Fast Nonnegative Matrix Factorization with Rank-one ADMM

Dongjin Song[*], **David A. Meyer**[†], **Martin Renqiang Min**[‡],
[*]Department of ECE, UCSD, La Jolla, CA, 92093-0409
dosong@ucsd.edu
[†]Department of Mathematics, UCSD, La Jolla, CA, 92093-0112
dmeyer@math.ucsd.edu
[‡]NEC Labs America, Princeton, NJ 08540
renqiang@nec-labs.com

Abstract

Nonnegative matrix factorization (NMF), which aims to approximate a data matrix with two nonnegative low rank matrix factors, is a popular dimensionality reduction and clustering technique. Due to the non-convex formulation and the nonnegativity constraints over the two low rank matrix factors (with rank $r > 0$), it is often difficult to solve NMF efficiently and accurately. Recently, the alternating direction method of multiplier (ADMM) was shown to be more accurate and efficient than classical approaches such as the multiplicative update rule (MUR) or alternating least square (ALS). Nevertheless, the computation of ADMM is proportional to the cube of the rank r because of the underlying matrix inverse problem and thus may be inefficient when r is relatively large. In this paper, we propose a rank-one ADMM to address this problem. In each step, we search for a rank-one solution of NMF based upon ADMM and utilize greedy search to obtain the low rank matrix factors. In this way, rank-one ADMM avoids the matrix inverse problem and the computation is only linearly proportional to r. Thorough empirical studies demonstrate that rank-one ADMM is more efficient and accurate than baseline approaches.

1 Introduction

In the past decade, nonnegative matrix factorization (NMF) [5][6] and its extensions have been widely applied for various applications, *e.g.*, face recognition [2][3], scene classification [10], social network analysis [11][12], bioinformatics [9], *etc*. Essentially, NMF aims to find two nonnegative low rank matrix factors $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{r\times d}$ so as to reconstruct a data matrix $X \in \mathbb{R}^{n\times d}$ of n samples and d features, *i.e.*,

$$\begin{aligned} \min_{U,V} \quad & \frac{1}{2}\|X - UV\|_F^2 \\ \text{s.t.} \quad & U \geq 0, V \geq 0 \end{aligned} \tag{1}$$

where r denotes rank, $\|\cdot\|_F$ is the Frobenius norm, and the inequalities are element-wise. Since NMF is non-convex and U and V are constrained to be nonnegative, it is difficult to solve NMF accurately and efficiently. To address this issue, Lee and Seung [6] developed multiplicative update rule (MUR) which is guaranteed to obtain a locally optimal solution. Since MUR converges very slowly, alternating least square (ALS) [8] and projected gradient method [7] were subsequently developed.

Recently, alternating direction method of multiplier (ADMM) [1] has shown its superiority over MUR and ALS with respect to both reconstruction accuracy and efficiency [15][14][13]. Nevertheless, since the computation of ADMM involves a matrix inverse problem, the complexity of ADMM is proportional to the cube of the rank r of two low rank matrix factors and may be inefficient when r is relatively large. To resolve this issue, we propose rank-one ADMM in this paper. The main idea

is to find a rank-one solution of NMF based upon ADMM and utilize greedy search to obtain the low rank matrix factors. In this way, rank-one ADMM avoids the matrix inverse problem and the computation is only linearly proportional to r.

2 Related Work

In this section, we briefly introduce three representative approaches for optimizating NMF, *i.e.*, multiplicative update rule (MUR) [6], alternating least square (ALS) [8], and alternating direction method of multiplier (ADMM) [1].

Notations: Given a matrix $X \in \mathbb{R}^{n\times d}$ of n rows and d columns, we use $\mathbf{x}_j \in \mathbb{R}^n$ to denote its j-th column, use $\mathbf{x}^i \in \mathbb{R}^{1\times d}$ to denote its i-th row, and use X_{ij} to denote the entry in i-th row and j-th column of X.

2.1 Multiplicative Update Rule

MUR proposed by Lee and Seung [6] is the most popular method for NMF and can be written as:

$$U_{ik} \leftarrow U_{ik}\frac{(XV^T)_{ik}}{(UVV^T)_{ik}} \tag{2}$$

and

$$V_{kj} \leftarrow V_{kj}\frac{(U^TX)_{kj}}{(U^TUV)_{kj}}. \tag{3}$$

By using MUR with nonnegative initializations of U and V, both U and V will remain nonnegative throughout iterations. Since MUR converges slowly in many real world applications, ALS [8] can be utilized as the substitution.

