Lab 3: Nonnegative Matrix Factorization

Ting Lin, 1700010644

May 22, 2020

Contents

In this lab, we surveyed several methods on non-negative matrix factorization (NMF). Most of introduced algorithms are concerning NMF under the Euclidean(Frobenius) metric, however, some of them can be naturally extended to KL divergence. We introduce methods based on Multiplicative Update, Alternative Least Square, Alternative Non-negative Least Square, Alternative Direction of Multiplier Methods. Moreover, a new algorithm based on Nonlinear Least Square while subproblem is solved by ADMM is proposed. In the sections, we will analyze the convergence and test their performance, and finally test them in the popular ORL and Yale database.

1 Problem Setting

Nonnegative matrix factorization(NMF) considers the following optimization problem,

$$\begin{aligned} &\min \|V - WH\|_F^2 \\ \text{s.t. } &0 < W \in \mathbb{R}^{m \times r}, \\ &0 < H \in \mathbb{R}^{r \times n} \end{aligned} \tag{1}$$

or based on KL divergence alternatively,

$$\begin{aligned} &\min D(V||WH) \\ \text{s.t. } &0 < W \in \mathbb{R}^{m \times r}, \\ &0 < H \in \mathbb{R}^{r \times n} \end{aligned} \tag{2}$$

Here

$$D(A||B) = \sum_{i,j} A_{ij} \log \frac{A_{ij}}{B_{ij}} - A_{ij} + B_{ij},$$

which might behaves singular when the entry of A or B is near zero.

We always assume that V is nonnegative otherwise we can consider $\mathcal{P}(V)$. Here and throughout this paper we denote $\mathcal{P}(V)$ by the projection: $[\mathcal{P}(V)]_{ij} = \max(V_{ij}, 0)$.

We recall some basic properties according to [1]:

1. NMF does not in general have a unique solution (up to scaling and permutation).
2. NMF is not identifiable, in the sense that the task is challenging even in the clear artificial low rank data.
3. Clearly, it is smooth but non-convex.
4. The gradient of W is $\nabla W = W(HH') - VH'$, and $\nabla H = W'WH - W'V$

2 Methods based on MU

In this section we introduce MU method and its variants, like [2, 4, 3, 5].

2.1 MU and its convergence analysis

The basic idea of MU is simple, we choose a suitable stepsize and do gradient descent. Concretely, consider the gradient descent:

$$H_{ij} = H_{ij} + \eta_{ij}[\nabla H]_{ij}$$

Set $\eta_{ij} = H_{ij}/[W'WH]_{ij}$ we obtain the update rule of H,

$$H_{ij} = H_{ij} \frac{[W'V]_{ij}}{[W'WH]_{ij}}$$

Similar for W's update,

$$W_{ij} = W_{ij}\frac{[V'H]_{ij}}{[WHH']_{ij}}$$

Here we introduce a more compact notation, let $\otimes$ and $\oslash$ be the element-wise multiplication and division operator, we can rewrite it as

$$H = H \otimes (W'V) \oslash (W'WH), \qquad W = W \otimes (V'H) \oslash (WHH')$$

The implementation please see **nmf_mu.m**.

The implementation is a little complex in the KL version:

$$H_{ij} = H_{ij}\frac{\sum_k W_{ki}V_{kj}/[WH]_{kj}}{\sum_k W_{ki}}$$

$$H_{ij} = H_{ij}\frac{\sum_k H_{jk}V_{ik}/[WH]_{ik}}{\sum_k H_{jk}}$$

also, see **nmf_mu.m**

The update factor is chosen such that the residual $\|V - WH\|$ ($D(V||WH)$ resp) is nonincreasing. The advantage of MU method is that no explicit projection operation is needed.

Proposition 1. *$\|V - WH\|$ ($D(V||WH)$ resp) is nonincreasing after each update.*

We summarize it into the following algorithm [2].

Algorithm 1 MU

```
Require: $V, k$
Initialize $W$ and $H$
while not convergence do
    $W_{ij} = W_{ij}\frac{[V'H]_{ij}}{[WHH']_{ij}}$
    $H_{ij} = H_{ij}\frac{[W'V]_{ij}}{[W'WH]_{ij}}$
end while
return $W, H$.
```

In practice, we set stop criterion as max iteration, and initialization step we use $W = rand(m, r); H = rand(r, n)$.

2.2 Modified and Accelerated MU

However, the original MU method is not efficient, and faced various problem. We introduce some techniques to alleviate them and make mu more powerful.

