J. Appl. Math. & Informatics Vol. **30***(2012), No. 3 - 4, pp. 365 - 380*
Website: http://www.kcam.biz

CONVERGENCE ANALYSIS OF THE EAPG ALGORITHM FOR NON-NEGATIVE MATRIX FACTORIZATION[†]

CHENXUE YANG[*] AND MAO YE

Abstract. Non-negative matrix factorization (NMF) is a very efficient method to explain the relationship between functions for finding basis information of multivariate nonnegative data. The multiplicative update (MU) algorithm is a popular approach to solve the NMF problem, but it fails to approach a stationary point and has inner iteration and zero divisor. So the elementwisely alternating projected gradient (eAPG) algorithm was proposed to overcome the defects. In this paper, we use the fact that the equilibrium point is stable to prove the convergence of the eAPG algorithm. By using a classic model, the equilibrium point is obtained and the invariant sets are constructed to guarantee the integrity of the stability. Finally, the convergence conditions of the eAPG algorithm are obtained, which can accelerate the convergence. In addition, the conditions, which satisfy that the non-zero equilibrium point exists and is stable, can cause that the algorithm converges to different values. Both of them are confirmed in the experiments. And we give the mathematical proof that the eAPG algorithm can reach the appointed precision at the least iterations compared to the MU algorithm. Thus, we theoretically illustrate the advantages of the eAPG algorithm.

AMS Mathematics Subject Classification : 15A60.
Key words and phrases : Non-negative matrix factorization, convergence, equilibrium point.

1. Introduction

Non-negative matrix factorization (NMF) is a very efficient method to explain the relationship between functions for finding basis information of multivariate nonnegative data [1, 2]. By adding the constraint that all the elements of the

Received April 10, 2011. Revised July 18, 2011. Accepted September 10, 2011. [*]Corresponding author. [†]This work was supported in part by the National Natural Science Foundation of China (60702071), 973 National Basic Research Program of China, (2010CB732501), Fundation of Sichuan Excellent Young Talents(09ZQ026-035) and Open Project of State Key Lab. for Novel Software Technology of Nanjing University.

matrix are non-negative, it's fully to guarantee the interpretability of the decomposition [3]. Besides, it simplifies the realization and takes little storage space.

The traditional NMF problem may be described as a linear and non-negative approximate representation. Given a non-negative data matrix $V \in R^{m\times n}$, NMF finds an approximate factorization

$$V \approx WH, \tag{1}$$

where W and H are $m\times r$ and $r\times n$ non-negative matrices, respectively. Usually, the positive integer r is chosen to be smaller than m or n, so that the size of W and H are smaller than the original matrix V.

In fact, NMF is a problem of solving optimization. To find the matrices W and H, the problem in (1) is commonly reformulated as minimizing the following cost function:

$$f(W, H) := \frac{1}{2} \parallel V - WH \parallel_F^2, \tag{2}$$

where $\parallel \bullet \parallel_F$ represents the Frobenius norm.

In 1999, NMF was first published in [1]. Recently, NMF has received much attention and been applied to many application areas including image databases [5], subsystem identification [6], spectral data analysis [10], blind source separation (BSS) [14], text data mining [7, 8], speech processing [11, 12] and the novel fast multi-objective evolutionary [23].

There are many types of NMF techniques in the literatures, such as gradient descent methods [15], alternating least squares [4],and the popular multiplicative update algorithm (MU) [1]. However, the MU algorithm fails to approach a stationary point, which has been shown in [9] by numerical examples. For overcoming the defects of the MU algorithm having inner iteration and zero divisor, Lin [13] proposed three algorithms on NMF problem, which are the columnwisely alternating gradient (cAG) algorithm, the columnwisely alternating projected gradient (cAPG) algorithm and the elementwisely alternating projected gradient (eAPG) algorithm. And the eAPG algorithm can reach the appointed precision at the least iterations compared to the MU, cAG and cAPG algorithms in most cases. The cost functions of the eAPG algorithm is non-increasing as that of the MU algorithm. Note that there does not exist the mathematical proof of convergence of the eAPG algorithm. Therefore, in this paper, we analyze the convergence of the eAPG algorithm and show the advantages theoretically.

Currently, many classical methods, which are focused on the convergence analysis of the algorithms for NMF, have be found in the literatures , such as projected gradient approaches [17], bound optimization [18] and so on. Furthermore, some algorithms to accelerate the convergence are designed in [19]. In this paper, we will utilize the fact that the equilibrium point is stable to prove the convergence of the eAPG algorithm. First, we prove the existence of the equilibrium point by the Squeeze Theorem. Then, a classic model proposed in [20], is used to obtain the equilibrium point. At last, by using the method in

[16], we construct the invariant sets to constrain the area of the initializations to guarantee the integrity of the convergence analysis. By using the method of the above theories, we prove the eAPG algorithm is locally convergent. In addition, the conditions, which satisfy that the non-zero equilibrium point exists and is stable, can cause that the algorithm converges to different values.

Our contributions can be summarized as follows: First, we prove the convergence of the eAPG algorithm and obtain the structure of the equilibrium points, which lay a foundation for better understanding of the eAPG algorithm. Second, we show the advantages of the eAPG algorithm theoretically. And give the mathematical proof that the eAPG algorithm can reach the appointed precision at the least iterations.