2.2 Alternating Least Square

ALS developed by Paatero and Taper [8] minimizes the least square cost function in Eq. 1 with respect to U and V, iteratively. The procedure is given as

$$U = \mathrm{P}_+\Big(XV^T(VV^T)^*\Big) \tag{4}$$

and

$$V = \mathrm{P}_+\Big((U^TU)^*U^TX\Big), \tag{5}$$

where * denotes pseudo-inverse and P_+ projects the matrix onto a nonnegative set, *i.e.*, $\mathrm{P}_+(U) = \max(U, 0)$. Although ALS has shown its efficiency for NMF, recent research indicates that alternating direction method of multiplier (ADMM) [1] is more effective and efficient than ALS [15][14].

2.3 Alternating Direction Method of Multipliers

The alternating direction method of multipliers (ADMM) [1] aims to solve convex optimization problems by splitting them into smaller pieces such that each part of them is easier to handle. It could also be used for non-convex problems such as NMF [15][14][13]. Specifically, ADMM introduces two auxiliary variables S and T and considers the following equivalent formulation:

$$\begin{aligned} \min_{U,V,S,T} \quad & \frac{1}{2}\|X - UV\|_F^2 \\ \text{s.t.} \quad & U - S = 0, \quad V - T = 0, \quad S \geq 0, \quad T \geq 0. \end{aligned} \tag{6}$$

The augmented Lagrangian function of Eq. 6 is given as:

$$L(U, V, S, T, \Lambda, \Pi) = \frac{1}{2}\|X - UV\|_F^2 + \langle \Lambda, U - S\rangle + \langle \Pi, V - T\rangle + \frac{\rho}{2}\|U - S\|_F^2 + \frac{\rho}{2}\|V - T\|_F^2 \tag{7}$$

where $\Lambda \in \mathbb{R}^{n\times r}$ and $\Pi \in \mathbb{R}^{r\times d}$ are Lagrange multipliers, $\langle\cdot\rangle$ is the matrix inner product, and $\rho > 0$ is the penalty parameter for the constraints. By minimizing L with respect to U, V, S, T, Λ, and Π, we can obtain the closed form solution for each step as shown in appendix (Algorithm 2).

Note that the main computations in classical ADMM are the inversion of an $r \times r$ matrix and XV^T (or U^TX). When r is relatively large (*e.g.*, 500 or 1000 in some real world applications), the matrix inverse problem will consume substantial time at each step and thus ADMM will converge slowly. To resolve this issue, rank-one ADMM is developed.

Algorithm 1: Rank-one ADMM for NMF

Input: $X \in \mathbb{R}^{n\times d}$, $\mathbf{u}_{i+1} \in \mathbb{R}^n$, $\mathbf{v}^{i+1} \in \mathbb{R}^{1\times d}$, $\mathbf{s}_{i+1} \in \mathbb{R}^n$, $\mathbf{t}^{i+1} \in \mathbb{R}^{1\times d}$, $\mathbf{p} \in \mathbb{R}^n$, $\mathbf{q} \in \mathbb{R}^{1\times d}$, ρ
Output: U and V

Set $\widehat{X} = X$;
Calculate $\mathbf{u}_0$, $\mathbf{v}^0$ based upon ADMM
For $i = 0$ to $r-1$,
  Set $k = 0$ (*index of iteration*)
  $\widehat{X} = \widehat{X} - \mathbf{u}_i\mathbf{v}^i$
  Repeat:
    $\mathbf{u}_{i+1}(k+1) = \Big[\widehat{X}\Big(\mathbf{v}^{i+1}(k)\Big)^T + \rho\mathbf{s}_{i+1}(k) - \mathbf{p}(k)\Big] / \Big[\mathbf{v}^{i+1}(k)\Big(\mathbf{v}^{i+1}(k)\Big)^T + \rho\Big]$
    $\mathbf{v}^{i+1}(k+1) = \Big[\mathbf{u}_{i+1}(k+1)\Big)^T\widehat{X} + \rho\mathbf{t}^{i+1}(k) - \mathbf{q}(k)\Big] / \Big[\Big(\mathbf{u}_{i+1}(k+1)\Big)^T\mathbf{u}_{i+1}(k+1) + \rho\Big]$
    $\mathbf{s}_{i+1}(k+1) = \mathrm{P}_+\Big(\mathbf{u}_{i+1}(k+1) + \mathbf{p}(k)/\rho\Big)$
    $\mathbf{t}^{i+1}(k+1) = \mathrm{P}_+\Big(\mathbf{v}^{i+1}(k+1) + \mathbf{q}(k)/\rho\Big)$
    $\mathbf{p}(k+1) = \mathbf{p}(k) + \rho\Big(\mathbf{u}_{i+1}(k+1) - \mathbf{s}_{i+1}(k+1)\Big)$
    $\mathbf{q}(k+1) = \mathbf{q}(k) + \rho\Big(\mathbf{v}^{i+1}(k+1) - \mathbf{t}^{i+1}(k+1)\Big)$
    $k = k+1$
  Until convergence.
  $U = [U, \mathbf{u}_{i+1}(k+1)]$, $V = [V; \mathbf{v}^{i+1}(k+1)]$, $S = [S, \mathbf{s}_{i+1}(k+1)]$, $T = [T; \mathbf{t}^{i+1}(k+1)]$.
End for.

