MULTIPLICATIVE ALGORITHM FOR CORRENTROPY-BASED NONNEGATIVE MATRIX FACTORIZATION

Ehsan Hosseini Asl[1], Jacek M. Zurada[1,2]

[1] Department of Electrical and Computer Engineering
University of Louisville, Louisville, KY, USA
ehsan.hosseiniasl@louisville.edu, jacek.zurada@louisville.edu

[2]IT Institute, University of Social Sciences
9 Sienkiewicza St., 90-113 Łódź, Poland

Abstract

Nonnegative matrix factorization (NMF) is a popular dimension reduction technique used for clustering by extracting latent features from high-dimensional data and is widely used for text mining. Several optimization algorithms have been developed for NMF with different cost functions. In this paper we evaluate the correntropy similarity cost function. Correntropy is a nonlinear localized similarity measure which measures the similarity between two random variables using entropy-based criterion, and is especially robust to outliers. Some algorithms based on gradient descent have been used for correntropy cost function, but their convergence is highly dependent on proper initialization and step size and other parameter selection. The proposed general multiplicative factorization algorithm uses the gradient descent algorithm with adaptive step size to maximize the correntropy similarity between the data matrix and its factorization. After devising the algorithm, its performance has been evaluated for document clustering. Results were compared with constrained gradient descent method using steepest descent and L-BFGS methods. The simulations show that the performance of steepest descent and L-BFGS convergence are highly dependent on gradient descent step size which depends on σ parameter of correntropy cost function. However, the multiplicative algorithm is shown to be less sensitive to σ parameterand yields better clustering results than other algorithms. The results demonstrate that clustering performance measured by entropy and purity improve the clustering. The multiplicative correntropy-based algorithm also shows less variation in accuracy of document clusters for variable number of clusters. The convergence of each algorithm is also investigated, and the experiments have shown that the multiplicative algorithm converges faster than L-BFGS and steepest descent method.

Key words: Nonnegative Matrix Factorization (NMF), Correntropy, Multiplicative Algorithm, Document Clustering

1 Introduction

Large size of data is one of the central issues in data analysis research. Processing these large amounts of data opens new issues related to data representation, disambiguation, and dimensionality reduction. A useful representation typically makes latent structure in the data explicit, and often reduces the dimensionality of the data so that additional computational methods can be applied. In this process it is important to reduce the data size without losing its most essential features. Therefore, a common ground in the various approaches of data mining is to replace the original data with a lower dimensional representation obtained via subspace approximation [1, 2, 4].

There are several methods to reduce the dimensionality of large data such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD) and Independent Component Analysis (ICA). Often the data to be analyzed is nonnegative, and the low-rank data are further required to be comprised of nonnegative values in order to avoid contradicting physical realities. However, these classical tools cannot guarantee to maintain the nonnegativity [1]. Therefore, an approach of finding reduced rank nonnegative factors to approximate a given nonnegative data matrix becomes a natural choice. The Nonnegative Matrix Factorization (NMF) approach allows to create a lower rank data out of original data, while maintaining nonnegativity of matrices entries [1, 2, 3].

The NMF technique approximates a data matrix A with the product of low rank matrices $W$and H, such that $A \approx WH$ and the elements of$W$ and$H$ are nonnegative [1,2]. If columns of$A$would be data samples, then the columns of Wcan be interpreted as basis or parts from which data samples are formed, while the columns of$H$ give the contribution of each basiswhich when combined form the corresponding data sample. In application of NMF to clustering, it is common to define clusters based on each basis vector, and assigning each data sample to a cluster based on basis contribution intensity which is found from matrix H.

Several cost functions have been used in the literature to implement the NMF for various types of applications and data type. Euclidean distance is the most common cost function used for many applications including text mining [1]. Kullback-Leibler divergence (KL-divergence) [1, 2], β-divergence [21, 22] are among other methods also used for different applications. However, the main issue is to find the matrix factors (W, H) that minimize the chosen cost function. There are several optimization algorithms in the literature to perform this optimum decomposition [3, 4, 8, 10, 11, 12]. Correntropy similarity function is a recently proposed cost function which has been used for different tasks of pattern recognition [23]. It has been introduced to NMF only recently in [24, 25, 26]. In this paper, a multiplicative algorithm for corren-

tropy-based NMF (MACB-NMF) has been developed and its performance has been investigated in comparison to general gradient descent method for document clustering application using several metrics.

