ARTICLE IN PRESS

Journal of King Saud University – Computer and Information Sciences xxx (2018) xxx–xxx

Contents lists available at ScienceDirect

Journal of King Saud University – Computer and Information Sciences

journal homepage: www.sciencedirect.com

Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization

Sandeep Kumar Raghuwanshi *, Rajesh Kumar Pateriya

Computer Science & Engineering, Maulana Azad National Institute of Technology Bhopal, M.P., India

ARTICLE INFO

Article history:
Received 19 January 2018
Revised 26 February 2018
Accepted 18 March 2018
Available online xxxx

Keywords:
Gradient Descent
Information filtering
Matrix factorization
Singular value decomposition
Stochastic gradient descent

ABSTRACT

The limitations of neighborhood-based Collaborative Filtering (CF) techniques over scalable and sparse data present obstacle for efficient recommendation systems. These techniques show poor accuracy and dismal speed in generating recommendations. Model-based matrix factorization is an alternative approach use to overcome aforementioned limitations of CF.

Singular value decomposition (SVD) is widely used technique to get low-rank factors of rating matrix and use Gradient Descent (GD) or Alternative Least Square (ALS) for optimization of its error objective function. Most researchers have focused on the accuracy of predictions but they did not accumulate the convergence rate of learning approach. In this paper, we propose a new filtering technique that implements SVD using Stochastic Gradient Descent (SGD) optimization and provides an accelerated version of SVD for fast convergence of learning parameters with improved classification accuracy. Our proposed method accelerates SVD in the right direction and dampens oscillation by adding a momentum value in parameters updates. To support our claim, we have tested our proposed model against the famed real world datasets (MovieLens100k, FilmTrust and YahooMovie). The proposed Accelerated Singular Value Decomposition (ASVD) outperformed the existing models and achieved higher convergence rate and better classification accuracy.

1. Introduction

In the present age of science, data are being collected from a wide range of sources like Ecommerce websites, Social networking, Entrainments sites, biological and health field. Since the collected data is voluminous, unstructured and dynamic in nature, it poses more and more complexity over the data processing system for its uses. Therefore, demand for efficient information filtering or recommendation system with effective methods for optimum computing resource utilization has been increased in recent time (Donoho, 2000). Filtering algorithms are the sole part of recommendation system that aims to predict preference of a user would assign to a given item. Researchers use various data analytics techniques to develop more efficient recommendation systems by leveraging the fetched (or gathered) data such as user profile, item profile and user-item interactions (user-generated star ratings, testimonials, recommendations, feedback, etc.) and develop fast and efficient information filtering systems (Sarwar et al., 2001). Filtering techniques generate prediction via three methods: content-based, collaborative based and hybrid based. Content-based filtering takes the help of item attributes in order to recommend other items with similar attributes. Collaborative makes use of a model, based on user activities, behaviour and user-item interaction relationship (rating/preference) and generates the recommendations upon similarities to other users. Hybrid method use mixture of content-based and collaborative based techniques to generate predictions. While each of above-mentioned methods has their own strengths and limitations. Collaborative Filtering does not need prior knowledge of items and is completely domain free. Thus, it has become the state of art of many modern and commercial recommendation systems.

The basic concept behind the working of collaborative filtering is: if two users share a common interest, they are more likely to share those preferences in future as well (Ivens and Alencar Paulo, 2015), (Ricci et al., 2011). Collaborative Filtering can

* Corresponding author.
E-mail address: sraghuwanshi@gmail.com (S.K. Raghuwanshi).

Peer review under responsibility of King Saud University.

Production and hosting by Elsevier

https://doi.org/10.1016/j.jksuci.2018.03.012
1319-1578/

Please cite this article in press as: Raghuwanshi, S.K., Pateriya, R.K. Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.03.012

further categorize in Neighbourhood-based, model-based and hybrid methods. Collaborative filtering techniques suffer from the issues of:

Cold Start: A history of previous data is needed to make accurate future predictions for a given user, which is absent in case of new user and new item.

Scalability: The Performance of traditional Collaborative Filtering techniques suffer from the increased size of users and items. The tremendous growth in database results to produce inefficient predictions with limited computing capabilities.

Sparsity: In most cases, the user-item rating matrix is sparse or contain very few entries, as active users rate a small subset of total items.

