An Application of Differential Evolution Algorithm-based Restricted Boltzmann Machine to Recommendation Systems

R. J. Kuo[1], J. T. Chen[2]

[1] Department of Industrial Management, National Taiwan University of Science and Technology, Taiwan
[2] Department of Big Data, Chunghwa Telecom Co., Ltd., Taiwan
rjkuo@mail.ntust.edu.tw; m10301001@mail.ntust.edu

Abstract

Due to growth of electronic commerce, currently, many customers prefer to buying products from the internet. Thus, recommendation system, like restricted Boltzmann machine (RBM), has become a good technique to recommend the right product to the potential customer. This can dramatically increase the customer loyalty. However, it is necessary to determine some parameters for RBM and enhance its computation performance. Therefore, this study intends to propose a hybrid algorithm which combines the cluster-based restricted Boltzmann machine (CRBM) with differential evolution (DE) algorithm to optimize the RBM's parameters for collaborative filtering. The CRBM applies a clustering algorithm to determine the size and elements for each mini-batch gradient descent method for the RBM. The proposed DE-based CRBM algorithm is validated using four benchmark datasets. The results are compared with those of batch RBM, mini-batch RBM, clustering RBM, PSO-based and GA-based clustering RBM. The experimental results reveal that optimizing the RBM's parameters using metaheuristic can obtain the better result. It also show that the proposed DE-based CRBM algorithm performs better than GA-based and PSO-based CRBM algorithms.

Keywords: Recommendation systems, Collaborative filtering, Restricted Boltzmann machine, *K*-means algorithm, Differential evolution algorithm

1 Introduction

The internet technology has facilitated the online activities including online shopping, streaming movie, online games, etc. Many companies have earned a lot of profits from the internet. Due to this success, many e-commerce companies compete to offer more varieties of products and let customers have more choices. This phenomena has caused customer information overloading [1]. Instead of giving useful information, too many information may lead to the paradox of choices. Therefore, choosing which information should be provided to the customers has become an important issue for most of companies. A recommendation system is a system which can be applied to solve this problem. It provides a recommendation or useful information to the customers based on their profiles and behaviors.

Nowadays, many applications which have huge number of items have applied the recommendation systems. This system provides the customers with a list of recommended items they might prefer. The system learns the customer behavior taken from the previous choices or the customer profiles, then recommends the items which have similar characteristics. The recommendation system also can provide the "rating predictions." For instance, the DVD rental Netflix displays the movie rating prediction. Google Play shows the "similar apps" when the user opens or downloads an app. It also shows the "Users also installed" which is the list of other apps download together by other users who installed the current apps. YouTube, Amazon, and Microsoft also apply the similar recommendation systems. Therefore, there have been many researches working on the recommendation systems. In general, the recommendation systems use two types of technologies. They are the content-based systems and collaborative filtering systems [2]. This study will only focus on the improvement of the collaborative filtering.

With advanced technology, computing power grows faster and faster. Algorithm can be designed as complex as human brain. Due to complex algorithm, more parameters come up in algorithm to let algorithm become more sophisticated and powerful, such as deep learning. Parameter setting has become a tough issue. In practice, most of parameter settings are set by experience. This paper overcomes this problem by combining clustering RBM algorithm with a metaheuristic algorithm, named differential evolution (DE) algorithm to optimize the RBMs' parameter and obtain a better prediction results.

The rest of this paper is organized as follows.

*Corresponding Author: R. J. Kuo; E-mail: rjkuo@mail.ntust.edu.tw
DOI: 10.3966/160792642020052103008

Section 2 briefly discuses some theories applied in this paper. Section 3 presents the proposed algorithm while Section 4 discusses the experiments conducted to validate the proposed algorithm. Finally, concluding remark is presented in Section 5.

2 Background

This section will briefly review the recommendation systems, restricted Boltzmann machine, differential evolutionary algorithm, and cluster analysis which will used in the current study.

2.1 Recommendation Systems

Internet and customization growing have motivated organizations to shift from the traditional world of mass production to the new world of customization with respect to their offerings [3]. Recommendation systems were introduced in 1992 when the first recommendation system, Tapestry, appeared [4]. Currently, recommendation systems have been widely applied to provide personalized information to the users according to users' behavior record. The recommendation systems has been applied in many different fields, such as on-line shopping, movies, music, and travel recommendations [5]. It not only offers personalized recommendation service and reduces the burden of information loading, but also raises the revenue. In general, the technologies applied in the recommendation systems can be grouped into two categories, content-based systems and collaborative filtering systems [6-7]. The content-based systems learn the users' behavior from the properties of the recommended items. On the other hand, collaborative filtering is to predict the interest of an active user based on the opinions of users with similar interests and it works by building a database of preference which is the user-item matrix for items by users [8]. Gan and Jiang [9] proposed a power function to adjust user similarity scores derived from ratings converted to binary to find relevant links between users and items. Kaššák et al. [10] proposed a method using cosine similarity between users for collaborative filtering and items features for content-based to generate top-N recommended items. Basically, collaborative filtering can be divided into three subsets: memory-based collaborative filtering, model-based collaborative filtering, and hybrid-based collaborative filtering [11]. Both of two methods have their own advantages. Thus, in order to cover both of advantages, hybrid method has been implemented. It uses different data including user-item matrix and content information as different aspects to build a probabilistic model [12]. However, there still have been some problems with recommendation systems.

The most well-known problem in recommendation systems is the cold start problem [13]. The cold start problem is related to user's preference or item's information is not available in the data set. Three situations in cold start problems: (a) recommend to new users, (b) recommend new items, and (c) recommend new items to new users. Many researchers tried to solve these problems, like using social tags to solve problem in item aspect [14] or using classification technique to solve cold start in user aspect [15]

2.2 Restricted Boltzmann Machine

Boltzmann machines have been proposed in 1985 [16]. It is then further improved by Smolensky [17] as the Restricted Boltzmann machine (RBM). Although RBM has been proposed for many years, in recent years, Hinton has revised this method and make it more powerful and useful. Nodaway, RBM has become a fundamental of deep learning's method. Restricted Boltzmann machine has been applied to many different fields including data preprocessing [18], text mining [19], image classification [20], and recommendation systems [21]. The application of RBM in recommendation systems was proposed by Salakhutdinov et al. [21] to solve the Netflix challenge recommendation's problems [22].

RBM is a special type in Boltzmann machines. Boltzmann machine can be introduced as a symmetrically connected network of stochastic processing units which can be seen as a type of neural network model [16]. A Boltzmann machine can be used to learn the importance of an unknown data's distribution based on sampling data. These sampling observations are used to train the model. Training Boltzmann machine can be seen as adjusting its parameters by learning probability distribution of data. However, learning a Boltzmann machine is computationally demanding. The learning problem can be solved by implementing restriction concept on the network. In Boltzmann machines, two types of units can be used. They can have visible units and potentially hidden units. RBMs always have both types of units, and these would be arranged in two layers.