This paper is structured as follows: A brief review of the eAPG algorithm is given in Section 2. In Section 3, we prove the existence of the equilibrium points of the eAPG algorithm and obtain their expressions by using the classic model in [20]. In Section 4, we construct the invariant sets to constrain the area of initializations and prove the convergence of the eAPG algorithm in Section 5. In Section 6, the simulations of the eAPG algorithm confirm our theories. Finally, a conclusion is given in Section 7.

2. Non-negative Matrix Factorization Algorithms

The multiplicative update (MU) algorithm is proposed by Lee and Seung [1] to solve the objective function (2), which is not defined well. In addition, it may appear inner iteration. To overcome these defects, Lin proposed the eAPG algorithm [13] which is described generally as follows:

1. Set $H = rand(r, n)$; $W = rand(m, r)$;
2. For iter = 1: maxiter, compute

$$H_{ij} = H_{ij} - \eta_{ij},\ for\ j = 1, 2, ..., n;\ i = 1, 2, ..., r, \tag{3}$$

$$W_{ij} = W_{ij} - \xi_{ij},\ for\ i = 1, 2, ..., m;\ j = 1, 2, ..., r, \tag{4}$$

where η_{ij} is calculated by

$$\eta_{ij} = \begin{cases} \eta^*_{ij}, & if\ h_{ij} - \eta^*_{ij} \geq 0\ and\ W_i \neq 0, \\ 0, & otherwise, \end{cases} \tag{5}$$

and

$$\eta^*_{ij} = \frac{(W_i)^T(WH_j - V_j)}{(W_i)^T W_i}, \tag{6}$$

and ξ_{ij} is defined by

$$\xi_{ij} = \begin{cases} \xi^*_{ij}, & if\ w_{ij} - \xi^*_{ij} \geq 0\ and\ H_j \neq 0, \\ 0, & otherwise, \end{cases} \tag{7}$$

and

$$\xi^*_{ij} = \frac{(H_j^T)^T(H^T W_i^T - V_i^T)}{(H_j^T)^T H_j^T}. \tag{8}$$

3. Existence of Equilibrium Points

Similarly, as the method in [16], before analyzing the convergence of the eAPG algorithm, we need to prove the existence of the equilibrium points and obtain the structures of them. Both H and W will be updated alternately using Eq. (3) and Eq. (4), but they are computed separately. Thus, we consider W being fixed firstly to prove that the equilibrium point of H exists. And by using the model in [20], we obtain its structure. Then, we fix H to prove that the equilibrium point of W exists, and obtain its structure in the same way.

From the description of W and H, we have the following expressions

$$WH_j = \begin{pmatrix} \sum\limits_{i=1}^{r} w_{1i}h_{ij} \\ \sum\limits_{i=1}^{r} w_{2i}h_{ij} \\ \vdots \\ \sum\limits_{i=1}^{r} w_{mi}h_{ij} \end{pmatrix} \tag{9}$$

and

$$(W_i)^T(WH_j - V_j) = \sum_{k=1}^{m} w_{ki}(\sum_{i=1}^{r} w_{ki}h_{ij} - v_{kj}) \ . \tag{10}$$

From Eq. (6), it follows that

$$\eta^*_{ij} = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij} - v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2}. \tag{11}$$

From Eq. (5), if $h_{ij} - \eta^*_{ij} \geq 0$, then

$$\eta_{ij} = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij} - v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2}. \tag{12}$$

From Eq. (3) and Eq. (12), we have the following algorithm

$$h_{ij} \longleftarrow h_{ij} - \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij} - v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2}. \tag{13}$$

To discuss the convergence of the eAPG algorithm for H, we only need to discuss the convergence of algorithm (13).

Definition 1. For the algorithm (13), a point $h_{ij} \in R$ is called an equilibrium if and only if it satisfies

$$\frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij} - v_{kj})}{\sum\limits_{k=1}^{m} (w_{ki})^2} = 0. \tag{14}$$

Denote

$$f(h_{ij}) = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij} - v_{kj})}{\sum\limits_{k=1}^{m} (w_{ki})^2}. \tag{15}$$

We assume all elements in H except h_{ij} , w_{ki} and v_{kj} in W and V are constants. In addition, in each iteration, both h_{ij} and w_{ij} will be updated alternately. Thus, it is reasonable to consider w_{ij} as a constant here for the convergent analysis of h_{ij}. Similarly, for the analysis of w_{ij}, we consider h_{ij} as a constant.