This paper is organized as follows. Section 2 introduces the correntropy cost function. Section 3 discusses some developed optimization algorithms for NMF. In section 4, a multiplicative update algorithm for correntropy cost function (MACB) is presented. Experiments on real data set are presented in Section 5. The discussion and conclusions are presented in Section 6.

2 Correntropy Similarity Function

Given a data matrix $A \in \mathbb{R}^{m\times n}$ and a positive integer $k < \min\{m.n\}$, find nonnegative factorization into matrices $W \in \mathbb{R}^{m\times k}$ and $H \in \mathbb{R}^{k\times n}$ as

$$\min_{W,H} D(A|WH) \; subject\; to\; W \geq 0, H \geq 0 \tag{1}$$

where:

$A \geq 0$expresses nonnegativity of the entries of A(and not semidefinite positiveness),

$D(A|WH)$isa measure for goodness of fit such that

$$D(A|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n} d([A]_{ij}|[WH]_{ij}) \tag{2}$$

where:

d(x|y) is a scalar cost function [22].

Several cost function are used and most of them use the Bregman divergence [7]. Generally, a divergence function is defined as follows

$$D_{\alpha}(a,b) = \begin{cases} a\dfrac{a^{\alpha}-b^{\alpha}}{\alpha} + b^{\alpha}(b-a) & : \alpha \in (0,1] \\ a(loga - logb) + (b-a) & : \alpha = 0 \end{cases} \tag{3}$$

where:

α is chosen to define the type of the divergence function.

Obviously, $D_1(a,b) = (a-b)^2$ is the Euclidean distance function, and $D_0(a,b)$ defines KL-divergence [13]. The most common function found in literature is shown below

$$D_{Euclidean}(A|WH) = \sum_{i=1}^{m}\sum_{j=1}^{n}\frac{1}{2}\left(A_{ij} - (WH)_{ij}\right)^2 \quad (4)$$

Using the above notation, the correntropy cost function is defined as

$$d_{correntropy}(a|b) = -exp\left(\frac{-(a-b)^2}{2\sigma^2}\right) \quad (5)$$

$$D_{correntropy}(A|WH) = -\sum_{i=1}^{m}\sum_{j=1}^{n}\exp\left(\frac{-\left(A_{ij} - (WH)_{ij}\right)^2}{2\sigma^2}\right) \quad (6)$$

where:

$\sigma$is a parameter of correntropy measure.

The optimization algorithms try to minimize the correntropy, since it is a similarity instead of distance between two elements. The algorithm for minimizing these cost functions is introduced in the next section.

3 Optimization Algorithms

A key issue of NMF factorization is to minimize the cost function while keeping elements of W and H matrices nonnegative. Another challenge is the existence of local minima due to non-convexity of $D(A|WH)$ in both $W$and H. Moreover, a unique solution to NMF problem does not exist, since for any invertible matrix B whose inverse is B^{-1}, a term $WBB^{-1}H$ could also be nonnegative. This is most probably the main reason for non-convexity of $D(A|WH)$ function [13].

Several algorithms exist for minimizing cost functions in the NMF context. Lee and Seung [1, 2] developed a multiplicative algorithm for solving Euclidean and KL-divergence in 2001. Sparse Coding and sparseness constraint which impose sparsity on H matrix was proposed by Hoyer in 2002 and 2004 [3, 5]. Alternating Least Square (ALS) [12], ALS using projected gradient descent (ALSPGRAD) [14], gradient descent with constrained least square (GD-CLS) [9], Quasi Newton method [11], Alternating Nonnegative Constrained Least Squares (ANLS) using active set and block principal pivoting [17, 20], Hierarchical Alternating Least Square (HALS) [19] was proposed for Euclidean cost function. Fevotte et al proposed several algorithms for minimizing β-divergence cost function [21, 22]. In 2012, Li et al convert general

Bregman divergence to Euclidean distance function using Taylor expansion and solve the corresponding function using HALS algorithm [25]. Du et al proposed a half-quadratic optimization algorithm to solve NMF based on correntropy cost function and developed a multiplicative algorithm for resulting weighted NMF [26].

In 2012, Ensari et al used general algorithms of Constrained Gradient Descent (CGD) method for solving the correntropy function [18] and compared the results with projected gradient descent method of Euclidean cost function [24, 25]. The major disadvantage of CGD is its dependency on σ parameter of correntropy cost function. As will be shown in the next section, the update rate of CGD algorithm is based on this parameter. In the next section, we derive the CGD algorithm based on multiplicative update rule which has adaptive update learning rate and less sensitivity to variation of σ parameter.