In the RBMs network graph, each visible unit would connect to all the hidden units which units are on. However, there are no connections between units in the same layer, and this restriction gives the RBM its name.

In the RBM, visible unit can be represented as binary data and it is connected with hidden unit with symmetric weight. And its joint configuration, (v, h), of the visible and hidden units has an energy [23] given by:

$$E(v,h) = -\Sigma_{i \in visible} a_i v_i - \Sigma_{j \in hidden} b_j b_j - \Sigma_{i,j} v_i h_j w_{ij}, \quad \textbf{(1)}$$

where v_i and h_j are the binary states of visible unit i and hidden unit j. The a_i and b_j are visible unit's biases and hidden unit's biases, while w_{ij} is the weight connecting visible and hidden units. The network assigns a probability to every possible pair of a visible vector and a hidden vector through energy function:

$$p(v,h)=\frac{1}{z}e^{-E(v,h)}, \quad \textbf{(2)}$$

where Z is given by summing over all possible pair of visible and hidden units:

$$Z=\Sigma_{v,h}e^{-E(v,h)}.$$
$$Z=\Sigma_{v,h}e^{-E(v,h)}. \quad \textbf{(3)}$$

The probability that the network assigns to a visible unit, v, is given by summing over all hidden units:

$$p(V)=\frac{1}{z}\Sigma_{h}e^{-E(v,h)}. \quad \textbf{(4)}$$

The derivative of the log probability of a training data with respect to a weight:

$$\frac{\partial \log p(V)}{\partial w_{ij}}=<v_ih_j>_{data}-<v_ih_j>_{model}, \quad \textbf{(5)}$$

where the angle brackets are used to represent expectations of specific distribution specified by script below. So, the learning rule can be easily preforming log probability of the training data by following equation:

$$\Delta w_{ij}=\varepsilon<v_ih_j>_{data}-<v_ih_j>_{model}, \quad \textbf{(6)}$$

where ε is a learning rate

Because hidden layer has no direct connection between each other in RBM, so it can get an unbiased sample of $<v_ih_j>_{data}$ easily. Given a randomly selected training data, υ, the binary state, h_j, of each hidden unit, j, is set to 1 with probability

$$p(h_j=1\,|\,v)=\sigma(b_j+\Sigma_i v_i w_{ij})=p(h_j=1\,|\,v)$$
$$=\sigma(b_j+\Sigma_i v_i w_{ij}), \quad \textbf{(7)}$$

where $\sigma(_X)$ is the logistic sigmoid function.

Also visible units do not have direct connections between each other in an RBM, so can easily get an unbiased sample of visible unit, when given hidden unit

$$p(v_i=1\,|\,v)=\sigma(a_i+\Sigma_j h_j w_{ij}). \quad \textbf{(8)}$$

However, it is difficult to get unbiased sample of $<v_ih_j>_{model}$. There is a faster learning way to get $<v_ih_j>_{model}$ proposed by Hinton [19]. Start by stetting visible units. Then hidden unit can be computed by equation 7. When hidden unit has been computed, a procedure called "reconstruction" would be triggered. It will compute visible unit by equation 8. Then update weight using equation 6. This kind of learning rule is much more closely approximating the gradient of another objective function called the Contrastive Divergence [24].

2.3 Differential Evolutionary Algorithm

In 1995, Storn and Price proposed the concept of the differential evolution (DE) algorithms [25]. DE algorithm is a population-based algorithm and is similar to GA. It uses concepts of mutation, cross over, choose, and recombination to improve the initial solution. With random initial solutions, it can generate a set of new vectors, which have the best solution so far, to get optimal solution and to avoid converging on the local minimum. Because of crossing over the group of best solution to replace some vector parameters, it can usually get better solution in each iteration. DE algorithm has been proven to has better performance than PSO and GA in 34 widely used benchmark [26]. Therefore, this paper combines DE algorithm with RBM to solve RBM's problem. The main steps are: (1) generate vectors randomly, (2) mutate vectors randomly and use cross over rate to decide whether mutate, and (3) Compute fitness and decide whether keep new vectors or not.

2.4 Cluster Analysis

Cluster analysis is a technique to explore data insight. Cluster analysis aims to separate data into a set of meaningful groups. In other words, clustering is using unsupervised data to classify which instances should be classified into some clusters [27]. A good clustering method will produce good result which has low similarity between each group and high similarity between group members [28]. Through clustering technique, it can easily help people understand data structure or explore data. Clustering has been applied in several fields, such as customer segmentation [29], image recognition [30], medicine [31], and industrial engineering [32].

3 Methodology

This section presents the proposed DE-based CRBM algorithm with parameters optimization in details.

3.1 Differential Evolution-based Cluster RBM Algorithm

This paper applies DE algorithm as the main method to optimize the RBMs' parameters. Firstly, the algorithm is used to find the initial parameters solution (Learning, Momentum, Weight decay) represented as the DE population. Secondly, RBM algorithm is applied to find each new solution obtained by the mutation, crossover and selection operators of the DE algorithm. Figure 1 shows the framework of the proposed algorithm.

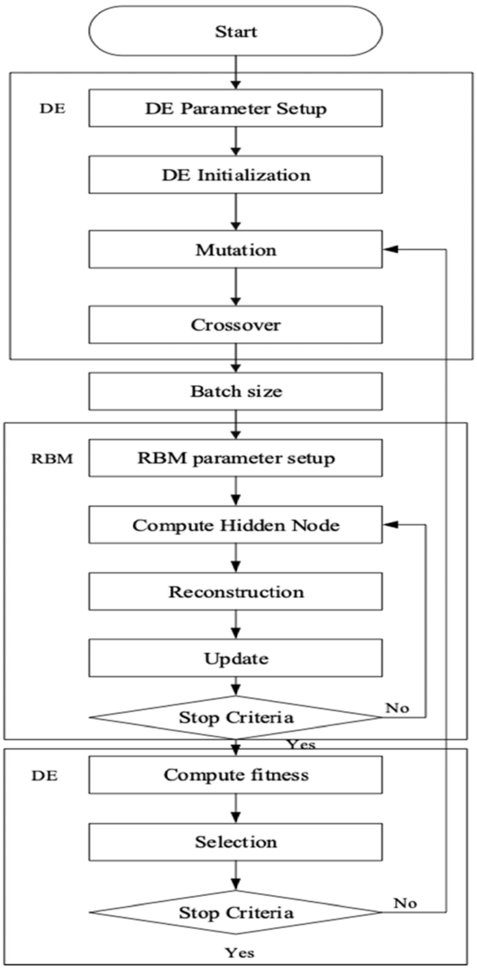


Figure 1. DE-based RBM algorithm

Since the proposed algorithm employs DE algorithm to find the best RBMs' parameters, the chromosome of DE algorithm comprises of the RBMs' parameters as illustrated in Figure 2. Detailed procedure of the proposed DE-based RBM algorithm is as the follows.