Theorem 1. *There exists $h_{ij}^{(0)} \in [0, \infty)$, such that $f(h_{ij}^{(0)}) = 0$, i.e., $h_{ij}^{(0)}$ is the equilibrium of algorithm (13).*

Proof. Obviously, $f(h_{ij})$ is a continuous function. For the initialization A in Eq. (14), let $h_{ij}^{(1)} > 0$ and $\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(1)} < v_{kj}$. From Eq. (10), we obtain $\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(1)} - v_{kj}) < 0$. Thus, from Eq. (15), there is

$$f(h_{ij}^{(1)}) = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(1)} - v_{kj})}{\sum\limits_{k=1}^{m} (w_{ki})^2} < 0.$$

Increasing $h_{ij}^{(1)}$, we can obtain another point $h_{ij}^{(2)}$, such that $\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(2)} > v_{kj}$. Similarly,

$$f(h_{ij}^{(2)}) = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(2)} - v_{kj})}{\sum\limits_{k=1}^{m} (w_{ki})^2} > 0.$$

According to the Intermediate Value Theorem, there exists $h_{ij}^{(0)}$, $h_{ij}^{(1)} < h_{ij}^{(0)} < h_{ij}^{(2)}$, such that

$$f(h_{ij}^{(0)}) = \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}^{(0)} - v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2} = 0.$$

□

From Theorem 1, algorithm (13) exists the equilibrium point $h_{ij}^{(0)}$. Next, we need to obtain the structure of $h_{ij}^{(0)}$ to analyze the stability of algorithm (13). From the classic model in [20], we have the following lemma.

Lemma 1. *In difference system $x(t+1) - x(t) = F(x(t))$, there exists an equilibrium point $x^\star$, if $F_1(x) \leq F(x) \leq F_2(x)$, where $F_1(x)$ and $F_2(x)$ are monotonous, which satisfies $\eta_1 F_2(x^\star) - F(x^\star) = F(x^\star) - \eta_2 F_1(x^\star) = 0$.*

Lemma 1 is used to obtain the structure of the equilibrium point. For simplicity, we denote

$$w_1 = \min_{1\leq k\leq m, 1\leq i\leq r}\{w_{ki}\},\ w_2 = \max_{1\leq k\leq m, 1\leq i\leq r}\{w_{ki}\}, \tag{16}$$

$$v_1 = \min_{1\leq k\leq m, 1\leq j\leq n}\{v_{kj}\},\ v_2 = \max_{1\leq k\leq m, 1\leq j\leq n}\{v_{kj}\}. \tag{17}$$

Note that all elements in the jth column of H except h_{ij} are constants, then we assume

$$P = \sum_{p=1}^{r} h_{pj} - h_{ij}. \tag{18}$$

Using these notations, we have the following theorem.

Theorem 2. *There exists a non-zero equilibrium point of algorithm (13), for some constants $0 \leq \eta_1 \leq 1$ and $\eta_2 \geq 1$, the structure of the equilibrium point is $h_{ij} = \dfrac{v_1}{\eta_1 w_2} - P$ or $h_{ij} = \dfrac{v_2}{\eta_2 w_1} - P$ in the condition $\eta_2 w_1 v_1 = \eta_1 w_2 v_2$.*

Proof. For the term

$$\sum_{i=1}^{r} w_{ki}h_{ij} = \sum_{p=1}^{i-1} w_{kp}h_{pj} + \sum_{q=i+1}^{r} w_{kq}h_{qj} + w_{ki}h_{ij},$$

from the above equations, we have

$$w_1(P + h_{ij}) - v_2 \leq \sum_{i=1}^{r} w_{ki}h_{ij} - v_{kj} \leq w_2(P + h_{ij}) - v_1.$$

In the above inequalities, only h_{ij} is variable and all the others are constants, there is

$$\frac{w_1[w_1(P+h_{ij})-v_2]}{w_2^2} \le \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}-v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2} \le \frac{w_2[w_2(P+h_{ij})-v_1]}{w_1^2}. \quad (19)$$

For the given initializations of W and H, we can choose two constants $0 \le \eta_1 \le 1$ and $\eta_2 \ge 1$, such that

$$\frac{w_1[\eta_2 w_1(P+h_{ij})-v_2]}{w_2^2} \le \frac{\sum\limits_{k=1}^{m} w_{ki}(\sum\limits_{i=1}^{r} w_{ki}h_{ij}-v_{kj})}{\sum\limits_{k=1}^{m}(w_{ki})^2} \le \frac{w_2[\eta_1 w_2(P+h_{ij})-v_1]}{w_1^2}. \quad (20)$$

Base on Lemma 1, from Eq. (14), we obtain the following two equations,

$$\frac{w_2[\eta_1 w_2(P+h_{ij})-v_1]}{w_1^2} = 0 \quad (21)$$

and

$$\frac{w_1[\eta_2 w_1(P+h_{ij})-v_2]}{w_2^2} = 0. \quad (22)$$

Eq. (21) has solution

$$h_{ij} = \frac{v_1}{\eta_1 w_2} - P. \quad (23)$$

And Eq. (22) has solution

$$h_{ij} = \frac{v_2}{\eta_2 w_1} - P. \quad (24)$$

From Theorem 1 and the above analysis, there exists an equilibrium point for the algorithm (13) if the non-zero solutions of Eq. (14) satisfy

$$\frac{v_1}{\eta_1 w_2} - P = \frac{v_2}{\eta_2 w_1} - P.$$

The above equation can be simplified to

$$\eta_2 w_1 v_1 = \eta_1 w_2 v_2, \quad (25)$$

which is the condition of existence of equilibrium.

