Computational Linear Algebra

Nonnegative Matrix Factorization

Benoît Müller

October 2023

Question a)

In the following code, we implement an alternating optimization algorithm, using the multiplicative update (MU) rules for W, H described in Theorem 1 of [3]:
NMF.m :

```matlab
function [W,H, err, time] = NMF(V,W0,H0,tol,maxit)
tic;
W = W0;
H = H0;
err = zeros(maxit+1,1);
time = zeros(maxit+1,1);
i = 2;
err(i) = norm(V-W*H,'fro');
time(i) = toc;
while(abs(err(i)-err(i-1))>tol && time(i)<maxit)
    i = i+1;
    H = H.*(W'*V)./(W'*W*H);
    W = W.*(V*H')./(W*H*(H'));
    err(i) = norm(V-W*H,'fro');
    time(i) = toc;
end
err = err(2:i);
time = time(2:i);
end
```

Question b)

We apply our algorithm to the face data base example in [4] with the following code: **begining of question_b.m :**

```matlab
m = 361;
n1 = 472;
n2 = 2429;
V1 = zeros(m,n1);
V2 = zeros(m,n2);
r = 7^2;
% load the images:
for i = 0:n1-1
```

```
    image = imread(sprintf("face/cmu_%04d.pgm",i));
    V1(:,i+1) = image(:);
end
for i = 1:n2
    image = imread(sprintf("face/face%05d.pgm",i));
    V2(:,i) = image(:);
end
% compute the factorization:
W0 = rand(m,r);
H0 = rand(r,n1);
tol = 1e-4;
maxit = 60;
[W,H,err,time] = NMF(V1,W0,H0,tol,maxit);
errrelat = (err-min(err))/(err(1)-min(err));

% shape a matrix to show the images:
W_mat = zeros(sqrt(r)*sqrt(m),sqrt(r)*sqrt(m));
a = sqrt(r);
b = sqrt(r);
col = 1;
for i = 1:a
    for j = 1:b
        W_mat(((i-1)*sqrt(m)+1):(i*sqrt(m)), ((j-1)*sqrt(m)+1):(j*sqrt(m)))...
        = reshape(W(:,col),sqrt(m),sqrt(m));
        col = col+1;
    end
end
figure
imshow(W_mat)

% test set done separately:
W1 = rand(m,r);
H1 = rand(r,n2);
tol = 1e-4;
maxit = 60;
[Wb,Hb,errb,timeb] = NMF(V2,W1,H1,tol,maxit);
errrelatb = (errb-min(errb))/(errb(1)-min(errb));

% test set done separately:
W_matb = zeros(sqrt(r)*sqrt(m),sqrt(r)*sqrt(m));
a = sqrt(r);
b = sqrt(r);
col = 1;
for i = 1:a
    for j = 1:b
        W_matb(((i-1)*sqrt(m)+1):(i*sqrt(m)), ((j-1)*sqrt(m)+1):(j*sqrt(m))) ...
        = reshape(Wb(:,col),sqrt(m),sqrt(m));
        col = col+1;
    end
end
figure
imshow(W_matb)
%R = W*H;
%im = R(:,1);
```

We plot the resulting basis of faces in an image:

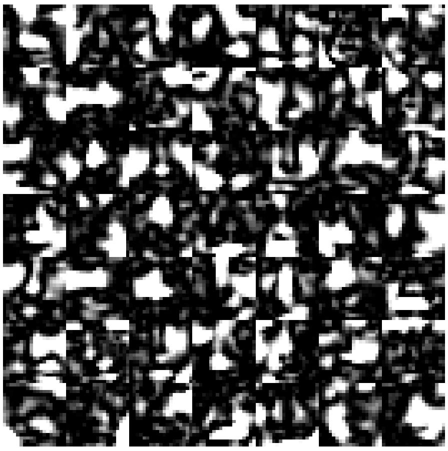

Figure 1: Basis a faces induced by the factorization done with NMF

Even if it is started at a random matrix, Figure [1] shows that the basis of the factorization is made of recognizable images: it contain different versions of mouths, noses, eyes, and some other facial parts. To check the efficiency of the factorization, we can compute $WH_{:,j}$ and compare it with the original image $V_{:,j}$

Question c)

