A novel initialization method for symmetric nonnegative matrix factorization

Jian-Qiang Wu and Hao-Xia Huang
School of computer, JiaYing University, Meizhou, 514015,China
E-mail: wjqjyu@163.com, hhxjyu@163.com

Symmetric nonnegative matrix factorization(SNMF) has widely employed in many areas of applications. Authors proposed a novel Lanczos tridiagonalization-based initialization method for SNMF, which can be combined with existing SNMF algorithms and achieve higher efficiency. Experiments show that the SNMF algorithm which combined the proposed initialization method can converge to a better solution.

Keywords: Symmetric Nonnegative Matrix Factorization; Initialization; Lanczos Tridiagonalization.

1. Introduction

Symmetric nonnegative matrix factorization (SNMF) [1] algorithms have been successfully employed in many areas of applications, such as image processing and machine learning. This problem can be described briefly as follow:

Denoted by $R_+^{n,n}$ the set of all square n-by-n nonnegative matrices, SNMF problem we consider is that for a given symmetric matrix $A \in R_+^{n,n}$ and $k \ll n$, finding $W \in R_+^{n,k}$ such that $A \approx WW^T$, W is a nonnegative matrix and WW^T is an approximation of the given symmetric matrix A in some distant metrics(in this paper, authors use Frobenius norm $\|.\|_F$ as the distant metrics).

In some cases, such as A is indefinite and has some negative eigenvalues, WW^T will not provide a good approximation for A because WW^T cannot absorb the subspaces associated with negative eigenvalues. However, the weighted SNMF $A = WSW^T$ can absorb the subspaces associated with both positive and negative eigenvalues. Also, we know that matrix have Choleskey factorization $A = LL^T$ if A is symmetric positive definition. Otherwise, one does $A = LDL^T$ where the diagonal matrix D takes care of the negative eigenvalues.

Most of existing NMF-related researches are focused on the factorization algorithm. Yang et.al proposed a nonnegative matrix factorization algorithm based a trained projection matrix [2]. Kannan et.al proposed distributed-memory parallel algorithm that computes the factorization by iteratively solving alternating non-negative least squares (NLS) subproblems [3]. The need to

Copyright © 2017, the Authors. Published by Atlantis Press.
This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

investigate good initialization strategies for NMF are mentioned [4], however, few of them referred to SNMF initialization strategies. Ding et al. [5] proposed an initialization method of SNMF, but only one of the factors (e.g. W) is initialized. Gloub et.al proposed a method to enhance the initialization of non-negative matrix factorization based on double singular value decomposition [6].

In this paper, authors proposed an initialization method in which both of the factors (W,S) are initialized based on Lanczos tridiagonalization and can be readily combined with available SNMF algorithms to achieve higher efficiency.

2. Our Work

In our scheme, the initialization method is based on Lanczos tridiagonalization, so we introduce how to get low-rank approximation matrix based on Lanczos tridiagonalization [7] briefly at first.

2.1. *Lanczos-based low-rank approximation*

Let b be a starting vector, for i=1,2…, compute Equ.(1):

$$\begin{cases} \beta_1 u_1 = b\,;\ \alpha_1 = u_1^T A u_1;\ \alpha_i = u_i^T A u_i \\ \beta_{i+1} u_{i+1} = (A - \alpha_i I)u_i - \beta_i u_{i-1} \end{cases} \quad (1)$$

β_{i+1} should be chosen nonnegative and satisfy $\|u_{i+1}\|=1$, $\|\bullet\|$ is either vector or matrix two-norm at the rest of the paper. Equ.(1) can be written compactly as:

$$U_{k+1}(\beta_1 e_1) = b, AU_k = U_{k+1}T_{k+1}(:,1:k) \quad (2)$$

Where $T_{k+1} \in R^{k+1,k+1}$, is symmetric tridiagonal matrix and $T_{k+1}(:,1:k)$ is T_{k+1} with last column deleted.

If we want to find a few dominant singular triplets of A, we must compute the singular value decomposition(SVD) of T_k. So the singular values of T_k are then used as approximations of the singular values of A and the singular vectors of T_k are combined with Lanczos vectors U_k to form approximations of the singular vectors of A [8]. But we are only interested in finding a low-rank approximation of A, thus, a direct approach can be used without computing the SVD of T_k. So it is natural to choose the low-rank approximation of A as follow:

$$J_k = U_k T_k U_k^T \quad (3)$$

Following the errors analysis in [9], it is straightforward to show that in finite precision arithmetic, (2) become

$$\hat{U}_{k+1}(\hat{\beta}_1 e_1) = b, \hat{A}\hat{U}_k = \hat{U}_{k+1}\hat{T}_{k+1}(:,1:k) + F_k \quad (4)$$

Where $||F_k|| = O(||A||_F \epsilon_M)$, $F_i = [f_1, f_2, \ldots, f_i]$ and ϵ_M is machine epsilon Also, we denote "^" the computed version of a quantity. The Lanczos process is described as follow.