3 Rank-one ADMM for NMF

In rank-one ADMM, we successively search for rank-one solution to NMF based upon ADMM and utilize greedy search to obtain the low rank matrix factors $U \in \mathbb{R}^{n\times r}$ and $V \in \mathbb{R}^{r\times d}$.

For initialization ($i = 0$), we can adopt classical ADMM to obtain a rank-one solution of NMF. For the $i+1$ step, since $U \in \mathbb{R}^{n\times i}$ and $V \in \mathbb{R}^{i\times d}$ are already known and can be written as concatenations of column vectors ($U = [\mathbf{u}_1, ..., \mathbf{u}_i]$) and row vectors ($V = [\mathbf{v}^1; ...; \mathbf{v}^i]$) respectively, rank-one ADMM can be formulated as:

$$\begin{aligned} \min_{\mathbf{u}_{i+1},\mathbf{v}^{i+1},\mathbf{s}_{i+1},\mathbf{t}^{i+1}} \quad & \frac{1}{2}\Big\|X - [\mathbf{u}_1, ..., \mathbf{u}_i, \mathbf{u}_{i+1}][\mathbf{v}^1; ...; \mathbf{v}^r; \mathbf{v}^{i+1}]\Big\|_F^2 \\ \text{s.t.} \quad & \mathbf{u}_{i+1} - \mathbf{s}_{i+1} = 0, \mathbf{v}^{i+1} - \mathbf{t}^{i+1} = 0, \mathbf{s}_{i+1} \geq 0, \mathbf{t}^{i+1} \geq 0. \end{aligned} \tag{8}$$

Since we have known $U = [\mathbf{u}_1, ..., \mathbf{u}_i]$ and $V = [\mathbf{v}^1; ...; \mathbf{v}^i]$, Eq. 8 can be simplified as:

$$\begin{aligned} \min_{\mathbf{u}_{i+1},\mathbf{v}^{i+1},\mathbf{s}_{i+1},\mathbf{t}^{i+1}} \quad & \frac{1}{2}\Big\|\widehat{X} - \mathbf{u}_{i+1}\mathbf{v}^{i+1}\Big\|_F^2 \\ \text{s.t.} \quad & \mathbf{u}_{i+1} - \mathbf{s}_{i+1} = 0, \mathbf{v}^{i+1} - \mathbf{t}^{i+1} = 0, \mathbf{s}_{i+1} \geq 0, \mathbf{t}^{i+1} \geq 0, \end{aligned} \tag{9}$$

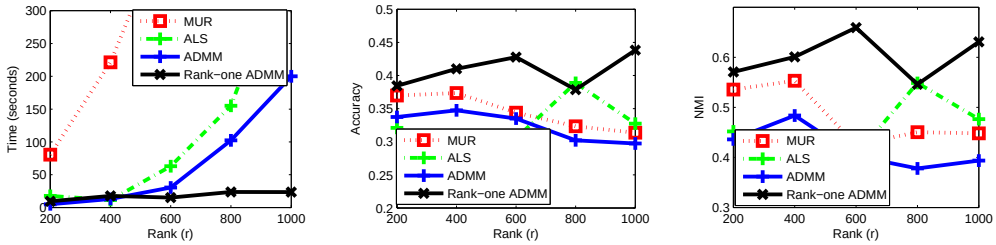
where $\widehat{X} = X - UV$. The augmented Lagrangian of Eq. 10 can be written as:

$$\begin{aligned} L(\mathbf{u}_{i+1}, \mathbf{v}^{i+1}, \mathbf{s}_{i+1}, \mathbf{t}^{i+1}, \mathbf{p}, \mathbf{q}) = & \frac{1}{2}\Big\|\widehat{X} - \mathbf{u}_{i+1}\mathbf{v}^{i+1}\Big\|_F^2 + \langle \mathbf{p}, \mathbf{u}_{i+1} - \mathbf{s}_{i+1}\rangle + \langle \mathbf{q}, \mathbf{v}^{i+1} - \mathbf{t}^{i+1}\rangle + \\ & \frac{\rho}{2}\|\mathbf{u}_{i+1} - \mathbf{s}_{i+1}\|_2^2 + \frac{\rho}{2}\|\mathbf{v}^{i+1} - \mathbf{t}^{i+1}\|_2^2. \end{aligned} \tag{10}$$