Therefore, from Lemma 1, for the algorithm (13), the structure of the equilibrium point is Eq. (23) or Eq. (24) which satisfies Eq. (25). □

Analogously, for a fixed H, we can get similar results with respect to W. Similarly, we have the following expressions,

$$(H_j^T)^T(H^T W_i^T - V_i^T) = \sum_{p=1}^{n} h_{jp}(\sum_{k=1}^{r} h_{kp}w_{ik} - v_{ip}). \quad (26)$$

From Eq. (8), it follows that

$$\xi_{ij}^* = \frac{\sum\limits_{p=1}^{n} h_{jp}(\sum\limits_{k=1}^{r} h_{kp} w_{ik} - v_{ip})}{\sum\limits_{p=1}^{n} (h_{jp})^2}. \tag{27}$$

From Eq. (7), if $h_{ij} - \xi_{ij}^* \geq 0$, then

$$\xi_{ij} = \frac{\sum\limits_{p=1}^{n} h_{jp}(\sum\limits_{k=1}^{r} h_{kp} w_{ik} - v_{ip})}{\sum\limits_{p=1}^{n} (h_{jp})^2}. \tag{28}$$

Hence, we need to discuss the convergence of the following algorithm,

$$w_{ij} \longleftarrow w_{ij} - \frac{\sum\limits_{p=1}^{n} h_{jp}(\sum\limits_{k=1}^{r} h_{kp} w_{ik} - v_{ip})}{\sum\limits_{p=1}^{n} (h_{jp})^2}. \tag{29}$$

Definition 2. For the algorithm (29), a point $w_{ij} \in R$ is called an equilibrium if and only if it satisfies

$$\frac{\sum\limits_{p=1}^{n} h_{jp}(\sum\limits_{k=1}^{r} h_{kp} w_{ik} - v_{ip})}{\sum\limits_{p=1}^{n} (h_{jp})^2} = 0. \tag{30}$$

Denote

$$h_1 = \min_{1 \leq j \leq r, 1 \leq p \leq n} \{h_{jp}\}, \; h_2 = \max_{1 \leq j \leq r, 1 \leq p \leq n} \{h_{jp}\},$$

and

$$Q = \sum_{k=1}^{r} w_{ik} - w_{ij},$$

where Q representatives as the sum of all elements in the ith row of H except h_{ij}. Contrarily, only w_{ij} is variable and all the others are constants, then we have

$$\frac{h_1[h_1(Q + w_{ij}) - v_2]}{h_2^2} \leq \frac{\sum\limits_{p=1}^{n} h_{jp}(\sum\limits_{k=1}^{r} h_{kp} w_{ik} - v_{ip})}{\sum\limits_{p=1}^{n} (h_{jp})^2} \leq \frac{h_2[h_2(Q + w_{ij}) - v_1]}{h_1^2},$$

which is similar to the inequality (19). From this expression, we have the following Theorem.

Theorem 3. *There exists a non-zero equilibrium point of algorithm (29), for some constants* $0 \leq \eta_1 \leq 1$ *and* $\eta_2 \geq 1$, *the structure of equilibrium point is* $w_{ij} = \dfrac{v_1}{\eta_1 h_2} - Q$ *or* $w_{ij} = \dfrac{v_2}{\eta_2 h_1} - Q$ *in the condition* $\eta_2 h_1 v_1 = \eta_1 h_2 v_2$.

Proof. The proof is similar to Theorem 2. □

From Eq. (25), the condition, which satisfies that the non-zero equilibrium point exists, is determined by η_1 and η_2. Then, the value of η_1 or η_2 can cause that the algorithm converges to different values, which will be verified in the experiments.

4. Invariant Sets

Definition 3 ([21])**.** A set S is called an invariant set for a dynamic system $x_{n+1} = f(x_n)$, if for any initial value $x_0 \in S$, the updates x_{n+1} of system starting from x_0 will remain in S for all $n \geq 0$.

From the proof of Theorem 2, we assume all the others are constants except h_{ij}, which can be decided in the upcoming discussion. We can prove Theorems 4 and 5.

Denote

$$H_1 = \{h_{ij} | h_{ij} \in R,\ \frac{v_1}{w_2} - P \leq h_{ij} \leq \frac{v_2}{w_1} - P\}, \tag{31}$$

$$W_1 = \{w_{ij} | w_{ij} \in R,\ \frac{v_1}{h_2} - Q \leq w_{ij} \leq \frac{v_2}{h_1} - Q\}. \tag{32}$$

Theorem 4. *Suppose* v_1, v_2, w_1, w_2 *and* P *are constants,* H_1 *is an invariant set of algorithm (13).*

Proof. Suppose

$$0 < \frac{v_1}{w_2} - P \leq h_{ij}(t) \leq \frac{v_2}{w_1} - P. \tag{33}$$

From algorithm (13), for the $(t+1)$th update, we have

$$h_{ij}(t+1) = h_{ij}(t) - \frac{\sum\limits_{k=1}^{m} w_{ki}\big(\sum\limits_{i=1}^{r} w_{ki} h_{ij}(t) - v_{kj}\big)}{\sum\limits_{k=1}^{m} (w_{ki})^2}. \tag{34}$$