Algorithm 1. Lanczos process in Matlab

Inputs: A symmetric matrix $A \in R_+^{n,n}$, positive vector $b \in R_+^n$, integer $0<k<n$.

Outputs: Rank-k matrix $U \in R^{n,k}$, $T \in R_+^{k,k}$.

Steps: 1. y=norm(b); u=b/y; f=A*u; x=u'*f; f=f-x*u; T(1,1)=x;

2. U=[u];

3. For j=2:k

y=norm(f); u0=u; u=f/y; f=A*u-u0*y; x=u'*f; f=f-x*u;

T(j, j-1)=T(j-1,j)=y; U=[U, u]; End;

2.2. *Our method of initialization of (W, S)*

For convenience, we denote by $A \geq B$ the componentwise inequality $\alpha_{i,j} > \beta_{i,j}$ for all elements of A and B. Given any vector or matrix X, its "positive section", $X_+ \geq 0$, will be defined to be the vector or matrix of the same size that contains the same values as X where X has nonnegative elements and 0 else- where. The "negative section" of X will be the matrix $X_- = X_+ - X$, $X_- \geq 0$. It follows that any vector or matrix X can be written as $X = X_+ - X_-$, and if $X \geq 0$ then $X_- = 0$. Our strategy is similar to the strategy in [6], so we first review some results.

Lemma 1. For any matrix $C \in R^{m,n}$ of rank one, rank(C_+), rank(C_-)$\leq$2.

Lemma 2. Let $C \in R^{m,n}$ have unit rank, so that $C=uv^T$ for some $u \in R^m$, $v \in R^n$. Also, let $\hat{u}_\pm := u_\pm/||u_\pm||$, $\hat{v}_\pm := v_\pm/||v_\pm||$ be the normalized positive and negative sections of u and v, and $\hat{\sigma}_\pm := ||u_\pm||\,||v_\pm||$, $\hat{\eta}_\pm := ||u_\pm||\,||v_\mp||$. Then the unordered singular value expansions C_+ and C_- are:

$$C_+ = \sigma_+ \hat{u}_+ \hat{v}_+^T + \sigma_- \hat{u}_- \hat{v}_-^T \qquad C_- = \eta_+ \hat{u}_+ \hat{v}_-^T + \eta_- \hat{u}_- \hat{v}_+^T \qquad (5)$$

Lemma 3. Given $C \in R^{m,n}$ have unit rank, if C contains both positive and negative elements, then rank(C_+)=rank(C_-)=2. If $C \geq 0$ (resp. $C \leq 0$), then rank(C_+) =1 (resp. rank(C_-)=1).

Lemma 4. $C_+ = \operatorname{argmin}_{G \in R_+^{m,n}} ||C - G||_F$ if $C \in R^{m,n}$.

Thus, the best nonnegative approximation of the unit rank matrix $C=uv^T$ (in terms of the Frobenius norm) would be C_+. Assume that A is a symmetric nonnegative matrix, in the above Lanczos process, we always take a positive vector b as the starting vector and set $\beta_i > 0$. Since A is symmetric nonnegative, then α_1 is nonnegative. But this cannot guarantee that J_k of (3) is symmetric and

nonnegative; thus we rewrite it as Euq.(6), where $C_i = u_i u_i^T$ and $E_i = u_{i+1} u_i^T$, both of them are unit rank matrices. In particular, every unit rank matrix C_i and E_i is approximated by corresponding nonnegative section C_{i+} and E_{i+}.

$$J_k = U_k T_k U_k^T = [u_1, u_2, \cdots u_k] \begin{pmatrix} \alpha_1 & \beta_2 & & \\ \beta_2 & \alpha_2 & \ddots & \\ \ddots & \ddots & & \beta_k \\ & & \beta_k & \alpha_k \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix}$$

$$= [u_1, u_2, \cdots u_k] \left[\begin{pmatrix} \alpha_1 & & \\ & \alpha_2 & \\ & \ddots & \\ & & \alpha_k \end{pmatrix} + \begin{pmatrix} 0 & & \\ \beta_2 & 0 & \\ & \ddots & \ddots \\ & & \beta_k \; 0 \end{pmatrix} + \begin{pmatrix} 0 & \beta_2 & & \\ & 0 & \ddots & \\ & & \ddots & \beta_k \\ & & & 0 \end{pmatrix} \right] \begin{pmatrix} u_1 \\ u_2 \\ \vdots \\ u_k \end{pmatrix}$$

$$= \sum_{i=1}^{k} \alpha_i C_i + \sum_{i=1}^{k-1} \beta_{i+1} (E_i + E_i^T) \quad (6)$$