Neighbourhood-based method, also known as a memory-based method, uses a user-item matrix to make predictions by finding similarity between users and items. The main advantages of memory based techniques are their simplicity, ease of implementation and easy to understand and explain. On the contrary, these techniques do not scale well and are limited to filter the sparse data. With the explosive growth of data over the internet, demand for scalable and efficient information filtering techniques has risen sharply (Bobadilla et al., 2013; Sarwar et al., 2000). Model-based filtering techniques came up as the solution for limitations of scalability and sparsity issues that are evident in memory based filtering techniques.

Model-based techniques use of data mining and machine learning techniques to generate predictions. These techniques build a latent model using offline learning and apply that model online to generate predictions. Sparsity is the main challenge of collaborative techniques as the user-item matrix contains very few entries, so, the first task is to estimate missing values of rating matrix with minimum accumulative error(s) to provide better predictions which makes it, a matrix completion problem.

Latent factor model combines the concepts of matrix factorization and dimension reduction (Koren et al., 2009). Much research has been done in matrix factorization techniques with low-rank approximation, such as principal component analysis (PCA) Singular value decomposition (SVD) (Koren et al., 2009), SVD++, Probabilistic Matrix Factorization (PMF) (Salakhutdinov and Mnih, 2007) and Non-Negative Matrix Factorization (NMF) (Hoyer, 2004; Hernando et al., 2016). SVD is the most popular and applied matrix factorization method in information filtering system (Koren et al., 2009; Bokde et al., 2014; Koren, 2008). As reported in the literature, the SVD can be implemented either using linear algebra or machine learning. The linear algebraic method does not perform well with sparse data matrix and not fulfils the promise of latent factor model in collaborative filtering. On the other hand, the SVD as machine learning uses Gradient Descent (GD) and Alternative Least Square (ALS) for optimization of error objective function (Koren et al., 2009; Bokde et al., 2014; Koren, 2008). All the existing techniques focus on the accuracy of predictions of rating matrix but they do not accumulate the convergence rate of learning approach. However, for efficient online systems convergence rate is a parameter of importance.

Therefore, this paper explores the implementation of SVD matrix factorization using stochastic gradient descent optimization and proposes an accelerated version of SVD for fast convergence of learning parameters. The major contributions of this paper are as follows: (i) It implements SVD as machine learning using Vanilla Gradient Descent and Stochastic gradient descent (ii) It proposes an accelerated SVD method based on using momentum based stochastic gradient descent optimization for fast and stable offline learning. (iii) It shows the implementation of SVD with a different version of gradient descent optimization techniques and compares the effectiveness of proposed model over convergence rate and overall error efficiency.

The rest of the paper is organized as follows. Section 2 introduces the concept of matrix factorization with dimensionality reduction and use it as model-based collaborative filtering approach of recommendation system. Section 3 presents SVD implementation as a machine learning approach with different optimization techniques. Section 4 provides an overview of experiment and analysis of outcome results. Finally, the Section 5 presents the conclusion and future scope of this work.

2. Background

Collaborative Filtering (CF) power comes as it does not require domain knowledge and does not need for massive data collection to build a recommendation models; as a result, it attracts many researchers in recent time and made significant progress in producing successful real-life recommendation systems, like Amazon, Netflix and google news. This paper focuses on model-based collaborative filtering characterize by matrix factorization with dimensionality reduction.

In the basic form, matrix factorization learns user and item latent factors from a user-item matrix and leads that knowledge for the recommendation. These techniques have become popular in recent time and showed excellent performance and sparse and highly scalable data sets (Koren et al., 2009). Matrix factorization models map users and items into a joint latent factor space of reduced dimensionality, where user-item ratings are mapped using the inner product of these latent factor vectors (Donoho, 2000; Zhou and Wu, 2016).

Dimensionality Reduction techniques are another model used in information filtering and recommendation systems. These techniques are inherited from clustering methods and maps a high dimensional input space into low dimensional latent space. In recommendation systems, user-item rating matrix data is highly redundant and correlated. As a result, using the low-rank approximation, the data matrix can be approximated quite well even with a small subset of values in original rating matrix (Koren, 2008). Fig. 1 shows the basic model of matrix factorization with dimensional reduction.