Learning Rate	Momentum	Weight decay
Number of Hidden nodes	F	CR

Figure 2. Solution representation

Step 1: Setup DE parameters

Set up parameters in differential evolution including population size. For mutation scale constant factor (F) and the crossover rate (CR), they will be determined by DE itself.

Step 2: Initialize DE

Initialize each particle with position $(X_{D,i,t})$ randomly. A population vector can be presented as:

$$X_{D,i,t} = (X_{1,i,t}, ..., X_{D,i,t}),$$
$$X_{D,i,t} = (X_{1,i,t}, ..., X_{D,i,t}), \quad \textbf{(9)}$$

where D is the number of population size.

Step 3: Implement mutation

The purpose of mutation is to expand the search space. First, select two different vectors $X_{\gamma 1}$ and $X_{\gamma 2}$ from population vector randomly. A new mutant vector $V_{i,t+1}$ is generated as follows:

$$V_{i,t+1} = X_{i,} + F(X_{\gamma 1} - X_{\gamma 2}), \quad \textbf{(10)}$$

where F is the mutation factor between (0, 1).

Step 4: Implement crossover

Crossover is the way to increase the diversity of the vectors and have chance to jump out local optimal.

$$U_{i,t+1} = (U_{1,i,t+1}, ..., U_{D,i,t+1}),$$
$$U_{i,t+1} = (U_{1,i,t+1}, ..., U_{D,i,t+1}), \quad \textbf{(11)}$$

$$\begin{cases} U_{j,i} = V_{j,i} \ i \ frand \le CRorj = rand(j) \\ U_{j,i} = X_{j,i} \ otherwise \end{cases} \quad \textbf{(12)}$$

In equation (12), rand in a uniform random number generator with outcome $\in [0,1]$. CR is crossover rate $\in [0,1]$ and $rand(j)$ is a randomly chosen index $\in [1, 2 ... D]$ which make sure $U_{j,i}$ gets at least one parameter from $V_{j,i}$. Crossover rate would determine the changing over rate with optimal vector and synthetic vector after disturbance.

Step 5: Setup batch and batch size

Due to difficulty of setting batch and batch size, the proposed method uses *K*-means algorithm to decide which data should be in the same batch and batch size.

Step 6: RBM parameter setup

Each DE vector contains RBM's parameter value including: learning rate, momentum, and weight decay. Besides these parameters, batch size and number of hidden nodes are also set in this step.

Step 7: Compute hidden nodes

Given training data, υ, the binary state, h_j, of each hidden unit j, is set to 1 with probability:

$$p(h_j = 1 \mid v) = \sigma p(b_j + \Sigma_i v_i w_{ij}), \quad \textbf{(13)}$$

where $\sigma(x)$ is the logistic sigmoid function.

To approximate the model's distribution, this step uses gibbs sampling to approximate the model's distribution

Step 8: Reconstruction

Because visible units do not have direct connections between each other in an RBM, so we can easily get an unbiased sample of visible unit, when given hidden unit

$$p(v_i = 1 \mid v) = \sigma(a_i + \sum_j h_j w_{ij}). \quad \textbf{(14)}$$

Step 9: Update

When computing the approximation of model distribution value, we will use previous steps result to update variables.

$$V_t = Momentum \times V_{t-1} + \varepsilon(\Delta w_{ij} - decay \times w_{ij}). \quad \textbf{(15)}$$

$$W_{t+1} = W_t + V_t. \quad (16)$$

$$Vh_t = Momentum \times Vh_{t-1} + \varepsilon(\Delta h_j - decay \times h_j). \quad (17)$$

$$b_{t+1} = b_t + Vh_t \cdot b_t = b_t + Vh_t. \quad (18)$$

$$Vh_t = Momentum \times Vv_{t-1} \cdot \varepsilon(\Delta v_i - decay \times v_i). \quad (19)$$

$$a_{t+1} = a_t + Vv_t. \quad (20)$$

where v_i and h_j are the binary states of visible unit i and hidden unit j. The a_i and b_j are visible unit's biases and hidden unit's biases, while w_{ij} is the weight connect visible and hidden. t is iteration, and V is the velocity of each variable.

Step 10: Compute Fitness

This step computes DE's fitness, and would use testing's MAE as fitness to be a criteria that DE weather select.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |, \quad (21)$$

where y_i is the real value of data and $\hat{y}_i$ is the forecasting value of data.

Step 11: Implement selection

Selection operation preserves the better solution in each iteration. In order to decide where the vector should become a member of the next iteration, the vector $U_{i,t+1}$ is compared with target vector x_i using the greedy criteria. If vector $U_{i,t+1}$'s fitness is smaller than x_i's fitness, then x_i change to $U_{i,t+1}$; otherwise, x_i will not change.

$$x_{i,t+1} = \begin{cases} U_{i,t+1} & iff(U_{i,t+1}) < f(x_{i,t}). \\ x_{i,t}, & otherwise \end{cases} \quad (22)$$

Step 12: Check if the number of iterations is satisfied

Stop if the stopping criterion is satisfied; otherwise, go back to Step 3.

4 Numerical Illustrations

This section will illustrate computational results using some benchmark datasets. The detailed discussion is as follows.

4.1 Datasets

In order to verify the proposed DE-based CRBM algorithm, four real-world datasets from Grouplens, Books-Crossing Community and UCI are employed. The first dataset, MovieLens 100k, consists of 100,000 ratings for 1,682 movies by 943 users, while the second one, MovieLens 1M, contains one million ratings for 3,952 movies by 6,040 users. Each rating is an integer between 1 (worst) and 5 (best). The Book-Crossing (BX) contains 278,858 users (anonymized but with demographic information) providing 1,149,780 ratings (explicit / implicit) about 271,379 books. Explicit means user rated 1(worst) to 10(best) and implicit means user interacted with an item. The Restaurant dataset contains 1161 ratings for 130 restaurants by 138 users. Each rating is an integer between 0 (worst) and 2 (best). In order to predict active users and items, Movielens 1M and Book-Crossing will be preprocessed. In order to define active users and items, this paper computes the frequencies of users rating times and items rating. This step is conducted since this paper more focuses on popular movies and frequent users. Table 1 shows the summary of each dataset.