4 Multiplicative Algorithm for Correntropy-based NMF

This section proposes a multiplicative algorithm for correntropy cost function (MACB). To minimize (6) using gradient descent algorithm, its gradient should be taken with respect to W and H matrices' elements which are parameters of cost function. The gradients $\nabla_W(D_\varphi), \nabla_H(D_\varphi)$are calculated as follows,

$$\nabla_W(D_\varphi(A\|WH)) = {}^1\!/_{\sigma^2}\left[\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\odot(WH-A)\right]H^T \quad (7)$$

$$\nabla_H(D_\varphi(A\|WH)) = {}^1\!/_{\sigma^2}\,W^T\left[(WH-A)\odot\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right] \quad (8)$$

where:

$\odot$is the element-wise product of two matrices.

As can be seen from Equations(7) and (8), the gradient formula involves the step size in the direction of gradient that is proportional to $1/\sigma^2$ parameter. Therefore, the gradient step variation could cause the solution to deviate from the limit points of the feasible region. This may result in unsatisfactory solution for $W$and H.

The multiplicative gradient descent approach is equivalent to updating each parameter by multiplying its value at previous iteration by the ratio of the negative and positive parts of the gradient of the cost function with regard to this parameter [2, 11]. Suppose there is a function $f(\theta)$ which should be minimized over θ. Gradient descent using multiplicative algorithm is equivalent to,

$$\theta \leftarrow \theta \frac{[\nabla f(\theta)]_-}{[\nabla f(\theta)]_+} \qquad (1)$$

where:

$$\nabla f(\theta) = [\nabla f(\theta)]_+ - [\nabla f(\theta)]_- \qquad (10)$$

and the summands are both nonnegative.This ensures nonnegativity of the parameter updates, provided initializationis with a nonnegative value. A fixed point $\theta^\star$ of the algorithm implies either $\nabla f(\theta_-) = 0$ or $\theta^\star = 0$[21, 22]. We apply this algorithm on Correntropy function gradients, Equations (7) and (8), and derive the update formula for W and H matrices respectively as follows,

$$W \leftarrow W \frac{[\nabla_W (D_\varphi(A\|WH))]_-}{[\nabla_W (D_\varphi(A\|WH))]_+} \qquad (11)$$

$$W \leftarrow W \odot \frac{\left[\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right) \odot A\right] H^T}{\left[\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right) \odot (WH)\right] H^T} \qquad (12)$$

$$H \leftarrow H \frac{[\nabla_H (D_\varphi(A\|WH))]_-}{[\nabla_H (D_\varphi(A\|WH))]_+} \qquad (13)$$

$$H \leftarrow H \odot \frac{W^T \left[A \odot \exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right]}{W^T \left[(WH) \odot \exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right]} \qquad (14)$$

As can be seen from Equations (12) and (14), the σ parameter is in numerator and denominator of update algorithm, which reduce the effect of variation of this parameter to the update algorithm. Although, we do not prove the non-increasing property of multiplicative update algorithm with Correntropy criterion analytically, the experimental results show that it is monotonic and non-increasing. It also give better results in comparison to other gradient descent methods. Therefore, MACB algorithm for NMF is as follows:

MACB-NMF Algorithm:

(1) Initialize W and H with nonnegative values, and scale the columns of W to unit norm.
(2) Iterate until convergence or for l iterations:

$$\text{(a)}\quad W_{ij} \leftarrow W_{ij} \frac{\left(\left[\left(\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right)\odot A\right]H^T\right)_{ij}}{\left(\left[\left(\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right)\odot (WH)\right]H^T\right)_{ij}+\epsilon} \quad for\ i\ and\ j\ \ [\epsilon = 10^{-9}]$$

$$\text{(b)}\quad H_{ij} \leftarrow H_{ij} \frac{\left(W^T\left[A\odot\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right]\right)_{ij}}{\left(W^T\left[(WH)\odot\exp\left(\frac{-(A-WH)^2}{2\sigma^2}\right)\right]\right)_{ij}+\epsilon} foriandj\ \ [\epsilon = 10^{-9}]$$

5 Experiments

This section outlines the design procedure of an experiment to test MACB algorithm. We employ Reuters Documents Corpus for document clustering. This original dataset contains 21578 documents and 135 topics or document clusters created manually. Each document in the corpus is been assigned one or more topics or category labels based on its content. The size of each cluster which is the number of documents it contains, range from less than ten to four thousand. For this experiment, documents associated with only one topic are used and topics which contain less than five documents are discarded [9]. Therefore, 8293 documents with 48 topics were left at the end. In order to evaluate the performance of the MACB for increasing complexity, i.e., the number of clusters to be created or the k parameter, ten different k values of $[2, 4, 6, 8, 10, 15, 20, 30, 40, 48]$ are chosen.

