model into unified system. Figure 5 shows three main ways of making hybrid recommendation systems [5]:

1. Ensemble design: the output from single algorithm is combined into a robust output.
2. Monolithic design: the combined recommendation algorithm is presented by using various data types.

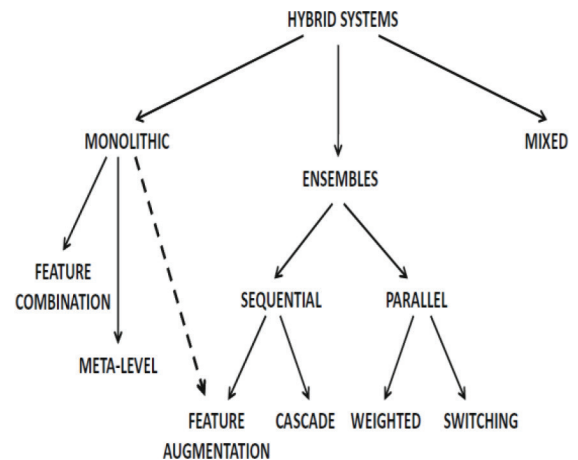


Figure 5 | The taxonomy of hybrid systems.



3. Mixed systems: these systems utilize multiple recommendation algorithms, but the items recommended by the several systems are unified system together side by side.

Another way that hybrid recommender systems can be classified into the following categories (seven groups): weighted, switching, cascade, feature augmentation, feature combination, meta-level and mixed. Generally, this is the view with more detail than another one [33].

## 4. PROPOSED MODEL

### 4.1. CB and CF

As aforementioned, the CB and CF use various sources of input, and they have different strengths in different scenarios. Table 1 summarizes the advantage and disadvantage between two these models.

Generally, content-based with textual feature is to overcome the cold-start problem. However, it does not use the relationship among user or item. In contrast, collaboration is more robustness by using this relationship. This is the reason the hybrid system with various sources is expected to gain many opportunities to achieve the best results.

### 4.2. Feature Combine Hybrid System

Because CB and CF use different source, textual in CB and rating source in CF, the question is how to combine two model in to unify model before using training data.

Figure 6 present this idea.

**Table 1** | Advantage and disadvantage of basis models in recommendation system

Categories	Collaboration		Input source	Advantage	Disadvantage
	User	Item			
Content-based	–	–	Textual description of item	• Item feature extract • Sparsity	• Non-collaboration • Cold-start
Collaborative filtering	Neighborhood-Based	User–User	Community ratings	• Simple • Easy to explain	• Small Data
	Collaborative Filtering (NBCF)	Item–Item			
	Model				
				• Cold-start • More accuracy	• Sparsity

Ratings matrix  $R$  with size  $(m \times n)$  is added  $d$  columns for features of items. Thus, the new ratings matrix is set size  $(m \times (n + d))$ , wherein  $d$  is the number of feature item and  $n$  is the number of items. Then the objective function is computed as follows with a parameter vector  $\theta$ :

$$J = \text{Collabrative object } (\theta) + \beta \cdot \text{Content object } (\theta) + \text{Regularization} \quad (8)$$

Notably, matrix  $R$  is an  $(m \times n)$  latent ratings matrix, and  $C$  is a  $(d \times n)$  content matrix, wherein each item is presented by  $d$  features. Examples include short reviews of items or properties of items. Since  $R$  is latent ratings matrix, missing values are set to be 0 value. Consequently,  $W$  is an  $(n \times n)$  item-item coefficient matrix wherein the rating values are predicted as  $\hat{R} = R.W$ . In this case, however, we can also predict the rating values as  $\hat{R} = C.W$ . Thus, instead of only optimizing  $\|R - RW\|^2$ , we implement an additional content-based value  $\|R - CW\|^2$ . Together with diagonal/non-negativity constraints and elastic regularization, the upgrade optimization model is computed as follows:

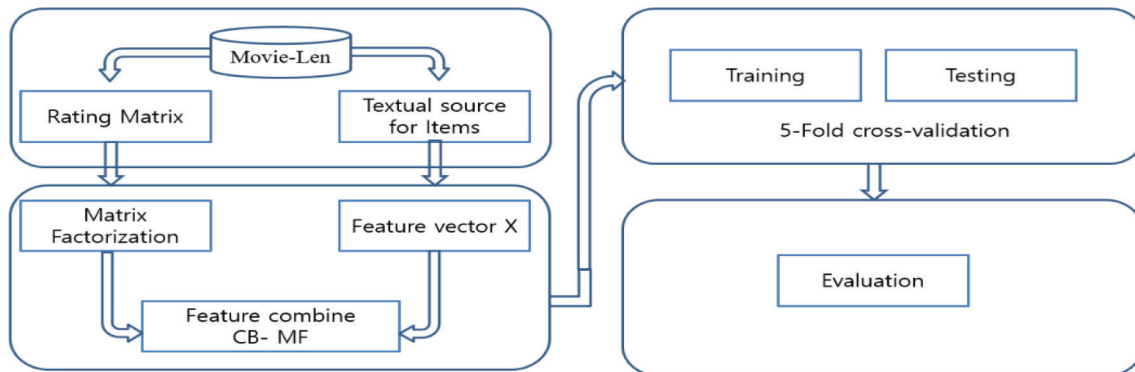
$$\text{Minimize } J = \|R - RW\|^2 + \beta \cdot \|R - CW\|^2 + \lambda \cdot \|W\|^2 + \lambda_1 \cdot \|W\| \quad (9)$$

subject to:

$$W \geq 0$$

$$\text{Diagonal } (W) = 0$$

In a tuning phase, the weight parameter  $\beta$  can be determined. Although the rating values can be predicted either as  $\hat{R} = C.W$  or  $\hat{R} = R.W$  or as only the latter prediction function is applied. Thus, the term  $\hat{R} = C.W$  is only utilized to refine as an additional regularize in the objective



**Figure 6** | Feature combine hybrid systems.

**Table 2** | The statistics of the two datasets

Features	Data	
	MovieLens-100 K	MovieLens-1 M
Users	943	6040
Items	1682	3952
Ratings	100,000	1,000,209
Ratings per user	106.4	165.6
Ratings per item	59.45	253.09
Rating sparsity	93.70%	95.81%

function. Moreover, the purpose of additional term is to upgrade the generalization power of algorithm for future predicting.

This approach can be utilized for combining any other instants of CF (optimization) model with CB models. For example, in the instant of matrix factorization, this method can use an  $(n \times k)$  shared item factor matrix  $X$ , a  $(d \times k)$  content factor matrix  $Z$  and  $(m \times k)$  user factor matrix  $U$  to identify the optimization model as follows:

$$\text{Minimize } J = \|R - U \cdot X^T\|^2 + \beta \cdot \|C - Z \cdot X^T\|^2 + \lambda(\|U\|^2 + \|X\|^2 + \|Z\|^2) \quad (10)$$

## 5. EVALUATION ISSUES AND DATA

### 5.1. Data

In the experiments, this study uses MovieLens-100M and MovieLens-100K which is presented in Table 2 to test hypothesis [2]. MovieLens-100K dataset includes about 943 users and 1.682 items with 100,000 rating values, and the dataset is download from the <https://grouplens.org/datasets/movielens/> website. All the rating are from 1 to 5 value. MovieLens-1M dataset is also bigger with 1.000.209 ratings from 3.952 movies and 6.040 users. Notably, each user ranked at least 20 movies [2].

### 5.2 Compare Methods

#### 5.2.1. Evaluation measurements

We use Root Mean Squared Error (RMSE) value, which are commonly used in machine learning to evaluate the accuracy of model. The formulation is as follows:

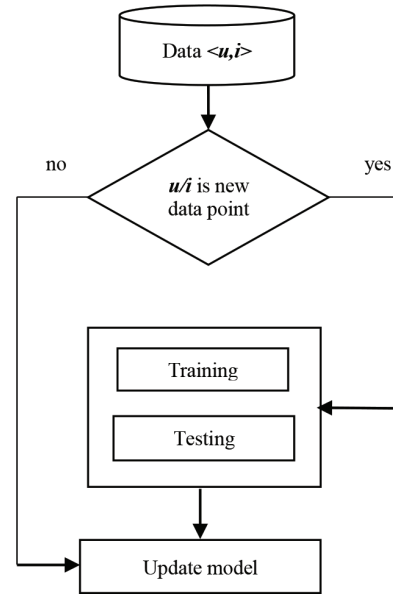
$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i,j} (R_{ij} - \hat{R}_{ij})^2} \quad (11)$$

The lower values of RMSE show the better prediction of the model.