Table 1. The characteristics of each dataset

Feature \ Datasets	MovieLens 100K	MovieLens 1M	Book-Crossing	Restaurants
Number of items	1,682	1000	1000	130
Number of users	943	1000	500	138
Total Ratings	100,000	358,793	8,867	1,161
Rating Interval	[1, 5]	[1, 5]	[1, 10]	[0, 2]

4.2 Performance Measurement

This study uses benchmark datasets to test the proposed recommendation algorithm's performance. In order to validate the performance, 5-fold cross validation is applied. This paper uses the MAE as the performance measurement indicator.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} | y_i - \hat{y}_i |, \quad (23)$$

where y_i is the real value of data and $\hat{y}_i$ is the forecasting value of data.

4.3 Parameter Setting

In this paper, the proposed algorithm is compared with Batch RBM (non-grouping) [28], Mini-batch RBM (grouping) [21], and cluster RBM (cluster-grouping). Batch RBM applies batch learning in its gradient descent method. When update RBM's weight and bias, it use all data as an instance to update. Mini-batch RBM means that its gradient descent method use mini-batch learning. When update RBM's weight and bias, it use separate all into several batch and use a batch as an instance to update. Cluster RBM means that its gradient descent method use mini-batch learning where its batch parameter is decide by clustering technique. Batch RBM's parameter setting follows the research of [33]. Mini batch RBM's parameter setting is from RBM's inventor work [34], while cluster RBM also follows RBM's inventor work but the batch setting follows clustering technique to decide batch number and each batch's member. In the proposed algorithm, number of particles or chromosomes is set as 20 and number of iterations is set as 20. Table 2 shows the details of the parameter setting for each method for different datasets.

Table 2. The parameters of MovieLens 100K, 1M, and Restaurants

Parameters \ Method	Batch RBM	Mini-Batch RBM	Cluster RBM
The number of Hidden unit	50	100	100
Momentum	0.6	0.5	0.5
Weight Decay	0.0002	0.0001	0.0001
Learning Rate	0.05	0.01	0.01
Batch number	1	10	Decide by clustering algorithm

For the Book-crossing dataset, the experiment results using parameter setting in Table 2 are not quite satisfying. Thus, further experiments are conducted using different level of the momentum and the learning rate. They levels tested for this dataset are 0.9, 0.8, 0.6, 0.5, 0.3, and 0.1. Table 3 shows the best parameter setting for Book-crossing dataset.

Table 3. The parameters of Books-crossing

Parameters \ Method	Batch RBM	Mini-Batch RBM	Cluster RBM
The number of Hidden units	50	100	100
Momentum	0.9	0.9	0.9
Weight Decay	0.0002	0.0001	0.0001
Learning Rate	0.5	0.1	0.6
Batch number	1	10	Decided by clustering algorithm

Furthermore, this paper decides the number of clusters for the *K*-means algorithm based on the sum-of-square error (SSE) and silhouette coefficient. Figure 3 to Figure 6 illustrate the silhouette coefficient values and SSE curves for each dataset. Table 4 shows the number of clusters based on the silhouette coefficient and SSE curves.