After creating clusters using NMF, the cluster is assigned to a most related document topic. For this purpose, a matrix which shows the distribution of all documents between each created cluster and dataset topics is created. The matrix's dimension is $k \times l$, which k is the number of clusters and l is the number of topics. This matrix is called Document Distribution Matrix (DDM). The maximum value at each column of DDM is found first. Then, the corresponding document topic related to this column is assigned to the NMF cluster related to the row number. At the end of this process, there may be some NMF clusters which are not assigned to any topic. Some of these clusters may contain large number of documents, and omitting them may reduce the accuracy metric. To assign these NMF clusters to a topic, the maximum value found in a row of DDM related to any of these NMF clusters is used for the topic assignment. It turns out that the related column of the founded value indicates the topic to be assigned. This method may results in assigning some of NMF clusters to more than one topic.

We evaluate the clustering performance with Accuracy, Root Mean Square Residual (RMSR), Entropy, Purity, and computational time metrics. Accuracy of clustering is assessed using the metric AC used by [4] is defined

$$AC = \sum_{i=1}^{n} \delta(d_i)/n \tag{15}$$

where:

$\delta(d_i)$ is set to 1 if d_i has the same topic label for both NMF cluster and the original topic, and otherwise set to 0,

n is the total number of documents in the collection.

The RMSR between A and $W$and H matrix is defined as:

$$RMSR = \sqrt{\frac{\sum_{ij}\left(A_{ij} - WH_{ij}\right)^2}{m * n}} \tag{16}$$

Total entropy for a set of clusters is calculated as the weighted mean of the entropies of each cluster weighted by the size of each cluster [8]. Using DDM, we compute p_{ij} for topic j, the probability that a member of cluster i belongs to topic j as $p_{ij} = n_{ij}/n_i$, where n_i is the number of objects in cluster i and n_{ij} is the number of documents of topic j in cluster i. Entropy of each cluster is defined as:

$$e_i = -\sum_{j=1}^{l} p_{ij} \log_2\left(p_{ij}\right) \tag{17}$$

where:

l is the number of topics.

Entropy of the full data set as the sum of the entropies of each cluster weighted by the size of each cluster:

$$e = \sum_{i=1}^{k} \frac{n_i}{n} e_i \tag{18}$$

where:

k is the number of NMF clusters,

n is the total number of documents.

Purity measures the extent to which each NMF cluster contained documents from primarily one topic [16]. *Purity* of a NMF clustering is obtained as a weighted sum of individual NMF cluster Purity values and is given by

$$P(S_i) = \frac{1}{n_i} max_j\left(n_i^j\right) \tag{19}$$

$$Purity = \sum_{i=1}^{K} \frac{n_i}{n} P(S_i) \tag{20}$$

where:

S_i is a particular NMF cluster of size n_i,
n_i^j is the number of documents of the $i-th$ topic that were assigned to the $j-th$ NMF cluster,
k is the number of clusters,
n is the total number of documents.

In general, the larger the Purity value, the better the clustering solution. We also compute the computational time taken by each minimization algorithms in terms of CPU time measured in second.

For performance evaluation of MACB, the results of this algorithm were compared to Steepest Descent (SD) and L-BFGS methods of gradient descent algorithm implemented in MATLAB [18], and robust Correntropy Induced Metric (rCIM) [26]. For each algorithm, three clustering experiments were executed based on normalization of W and H matrices. As mentioned before, NMF does not have a unique solution, and it is better to normalize either W or H to have a consistent factorization of a particular dataset when using different algorithms. This procedure is also taken to investigate the effect of normalization of these W and H matrices on the clustering result. Therefore, we implement three experiments for each algorithm, one without normalization, another using normalization of W matrix's columns, and the last one with normalization on each row of H matrix.