From the inequality (19), it follows that

$$h_{ij}(t) - \frac{w_2\big[w_2\big(P + h_{ij}(t)\big) - v_1\big]}{w_1^2} \leq h_{ij}(t+1) \leq h_{ij}(t) - \frac{w_1\big[w_1\big(P + h_{ij}(t)\big) - v_2\big]}{w_2^2},$$

where v_1, v_2, w_1, w_2 and P are constants. And the above inequality follows that

$$h_{ij}(t)\big(1 - \frac{w_2^2}{w_1^2}\big) - \frac{w_2^2}{w_1^2}P + \frac{w_2}{w_1^2}v_1 \leq h_{ij}(t+1) \leq h_{ij}(t)\big(1 - \frac{w_1^2}{w_2^2}\big) - \frac{w_1^2}{w_2^2}P + \frac{w_1}{w_2^2}v_2.$$

From the inequality (33), we have

$$(\frac{v_1}{w_2}-P)(1-\frac{w_2^2}{w_1^2})-\frac{w_2^2}{w_1^2}P+\frac{w_2}{w_1^2}v_1 \leq h_{ij}(t+1) \leq (\frac{v_2}{w_1}-P)(1-\frac{w_1^2}{w_2^2})-\frac{w_1^2}{w_2^2}P+\frac{w_1}{w_2^2}v_2.$$

Thus, the above inequality can be simplified to

$$\frac{v_1}{w_2}-P \leq h_{ij}(t+1) \leq \frac{v_2}{w_1}-P. \tag{35}$$

which is same as the inequality (33). The inequality (35) shows that for any $t \geq 0$, if $\frac{v_1}{w_2}-P \leq h_{ij}(t) < \frac{v_2}{w_1}-P$, there always exists $\frac{v_1}{w_2}-P \leq h_{ij}(t+1) \leq \frac{v_2}{w_1}-P$. Therefore, H_1 is an invariant set of algorithm (13). □

Theorem 5. *Suppose v_1, v_2, h_1, h_2 and Q are constants, W_1 is an invariant set of algorithm (29).*

Proof. The proof can be omitted since it's similar to Theorem 4. □

Theorem 4 and 5 guarantee that any trajectory of algorithm (13) and algorithm (29) starting from any points in the invariant sets H_1 and W_1 will stay in H_1 and W_1 correspondingly. In the following section, we will prove the equilibrium point is stable to complete the proof of convergence.

5. Convergence Analysis

Theorem 6. *For the algorithm (13), the non-zero equilibrium point is stable if it satisfies the condition Eq. (25) and is restricted in the invariant set (31).*

Proof. From Eq. (34) and Eq. (15), it follows that

$$h_{ij}(t+1)=h_{ij}(t)-f(h_{ij}(t)). \tag{36}$$

To discuss the convergence of algorithm (13), from Eq. (14), we just need to discuss the stability of $f(h_{ij})$. From the inequality (20), for simplicity, we denote

$$G_1(h_{ij})=\frac{w_2[\eta_1 w_2(P+h_{ij})-v_1]}{w_1^2} \tag{37}$$

and

$$G_2(h_{ij})=\frac{w_1[\eta_2 w_1(P+h_{ij})-v_2]}{w_2^2}, \tag{38}$$

which have

$$G_2(h_{ij}) \leq f(h_{ij}) \leq G_1(h_{ij}).$$

Using Lemma 1 to obtain the equilibrium point, $G_1(h_{ij})$ and $G_2(h_{ij})$ must satisfy monotonicity. In addition, the terms that we substitute the equilibrium point into the derivatives of $G_1(h_{ij})$ and $G_2(h_{ij})$ need to be less than 1 [21].

First, we discuss the monotonicity of the following $G_1(h_{ij})$ and $G_2(h_{ij})$. From Eq. (37) and Eq. (38), there is

$$G_1'(h_{ij})=\frac{\eta_1 w_2^2}{w_1^2}>0,$$

and similarly

$$G_2'(h_{ij}) = \frac{\eta_2 w_1^2}{w_2^2} > 0.$$

Substituting the non-zero solutions of Eq. (23) and Eq. (24) into the above equations, we obtain

$$\frac{d(G_1(h_{ij}))}{dh_{ij}}\Big|_{\frac{v_1}{\eta_1 w_2} - P} = \frac{\eta_1 w_2^2}{w_1^2} < 1$$

and

$$\frac{d(G_2(h_{ij}))}{dh_{ij}}\Big|_{\frac{v_2}{\eta_2 w_1} - P} = \frac{\eta_2 w_1^2}{w_2^2} < 1.$$

Therefore, for some constants $0 \leq \eta_1 \leq 1$ and $\eta_2 \geq 1$, if it satisfies the following inequality

$$\eta_1 w_2^2 < w_1^2 \leq \eta_2 w_1^2 < w_2^2, \tag{39}$$

the equilibrium point is stable. Therefore, base on Lemma 1, the function $f(h_{ij})$ is stable at the equilibrium point $\dfrac{v_1}{\eta_1 w_2} - P$ or $\dfrac{v_2}{\eta_2 w_1} - P$ in the invariant set H_1 if it satisfies the condition Eq. (39). □

By the same way, we have the stability theorem for w_{ij} in algorithm (29).