where $\mathbf{p} \in \mathbb{R}^n$ and $\mathbf{q} \in \mathbb{R}^{1\times d}$ are Lagrangian multipliers, and $\rho > 0$ is the penalty parameter for the constraints. By minimizing L with respect to $\mathbf{u}_{i+1}$, $\mathbf{v}^{i+1}$, $\mathbf{s}_{i+1}$, $\mathbf{t}^{i+1}$, $\mathbf{p}$, and $\mathbf{q}$, we can obtain the closed form solution of each step as shown in Algorithm 1.

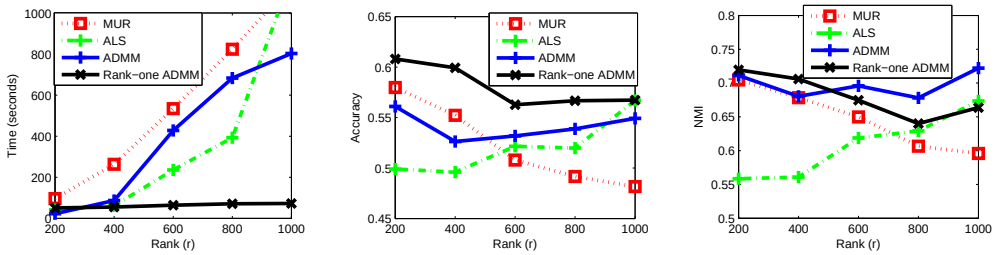
In Algorithm 1, $\mathbf{u}_{i+1}$ and $\mathbf{v}^{i+1}$ are randomly initialized. $\mathbf{s}_{i+1}$, $\mathbf{t}^{i+1}$, $\mathbf{p}$, and $\mathbf{q}$ are set to zero vectors of appropriate lengths. The stopping criterion for the inner loop of rank-one ADMM is met if the objective in Eq. 8 does not improve relative to a tolerance value.

In the inner loop, the main computation are the $\mathbf{u}_{i+1}(k+1)$ updating and $\mathbf{v}^{i+1}(k+1)$ updating with computational complexity of $O(n(d+1))$ and $O((n+1)d)$, respectively. Assuming that on



(a) Runing time (seconds) vs. rank (b) Accuracy vs. rank (c) NMI vs. rank

Figure 1: Experiment results on UMIST dataset.



(a) Runing time (seconds) vs. rank (b) Accuracy vs. rank (c) NMI vs. rank

Figure 2: Experiment results on Coil20 dataset.

average the inner loop takes k iterations to achieve convergence, the total computational complexity of rank-one ADMM will be $O(2ndrk + nrk + drk)$ which eliminates the matrix inverse problem in classical ADMM (as shown in Algorithm 2 of Appendix).

4 Experiments

In this section, we evaluate the effectiveness and efficiency of our proposed approach by comparing with three baseline approaches, *i.e.*, MUR, ALS, and ADMM. In particular, we implement all four approaches over two publicly available datasets, i.e., UMIST and Coil20. UMIST dataset contains 575 (size of 40×40) face images of 20 different persons. Coil20 dataset includes 1440 (size of 32×32) object images of 20 different classes.

The experiment results over UMIST and Coil20 are shown in Figure 1 and Figure 2, respectively. Figure 1(a) and 2(a) show the running time of each approach for convergence versus the rank r. we observe that when the rank r varies from 200 to 1000, rank-one ADMM generally converges faster than other approaches, especially when r is large. This is because the computation of ALS and ADMM is proportional to the cube of the rank due to the underlying matrix inverse problem; while the computation of rank-one ADMM is only linearly proportional to the cube of the rank. Moreover, we evaluate effectiveness of the low rank representations of four approaches based upon clustering *accuracy* and *normalized mutual information* (NMI) in Figure 1(b), Figure 1(c), Figure 2(b), and Figure 2(c). We observe that rank-one ADMM generally achieves better accuracy and NMI than the other approaches. This may because rank-one ADMM tends to yield more sparse low rank representations than other approaches.

5 Conclusion

In this paper, we proposed a rank-one alternating direction method of multiplier (ADMM) for non-negative matrix factorization (NMF). In each step, rank-one ADMM seeks for a rank-one solution of NMF based upon ADMM and utilizes greedy search to obtain low rank matrix factors. In this way, rank-one ADMM avoids the underlying matrix inverse problem and the computation is only linearly proportional to the rank r. We conducted empirical studies based upon two publicly available datasets, i.e., UMIST and Coil20. Our results demonstrated that rank-one ADMM is more efficient and effective than MUR, ALS, and traditional ADMM.