We will use the notation $\|.\|$ for the Frobenius matrix norm:

$$\|A\| = \sqrt{\sum_{i,j} A_{ij}^2}.$$

The matrix operators $\odot$ and $\oslash$ denote respectively the element-wise product and division:

$$(A \odot B)_{ij} = A_{ij}Bij$$

$$(A \oslash B)_{ij} = \frac{A_{ij}}{Bij}$$

Theorem 1. *The Frobenious norm $\|V - WH\|$ is nonincreasing under the folwing updates:*

$$H \mapsto H \odot (W^\top V) \oslash (W^\top WH), W \mapsto W \odot (VH^\top) \oslash (WHH^\top)$$

and the value of the norm is invariant under these updates if and only if W and H are stationary points.

The general idea of the proof is the use of an auxiliary function, and therefore we have to introduce to definition of an auxiliary function, and then we give two lemmas about this type of functions that will be used to prove the theorem.

Definition 1. $G(h, h')$ is an auxiliary function for $F(h')$ if the conditions

$$G(h, h') \geq F(h)$$
$$G(h, h) = F(h)$$

are satisfied.

The following lemma is the key of the proof of theorem 1, and is by itself the reason why we use this useful concept of auxiliary function.

Lemma 1. *If G is an auxiliary function, the F is non-increasing under the update :*

$$h^{t+1} = \operatorname*{argmin}_h G(h, h^t)$$

Proof. From the definition of an auxiliary function we have $F(h^{t+1}) \leq G(h^{t+1}, h^t)$, but following from the definition of h^{t+1}, we immediately have $G(h^{t+1}, h^t) \leq G(h^t, h^t) = F(h^t)$, where the last equality also comes from the definition of an auxiliary function. □

We observe that $F(h^{t+1}) = F(h^t)$ if h^t is a local minimum of $G(h, h^t)$. If the derivatives of F exist and are continuous in a small neighborhood of h^t, we deduce that $\nabla F(h^t) = 0$. Therefore, if we iterate the update from the lemma 1, we obtain a sequence of estimates that has to converge to a local minimum $h_{min} = \operatorname{argmin}_h F(h)$ of the objective function.

Lemma 2. *With $K(h^t)$ the diagonal matrix $K_{ij}(h^t) = \delta_{ij}\frac{(W^\top Wh^t)_i}{h_i^t}$, we have that*

$$G(h, h^t) = F(h^t) + (h - h^t)\nabla F(h^t) + \frac{1}{2}(h - h^t)^\top K(h^t)(h - h^t)$$

is an auxiliary function for

$$F(h) = \frac{1}{2}\|v - Wh\|^2$$

Proof. We directly see that $G(h, h) = F(h) + 0 + \frac{1}{2}0 = F(h)$,
so it remains to show that $G(h, h^t) \geq F(h)$. The function G is defined to look like a quadratic Taylor expansion of F centered in h. We want then to compare then the quadratic term with the actual hessian. We first compute the gradient of F:

$$\begin{aligned}
\frac{F(h + \epsilon d) - F(h)}{\epsilon} &= \frac{1}{2\epsilon}\|v - W(h + \epsilon d)\|^2 - \frac{1}{2\epsilon}\|v - Wh\|^2 \\
&= \frac{1}{2\epsilon}(\|v - Wh\|^2 - 2(v - Wh)^\top W\epsilon d + \|\epsilon d\|^2) - \frac{1}{2\epsilon}\|v - Wh\|^2 \\
&= (Wh - v)^\top Wd + \epsilon\frac{1}{2}\|d\|^2 \\
&\xrightarrow[\epsilon \to 0]{} (v - Wh)^\top Wd \\
&= (W^\top(Wh - v))^\top d \\
&= (W^\top Wh - W^\top v)^\top d.
\end{aligned}$$