Theorem 7. *For the algorithm (29), the non-zero equilibrium point exists and is stable if it's restricted in the invariant set (32).*

From Theorem 2 and Theorem 6, we have that the equilibrium point of algorithm (13) is stable, and from Theorem 4 we know that all the initializations are constrained in the invariant sets, which means the equilibrium point of algorithm (13) is stable in H_1.

According to the Theorem 4 in [22], we obtain the eAPG algorithm is locally convergent at the equilibrium points in the invariant sets. There are two conditions to constrain the convergence of the eAPG algorithm. For h_{ij}, one condition comes from the restrictions on invariant set H_1, and another condition comes from the conditions of existence of equilibrium. From the inequality (39), the condition, which satisfies that the non-zero equilibrium point is stable, is also determined by η_1 and η_2. And the value of η_1 or η_2 can cause that the algorithm converges to different values. Now, we will confirm this circumstance in the experiments.

6. Simulation and Discussions

With various statistical distributions, NMF algorithms have been extensively used to test data for signals and images. In this section, we present numerical experimental results to illustrate the convergence of the eAPG algorithm and analyze the importance of the two convergence conditions, especially Eq. (25) and the inequality (39).

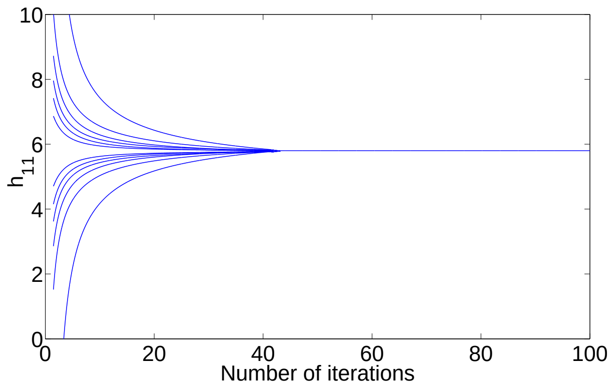

FIGURE 1. The convergence of h_{11}.

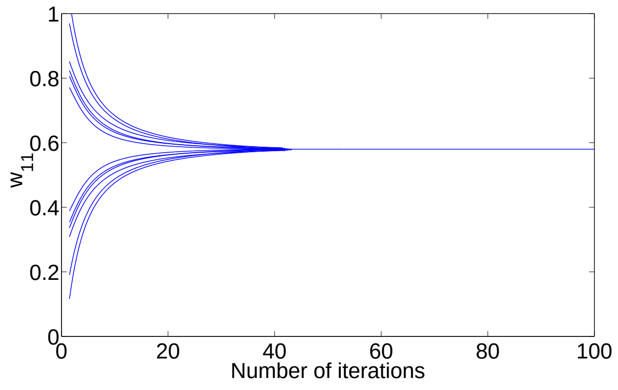

FIGURE 2. The convergence of w_{11}.

For NMF algorithms in the practical application, we want to choose suitable initializations to obtain the correct data separation by testing several groups data. To estimate the original sources in a very small error, the stable conditions Eq. (31), Eq. (32) and the inequality (25) are satisfied in the initializations. Then we have the following description of the algorithm steps [16].

1. Initialize the vector h_{ij} and non-negative matrix $W = \{w_{ki}\}_{mr}$ with positive decimal numbers;

2. According the computing result of P, Q and $\{v_{kj}\}_{m\times n}$, select suitable $\alpha = \dfrac{\eta_1}{\eta_2}$ for the test;

3. To avoid the divergence, set the threshold for h_{ij} and w_{ij} to 0.001-0.005, testing the convergence condition;

4. Compute h_{ij} and w_{ij} according to the algorithm (13) and (29) to obtain the factorization of v_{pj}.

5. If h_{ij} and w_{ij} have not converged, go back to step 3.

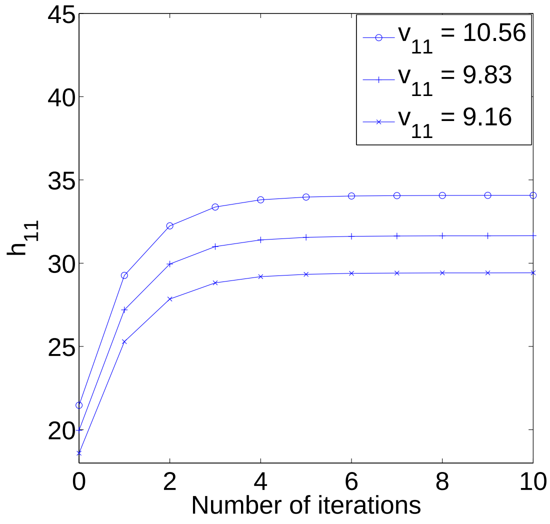

FIGURE 3. The convergence of the algorithm (40) with different v_{11}.