Modified MU In [3], two main difficulties of MU is declared:

1. The denominator of the step size may be zero.
2. If the numerator if zero, and the gradient is negative, then H_{ij}^k will not be changed. Hence the convergence analysis fails, and it often occurs in numerical results.

Therefore, the authors proposed the modified step size to

$$\bar{H} \oslash (W'WH + \delta)$$

where

$$\bar{H} = H \otimes [\nabla H \geq 0] + \max(H, \sigma) \otimes [\nabla H < 0].$$

The detailed algorithm is shown below.

Algorithm 2 Modified MU

```
Require: $V, k$
Initialize $W$ and $H$
while not convergence do
    $\bar{H} = H \otimes [\nabla H \geq 0] + \max(H, \sigma) \otimes [\nabla H < 0].$
    $\bar{W} = W \otimes [\nabla W \geq 0] + \max(W, \sigma) \otimes [\nabla W < 0].$
    $H = H - \bar{H} \oslash (W'W\bar{H} + \delta) \otimes \nabla H$
    $W = W - \bar{W} \oslash (\bar{W}HH' + \delta) \otimes \nabla W$
    normalize $W$ and $H$, such that the column sum of $W$ is one.
end while
return $W, H$.
```

In [3], several properties about this method is discussed.

Proposition 2. *If the initial value is nonnegative (strictly positive), then nonnegativity and strit positivity will be preserved after each update.*

Proposition 3. *The loss $\|V - WH\|$ is nonincreasing after each modified MU update. The sequence W^k, H^k is pre-compact, and each of its accumulated point satisfies KKT condition.*

2.3 Accelerated MU

The acceleration techniques is introduced and analyzed in [4, 5]. We mainly focus on the work in [4].

For any given ρ_W and ρ_H, we introduce the following algorithm:

Algorithm 3 Accelerated MU

```
Require: $V, k, \delta$
Initialize $W$ and $H$
while not convergence do
    $\varepsilon = 1, \gamma = 1$
    while iter¡[$1 + rho_W \alpha$] and $\gamma > \delta\varepsilon$ do
        $W^- = W$
        $W = W \otimes (V'H) \oslash (WHH')$
        if iter==1 then
            $\varepsilon = \|W - W^-\|\|_F$
        end if
        $\gamma = \|W - W^-\|\|_F$
    end while
    $\varepsilon = 1, \gamma = 1$
    while iter¡[$1 + rho_H \alpha$] and $\gamma > \delta\varepsilon$ do
        $H^- = H$
        $H = H \otimes (W'V) \oslash (W'WH)$
        if iter==1 then
            $\varepsilon = \|H - H^-\|\|_F$
        end if
        $\gamma = \|H - H^-\|\|_F$
    end while
end while
return $W, H$.
```

Here $\rho_W = 1 + \frac{mn+nr}{mr+m}$, $\rho_H = 1 + \frac{mn+mr}{nr+n}$. δ is chosen to control the stop criterion and is chosen as 0.1 in our code (see **mnf_muacc.m**). Notice that the method is just replace an alternative update to a new update strategy with condition, hence all the convergence analysis makes sense in the accelerated version.

2.4 Experiments

3 Methods based on ALS

The idea of Alternative (Nonnegative) Least Square is instead of solving the original problem, we optimize two linear problem alternatively.

$$\min_{W>0} \|V - WH^*\| \tag{3}$$

$$\min_{H>0} \|V - W^*H\| \tag{4}$$

clearly, if we can solve the linear problem correctly, then the series W^k, H^k must be non-increasing. Two ways was attempted in the literature. We will discuss the methods based on solving subproblem with or without constraints. In this section we introduce solve ALS plus a suitable projection step, and more general algorithms in constraint linear programming is explored in

3.1 Naive ALS

In Naive ALS, we only need to compute the least square solution and project it into positive cone.

Algorithm 4 MU

```
Require: $V, k$
Initialize $W$ and $H$
while not convergence do
    $H = (W'W)^{-1}(W'V)$
    $H = \mathcal{P}(H)$
    $W = VH'(HH')^{-1}$
    $W = \mathcal{P}(W)$
end while
return $W, H$.
```

3.2 Projected BB method based ALS

We solve the subproblem by a projected BB, which is first introduced in [7]. Which is slightly different from global BB method, since we have to project our result into positive cone after each line search.