Model-based filtering combines latent factor and dimensionality reduction to generate matrix factorization technique (Koren, 2008). Singular value decomposition is the well-studied and established techniques in the field of information retrieval and deal efficiently with challenges of collaborative filtering like accuracy, sparsity, and scalability (Zhou and Wu, 2016). In basic form SVD is characterized as for a given rating matrix R of size $m \times n$ is defined as:

$$R = P\Sigma Q^T \tag{1}$$

where P is the size of $m \times n$, Σ is the of $n \times n$ and Q is the size of $n \times n$. Apart from that P and Q are the orthonormal matrices and Σ is a diagonal matrix. The solution to the singular value decomposition is to let $\Sigma = \mathrm{diag}(\sigma_1, \sigma_2, \sigma_3, \ldots\ldots\sigma_n)$, where σ_i is the ith largest eigenvalue of RR^T, the columns of P are defined to be the eigenvectors of RR^T, and the columns of Q are defined to be the

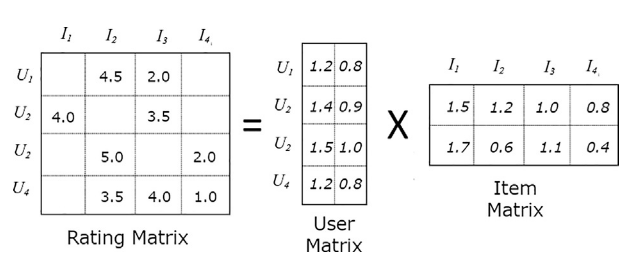

Fig. 1. An example of matrix factorization with dimensional reduction.

Please cite this article in press as: Raghuwanshi, S.K., Pateriya, R.K. Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.03.012

eigenvectors of R^TR. Given this solution to SVD, it is well known that $P_k\sum_k Q_k^T$ is the best rank k approximation to R under the Frobenius norm. Where the Frobenius norm is calculated by summing the squares of all elements of the matrix (Koren, 2008; Loukhaoukha, 2012; Loukhaoukha et al., 2011).

$$F(R - P_k\sum_k Q_k^T) = \sum_{i=1}^{m}\sum_{j=1}^{n}(R_{mn} - (P_k\sum_k Q_k^T)_{mn})^2 \tag{2}$$

SVD produced a surprising result in the field of information retrieval for identifying latent features (KIM and YUM, 2005) but raises some questions when applying in the domain where the high percentage of values are missing in rating matrix. Some methods use imputation techniques to fill the missing value of rating matrix and then applying SVD for collaborative filtering (Koren et al., 2009; Bell et al., 2007) However, these approaches are expensive and sometimes error-prone due to inaccurate imputation. Hence recent research work (Salakhutdinov and Mnih, 2007; Paterek, 2007; Aggarwal, 2016) suggest direct modelling of SVD matrix factorization and define as:

Let a rating matrix R of order $m \times n$ is approximately factorized into two matrices of P and Q of order $m \times k$ and $n\times$ k respectively where k $\ll$ (m, n) as:

$$R \approx PQ^T \tag{3}$$

Each of matrix P represents latent user factor, and each column of Q matrix represent the latent component. The approximate rating r_{ij} of matrix R is calculated by the product of vector p_iR^k to q_jR^k as (Koren et al., 2009; Aggarwal, 2016):

$$r_{ij} \approx \sum_k p_{ik}q_{jk}^T \tag{4}$$

To learn latent vector pi and qj, maximum likelihood approach is to minimize the squared error of the set of known ratings (Koren et al., 2009; Aggarwal, 2016).

$$Objective : e_{ij}^2 = \min_{p_i q_j}\sum_{i,j\in I}\sum_k (r_{ij} - p_{ik}q_{jk}^T)^2 \tag{5}$$

where I is set of known user-item ratings. In case of sparse data, the observed set I of ratings is small, which can cause overfitting. Overfitting is a common problem of machine learning in classification. It is behavior of prediction model produce analysis, too closely for a specific data points. It happens mainly when the training data contains few values which is a very common in online product recommendation systems. To address the problem of overfitting, many techniques like cross validation, pruning, regularization etc. are proposed and used. Regularization is one of the powerful mathematical approach to reduce the overfitting by adding penalties in model for extreme parameter values. Therefore, it is adequate to add a regularization term $\frac{\beta}{2}(P^2 + Q^2)$ to objective function expressed in Eq. (5) with $\beta > 0$ is regularization parameter (Aggarwal, 2016). Two approaches are popular to optimize or minimize objective function expressed in Eq. (6), Stochastic Gradient Descent (SGD) and Alternative Least Square (ALS) (Koren et al., 2009; Aggarwal, 2016).