In experiment 1, we first select a group of data to demonstrate the analysis results. Without loss of generality, we only consider the convergence of w_{11} and h_{11} with different initializations. From algorithm (13), it follows that

$$h_{11} \longleftarrow h_{11} - \frac{\sum\limits_{k=1}^{m} w_{k1}(\sum\limits_{i=1}^{r} w_{ki}h_{i1} - v_{k1})}{\sum\limits_{k=1}^{m} (w_{k1})^2}. \tag{40}$$

We test the algorithm (40) for a randomly generated full column rank matrix $V \in R^{100\times 100}$. For initializations of $P = \sum\limits_{p=1}^{r} h_{p1} - h_{11} = 1.01$, with the condition $\eta_1 w_2^2 < w_1^2 \leq \eta_2 w_1^2 < w_2^2$, we select $\alpha = 0.01$. Here, we show the convergence of h_{11} in Figure 1 and the convergence of w_{11} in Figure 2, respectively. Therefore, for different initializations of h_{11}, it always converges to the same constant if the condition $\eta_1 w_2^2 < w_1^2 \leq \eta_2 w_1^2 < w_2^2$ is satisfied. And w_{11} has the same result. The convergence of the eAPG algorithm is verified.

In experiment 2, we test the algorithm (40) for a randomly generated full column rank matrix V and the matrix $W \in R^{100\times 100}$. For initializations of $P = 1.01$, with the condition $\eta_1 w_2^2 < w_1^2 \leq \eta_2 w_1^2 < w_2^2$, we select $w_1 = 1.21$, $w_2 = 3.01$ and $\eta_1 = 0.1$. To show the simulation result, we change v_{11}. Figure 3 shows that the trend and the number of iterations are almost the same with different values of v_{11}. Thus, the change of v_{11} affect little the convergence of the algorithm.

To improve the speed and accuracy of the algorithm, a good initialization is very important. If the initializations are selected improperly, even it satisfies the

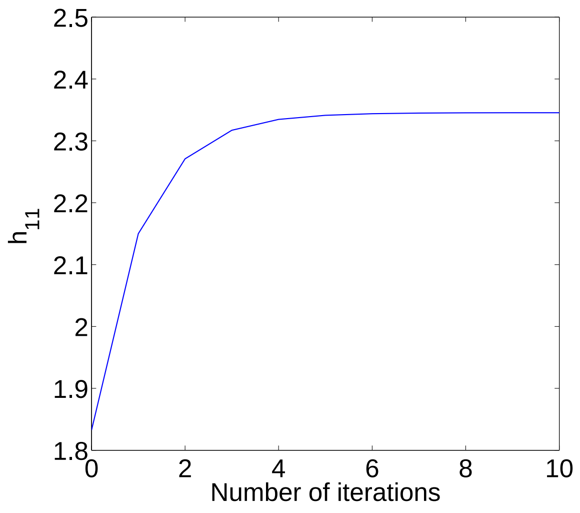

FIGURE 4. The convergence of the algorithm (40) with $\eta_1 = 0.1$.

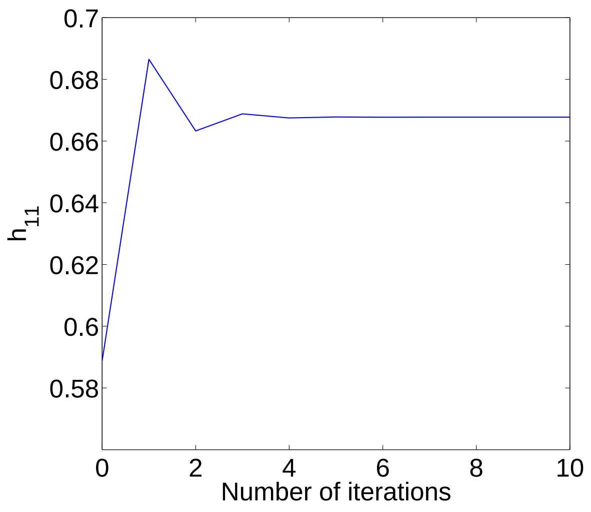

FIGURE 5. The convergence of the algorithm (40) with $\eta_1 = 0.2$.

condition Eq. (25), the algorithm may iterate many times to converge. Then, we want to find which element can affect the initialization in the eAPG algorithm.

In experiment 3, we test the algorithm (40) for the same matrix $V \in R^{100\times 100}$ as that in experiment 2. From Eq. (25), we can find that the condition, which satisfies that the non-zero equilibrium point exists, is determined by η_1 and η_2. With different initializations of η_1, the convergence of the algorithm (40) is shown in Figure 4 and Figure 5.

Figure 4 shows that the algorithm (40) converges to 2.34 in the condition $\eta_1 = 0.1$, and Figure 4 shows that the algorithm (40) converges to 0.668 in the condition $\eta_1 = 0.2$. Different values of η_1 and η_2 will affect that the algorithm converge to different values. Therefore, it's important to select the appropriate η_1 and η_2.

7. Conclusion

In this paper, we analyze the convergence of the eAPG algorithm by utilizing the fact that the equilibrium point is stable. We obtained the structure of the equilibrium points and the convergence conditions. In addition, using the conditions of the initialization constraints can accelerate the convergence of the algorithm. In the end, simulations illustrate the advantages of the eAPG algorithm and support our theories.