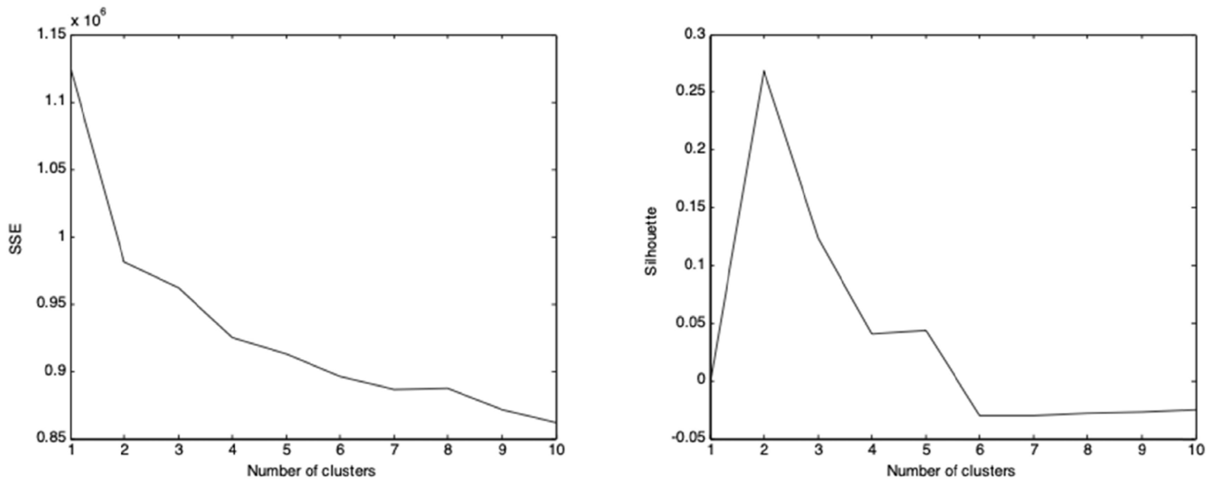


Figure 3. Silhouette coefficient curve and SSE curve for MovieLens 100K

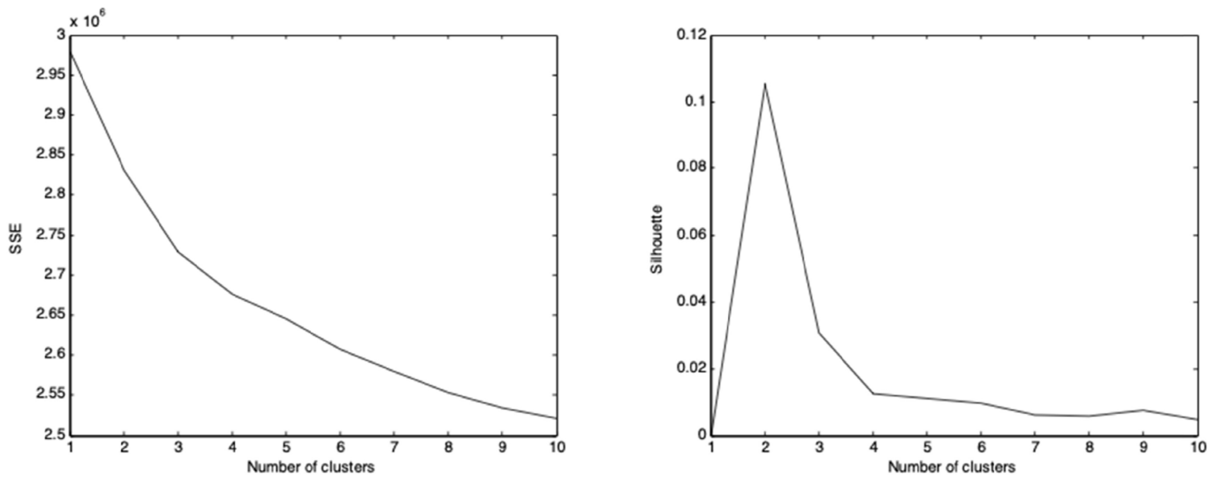


Figure 4. Silhouette coefficient curve and SSE curve for MovieLens 1M

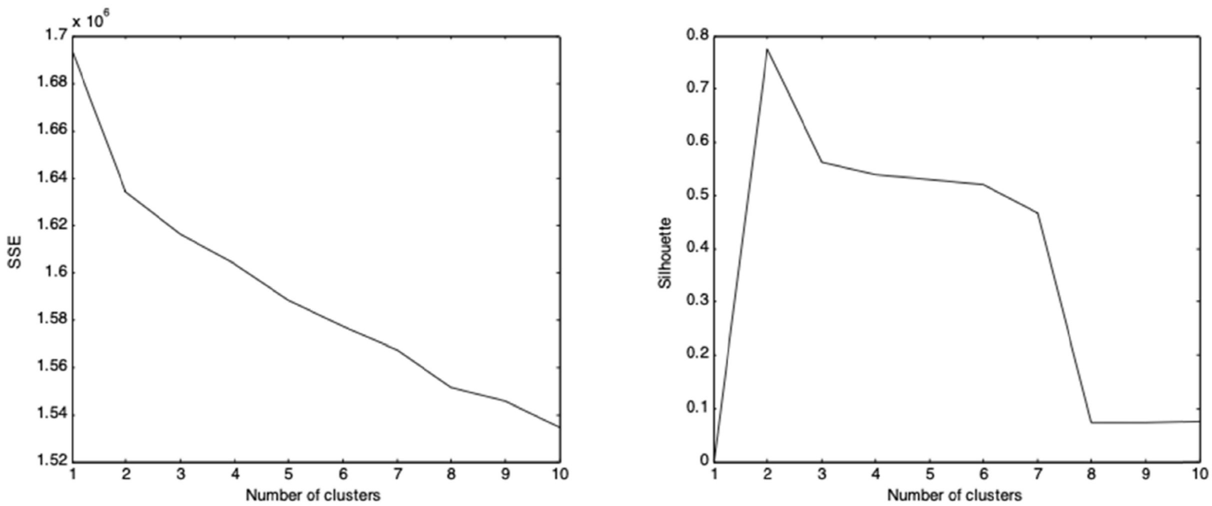


Figure 5. Silhouette coefficient curve and SSE curve for Book-Crossing

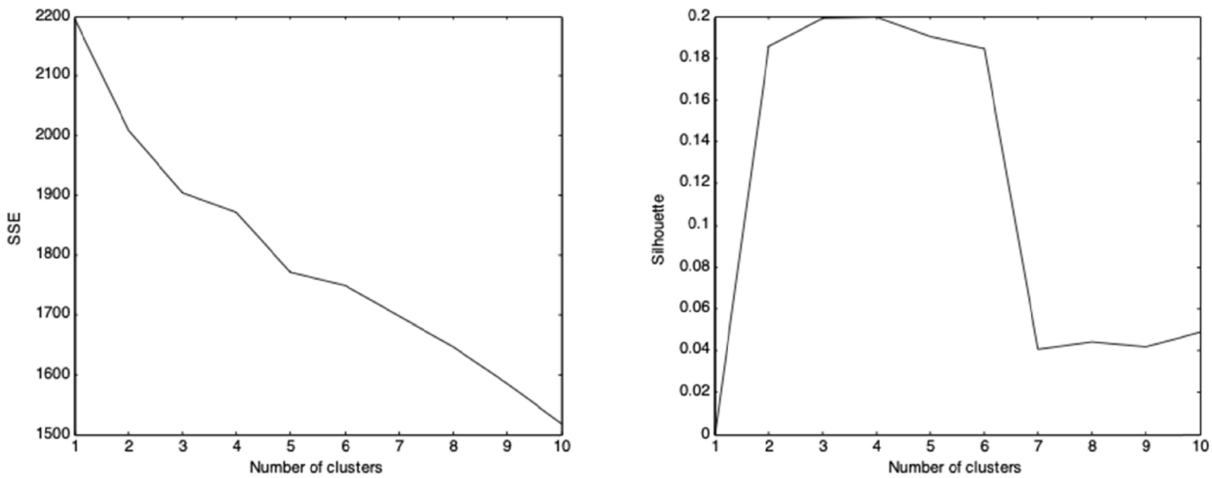


Figure 6. Silhouette coefficient curve and SSE curve for Restaurant

Table 4. The optimal K for each dataset

Datasets Parameters	MovieLens 100K	MovieLens 1M	Book-Crossing	Restaurants
Optimal K	2	2	2	3

4.4 Computational Results

In this study, every algorithm was run 10 times for every dataset. The evaluation is made based on the average of mean absolute error (MAE). The proposed methods are compared with Batch RBM, Mini-Batch RBM, CRBM, GA-based CRBM, and PSO-based CRBM. Tables 5 and 6 show the MAE values for training and testing for every dataset.

According to the result for MovieLens 100K, the proposed metaheuristic-based algorithms are better than the other algorithms without metaheuristics. Among all of the tested algorithm, DE-based CRBM can provide the best results for both training and testing data. The DE-based CRBM obtains better average MAE than other algorithms. Although the best result is obtained by GA-based CRBM, the differences between the best result of DE-based CRBM and GA-based CRBM is not significant. In terms of standard deviation, BRBM and MRBM algorithms are more stable than other algorithms.

For MovieLens 1M dataset, the proposed DE-based CRBM algorithm is also superiors than other algorithms in terms of average, the best, and the worst MAE values for both training and testing sets. Similar results also are happened for Book-crossing and Restaurant datasets. The proposed DE-based CRBM also performs better than other algorithms. Unlike the first three datasets, for the Restaurant dataset, DE-based CRBM algorithm not only has smaller MAE, but also the most stable result.

In overall, the experimental results reveal that the combining metaheuristics and RBM algorithm provides better results than only RBM algorithms. It is because the RBM's parameters have significant influence to the result. Thus, providing a better parameter setting for RBM is important. Furthermore, among three different metaheuristic-based RBM algorithms tested in this paper, DE-based CRBM algorithm is more promising than GA-based and PSO-based CRBM algorithms.

A further analysis is made based on the convergence histories of each algorithm. This paper uses convergence histories of MovieLens 100K dataset to illustrate the convergence analysis. Figure 7 and Figure 8 indicate that BRBM and MRBM algorithms converge faster than CRBM. For metaheuristic-based RBM algorithms, all of them can converge relatively faster than BRBM, MRBM and CRBM algorithms.

5 Conclusions

Nowadays, more consumers begin to change their consumption habit. They begin to rely on online-shopping instead of visiting physical stores. Due to this trend, more companies use recommendation systems to increase consumer's loyalty. Recommendation systems not only increase the revenue for company but also reduce consumer's information burden. Therefore, prediction on recommendation systems becomes more and more important. This study has proposed a novel RBM algorithm which combines DE, *K*-Means and RBM algorithms.