References

[1] Boyd, S., Parikh, S., Chu, E., Peleato, B., and Eckstein, J. (2011) Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends ®in Machine Learning*, 3(1): 1-122.

[2] Guan, N., Tao, D., Luo, Z., Yuan, B. (2011) Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent. *IEEE Transactions on Image Processing* , vol. 20, no. 7, pp. 2030-2048.

[3] Guan, N., Tao, D., Luo, and Shawe-Taylor, J. (2012) MahNMF: Manhattan non-negative matrix factorization. *arXiv preprint arXiv:1207.3438*

[4] Kim, J. and Park, H. (2011) Fast nonnegative matrix factorization: An active-set-like method and comparisons. *SIAM Journal on Scientific Computing* 33(6): 3261-3281, 2011.

[5] Lee, D. and Seung, H. (1999) Learning the parts of objects by non-negative matrix factorization. *Nature* 401:788-799.

[6] Lee, D. and Seung, H. (2001) Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems 13*, pp. 556-562. Cambridge, MA: MIT Press.

[7] Lin, C.-J. (2007) Projected gradient methods for nonnegative matrix factorization. *Neural Computation* 19(10):2756-2779.

[8] Paatero, P. and Tapper, U. (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics 5* 111-126.

[9] Min, M. R., Ning, X., Cheng, C., and Gerstein, M. (2014) Interpretable sparse high-order Boltzmann machines. The 17th International Conference on Artificial Intelligence and Statistics.

[10] Song, D. and Tao, D. (2010) Biologically inspired feature manifold for scene classification. *IEEE Transactions on Image Processing* , vol. 19, no. 1, pp. 174-184.

[11] Song, D. and David, A. M. (2014) A model of consistent node types in signed directed social networks. *IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 72-80.

[12] Song, D. and David, A. M. (2014) Recommending positive links in signed social networks by optimizing a generalized AUC. *The Twenty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*

[13] Sun, L. and Févotte, C., (2014) Alternating direction method of multipliers for non-negative matrix factorization with the beta-divergence. *IEEE International Conference on Acoustics, Speech, and Signal Processing*.

[14] Xu, Y., Yin, W., Wen, Z., and Zhang, Y. (2012) An alternating direction algorithm for matrix completion with nonnegative factors. *Frontiers of Mathematics in China*, 7(2): 365-384.

[15] Zhang, Y. (2010) . An alternating direction algorithm for nonnegative matrix factorization. *Technical Report,* Rice University.

Appendix

Algorithm 2: ADMM for NMF

1: **Input:** $X \in \mathbb{R}^{n\times d}$, $U \in \mathbb{R}^{n\times r}$, $V \in \mathbb{R}^{r\times d}$, $S \in \mathbb{R}^{n\times r}$, $T \in \mathbb{R}^{r\times d}$, ρ
2: **Output:** U and V
3: Set $k = 0$ (*index of iteration*)
4: Repeat:
5: $U(k+1) = \Big(XV(k)^T + \rho S(k) - \Lambda(k)\Big)\Big(V(k)V(k)^T + \rho I\Big)^{-1}$
6: $V(k+1) = \Big(U(k+1)^T U(k+1) + \rho I\Big)^{-1}\Big(U(k+1)^T X + \rho T(k) - \Pi(k)\Big)$
7: $S(k+1) = \mathrm{P}_+\Big(U(k+1) + \Lambda(k)/\rho\Big)$
8: $T(k+1) = \mathrm{P}_+\Big(V(k+1) + \Pi(k)/\rho\Big)$
9: $\Lambda(k+1) = \Lambda(k) + \rho\Big(U(k+1) - S(k+1)\Big)$
10: $\Pi(k+1) = \Pi(k) + \rho\Big(V(k+1) - T(k+1)\Big)$
11: $k = k + 1$.
12: Until convergence.

In Algorithm 2, U and V are randomly initialized. S , T, Λ, and Π are set to zero matrices of appropriate sizes. The stopping criterion for ADMM is met if the objective in Eq. 6 does not improve relative to a tolerance value. In each step, the main computation are $U(k+1)$ updating and $V(k+1)$ updating, which are $O(ndr+dr+r^3)$ and $O(ndr + nr + r^3)$, respectively. Assuming ADMM takes k iteration to achieve convergence, the total computational complexity for ADMM will be $O(2ndrk + nrk + drk + kr^3)$.