Since σ value has an effect on update learning rate of SD, L-BFGS and rCIM algorithms, improper selection of σcould result in poor clustering. However, σ value have a small effect on MACB update algorithm, because the effect of σ is significantly decreased by the division in formula of MACB algorithm. Moreover, the learning rate is adaptive and is proportional to W and H matrices in each step of MACB algorithm. By implementing several experiments, we realize that the best value which yields the highest AC, lowest Entropy and highest Purity in clustering for each algorithm is $\sigma = 1$. We continue the experiment with three methods of normalization for MACB algo-

rithm and compare them to W-normalized case (normalization on each column of W matrix) for SD, L-BFGS, and rCIM algorithms with $\sigma = 1$ for three algorithms of optimization. AC, Entropy and Purity of clustering are shown in Figure 1-3 respectively,

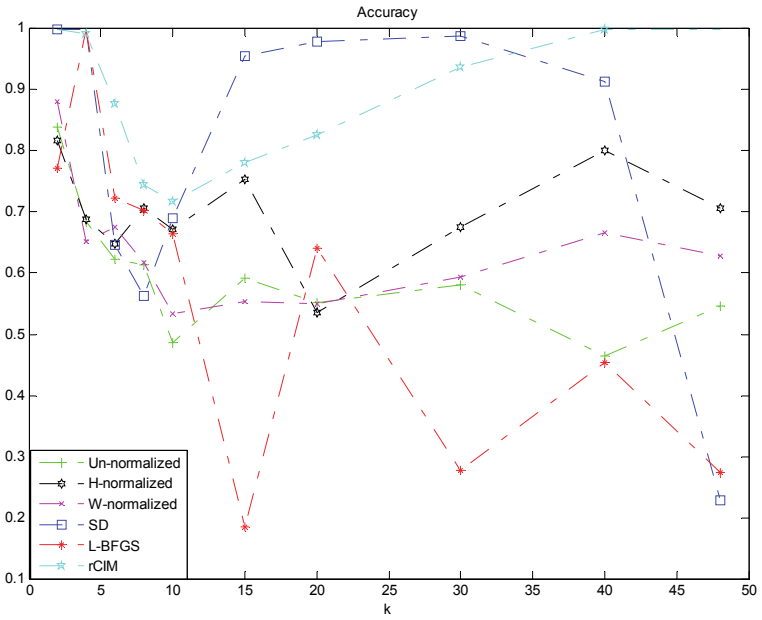

Figure 1. Accuracy of SD, L-BFGS, rCIM, and MACB algorithm

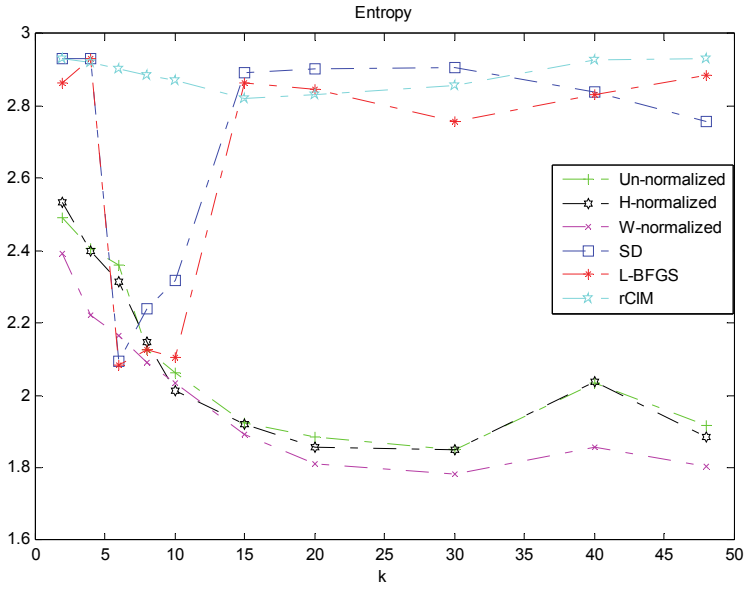

Figure 2. Entropy of SD, L-BFGS, and MACB algorithm

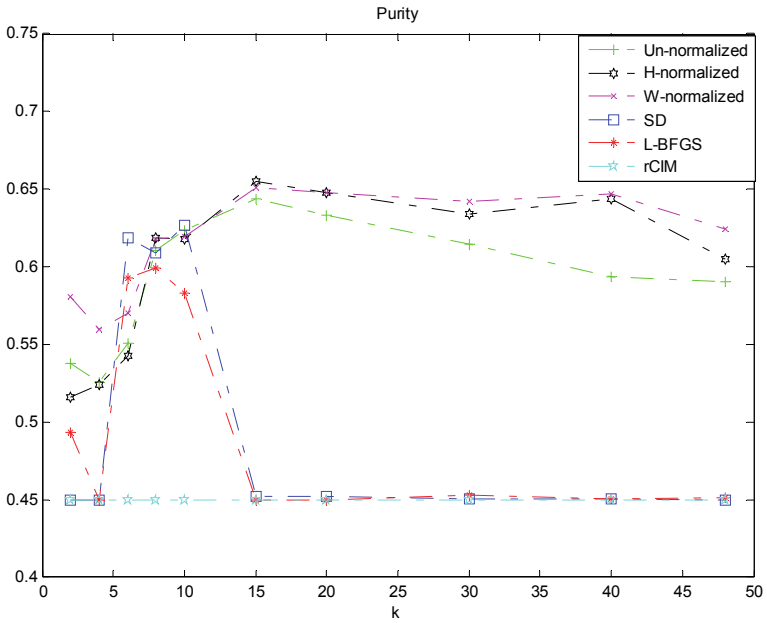

Figure 3. *Purity* of SD, L-BFGS, and MACB algorithm

It is clear that MACB algorithm yields smaller Entropy and higher Purity for all values of k. However, SD, L-BFGS, and rCIM algorithms have low Entropy and high Purity only for $k = [6,8,10]$. On the other hand, MACB have a consistent change in AC, Entropy, and Purity for different values of k. Moreover, as k increase, the quality of clustering improves for MACB. To have a good comparison between all algorithms, we select two values of k which results in highest AC, lowest Entropy and highest Purity. According to Fig.1-3, these metrics occurs in $k = [15, 19]$. Therefore we tabulate the clustering result of each algorithm for corresponding k values in Table 1 and 2.