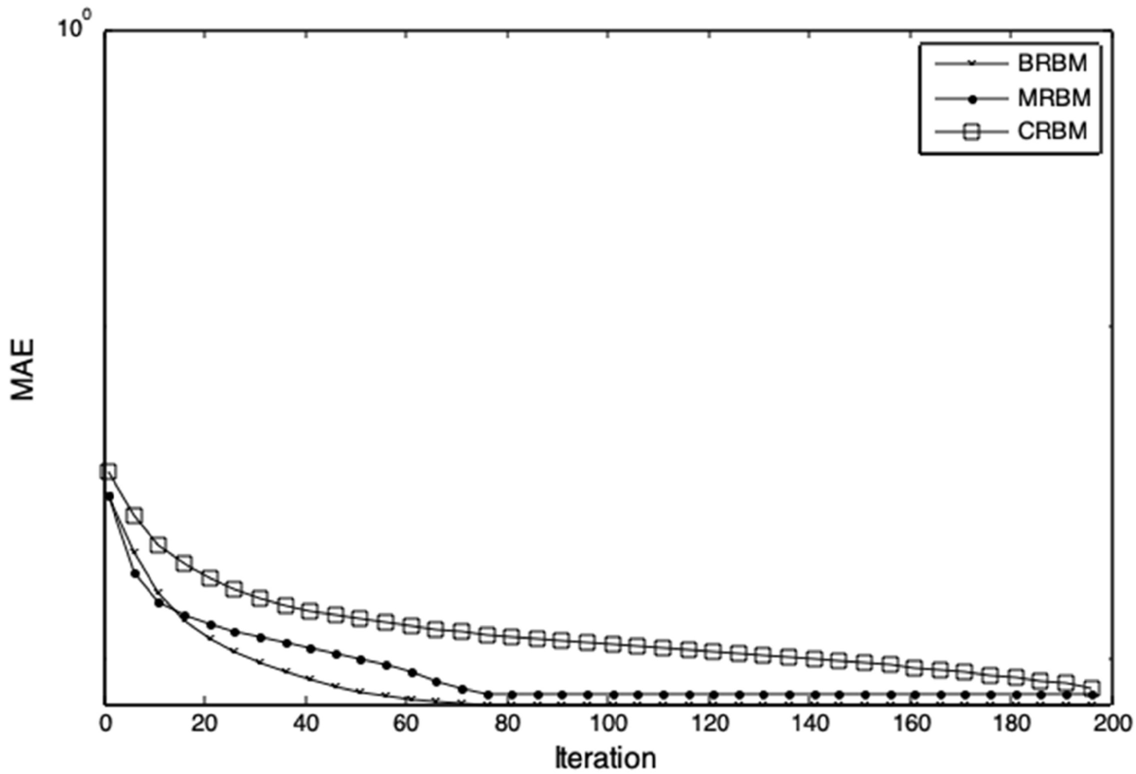


Figure 7. Convergence curves of RBM Method for MovieLens 100K dataset

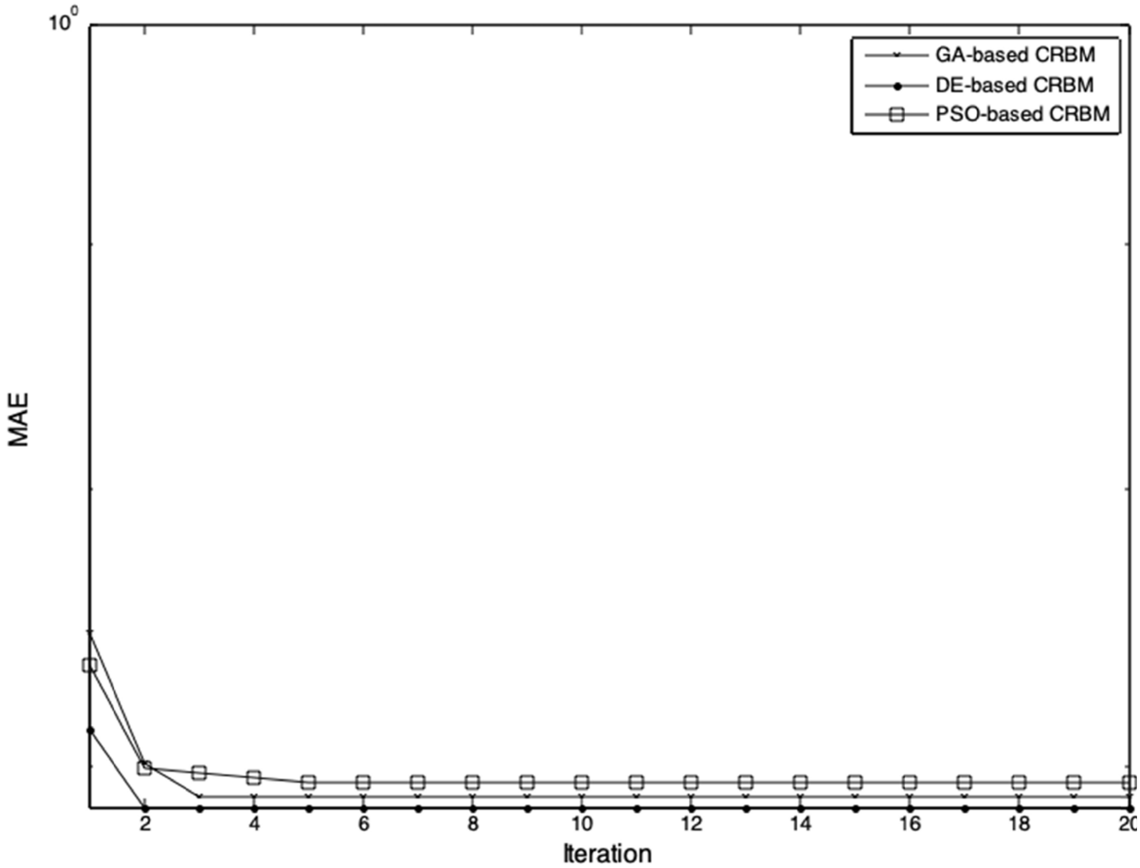


Figure 8. Convergence curves of Metaheuristic-based algorithm for MovieLens 100K dataset

The RBM involves some parameters which influence the result. The proposed algorithm aims to provide the best parameter setting for the RBM so then it can obtain a better result. The proposed algorithm is verified using four datasets from Grouplens, Books-Crossing Community, and UCI machine learning repository. The results are compared with two original RBM algorithms including Batch RBM, Mini Batch RBM. The experiment results reveal that metaheuristic-based RBM algorithm is superior to only RBM-based algorithms. It is because metaheuristics algorithm successfully provides the best parameters for the RBM algorithm. Furthermore, among three metaheuristics-based RBM algorithm tested in this paper, DE-based CRBM algorithm is the most promising algorithm. It is shown from the average, the best and the worst results which are always better than both GA-based CRBM and PSO-based CRBM algorithms. The proposed DE-based Cluster RBM can enhance the capability of searching optimal parameters and avoid local optimal. Furthermore, this paper proposes to apply clustering algorithm to determine the batch size and member. However, in the future study, different metaheuristics can also be applied.