As a result we get $\nabla F(h) = W^\top(Wh - v) = W^\top Wh - W^\top v$, from which we see the form of the hessian $\nabla^2 F(h) = W^\top W$. We conclude that F admit the exact finite Taylor development

$$F(h) = F(h^t) + (h - h^t)W^\top \nabla F(h) + \frac{1}{2}(h - h^t)^\top W^\top W(h - h^t),$$

and it differ from the definition of G only for the quadratic term:

$$G(h, h^t) - F(h) = (h - h^t)^\top (K(h^t) - W^\top W)(h - h^t)$$

and it is nonnegative for any h, h^t if and only if $K(h^t) - W^\top W$ is semipositive definite. We find that

$$\begin{aligned}
h^\top K(h^t)h &= \sum_{ij} h_i^\top K_{ij}(h^t)h_j \\
&= \sum_{ij} h_i \delta_{ij} \frac{(W^\top Wh^t)_i}{h_i^t} h_j \\
&= \sum_i h_i \frac{(W^\top Wh^t)_i}{h_i^t} h_i \\
&= \sum_i h_i^2 \frac{\sum_j (W^\top W)_{ij} h_j^t}{h_i^t} \\
&= \sum_{ij} h_i^2 \frac{h_j^t}{h_i^t} (W^\top W)_{ij} \\
&= \sum_{ij} \left(\frac{h_i}{h_i^t}\right)^2 h_j^t h_i^t (W^\top W)_{ij} \\
&= \sum_{ij} \frac{1}{2}\left(\left(\frac{h_i}{h_i^t}\right)^2 + \left(\frac{h_j}{h_j^t}\right)^2\right) h_j^t h_i^t (W^\top W)_{ij} && \text{(mean with the transposed summing)} \\
&\geq \sum_{ij} \frac{h_i}{h_i^t} \frac{h_i}{h_i^t} h_j^t h_i^t (W^\top W)_{ij} && \text{(Young inequality)} \\
&= \sum_{ij} h_i (W^\top W)_{ij} h_j \\
&= h^\top W^\top Wh.
\end{aligned}$$

This proves indeed the semipositivity since $0 \leq h^\top K(h^t)h - h^\top W^\top Wh = h^\top(K(h^t) - W^\top W)h$, we obtain $G(h, h^t) \geq F(h)$, and G is an auxiliary function. □

Now we come to the proof of the theorem :

Proof. Recall the result of the lemma 1, i.e $h^{t+1} = \operatorname{argmin}_h G(h, h^t)$. We replace in this equation $G(h, h^t)$ by the result from lemma 2, which is :

$$G(h, h^t) = F(h^t) + (h - h^t)\nabla F(h^t) + \frac{1}{2}(h - h^t)^\top K(h^t)(h - h^t)$$

And now we get that $h^{t+1} = h^t - K(h^t)^{-1}\nabla F(h^t)$. Indeed, minimizing G with respect to h is a convex optimisation problem, which can be solved by computing the gradient of G, and solving $\nabla G(h, h^t) = 0$, to get h^{t+1}.

Since G is an auxiliary function, we get that F is non-increasing under this update rule, according to

lemma 1.
We now have :

$$\begin{aligned} h^{t+1} &= h^t - K(h^t)^{-1}\nabla F(h^t) \\ \implies h^{t+1} &= h^t - \frac{1}{2}K(h^t)^{-1}\|v - Wh^t\|^2 \\ \implies h_a^{t+1} &= h_a^t \frac{(W^Tv)_a}{(W^TWh^t)_a} \end{aligned}$$

Where the last equality is obtained by a direct computation and using the fact that $K(h^t)$ is a diagonal matrix and thus its inverse is defined by $K_{ab}(h^t)^{-1} = \delta_{ab}\frac{h_a^t}{(WW^Th^t)_a}$.
Now to end the proof, we reverse the roles of W and H in lemma 1 and 2, and similarly, F can be shown to be non-increasing under the update rules for W. □

Question d)