Tables 1 and 2 indicate that MACB algorithm give better Entropy and Purity in comparison to the other algorithms. The RMSR metric is also small for MACB algorithm, while this metric is too large for SD, L-BFGS and rCIM. This indicates a large error between $WH$and$A$. One may notice that the computational time of MACBand rCIM algorithms is higher than SD and L-BFGS algorithms. The reason is that in each step of algorithm, there are two multiplications and divisions for updating W and H in MACB and rCIM algorithms, which do not exist in SD and L-BFGS algorithms. The multiplication and division of these large matrices are highly computational and time consuming.

As a result, we can conclude that the computed W and H matrices using MACB algorithm offer the best approximation of documents dataset among other correntropy-based NMF. The minimization of correntropy cost function for 40 iterations is shown in Fig.4 for all algorithms. It demonstrates that MACB algorithm has a faster convergence than SD, L-BFGS and rCIM algorithms. Gradient minimization curve for $k = 20,30,40,48$ is shown in Figure 5. It indicates that as the value of k increases, the gradient minimizes more slowly. This implies that the algorithm reaches the limit point of feasible region, and the constraint of nonnegativity does not allow the optimization algorithm to converge. We propose that other algorithms like alternating least square method with nonnegativity constraint and hierarchical ALS could be investigated on this case for future work.

Table 1. Comparison between performance of different NMF algorithms, k=15

Algorithm	**RMSR**	**Accuracy**	**Entropy**	**Purity**	**CPU time (sec)**
SD	1983	0.9401	2.8834	0.4582	552
L-BFGS	2517	0.1469	2.8634	0.4496	602
MACB (W-normalized)	0.3328	0.5530	1.8920	0.6514	2353
MACB (H-normalized)	0.3328	0.7528	1.9191	0.6551	2353

Table 2. Comparison between performance of different NMF algorithms, k=20

Algorithm	**RMSR**	**Accuracy**	**Entropy**	**Purity**	**CPU time (sec)**
SD	53594	0.8961	2.8616	0.4527	535
L-BFGS	17.75	0.6274	2.8399	0.4496	605
Multiplicative (W-normalized)	0.9776	0.5507	1.8094	0.6475	2513
Multiplicative (H-normalized)	0.9776	0.5360	1.8567	0.6479	2513