The second measurements we use in experiment is learning time. This value is to evaluate the speed-up of learning algorithms.

#### 5.2.2. Proposed evaluation method

To evaluate in incremental model, we design a prequential approach. Figure 7 shows the steps with every observed point data  $\langle u, i \rangle$ , presenting a rating interaction between user  $u$  and item  $i$ :

**Figure 7** | Real-data evaluation approach.

This approach provides following advantages:

- It allows management of incremental system.
- Offline evaluation can be integrated in online evaluation.
- Both new user and item instance is suitable to evaluation.

## 6. RESULTS AND DISSCUSION

The results of learning time and accuracy are presented in Tables 3 and 4 with the MovieLens-1M data and MovieLens-100K respectively. Moreover, the comparing average value among algorithms are shown in Figure 8.

### 6.1. About the Accuracy

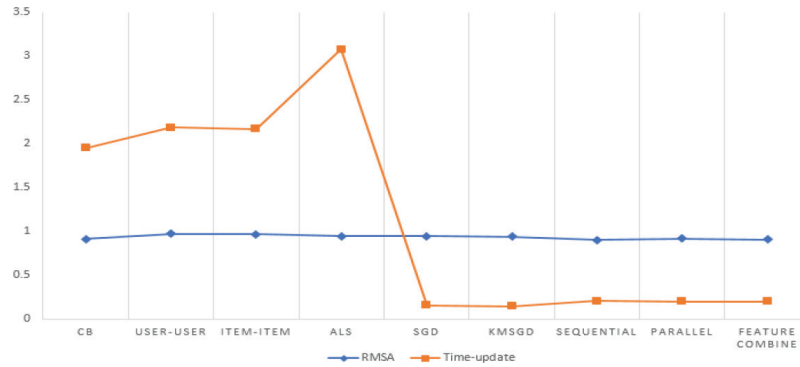
Very impressively, the simple algorithm gains a best result with 0.914 in RMSA. In CF algorithms, the divergence is not significant and the KMSGD with 0.949 approximately same with 0.948 of SGD. However, the difference is highlighted in hybrid system group, especially featured hybrid system. If we only focus on CF algorithms [Neighborhood-Based Collaborative Filtering (NBCF) and Model-Based Collaborative Filtering (MFCF)], algorithm with SGD shows the best result. This means that the more complex

**Table 3** Overall results of MovieLens-100K data

Measures	CB	Collaboration-Based-Filtering (CF)					Hybrid system		
		Neighborhood-Based CF (NBCF)		Model-Based CF (MFCF)			Ensemble		Monolithic
		User–User	Item–Item	ALS	SGD	KMSGD	Sequential	Parallel	Featured hybrid system
RMSA	0.914	0.976	0.968	0.951	0.948	0.949	0.894	0.896	0.890
Learning time (ms)	0.886	1.186	1.166	1.986	0.118	0.101	0.108	0.106	0.104

**Table 4** Overall results of MovieLens-1M data

Measures	CB	Collaboration-Based-Filtering (CF)					Hybrid system		
		Neighborhood-Based CF (NBCF)		Model-Based CF (MFCF)			Ensemble		Monolithic
		User–User	Item–Item	ALS	SGD	KMSGD	Sequential	Parallel	Featured hybrid system
RMSA	0.916	0.975	0.966	0.944	0.940	0.936	0.912	0.918	0.910
Learning time (ms)	3.016	3.180	3.168	4.168	0.201	0.198	0.204	0.201	0.199

**Figure 8** Comparing average RMSA and time update value among algorithms.

model, the better performance recommendation system exception the impressive of CB. To explain this situation, the authors focus on data. In Movie-Lens data, the features of item are presented in 19 binary value (19 types of movies) with 1 for the kind of that type movie. TF-IDF algorithms become non-meaningful in extracting features in CB model.

For this reason, the same thing is proved with the 0.916 RMSA value of CB in Movie-Lens 1M.

However, featured hybrid system still gains the best value 0.910. It is assumed that, SGD algorithm is not much meaningful in predict model but the hybrid system is a best choice to recommendation system research.