We implement in the following code the acceleration technique described in Algorithm 3 of [2] for the MU rule from question a). We use the parameters $\alpha = 2$ and $\epsilon = 0.1$.
ANMF.m :

```
function [W,H,err, time] = ANMF(V,W0,H0,tol,maxit,alpha,epsilon)
tic;
W = W0;
H = H0;
K = nnz(V);
n = size(H0);
n = n(2);
m = size(W0);
m = m(1);
r = size(W0);
r = r(2);
rhow = 1+ (K+n*r)/(m*r+m);
rhoh = 1+ (K+m*r)/(n*r+n);
err = zeros(maxit+1,1);
time = zeros(maxit+1,1);
maxw = floor(1+alpha*rhow);
maxh = floor(1+alpha*rhoh);
i = 2;
err(i) = norm(V-W*H,'fro');
time(i) = toc;
while(abs(err(i)-err(i-1))>tol && time(i)<maxit )
    i = i+1;


    A = V*H';
    B = H*H';
    W_iter = W;
    W_1 = W_iter.*(A./(W_iter*B));
    W_l = W_1;
    for j = 2:maxw
        W_old = W_l;
        W_l = W_l.*(A./(W_l*B));
        if norm(W_l-W_old,'fro') <= epsilon*norm(W_1-W_iter,'fro')
            break
        end
    end
    W = W_l;


    A = W'*V;
    B = W'*W;
    H_iter = H;
    H_1 = H_iter.*(A./(B*H_iter));
    H_l = H_1;
    for j = 2:maxh
        H_old = H_l;
        H_l = H_l.*(A./(B*H_l));
        if norm(H_l-H_old,'fro') <= epsilon*norm(H_1-H_iter,'fro')
            break
        end
    end
    H = H_l;
    err(i) = norm(V-W*H,'fro');
    time(i) = toc;

end
err = err(2:i);
time = time(2:i);

end
```

We now show the resulting new basis as well as how this accelerated MU algorithm compare in terms of time and approximation error to the standard MU algorithm. Here is the code:
Suite of question_b.m :

```
%imshow(im,[])

% compute the accelerated factorization:
alpha = 2;
epsilon = 0.1;
[Wa,Ha,erra,timea] = ANMF(V1,W0,H0,tol,maxit,alpha,epsilon);
errrelata = (erra-min(erra))/(erra(1)-min(erra));

% shape a matrix to show the images:
W_mata = zeros(sqrt(r)*sqrt(m),sqrt(r)*sqrt(m));
a = sqrt(r);
b = sqrt(r);
col = 1;
for i = 1:a
    for j = 1:b
        W_mata(((i-1)*sqrt(m)+1):(i*sqrt(m)), ((j-1)*sqrt(m)+1):(j*sqrt(m))) ...
        = reshape(Wa(:,col),sqrt(m),sqrt(m));
        col = col+1;
    end
end
figure
imshow(W_mata)

% test set done separately:
[Wba,Hba,errba,timeba] = ANMF(V2,W1,H1,tol,maxit,alpha,epsilon);
errrelatba = (errba-min(errba))/(errba(1)-min(errba));

% shape a matrix to show the images:
W_matba = zeros(sqrt(r)*sqrt(m),sqrt(r)*sqrt(m));
a = sqrt(r);
```

```
b = sqrt(r);
col = 1;
for i = 1:a
    for j = 1:b
        W_matba(((i-1)*sqrt(m)+1):(i*sqrt(m)), ((j-1)*sqrt(m)+1):(j*sqrt(m))) ...
        = reshape(Wba(:,col),sqrt(m),sqrt(m));
        col = col+1;
    end
end
figure
imshow(W_matba)

% plots of the errors:
figure
semilogy(time(1:500),errrelat(1:500))
hold on
semilogy(timea(1:500),errrelata(1:500))
hold off
title('Convergence of the relative error for the test set')
xlabel('time')
ylabel('error')
legend('MU','Accelerated MU')

figure
semilogy(timeb(1:500),errrelatb(1:500))
hold on
semilogy(timeba(1:500),errrelatba(1:500))
hold off
title('Convergence of the relative error for the train set')
xlabel('time')
ylabel('error')
legend('MU','Accelerated MU')

%R = W*H;
```

We obtain the Figure [2] which show the same properties as before, we get images representing faces, with the difference that the matrix are more sparse, indicating that the algorithm isolate more strongly the parts of the faces that are selected:

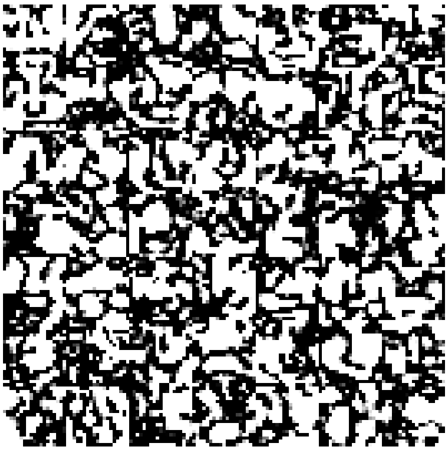

Figure 2: Basis a faces induced by the factorization done with ANMF

For the error, since the Frobenius error doesn't converge to zero, we plot the relative error: we translate by subtracting the minimum value of both curves and rescale by the first value (it is the same for both curves) so it begins at one. Here is the plot in Figure [3]:

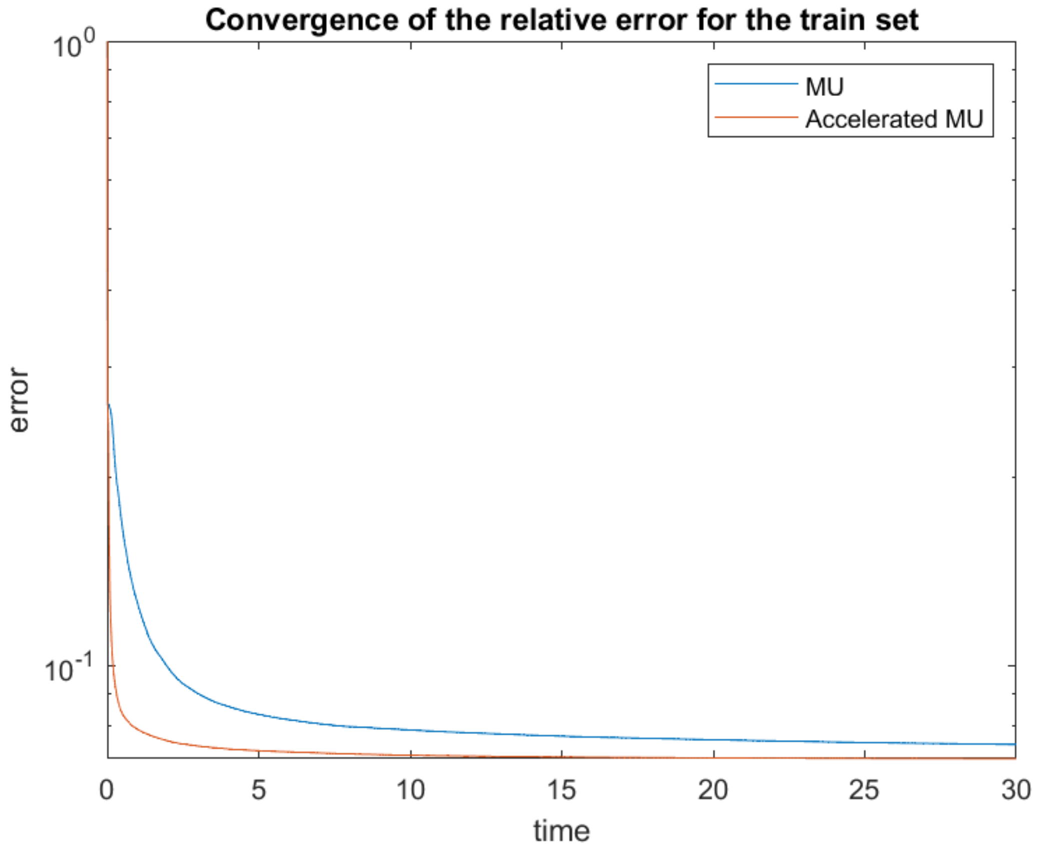

Figure 3: Evolution of the relative error during time

We see that the error is way better in term of time for the accelerated version. The value obtained by the simple method after 30 second is attained by the accelerated method after only two or three seconds.