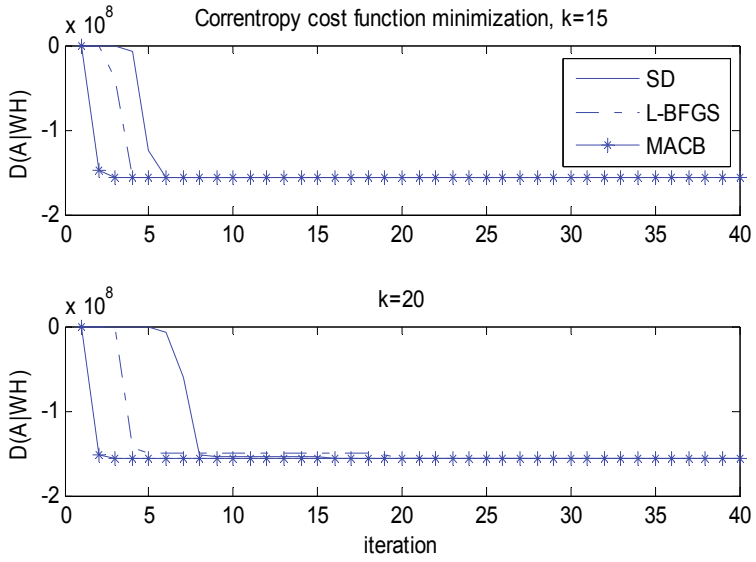

Figure 4. Correntropy cost function minimization curve

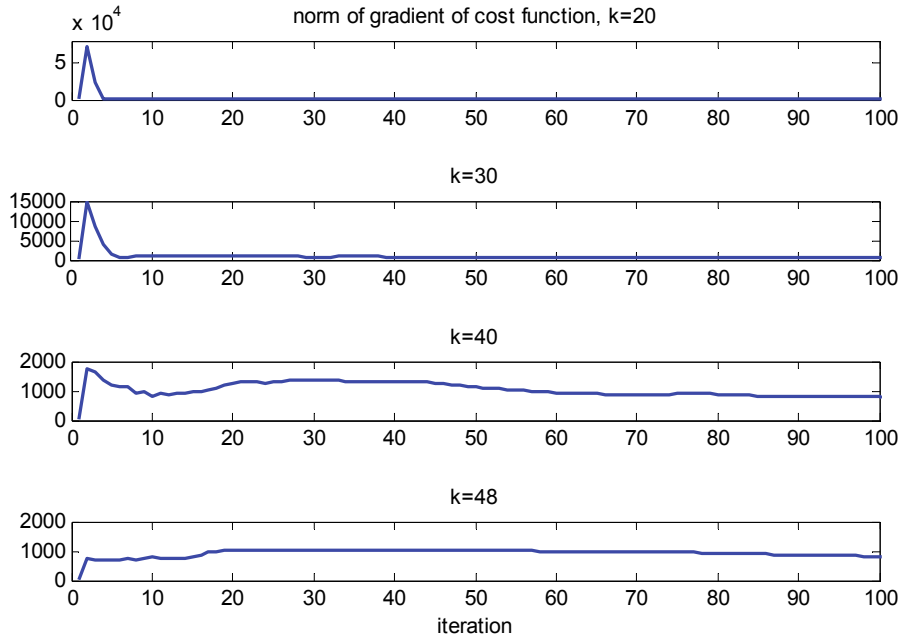

Figure 5. History of norm of cost function'sgradient

6 Conclusion

In this paper, a multiplicative algorithm for NMF based on correntropy cost function is developed. Its performance was tested on the Reuters Document Corpus for document clustering. The clustering result is also compared to gradient descent algorithm using SD and L-BFGS algorithms using common clustering evaluation measures. The minimization curve and curve of gradient's norm of cost function are also investigated. The result proves that MACB algorithm gives better clustering performance in terms of Entropy and Purity and also faster convergence than other two methods. However, it shows that by increasing the number of NMF clusters (k value), gradient curve of cost function does not converge appropriately. For future work, we propose that other minimization algorithms like ALS, ANLS, and HALS could be used for improving this problem.