References

1. D.D. Lee, and H.S. Seung, *Learning the parts of objects by non-negative matrix factorization*, Nature **401** (2002), 788-791.
2. P. Paatero, and U. Tapper, *Positive matrix factorization: A non-negative factor model with optimal utilization of error*, Environmetrics **5** (1994), 111-126.
3. D.D. Lee, and H.S. Seung, *Algorithms for Non-negative Matrix Factorization*, Advances in Neural Information Processing, MIT Press **13** (2001), 556-562.
4. M. Berry, and M. Browne, *Algorithms and applications for approximate nonnegative matrix factorization*, Computational Statistics and Data Analysis **52** (2006), 155-173.
5. M. Chu, and F. Diele, *Optimality, computation, and interpretation of nonnegative matrix factorizations*, Siam Journal on Matrix Analysis **4** (2004), 8030-8048.
6. P.M. Kim, and B. Tidor, *Subsystem identification through dimensionality reduction of large-scale gene expression data*, Genome Research **13** (2003), 1706-1718.
7. L.K. Saul, F. Sha and D.D. Lee, *Statistical signal processing with nonnegativity constraints*, Proceedings of EuroSpeech **2** (2003), 1001-1004.
8. F. Shahnaz, M.W. Berry, V.P. Pauca, and R. Plemmons, *Document clusting using nonnegative matrix factorization*, Information Processing and Management **42** (2006), 373-386.
9. E.F. Gonzales, and Y. Zhang, *Accelerating the Lee-Seung algorithm for non-negative matrix factorization*, Dept. Comput. Appl. Math., Rice Univ., Houston, TX, Tech. Rep ,2005.
10. V.P. Pauca, J. Piper, and R.J. Plemmons, *Nonnegative matrix factorization for spectral data analysis*, Linear Algebra and its Applications **416** (2006), 29-47.
11. F. Sha, and L.K. Saul, *Real-time pitch determination of one or more voices by nonnegative matrix factorization*, Advances in Neural Information Processing Systems, online papers (2005).
12. P. Smaragdis, and J.C. Brown, *Non-negative matrix factorization for polyphonic music transcription*, Proceedings of IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (2003), 177-180.
13. L. Lin, *Alternative gradient algorithms with applications to nonnegative matrix factorizations*, Applied Mathematics and Computation (2010), 1763-1770.
14. A. Cichocki, R. Zdunek, and S. Amari, *New algorithms for non-negative matrix factorization in applications to blind source separation*, ICASSP, Toulouse, France (2006), 621-625.
15. C.J. Lin, *Projected gradient methods for non-negative matrix factorization*, Tech. Report Information and Support Service ISSTECH (2005), 95-013.

16. S.M. Yang, and Z. Yi, *Convergence analysis of non-negative matrix factorization for BSS algorithm*, Neural Process Lett (2010), 45-64.
17. A. Cichocki, and R. Zdunek, *Multilayer non-negative matrix factorization using project gradient approaches*, International Journal of Neural Systems (2007), 431-446.
18. R. Salakhutdinov, S.T. Roweis, and Z. Ghahramani, *The convergence of bound optimization algorithms*, International Conference on Uncertainty in Artificial Intelligence **19** (2003).
19. C.J. Lin, *On the convergence of multiplicative update algorithms for nonnegative matrix factorization*,Transactions on Neural Networks **18** (2007), 1589-1596.
20. V.L. Kocic, and G. Ladas, *Global behavior of nonlinear difference equations of higher order*, Kluwer Academic Publishers, 1953.
21. J.K. Hale, *Theory of functional differential equations*, Lecture Notes in Mathematics (1971), 183-1971.
22. J. Dai, and S. Meyn, *Stabilitity and convergence of moments for multiclass queueing networks via fluid limit models*, Translation on Automatic Control (1995), 1889-1904.
23. Y.B. Liu, *A novel fast multi-objective evolutionary algorithm for QoS multicast routing in manet*, International Journal of Computational Intelligence Systems **3** (2009), 288.

Chenxue Yang received B.S. degree from Mathematics Science University of Electronic Science and Technology of China. Since 2011 she studied for Ph.D. at school of Computer Science and Engineering, University of Electronic Science and Technology of China, P.R. China and State Key Lab. for Novel Software TechnologyNanjing University, P.R. China. She research interests include numerical optimization and computer vision.

Department of Computer Science and Engineering, University of Electronic Science and Technology of China, P.R. China and State Key Lab. for Novel Software TechnologyNanjing University, P.R. China.
e-mail: yang.chenxue@163.com

Mao Ye received his B.S. degree in mathematics from Sichuan Normal University, Chengdu, China, in 1995, M.S. degree inmathematics from University of Electronic Science and Technology of China, Chengdu, China, in 1998, and Ph.D. degree in mathematics from ChineseUniversity ofHong Kong, in 2002. He is currently a professor and director of CVLab. His current research interests include machine learning and computer vision. In these areas, he has published over 70 papers in leading international journals or conference proceedings.

Department of Computer Science and Engineering, University of Electronic Science and Technology of China, P.R. China and State Key Lab. for Novel Software TechnologyNanjing University, P.R. China.
e-mail: yem_mei29@hotmail.com