$$Objective : e_{ij}^2 = \min_{p_i q_j}\sum_{i,j\in I}\sum_k (r_{ij} - p_{ik}q_{jk}^T)^2 + \frac{\beta}{2}\sum_k(\|P\|^2 + \|Q\|^2) \tag{6}$$

3. Matrix factorization and optimization

This section discusses the matrix factorization algorithms using SVD run for the experiments.

3.1. Vanilla gradient descent

This method computes the gradient of error $e_{ij} = r_{ij} - \hat{r}_{ij}$ where $\hat{r}_{ij} = +pq^T$ with μ as bias parameter used to improve the efficiency of method (user mean rating) and iterates through all ratings in training set. This method is slow as it requires to compute gradient for whole training set to perform just one update but guaranteed to converge global minima in full convex surface (Qian, 1999) The objective function used for this algorithm is described in Eq. (6) and learning parameters p and q are updated by equations as given below with α as learning rate ($0 < \alpha < 1$). The basic limitation of this optimization is it does not allow in an online scenario where the new examples are on the fly

$$\frac{\partial e_{ij}}{\partial p} = \sum_{i,j\in I}(r_{ij} - \mu - p_{ik}q_{kj})\cdot(-q_{kj}) + \beta\cdot p_{ik} = -\sum_{i,j\in I} e_{ij}\cdot q_{kj} + \beta.p_{ik}$$

$$\frac{\partial e_{ij}}{\partial q} = \sum_{i,j\in I}(r_{ij} - \mu - p_{ik}q_{kj})\cdot(-p_{ik}) + \beta\cdot q_{kj} = -\sum_{i,j\in I} e_{ij}\cdot p_{ik} + \beta.q_{kj}$$

Learning parameters update equation are:

$$p_{ik} = p_{ik} + \alpha\cdot\sum_{i,j\in I} e_{ij}\cdot q_{kj} + \beta\cdot p_{ik}$$

$$q_{kj} = q_{kj} + \alpha\cdot\sum_{i,j\in I} e_{ij}\cdot p_{ik} + \beta\cdot q_{kj}$$

3.2. Stochastic gradient descent

This method in contrast to gradient descent updates learning parameter by shuffling the training set values for each iteration. The gradient is computed over a set of training example and used for learning update in batch form in each step. Gradient calculation and parameter update are similar to above method except gradient is applied over parameters in batch form. Stochastic gradient descent results are smooth and used in the online system but have some limitations. Learning rate selection can be difficult in SGD as too small learning rate pose a slow rate of convergence whereas too large learning rate can cause fluctuation around minima. Also, constant learning rate for updating all parameter is not good practice for sparse value dataset. This method shows poor performance in navigating ravines, which is very common around local minima with convex surfaces as shown in Fig. 2(a).

Learning parameters update equations are:

$$\left.\begin{aligned} P_{grad} &= \tfrac{\partial e_{ij}}{\partial p} + = \left(-\sum_{i,j\in I} e_{ij}\cdot q_{kj} + \beta\cdot p_{ik}\right) \\ Q_{grad} &= \tfrac{\partial e_{ij}}{\partial q} + = \left(-\sum_{i,j\in I} e_{ij}\cdot p_{ik} + \beta\cdot q_{kj}\right)\end{aligned}\right\}\text{Batch of Size B}$$

Parameter Update

$$p_{ik} = p_{ik} + \alpha\cdot P_{grad}/B$$

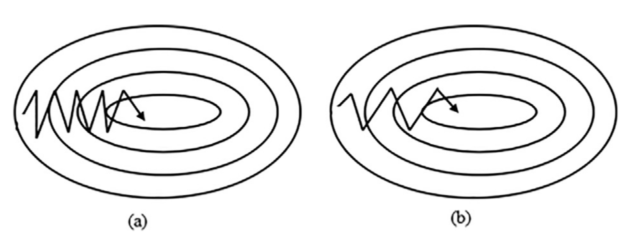

Fig. 2. Stochastic gradient descent (a) without momentum (b) with momentum.