References

1. Lee D.D.,Seung H.S., 1999,*Learning the parts of objects by non-negative matrix factorization*, Nature, 401, 6755, pp. 788-791.
2. Seung D.,Lee L., 2001,*Algorithms for non-negative matrix factorization,* Advances in neural information processing systems, 13, pp. 556-562.
3. Hoyer P.O., 2002, *Non-negative sparse coding,* Proc. of 12th IEEE Workshop on Neural Networks for Signal Processing, pp. 557-565.
4. Xu W., Liu X., Gong Y., 2003, *Document clustering based on non-negative matrix factorization*, Proc. of the 26th Annual Int. ACM SIGIR Conf. on Research and development in informaion retrieval, pp. 267-273.
5. Hoyer P.O., 2004,*Non-negative matrix factorization with sparseness constraints*, The Journal of Machine Learning Research, 5, pp. 1457-1469.
6. Pauca V.P., Shahnaz F., Berry M.W., PlemmonsR.J., 2004,*Text mining using non-negative matrix factorizations,* Proc. SIAM Int. Conf. on Data Mining, Orlando FL, pp. 22-24.
7. Sra S., DhillonI.S., 2005, *Generalized nonnegative matrix approximations with Bregman divergences*, Advances in neural information processing systems, pp. 283-290.
8. Tan P.N., Steinbach M., Kumar V., 2006,*Introduction to Data Mining*, Pearson Addison Wesley.
9. Shahnaz F., Berry M.W., Pauca V.P., PlemmonsR.J., 2006, *Document clustering using nonnegative matrix factorization*, Information Processing & Management, 42, 2, pp. 373-386.
10. Liu W., PokharelP.P., Principe J.C., 2006,*Correntropy: A localized similarity measure*, Int. Joint Conf. on Neural Networks, pp. 4919-4924.
11. Zdunek R., Cichocki A., 2006,*Non-negative matrix factorization with quasi-Newton optimization*, Int. Conf. on Artificial Intelligence and Soft Computing, Springer Berlin Heidelberg, 4029, pp. 870-879.
12. Berry M.W., Browne M., LangvilleA.N., Pauca V.P., PlemmonsR.J., 2007, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics & Data Analysis, 52, 1, pp. 155-173.
13. KompassR., 2007,*A generalized divergence measure for nonnegative matrix factorization*, Neural computation, 19, 3, pp. 780-791.
14. Lin C.J., 2007, *Projected gradient methods for nonnegative matrix factorization*, Neural computation, 19, 10, pp. 2756-2779.
15. Liu W., PokharelP.P., Príncipe J.C., 2007, *Correntropy: properties and applications in non-Gaussian signal processing*, IEEE Trans. on Signal Processing, 55, 11, pp. 5286-5298.
16. Ding C., Li T., Peng W., Park H., 2006, *Orthogonal nonnegative matrix t-factorizations for clustering*, Proc. of the 12th ACM SIGKDD Int. Conf. on Knowledge discovery and data mining, ACM, pp. 126-135.
17. Kim H.,Park H., 2008,*Nonnegative matrix factorization based on alternating nonnegativity constrained least squares and active set method*, SIAM Journal on Matrix Analysis and Applications, 30, 2, pp. 713-730.
18. Matlab Software by Mark Schmidt, www.di.ens.fr/~mschmidt/Software/minConf.html

19. Cichocki A.,Anh-HuyP., 2009, *Fast local algorithms for large scale nonnegative matrix and tensor factorizations*, IEICE Trans. on fundamentals of electronics, communications and computer sciences,92, 3, pp. 708-721.
20. Kim J.,Park H., 2011, *Fast nonnegative matrix factorization: An active-set-like method and comparisons*, SIAM Journal on Scientific Computing, 33, 6, pp. 3261-3281.
21. FévotteC., BertinN., DurrieuJ.L., 2011,*Nonnegative matrix factorization with the itakura-saito divergence: With application to music analysis*, Neural computation, 21, 3, pp. 793-830.
22. FévotteC., IdierJ., 2011, *Algorithms for nonnegative matrix factorization with the β-divergence*, Neural Computation, 23, 9, pp. 2421-2456.
23. He R., Zheng W.S., Hu B.G., 2011, *Maximum correntropy criterion for robust face recognition*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 33, 8, pp. 1561-1576.
24. Ensari T., Chorowski J., Zurada J.M., 2012, *Correntropy-Based document clustering via nonnegative matrix factorization*, Artificial Neural Networks and Machine Learning–ICANN 2012, Springer Berlin Heidelberg, pp. 347-354.
25. Ensari T., ChorowskiJ., Zurada J.M., 2012,*Occluded Face Recognition Using Correntropy-Based Nonnegative Matrix Factorization*, 11th International Conference on Machine Learning and Applications (ICMLA), 1, pp. 606-609.
26. Du L., Li X., Shen Y.D., 2012, *Robust Nonnegative Matrix Factorization via Half-Quadratic Minimization*, IEEE 12th International Conference on Data Mining (ICDM), pp. 201-210.