From (6), we can bound the error by computing the Frobenius norm of the residual $R = A - WSW^T$. Let $C_i = C_{i+} - C_{i-}$ and $E_i = E_{i+} - E_{i-}$, we definite $\{\sigma_j(C_{i+}), m_j(C_{i+}), m_j(C_{i+})\}$ and $\{\mu_j(E_{i+}), x_j(E_{i+})$ are singular triplets of C_{i+} and E_{i+}, j = 1, 2. Since $\alpha_i = u_i^T A u_i$, it is said that the diagonal elements of T maybe negative. In this paper, we set S =|T| and D = diag(0, θ_2 , ⋯; θ_k) (if α_i > 0, then θ_i = 0; else $\theta_i = 2\alpha_i$), then T =|T|+D = S+D. We set vector b is a positive vector, thus, $\alpha_1 = u_1^T A u_1$ is positive and J_k can be rewritten as:

$$J_k = \sum_{i=1}^{k} (|\alpha_i| + \theta_i) u_i u_i^T + \sum_{i=1}^{k-1} \beta_{i+1} (u_{i+1} u_i^T + u_i u_{i+1}^T)$$

$$= \sum_{i=1}^{k} (|\alpha_i| + \theta_i) C_i + \sum_{i=1}^{k-1} \beta_{i+1} (E_i + E_i^T)$$

$$= \alpha_1 C_1 + \sum_{i=2}^{k} |\alpha_i| C_{i+} - \sum_{i=2}^{k} |\alpha_i| C_{i-} + \sum_{i=1}^{k-1} \beta_{i+1} \left(E_{i+} + E_{i+}^T\right)$$

$$- \sum_{i=1}^{k-1} \beta_{i+1} \left(E_{i-} + E_{i-}^T\right) + \sum_{i=2}^{k} \theta_i C_i$$

$$= \alpha_1 C_1 + \sum_{i=2}^{k} |\alpha_i| \sigma_1(C_{i+}) m_1(C_{i+}) m_1(C_{i+})^T$$

$$+ \sum_{i=1}^{k-1} \beta_{i+1} \mu_1(E_{i+}) (x_1(E_{i+}) y_1(E_{i+})^T + y_1(E_{i+}) x_1(E_{i+})^T) + G \quad (7)$$

Where G is defined as (8) and we can choose (W, S) as(9):

$$G = \sum_{i=2}^{k} |\alpha_i| \sigma_2(C_{i+}) m_2(C_{i+}) m_2(C_{i+})^T + \sum_{i=1}^{k-1} \beta_{i+1} \mu_2(E_{i+}) (x_2(E_{i+}) y_2(E_{i+})^T$$

$$+ y_2(E_{i+}) x_2(E_{i+})^T) - \sum_{i=2}^{k} |\alpha_i| C_{i-} - \sum_{i=1}^{k-1} \beta_{i+1} \left(E_{i-} + E_{i-}^T\right) + \sum_{i=2}^{k} \theta_i C_i \quad (8)$$

$$WSW^T = \alpha_1 C_1 + \sum_{i=2}^{k} |\alpha_i| \sigma_1(C_{i+}) m_1(C_{i+}) m_1(C_{i+})^T$$

$$+ \sum_{i=1}^{k-1} \beta_{i+1} \mu_1(E_{i+}) (x_1(E_{i+}) y_1(E_{i+})^T + y_1(E_{i+}) x_1(E_{i+})^T) = J_k - G \quad (9)$$

2.3. *SNMF algorithms*

Lee and Seung suggested two useful algorithms and have been employed in many areas [2]. For weighted SNMF $A = WSW^T$, the update rules are:

$$\begin{cases} W_{ik} \leftarrow W_{ik} \cdot (AWS)_{ik} / ((WSW^TWS)_{ik} + \varepsilon) \\ S_{ik} \leftarrow S_{ik} \cdot (W^TAW)_{ik} / ((W^TWSW^TW)_{ik} + \varepsilon) \end{cases} \quad (10)$$

In here, ε is a small constant.