$q_{kj} = q_{kj} + \alpha \cdot Q_{grad}/B$

4. Proposed model and algorithm

The major limitation of stochastic gradient is that it shows poor performance in navigating ravines, which is very common around local minima with convex surfaces and oscillates towards the local minima (Qian, 1999) presented a method that helps SGD to accelerate in the right direction and dampens oscillations by adding a fraction of update of the last step to the current step.

Momentum term helps to update in the direction whose gradients point in the same direction and reduces updates where gradients change direction resultant in improvement of convergence rate and lower overall oscillation over error surface. By Fig. 2 one can easily visualize the phenomenon as mentioned above. Momentum method works by adding a fraction term λ to of the last update of learning parameters to the current update (Duchi et al., 2011). Value of λ(momentum) lies between 0 and 1 and kept as 0.8 or 0.9. Gradients are calculated similar to Vanilla gradient method, but parameters update is changed due to the introduction of momentum term as follows:

$$\boldsymbol{p}^{n+1}_{temp} = \lambda \cdot \boldsymbol{p}^{n}_{temp} + \boldsymbol{\alpha} \cdot (\boldsymbol{e_{ij}} \cdot \boldsymbol{q_{kj}} - \beta \cdot \boldsymbol{p_{ik}})\boldsymbol{p_{ik}} = \boldsymbol{p_{ik}} - \boldsymbol{p}^{n+1}_{temp} \tag{7}$$

and

$$\boldsymbol{q}^{n+1}_{temp} = \lambda \cdot \boldsymbol{q}^{n}_{temp} + \boldsymbol{\alpha} \cdot (\boldsymbol{e_{ij}}\beta \cdot \boldsymbol{p_{ik}} - \cdot\boldsymbol{q_{kj}})q_{kj} = q_{kj} - q^{n+1}_{temp} \tag{8}$$

Algorithm: SVD Learning Using Gradient Descent with Regularization

Input: Rating Matrix: R, Learning Rate: α, Regularization parameter: β, Momentum λ, Maxepoch: N
Output: Factorized Matrix: P and Q
begin
Initialize: P and Q Randomly
$I = \{(i, j): r_{ij}$ is known$\}$
while (not converge OR n <= N) do:
Compute error $e_{ij} = (r_{ij} - \hat{r_{ij}}) \in \mathrm{I}$
for each user-latent pair (i, k) do:
$p^{n+1}_{temp} = \lambda \cdot p^{n}_{temp} + \alpha \cdot (e_{ij} \cdot q_{kj} - \beta \cdot p_{ik})$
$p_{ik} = p_{ik} - p^{n+1}_{temp}$
for each item-latent pair (j, k) do:
$q^{n+1}_{temp} = \lambda \cdot q^{n}_{temp} + \alpha \cdot (e_{ij} \cdot p_{ik} - \beta \cdot q_{kj})$
$q_{kj} = q_{kj} - q^{n+1}_{temp}$
end
return P and Q
end

5. Experiment and result discussion

Matrix Factorization methods are the state of art of many recommendation systems. These methods deal efficiently with scalability and sparsity and other challenges of recommendation systems. SVD is the most popular technique of matrix factorization and applied as a machine learning approach over sparse dataset as a maximum likelihood approximation. The existing methods provide slow convergence rate over online systems. Therefore, we have proposed a momentum based gradient descent technique to optimize SVD learning for fast convergence and stability.

The effectiveness of proposed method is evaluated on the popular real world datasets. We have implemented three versions of SVD using different methods of gradient optimization. We have proposed momentum based gradient descent optimization of SVD matrix factorization and compares its result with other two optimization methods over convergence rate and accuracy parameters. All the algorithms are implemented in Python 3.6 on a personal computer with 2.6 GHz IntelCore i5 with 8 GB RAM capacity.

5.1. Dataset

For showing the effectiveness of our proposed optimization method, we used three real world datasets as shown in Table 1. These three datasets are popular and mostly use in the testing of effectiveness of proposed collaborative filtering methods. We randomly divided our dataset into training and testing sets (ratio 75:25) and measures the evaluation parameters over them.