Algorithm 5 PBBNLS

```
Require: $A, B, X, \nabla X, \rho$
Set $\gamma, M, \lambda_{max}, \lambda_{min}$
Set $Q(X) - (A'B, X) + 0.5(X, A'AX)$
$\lambda = 1/\|\nabla X\|_\infty$
for i = 1:iter do
    $\alpha = 1$
    $X^+ = X - \lambda \nabla X$
    $X^+ = \mathcal{P}(X^+)$
    $Q^+ = Q(X^+)$
    while $\lambda > \lambda_{min}$ and $Q^+ < \max_{i-M<k<i} Q(X_i) + \gamma\alpha(\nabla X, D)$ do
        $\alpha = \alpha/4$
        $X^+ = X - \alpha\lambda\nabla X$
        $X^+ = \mathcal{P}(X^+)$
        $Q^+ = Q(X^+)$
    end while
    $s = X^+ - X$
    $\nabla X^+ = A'AX - A'B$
    $y = \nabla X^+ - \nabla X$
    if $(s, y) < \varepsilon$ then
        $\lambda = \lambda_{\max}$
    else
        $\lambda = \min(\lambda_{\max}, \max(\lambda_{\min}, (s, s)/(s, y)))$
    end if
    $\nabla X = \nabla X^+, X_i = X = X^+$
end for
return $X, \nabla X$
```

The whole algorithm uses PBBNLS to update each steps.

Algorithm 6 APBB

```
Require: $V, k$
Initialize $W$ and $H$
while not convergence do
    $[W, \nabla W] = PBBNLS(H', V', W', \nabla W')$
    $W = W'$
    $\nabla W = \nabla W'$
    $[H, \nabla H] = PBBNLS(W, V, H, \nabla H)$
end while
return $W, H$.
```

3.3 Projected Gradient Descent based ALS

[6] proposed a projected gradient descent method to solve the subproblem.

3.4 Hierarchical ALS and its acceleration

Hierarchical ALS [8] solves subproblem by LS with rank one modification. More precisely, we solve the linear problem column by column in W, and row by row in H in one epoch's update. Here is the algorithm, written in MATLAB format in order to make the idea of row/column by row/column more clear.

```
Algorithm 7 HALS
Require: $V, k$
Initialize $W$ and $H$
while not convergence do
    $VtW = V'W, WtW = W'W$
    for $k = 1 : r$ do
        $tmp = VtW(:,k)' - (WtW(:,k) * H) + WtW(k,k) * H(k,:)$
        $H(k,:) = \mathcal{P}(tmp/WtW(k,k))$
    end for
    $VHt = VH', HHt = HH'$
    for $k = 1 : r$ do
        $tmp = (VHt(:,k) - (W * HHt(:,k)) + (W(:,k) * HHt(k,k)))$
        $W(:,k) = \mathcal{P}(tmp/HHt(k,k))$
    end for
end while
return $W, H$.
```

Also, an accelerated version is provided in [4] and implemented in this lab, see **nmf_halsacc.m**

3.5 Numerical Results

4 Methods based on ANLS

4.1 Block Pivoting

4.2 Active Set

4.3 Grouped Active Set

5 Methods based on ADMM

In this section we introduce Alternative Direction of Multiplier Method, which is powerful in handling constraint problem and non-smooth problem.

5.1 a short introduction to ADMM framework

Suppose we aim to solve a function

$$min_X f(X)$$

where X is subjecting to some constraints. We introduce an auxiliary variable Z, and consider the following augmented Lagrangian

$$L(X, Z, \alpha) = g(X, Z) + (\alpha, X - Z) + \frac{\rho}{2}\|X - Z\|^2.$$

Here $g(X, Z)$ is a splitting of $f(X)$, i.e. $g(X, X) = X$.

The ADMM provide the following framework: Usually two arg min problem does not have the closed form solution, however, if there is a splitting such that both subproblem is easy to solve, then the algorithm might be very effective, at least in convex optimization. In practice, especially in nonconvex problem, such algorithm is also widely used while some rigorous convergence analysis is lacked.

Algorithm 8 General ADMM

```
Require: $f, X$
Set $L$ be the augmented Lagrangian.
$Z = X$, $\alpha = 0$.
while not converge do
    $Z = \arg\min L(X, Z, \alpha)$
    $X = \arg\min L(X, Z, \alpha)$
    $\alpha = \alpha + \rho\mu(X - Z)$
end while
```

5.2 Naive ADMM

We first apply ADMM naively, the algorithm is introduced in [**?**] and we use the form shown in [9], yielding the following algorithm.