The proposed DE-based CRBM algorithm does not aim to solve the data with sparsity. However, the real world applications might face this problem. Thus, instance selection and non-active user problems should be considered. Furthermore, The RBM algorithm in this paper uses sigmoid function as the active function. In some cases, using sigmoid function might squeeze the output of high value input. Thus, further study using other active functions should be evaluated. In addition, dynamic learning rate also should be tested instead of static learning rate. In addition, the proposed method can be applied to different areas such as web page recommendation [35].

Acknowledgements

The current study is partially supported by Minister of Science and Technology of Taiwanese Government under contract number: MOST105-2410-H-011-017-MY3. The authors would like to appreciate this support.

Table 5. The computation results of training MAE for four datasets

Methods		MovieLens 100K	MovieLens 1M	Books-Crossing	Restaurants
BRBM	Average	0.78378	0.75267	1.09181	0.28232
	Best	0.78348	0.75266	1.08464	0.28077
	Worst	0.78410	0.75270	1.09784	0.28343
	SD	0.00022	0.00001	0.00446	0.00091
MRBM	Average	0.78127	0.76207	1.21202	0.13334
	Best	0.69462	0.76207	1.21200	0.13200
	Worst	0.79363	0.76208	1.21205	0.13503
	SD	0.03081	**0.00000***	**0.00002***	0.00092
CRBM	Average	0.77991	0.74559	1.26793	0.21853
	Best	0.77839	0.74555	1.26788	0.21726
	Worst	0.78110	0.74567	1.26801	0.21941
	SD	**0.00079***	0.00004	0.00004	**0.00068***
DE-based CRBM	Average	**0.58400***	**0.68607***	**0.30909***	**0.07587***
	Best	0.56101	**0.67674***	**0.24217***	**0.05951***
	Worst	0.61696	**0.70124***	**0.39188***	**0.09417***
	SD	0.01584	0.00692	0.04132	0.00996
GA-based CRBM	Average	0.65339	0.69217	0.36604	0.10492
	Best	0.57181	0.67537	0.33064	0.07961
	Worst	0.69919	0.70750	0.44141	0.12696
	SD	0.04256	0.01015	0.03228	0.01523
PSO-based CRBM	Average	0.61875	0.69970	0.35675	0.11829
	Best	**0.49310***	0.68819	0.25433	0.09774
	Worst	**0.66040***	0.71479	0.41966	0.21990
	SD	0.04793	0.00738	0.04888	0.03659

**the best result*

Table 6. The computation results of testing MAE for four datasets

Methods		MovieLens 100K	MovieLens 1M	Books-Crossing	Restaurants
BRBM	Average	0.81066	0.761982	1.39630	0.47449
	Best	0.81009	0.759175	1.39394	0.47164
	Worst	0.81116	0.764264	1.39871	0.47578
	SD	0.00028	0.001916	0.00175	0.00115
MRBM	Average	0.81436	0.764776	1.42247	0.38211
	Best	0.81423	0.761439	1.42238	0.37852
	Worst	0.81450	0.766844	1.42252	0.38624
	SD	0.00009	0.002027	**0.00004***	0.00210
CRBM	Average	0.81301	0.76024	1.50070	0.43144
	Best	0.81285	0.757568	1.50061	0.42963
	Worst	0.81320	0.762214	1.50081	0.43370
	SD	**0.00010***	**0.001704***	0.00005	**0.00108***
DE-based CRBM	Average	**0.69629***	**0.699084***	**1.19188***	**0.28712***
	Best	**0.67643***	**0.678071***	1.17747	**0.27364***
	Worst	**0.70820***	**0.701794***	**1.20611***	**0.30197***
	SD	0.00954	0.01637	0.01017	0.00725
GA-based CRBM	Average	0.71443	0.70965	1.22065	0.31317
	Best	0.69436	0.68774	**1.20365***	0.29180
	Worst	0.74942	0.732075	1.24525	0.32926
	SD	0.01654	0.018708	0.01394	0.01195
PSO-based CRBM	Average	0.70611	0.70822	1.19843	0.32417
	Best	0.69796	0.68578	1.17720	0.31441
	Worst	0.71879	0.73605	1.21754	0.33296
	SD	0.00642	0.02016	0.01090	0.00533

**the best result*

References

[1] D. M. Levy, Information Overload, *The Handbook of Information and Computer Ethics*, pp. 497, May, 2008.

[2] A. Rajaraman, J. D. Ullman, *Minig of Massive Datasets*, Cambridge University Press, 2012.

[3] J. B. Schafer, J. Konstan, J. Riedl, Recommender Systems in e-commerce, *The 1st ACM Conference on Electronic Commerce*, Denver, Colorado, 1999, pp. 158-166.

[4] I. Portugal, P. Alencar, D. Cowan, The Use of Machine Learning Algorithms in Recommender Systems: A Systematic Review, *Expert Systems with Applications*, Vol. 97, pp. 205-227, May, 2018.

[5] H. Su, K. Zheng, J. Huang, H. Jeung, L. Chen, X. Zhou, Crowdplanner: A Crowd-based Route Recommendation System, *IEEE 30th International Conference on Data Engineering*, Chicago, IL, USA, 2014, pp. 1144-1155.

[6] M. Balabanović, Y. Shoham, Fab: Content-based, Collaborative Recommendation. *Communications of the ACM*, Vol. 40, No. 3, pp. 66-72, March, 1997.

[7] J. Lu, D. Wu, M. Mao, W. Wang, G. Zhang, Recommender System Application Developments: A Survey, *Decision Support Systems*, Vol. 74, pp. 12-32, June, 2015.

[8] M. K. Najafabadi, A. H. Mohamed, M. N. R., A Survey on Data Mining Techniques in Recommender Systems, *Soft Computing*, Vol. 23, No. 2, pp. 627-654, November, 2017.

[9] M. Gan, R. Jiang, Improving Accuracy and Diversity of Personalized Recommendation through Power Law Adjustments of User Similarities, *Decision Support Systems*, Vol. 55, No. 3, pp. 811-821, June, 2013.

[10] O. Kaššák, M. Kompan, M. Bieliková, Personalized Hybrid Recommendation for Group of Users: Top-N Multimedia Recommender, *Information Processing & Management*, Vol. 52, No. 3, pp. 459-477, May, 2016.