5.2. Evaluation metric

The true purpose of the recommendation system is to predict the preference of an active user over the item. We use Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) to measure the closeness of prediction of implemented versions of SVD. We also considered three metrics i.e. Precision, Recall and F measure evaluation to measure the accuracy of SVD classification for prediction and recommendations. These metrics are defined as follows.

$$\boldsymbol{MAE} = \frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\mathbf{i}=\mathbf{1},}^{\boldsymbol{m}}\sum_{\mathbf{j}=\mathbf{1}}^{\boldsymbol{n}}|\boldsymbol{r_{i,j}} - \widehat{\boldsymbol{r_{i,j}}}|$$

$$\boldsymbol{RMSE} = \sqrt{\frac{\mathbf{1}}{\boldsymbol{N}}\sum_{\mathbf{i}=\mathbf{1},}^{\boldsymbol{m}}\sum_{\mathbf{j}=\mathbf{1}}^{\boldsymbol{n}}(\boldsymbol{r_{i,j}} - \widehat{\boldsymbol{r_{i,j}}})^2}$$

5.3. Results

We ran our experiment to test the for the proposed model of SVD with latent feature value K ranging between 10 and 50 in a step size of 10. Existing models of SVD with Vanilla and Stochastic gradient descent were implemented over three real world datasets (Movielens100K, FilmTrust and YahooMovie) with α (learning rate) and (regularizing parameter) as 0.01 and 0.1 respectively. The proposed model used the same set of parameters with λ (Momentum factor) of 0.8 value.

To test the convergence rate of implemented model of SVD, we ran the experiment for 500 iterations over training data and calculate the final error objective error at the interval of 10 iterations each. Fig. 3 shows the outcome of the experiment over the three datasets. It is evident in the Fig. 3 that proposed ASVD converged with higher rate (fast convergence) in comparison to RSVD and SSVD and outperformed the other two models.

Further, it can also be observed from Table 2 that for different set of latent features ranging from 10 to 50 with a step size of 10, the proposed ASVD model have less variation in MAE and RMSE values as compared to the existing models of SVD. The algorithmic parameters are set to values as shown in Table 2 while running the experiment for 500 iterations. It is clear from Table 2 values that proposed ASVD provides stable behavior for the latent features compared to other two existing methods over all used datasets.

Furthermore, the precision, recall and F-measure metrics are computed for three models classification and the corresponding plots are also shown in Fig. 4. The ASVD outperforms in the matter of accuracy of classifications over the other two models as it provides better precision, recall, and F measure values for all used

Please cite this article in press as: Raghuwanshi, S.K., Pateriya, R.K. Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.03.012

Table 1
Datasets description.

Dataset	# of Users	# of Items	Total ratings	Range	Sparsity
MovieLens 100K[a]	943	1682	100,000	[1, 2,..5]	93.6953
FilmTrust[b]	1508	2071	35,497	[0.5, 1…4]	98.8634
YahooMovie[c]	7642	11,926	211,231	[1, 2…5]	99.7680

[a] http://www.grouplens.org/dataset/movielens/100k.
[b] http://www.librec.net/datasets/filmtrust.zip.
[c] http://webscope.sandbox.yahoo.com.

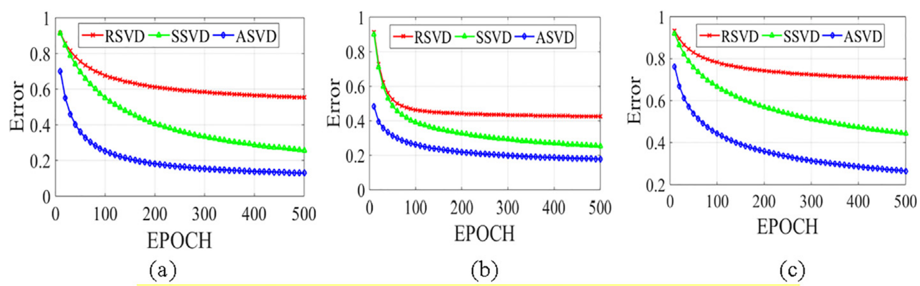

Fig. 3. Error vs epoch (a) MovieLens (b) FilmTrust (c) YahooMovie.

Table 2
Experimental results.