3. Numerical Experiment

Choosing different positive starting vector b in Lanczos process, our algorithms may contain a little randomization and produce different (W, S) or W, thus, we always take $b = (1, 1, \ldots, 1)^T$. All computation is done using Matlab version 7 on an computer with Intel (R) CPU @1.86G HZ and1.5 EMS memory.

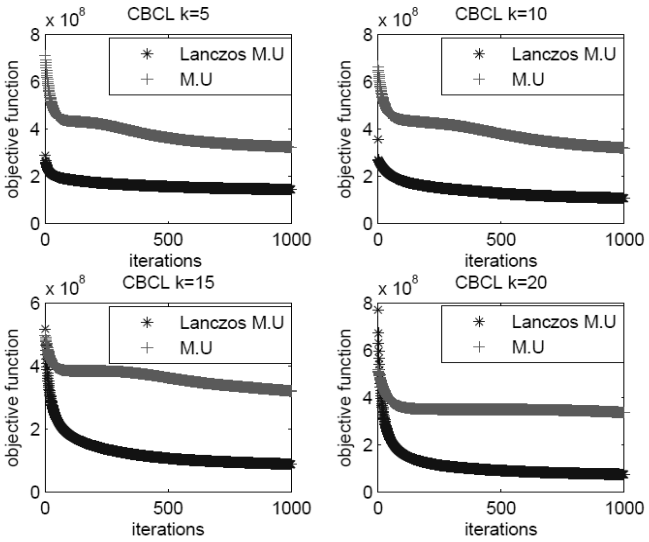

Fig. 1 Comparison of objective function for CBCL images between Lanczos M.U and M.U

We call the SNMF algorithm mentioned in section 2.3 M.U and the one combined with our initialization method Lanczos M.U. In experiment, the initial values obtained from their initial stages and setting the number of iterations to 1000. In M.U algorithm, we set W = rand(500, k), S = rand(k) as initialization pair (W, S). We take CBCL images as an example (http://cbcl.mit.edu/cbcl/soft-waredatasets/FaceData2.html); the size of matrix C is 361×2429. We set A =

CC^T and A is a 361 × 361 matrix, as well as, we use the Lanczos M.U and M.U on it and compare the objective function between them. In this experiment, we take parameter k = 5, 10, 15, 20.

After our initialization, we can see the objective function converges much faster and gets much smaller objective values than the one that without initialization process. In the figures, the curve with mark '+' is the result of M.U and the darker one with mark '*' is the result of Lanczos M.U.

4. Conclusion

Based on the Lanczos tridiagonalization process and nonnegative approximation of rank-one matrices, we derive a novel initialization algorithm for SNMF. This algorithm can be combined with SNMF algorithm and may contain a little randomization because of choice of the starting vector b. From experiments, we can see the SNMF algorithm with our initialization can converges faster and get small objective values than the one which is without initialization.

References

1. D. D. Lee, H. S. Seung, Algorithms for nonnegative matrix factorization, Adv. Neural Inf. Process. Syst. 13(2001).
2. Yang Z, Oja E. Linear and nonlinear projective nonnegative matrix factorization.[J]. IEEE Transactions on Neural Networks, 2010, 21(5):734-749.
3. Kannan R, Ballard G, Park H. A high-performance parallel algorithm for nonnegative matrix factorization[C], ACM Sigplan Symposium on Principles and Practice of Parallel Programming. ACM, 2015.
4. M.Cooper, J.Foote, Summarizing video via similarity matrix decomposition, in: Proceedings of IEEE workshop on Multimedia Singal Processing 2002.
5. Chris Ding, X.f He. On the Equivacence of Nonnegative Matrix Factorization and Spectral clustering, http://citeseerx.ist.psu.edu/viewdoc /download
6. Boutsidis C, Gallopoulos E. SVD based initialization: A head start for nonnegative matrix factorization[J]. Pattern Recognition, 2008, 41(4):1350-1362.
7. G. H. Gloub and C. F. VanLoan, Matrix Computation, 3rd, The Johns Hopkins University Press, Baltimore, MD. 1996.
8. J. Cullum, R. A. Willoughby, and M. Lake, A Lanczos algorithm for computing the singular values and corrsponding singular vectors of a matrix, ACM Trans.Math. Software,7(1981), pp. 140-169.
9. C. C. Paige, Errors analysis of the Lanczos algorithm for tridiagonalizaing a symmetric matrix, J. Inst. Math. Appl. 18(1976), pp. 341-349.