[11] F. O. Isinkaye, Y. O. Folajimi, B. A. Ojokoh, Recommendation Systems: Principles, Methods and Evaluation, *Egyptian Informatics Journal*, Vol. 16, No. 3, pp. 261-273, November, 2015.

[12] A. Popescul, D. M. Pennock, S. Lawrence, Probabilistic Models for Unified Collaborative and Content-based Recommendation in Sparse-data Environments*, The Seventeenth Conference on Uncertainty in Artificial Intelligence, Seattle, Washington*, August, 2001, pp. 437-444.

[13] A. I. Schein, A. Popescul, L. H. Ungar, D. M. Pennock, Methods and Metrics for Cold-start Recommendations, *The 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Tampere, Finland, August, 2002, pp. 253-260.

[14] Z.-K. Zhang, C. Liu, Y.-C. Zhang, T. Zhou, Solving the Cold-start Problem in Recommender Systems with Social Tags, *Europhysics Letters*, Vol. 92, No. 2, pp. 28002, November, 2010.

[15] B. Lika, K. Kolomvatsos, S. Hadjiefthymiades, Facing the Cold Start Problem in Recommender Systems, *Expert Systems with Applications*, Vol. 41, No. 4, Part 2, pp. 2065-2073, March, 2014.

[16] D. H. Ackley, G. E. Hinton, T. J. Sejnowski, A Learning Algorithm for Boltzmann Machines, *Cognitive Science*, Vol. 9, No. 1, pp. 147-169, January-March, 1985.

[17] P. Smolensky, Information Processing in Dynamical Systems: Foundations of Harmony Theory, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, Vol. 1: Foundations, pp. 194-281, February, 986.

[18] G. E. Hinton, R. R. Salakhutdinov, Reducing the Dimensionality of Data with Neural Networks, *Science*, Vol. 313, No. 5786, pp. 504-507, July, 2006.

[19] P. Xie, Y. Deng, E. Xing, Diversifying Restricted Boltzmann Machine for Document Modeling, *The 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Sydney, Australia, August, 2015, pp. 1315-1324.

[20] J. Wang, W. Wang, R. Wang, W. Gao, Image Classification Using RBM to Encode Local Descriptors with Group Sparse Learning, *IEEE International Conference on Image Processing*, Quebec City, QC, Canada, September, 2015, pp. 912-916.

[21] R. Salakhutdinov, A. Mnih, G. Hinton, Restricted Boltzmann Machines for Collaborative Filtering, T*he 24th International Conference on Machine Learning*, Corvalis, Oregon, June, 2007, pp. 791-798.

[22] A. Fischer, C. Igel, Training Restricted Boltzmann Machines: An Introduction, *Pattern Recognition*, Vol. 47, Vol. 1, pp. 25-39, January, 2014.

[23] J. J. Hopfield, Neural Networks and Physical Systems with Emergent Collective Computational Abilities, *The National Academy of Sciences*, Vol. 79, No. 8, pp. 2554-2558, April, 1982.

[24] G. E. Hinton, Training Products of Experts by Minimizing Contrastive Divergence, *Neural Computation*, Vol. 14, No. 8, pp. 1771-1800, August, 2002.

[25] R. Storn, K. Price, *Differential Evolution- A Simple and Efficient Adaptive Scheme for Global Optimization over Continuous Spaces*, Vol. 3: ICSI Berkeley, 1995.

[26] Q. Duan, T. W. Liao, H. Yi, A Comparative Study of Different Local Search Application Strategies in Hybrid Metaheuristics, *Applied Soft Computing*, Vol. 13, No. 3, pp. 1464-1477, March, 2013.

[27] P.-N. Tan, M. Steinbach, V. Kumar, *Introduction to Data Mining*, Pearson Education, 2014.

[28] A. K. Jain, M. N. Murty, P. J. Flynn, Data Clustering: A Review, *ACM Computing Surveys*, Vol. 31, No. 3, pp. 264-323, September, 1999.

[29] I. Bose, X. Chen, Detecting the Migration of Mobile Service Customers Using Fuzzy Clustering, *Information & Management*, Vol. 52, No. 2, pp. 227-238, March, 2015.

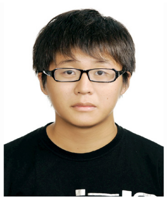
[30] Z. Feng, B. Zhang, Fuzzy Clustering Image Segmentation Based on Particle Swarm Optimization, *Telecommunication Computing Electronics and Control*, Vol. 13, No. 1, pp. 128-136, March, 2015.

[31] A. Nikfarjam, A. Sarker, K. O'Connor, R. Ginn, G. Gonzalez, Pharmacovigilance from Social Media: Mining Adverse Drug Reaction Mentions Using Sequence Labeling with Word Embedding Cluster Features, *Journal of the American*

Medical Informatics Association, Vol. 22, No. 3, pp. 671-681, May, 2015.
[32] R. Kuo, Y. Syu, Z.-Y. Chen, F.-C. Tien, Integration of Particle Swarm Optimization and Genetic Algorithm for Dynamic Clustering, *Information Sciences*, Vol. 195, pp. 124-140, July, 2012.
[33] K. Georgiev, P. Nakov, A Non-iid Framework for Collaborative Filtering with Restricted Boltzmann Machines, *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, 2013.
[34] G. Hinton, A Practical Guide to Training Restricted Boltzmann Machines, *Momentum*, Vol. 9, No. 1, pp. 926, 2010.
[35] T. Gopalakrishnan, P. Sengottuvelan, A. Bharathi, R. Lokeshkumar, An Approach to Webpage Prediction Method Using Variable Order Markov Model in Recommendation Systems, *Journal of Internet Technology*, Vol. 19, No. 2, pp. 415-424, March, 2018.

Biographies

R. J. Kuo received the M.S. degree in Industrial and Manufacturing Systems Engineering from Iowa State University, Ames, IA, in 1990 and the Ph.D. degree in Industrial and Management Systems Engineering from the Pennsylvania State University, University Park, PA, USA in 1994. Currently, he is the Distinguished Professor in the Department of Industrial Management at National Taiwan University of Science and Technology, Taiwan. He has published more than 100 papers in international journals. His research interests include architecture issues of computational intelligence and their applications to intelligent manufacturing, intelligent retailing, electronic business, logistics and supply chain management and decision support systems.



J. T. Chen received the M.S. degree in Industrial Management from National Taiwan University of Science and Technology, Taiwan. Currently, he works for Chunghwa Telecom Co., Ltd as Data Scientist. His research interests include architecture issues of computational intelligence and their applications to big data analytics, data mining and machine learning. He applied data science and machine learning technique on telecommunication's data to reduce on-line advertising cost.