Dataset: Movie Lens 100 K, Parameter: Learning Rate: α = 0.01, Regularization parameter: = 0.1, Momentum factor = 0.8, Maxepoch: N = 500				
Parameter	No. of Features	RSVD	SSVD	ASVD
Dataset: MovieLens100k				
MAE	10	2.098	2.306	3.126
	20	1.885	2.003	3.128
	30	2.421	2.033	3.138
	40	3.280	2.145	3.185
	50	**4.146**	**2.321**	***3.228***
RMSE	10	2.415	2.566	3.320
	20	2.284	2.323	3.341
	30	2.864	2.387	3.392
	40	3.856	2.525	3.412
	50	**4.951**	**2.742**	***3.432***
Dataset: FilmTrust				
MAE	10	2.627	2.674	2.874
	20	2.754	2.692	2.867
	30	3.014	2.786	2.899
	40	3.297	2.899	2.932
	50	**3.606**	**2.990**	***2.955***
RMSE	10	2.834	2.867	2.032
	20	2.959	2.915	3.037
	30	3.219	2.998	3.067
	40	3.652	3.132	3.111
	50	**4.220**	**3.278**	***3.154***
Dataset: YahooMovie				
MAE	10	2.323	2.065	1.591
	20	3.957	3.101	2.101
	30	5.619	4.017	2.562
	40	7.327	4.865	2.984
	50	**8.956**	**5.682**	***3.365***
RMSE	10	2.814	3.046	3.640
	20	2.467	2.809	3.556
	30	3.281	3.061	3.602
	40	4.653	3.560	3.709
	50	**6.148**	**4.147**	***3.837***

RSVD – Singular Value Decomposition with Using Vanilla Gradient Descent.
SSVD – Singular Value Decomposition with Using Stochastic Gradient Descent.
ASVD – Singular Value Decomposition with Proposed Momentum Stochastic Gradient Descent.

Please cite this article in press as: Raghuwanshi, S.K., Pateriya, R.K. Accelerated Singular Value Decomposition (ASVD) using momentum based Gradient Descent Optimization. Journal of King Saud University – Computer and Information Sciences (2018), https://doi.org/10.1016/j.jksuci.2018.03.012

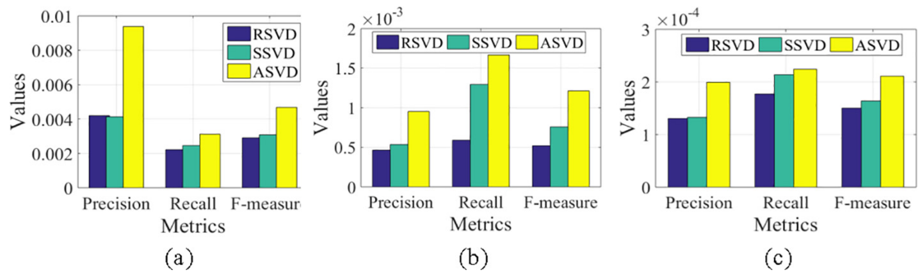

Fig. 4. Precision, recall and F-measures (a) MovieLens (b) FilmTrust (c) YahooMovie.

three datasets. As final note one can conclude that, our simulation study proved, that proposed ASVD method provides better performance regarding the speed of convergence and accuracy of classification.

6. Conclusion

This paper implements a matrix factorization method as a machine learning approach. The main challenges of machine learning approaches are their convergence rate and optimization method to get better accuracy and performance. Therefore, this paper proposes a new method to implement a well-known matrix factorization method SVD with acceleration using a momentum factor that provides a fast convergence with better stability and performance. The effectiveness of our method is evaluated on the popular three real world datasets named MovieLens100k, FilmTrust and YahooMovie. The experimental results show that the proposed ASVD outperformed other models of SVD using RSVD(-vanilla) and SSVD(stochastic). In our simulation study we observed that ASVD worked well in convergence rate, stability measures and classification accuracy. This method can be implemented as a collaborative filtering technique of information retrieval. In future, an adaptive learning rate based method can be used with SVD to get better results and convergence.

References

Aggarwal, C.C., 2016. Model-based collaborative filtering. In: Recommender Systems, Springer International Publishing, Cham, pp. 71–138. https://doi.org/10.1007/978-3-319-29659-3_3.