## 6.2. About the Learning Time

The learning-time is the major part in this study. In big data, a power predicted systems become un-useful if they cost too much time to process. The hypothesis is that the more complex model, the slower in training data. This hypothesis is proved with the learning time value of CB, NBCF and ALS in MFCF with 0.886, 1.186, 1.166 and 1.986 ms respectively in MoviesLen-100K [2]. Very impressively,

the algorithm with SGD, KMSGD and featured hybrid system is significantly better with 0.118, 0.101 and 0.104 ms respectively.

The results are the same with MovieLens-1M data. It means that the hypothesis the more complex model, the slower in training data is not correct. However, it became a big problem if we can use the incremental algorithm in model. It is proved with the Movie-Len 1M, the learning time of ALS compared with CB is significantly bigger: 4.168 and 3.016 ms. While the featured hybrid system continuously presents a lowest value with 0.199 ms.

Figure 8 presents a visual view about the comparing effectiveness among models. It can lead to conclusion that incremental algorithm and hybrid system deserve to depth research in recommendation system.

## 7. CONCLUSION

Stem from SGD algorithm, this study adjusts its mechanics in mini group and momentum. With experimental results, the proposed model combining feature of CB and the robustness of matrix factorization using incremental algorithm gain the better results in both learning time and accuracy with another state-of-the art algorithms.

The contribute of study is also present in a novel evaluation approach in real data. Although this approach is still simple, at least it presents a novel direction research in recommendation system.

The limitations in this study is relevant to various data. In the real system, the data with new users, new item and ranking is continuously updated. Moreover, the evaluation approach in this study is just an initial idea for the power evaluation approach in the future.

In the future, the authors will consider to apply parallel and distributed data analysis. Furthermore, the hybrid recommendation system is also a highlighted research combined with deep learning algorithms such as the convolutional neural network and attempt to further develop their performance.

## CONFLICTS OF INTEREST

The authors declare they have no conflicts of interest.