Appendix

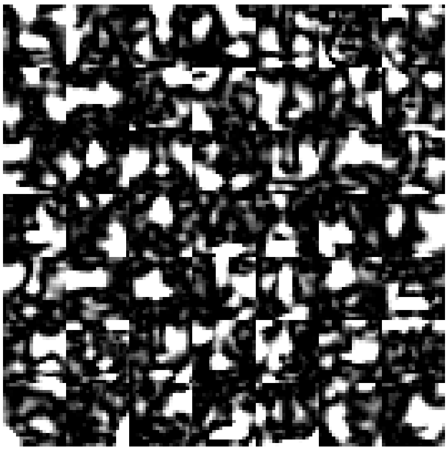

Figure 4: Basis of faces resulting from NMF for the test set

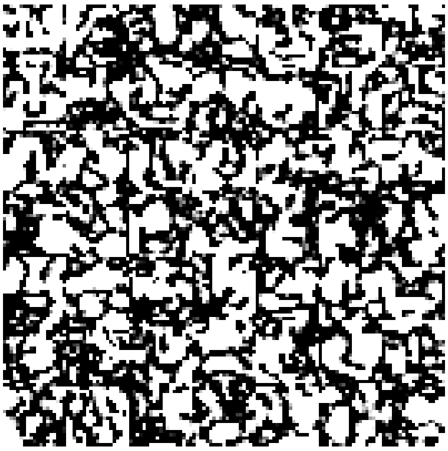

Figure 5: Basis of faces resulting from ANMF for the test set

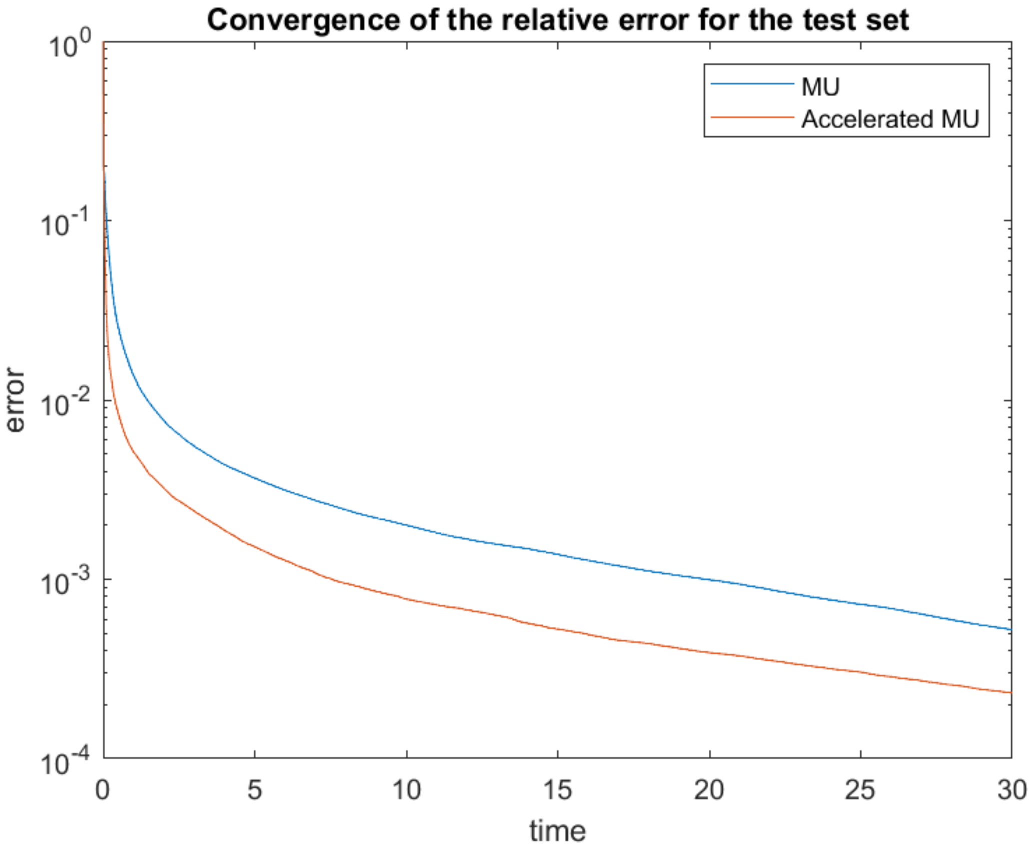

Figure 6: Evolution of the relative error during time for the test set

References

[1] MIT Center For Biological and Computation Learning. *CBCL Face Database*. URL: `http://www.ai.mit.edu/projects/cbcl.old/`.

[2] N. Gillis and F. Glineur. "Accelerated multiplicative updates and hierarchical ALS algo- rithms for nonnegative matrix factorization". In: *Neural Computation, 24(4)* (2012), pp. 1085–1105.

[3] Daniel D. Lee and H. Sebastian Seung. "Algorithms for Non-negative Matrix Factorization". In: *Advances in Neural Information Processing Systems (13)* (2003), pp. 556–562.

[4] Daniel D. Lee and H. Sebastian Seung. "Learning the parts of objects by non-negative matrix factorization". In: *Nature (401)* (1999), pp. 788–791.