Bell, R., Koren, Y., Volinsky, C., 2007. Modeling relationships at multiple scales to improve accuracy of large recommender systems. In: Proc. 13th ACM SIGKDD Int. Conf. Knowl. Discov. data Min. – KDD '07 95. https://doi.org/10.1145/1281192.1281206.

Bobadilla, J., Ortega, F., Hernando, A., Gutiérrez, A., 2013. Recommender systems survey. Knowl.-Based Syst. 46, 109–132. https://doi.org/10.1016/j.knosys.2013.03.012.

Bokde, D., Girase, S., Mukhopadhyay, D., 2014. Role of matrix factorization model in collaborative filtering algorithm: a survey. Int. J. Adv. Found. Res. Comput. 1.

Donoho, D.L., 2000. Aide-Memoire. High-dimensional data analysis: the curses and blessings of dimensionality. In: Am. Math. Soc. Lect. Challenges 21st Century 1–33.

Duchi, J., Edu, J.B., Hazan, E., Singer, Y., 2011. Adaptive subgradient methods for online learning and stochastic optimization. J. Mach. Learn. Res. 12, 2121–2159.

Hernando, A., Bobadilla, J., Ortega, F., 2016. A non negative matrix factorization for collaborative filtering recommender systems based on a Bayesian probabilistic model. Knowl.-Based Syst. 97, 188–202. https://doi.org/10.1016/j.knosys.2015.12.018.

Hoyer, P.O., 2004. Non-negative matrix factorization with sparseness constraints 5, 1457–1469. https://doi.org/10.1109/ICMLC.2011.6016966.

Kim, D., Yum, B., 2005. Collaborative filtering based on iterative principal component analysis. Expert Syst. Appl. 28, 823–830. https://doi.org/10.1016/j.eswa.2004.12.037.

Koren, Y., 2008. Factorization meets the neighborhood a multifaceted Collaborative Filtering Model. In: Proceeding of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD 08. p. 426. https://doi.org/10.1145/1401890.1401944.

Koren, Y., Bell, R., Volinsky, C., 2009. Matrix factorization techniques for recommender systems. Computer (Long. Beach. Calif). 42, 30–37. https://doi.org/10.1109/MC.2009.263.

Loukhaoukha, K., 2012. On the security of digital watermarking scheme based on SVD and tiny-GA. J. Inf. Hiding Multimed. Signal Process. 3, 135–141.

Loukhaoukha, K., Chouinard, J.-Y., Taieb, M.H., 2011. Optimal image watermarking algorithm based on LWT-SVD via multi-objective ant colony optimization. J. Inf. Hiding Multimed. Signal Process. 2, 303–319.

Paterek, A., 2007. Improving regularized singular value decomposition for collaborative filtering. KDD Cup Work 2–5. https://doi.org/10.1145/1557019.1557072.

Portugal Ivens, Alencar Paulo, C.D., 2015. The use of machine learning algorithms in recommender systems : a systematic review. eprint arXiv:1511.05263.

Qian, N., 1999. On the momentum term in gradient descent learning algorithms. Neural Networks. https://doi.org/10.1016/S0893-6080(98)00116-6.

Ricci, F., Rokach, L., Shapira, B., 2011. Introduction to recommender systems handbook. In: Recommender Systems Handbook. Springer US, Boston, MA, pp. 1–35. https://doi.org/10.1007/978-0-387-85820-3_1.

Salakhutdinov, R., Mnih, A., 2007. Probabilistic matrix factorization. In: Proceedings of the 20th International Conference on Neural Information Processing Systems, NIPS'07. Curran Associates Inc., USA, pp. 1257–1264.

Sarwar, B., Karypis, G., Konstan, J., Reidl, J., 2001. Item-based collaborative filtering recommendation algorithms. In: Proc. Tenth Int. Conf. World Wide Web – WWW '01, pp. 285–295. https://doi.org/10.1145/371920.372071

Sarwar, B.M., Karypis, G., Konstan, J.A., Riedl, J.T., 2000. Application of dimensionality reduction in recommender system - a case study, architecture. https://doi.org/10.1.1.38.744.

Zhou, X., Wu, S., 2016. Rating LDA model for collaborative filtering. Knowl.-Based Syst. 110, 135–143. https://doi.org/10.1016/j.knosys.2016.07.020.