The augmented Lagrangian function is denoted as

$$L(W, H, S, T, \Lambda, \Pi) = \frac{1}{2}\|X - WH\|_F^2 + (\Lambda, W - S) + (\Pi, V - T) + \frac{\rho}{2}\|U - S\|_F^2 + \frac{\rho}{2}\|V - T\|_F^2$$

Algorithm 9 Naive ADMM

```
Require: $f, X$
Set $L$ be the augmented Lagrangian.
$Z = X$, $\alpha = 0$.
while not converge do
    $W = (VH' + \rho S - \Lambda)(HH' + \rho I)^{-1}$
    $H = (W'W + \rho I)^{-1}(W'V + \rho T - \Pi)$
    $S = \mathcal{P}(W + \Lambda/\rho)$
    $T = \mathcal{P}(H + \Pi/\rho)$
    $\Lambda = \Lambda + \rho(W - S)$
    $\Pi = \Pi + \rho(H - T)$
end while
```

5.3 Alternating Optimizing ADMM

AO-ADMM is use ADMM to solve the subproblem in ALS, introduced in [10]. We only introduce how to solve the subproblem by ADMM, in the following algorithm.

Algorithm 10 ADMM-LS-UPDATE

```
Require: $Y, W, H, r$
$G = W'W$
$Haux = H$, $\alpha = 0$
$\rho = tr(G)/k$
for $i = 1 : iter$ do
    $Haux = (G + \rho I)^{-1}(W'Y + \rho(H + \alpha))$
    $H = Haux - \alpha$
    $\alpha = \alpha + H - Haux$
end for
```

6 A new method: LMF-ADMM

In this section we propose a new method based on LMF method of Nonlinear Least Square, and use ADMM to solve the subproblem.

We first recall LMF method, regarding NMF into a nonlinear least square problem: Suppose we have x_k and $J_k = \nabla r(x_k)$. Here $r(x) : \mathbb{R}^m \to \mathbb{R}^n$ is the residual function and our goal is to minimize $r'r$

Then we solve the following question to obtain the next point:

$$\min \|J_k(x - x_k) - r_k\|_F^2 + \nu\|x - x_k\|_F^2$$

Without constraint, the minimizer has a closed form:

$$x_k + (J_k'J_k + \nu I)^{-1}(J_k'r_k)$$

That is equivalent to LMF method. Inspired by this, we proposed an ANLS approach of LMF method.

Algorithm 11 LMF

```
$R = V - WH$
for $i = 1 : iter$ do
    $[W^+, H^+] = admm_update(W, H)$
    $R^+ = V - W^+H^+$
    $dW = W^+ - W, dH = H^+ - H$
    $\Delta f = \|R\|_F^2 - \|R^+\|_F^2$
    if $\Delta f > 0$ then
        $W = W^+, H = H^+, R = R^+, \mu = \mu/2$
        $\nabla W = WHH' - VH', \nabla H = W'WH - W'V$
    else
        $\mu = 2\mu$
    end if
end for
```

7 Overall Numerical Experiment

References

[1] Gillis N. Nonnegative matrix factorization: Complexity, algorithms and applications[J]. Unpublished doctoral dissertation, Universit catholique de Louvain. Louvain-La-Neuve: CORE, 2011.

[2] Lee D D, Seung H S. Algorithms for non-negative matrix factorization[C]//Advances in neural information processing systems. 2001: 556-562.

[3] Lin C J. On the convergence of multiplicative update algorithms for nonnegative matrix factorization[J]. IEEE Transactions on Neural Networks, 2007, 18(6): 1589-1596.

[4] N. Gillis and F. Glineur,"Accelerated Multiplicative Updates and hierarchical ALS Algorithms for Nonnegative Matrix Factorization"

[5] Gonzalez E F, Zhang Y. Accelerating the Lee-Seung algorithm for nonnegative matrix factorization[R]. 2005.

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

[7] Han, Lixing Neumann, Michael Prasad, Upendra. (2010). Alternating projected Barzilai-Borwein methods for Nonnegative Matrix Factorization. Electronic transactions on numerical analysis ETNA. 36. 54-82.

[8] Cichocki A, Phan A H. Fast local algorithms for large scale nonnegative matrix and tensor factorizations[J]. IEICE transactions on fundamentals of electronics, communications and computer sciences, 2009, 92(3): 708-721.

[9] Song D, Meyer D A, Min M R. Fast nonnegative matrix factorization with rank-one admm[C]//NIPS 2014 Workshop on Optimization for Machine Learning (OPT2014). 2014.

[10] Huang K, Sidiropoulos N D, Liavas A P. A flexible and efficient algorithmic framework for constrained matrix and tensor factorization[J]. IEEE Transactions on Signal Processing, 2016, 64(19): 5052-5065.