## REFERENCES

- [1] L. Sharma, A. Gera, A survey of recommendation system: research challenges, *Int. J. Eng. Trends Technol.* 4 (2013), 1989–1992.
- [2] S.T. Nguyen, H.Y. Kwak, S.H. Lee, G.Y. Gim, Using stochastic gradient decent algorithm for incremental matrix factorization in recommendation system, *Proceedings of the 2019 20th IEEE/ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD)*, IEEE, Toyama, Japan, 2019, pp. 308–319.
- [3] L. Ravi, S. Vairavasundaram, A collaborative location based travel recommendation system through enhanced rating prediction for the group of users, *Comput. Intell. Neurosci.* 2016 (2016), 1291358.
- [4] E. Çano, M. Morisio, Hybrid recommender systems: a systematic literature review, *Intell. Data Anal.* 21 (2017), 1487–1524.
- [5] C.C. Aggarwal, *Recommender systems*, Vol. 1, Springer International Publishing, Cham, 2016.
- [6] M.D. Ekstrand, J.T. Riedl, J.A. Konstan, Collaborative filtering recommender systems, *Found. Trends Human-Comput. Interact.* 4 (2010), 81–173.
- [7] F.J. Martin, Recsys' 09 industrial keynote: top 10 lessons learned developing deploying and operating real-world recommender systems, *Proceedings of the Third ACM Conference on Recommender Systems*, ACM, New York, USA, 2009, pp. 1–2.
- [8] L. Zhang, T. Luo, F. Zhang, Y. Wu, A recommendation model based on deep neural network, *IEEE Access* 6 (2018), 9454–9463.
- [9] J. Bennett, S. Lanning, The netflix prize, *Proceedings of KDD Cup and Workshop*, 2007, p. 35.
- [10] L. Bottou, Large-scale machine learning with stochastic gradient descent, In: Y. Lechevallier, G. Saporta (Eds.), *Proceedings of COMPSTAT'2010*, Physica-Verlag HD, Berlin, Heidelberg, 2010, pp. 177–186.
- [11] A. Gepperth, B. Hammer, Incremental learning algorithms and applications, *European Symposium on Artificial Neural Networks (ESANN)*, Bruges, Belgium, 2016.
- [12] B.M. Marlin, R.S. Zemel, Collaborative prediction and ranking with non-random missing data, *Proceedings of the Third ACM Conference on Recommender Systems*, ACM, New York, USA, 2009, pp. 5–12.
- [13] S.M. McNee, J. Riedl, J.A. Konstan, Being accurate is not enough: how accuracy metrics have hurt recommender systems, *CHI'06 Extended Abstracts on Human Factors in Computing Systems*, 2006, pp. 1097–1101.
- [14] F. Ricci, L. Rokach, B. Shapira, *Introduction to recommender systems handbook*, *Recommender Systems Handbook*, Springer, Boston, MA, 2011, pp. 1–35.
- [15] D.L. Silver, Machine lifelong learning: challenges and benefits for artificial general intelligence, *Proceedings of the International Conference on Artificial General Intelligence*, Springer, Berlin, Heidelberg, 2011, pp. 370–375.
- [16] B. Hammer, M. Toussaint, Special issue on autonomous learning, *Künstl. Intell.* 29 (2015), 323–327.
- [17] E. Menegatti, K. Berns, N. Michael, H. Yamaguchi, Special issue on intelligent autonomous systems, *Robot. Autonom. Syst.* 74 (2015), 297–298.
- [18] S. Thrun, Toward robotic cars, *Commun. ACM* 53 (2010), 99–106.
- [19] J.R. Rico-Juan, J.M. Iñesta, Adaptive training set reduction for nearest neighbor classification, *Neurocomputing* 138 (2014), 316–324.
- [20] C. Miranda, A.M. Jorge, Incremental collaborative filtering for binary ratings, *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*, Vol. 1, IEEE, Sydney, NSW, Australia, 2008, pp. 389–392.
- [21] J. Vinagre, A.M. Jorge, Forgetting mechanisms for scalable collaborative filtering, *J. Braz. Comput. Soc.* 18 (2012), 271–282.
- [22] B.M. Sarwar, G. Karypis, J. Konstan, J. Riedl, Incremental SVD-based algorithms for highly scaleable recommender systems, *Proceedings of the Fifth International Conference on Computer and Information Technology*, 2002, pp. 345–354.
- [23] M.W. Berry, S.T. Dumais, G.W. O'Brien, Using linear algebra for intelligent information retrieval, *SIAM Rev.* 37 (1995), 573–595.
- [24] E. Diaz-Aviles, L. Drumond, L. Schmidt-Thieme, W. Nejdl, Real-time top-n recommendation in social streams, *Proceedings of the Sixth ACM Conference on Recommender Systems*, 2012, pp. 59–66.
- [25] R.M. Bell, Y. Koren, Scalable collaborative filtering with jointly derived neighborhood interpolation weights, *Proceedings of the Seventh IEEE International Conference on Data Mining (ICDM)*, IEEE, Omaha, NE, USA, 2007, pp. 43–52.
- [26] G. Takács, I. Pilászy, B. Németh, D. Tikk, Scalable collaborative filtering approaches for large recommender systems, *J. Mach. Learn. Res.* 10 (2009), 623–656.
- [27] G. Ling, H. Yang, I. King, M.R. Lyu, Online learning for collaborative filtering, *Proceedings of the 2012 International Joint Conference on Neural Networks (IJCNN)*, IEEE, Brisbane, QLD, Australia, 2012, pp. 1–8.
- [28] S. Ruder, An overview of gradient descent optimization algorithms, *arXiv preprint arXiv:1609.04747*, 2016.
- [29] R. Battiti, Accelerated backpropagation learning: two optimization methods, *Complex Syst.* 3 (1989), 331–342.
- [30] R. Gemulla, E. Nijkamp, P.J. Haas, Y. Sismanis, Large-scale matrix factorization with distributed stochastic gradient descent, *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, New York, USA, 2011, pp. 69–77.
- [31] M.J. Pazzani, D. Billsus, Content-based recommendation systems. In: P. Brusilovsky, A. Kobsa, W. Nejdl (Eds.), *The Adaptive Web*, Springer, Berlin, Heidelberg, 2007, pp. 325–341.
- [32] G. Adomavicius, A. Tuzhilin, Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions, *IEEE Trans. Knowl. Data Eng.* 17 (2005), 734–749.
- [33] R. Burke, Hybrid recommender systems: survey and experiments, *User Model. User Adapt. Interact.* 12 (2002), 331–370.