PCA vs Non-negative Matrix Factorization

1. Result of PCA

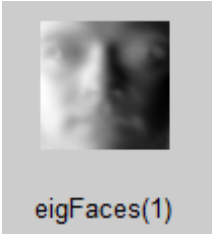

Figure 1:The fisrt image of eigFaces get by PCA

Figure 2:The eigFaces get by PCA

I display the factorization results of PCA as 7*7 grid with a anti-color processing .

From the figure, we can see that the eigfaces resemble distorted versions of wholefaces.

2. Result of NMF

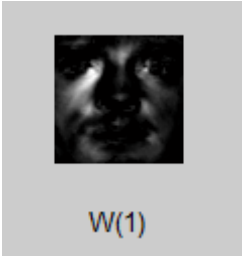

Figure 3:The fisrt image of base_images (W) get by NMF

Figure 4:The base_images(W) get by NMF

I display the factorization results of NMF as 7*7 grid with a anti-color processing.

From the figure, we can see that NMF leads to sparse representation, resulting in discovering a basis consisting localized features that correspond better with intuitive notions of the parts of faces.

原图

重构图 by NMF

Figure 5: Conmparison of the original image_1 and the restructed image_1 get by NMF

From the figure, we can see that the the restructed image get by NMF is alsmost the same as the original one.

3. The Comparison of PCA and NMF

PCA constrains the columns of W to be orthonormal and the rows of H to be orthogonal to each other. This relaxes the unary constraint of VQ, allowing a distributed representation in which each face is approximated by a linear combination of all the basis images, or eigenfaces. Although eigenfaces have a statistical interpretation as the directions of largest variance, many of them do not have an obvious visual interpretation. This is because PCA allows the entries of W and H to be of arbitrary sign. As the eigenfaces are used in linear combinations that generally involve complex cancellations between positive and negative numbers, many individual eigenfaces lack intuitive meaning.

NMF does not allow negative entries in the matrix factors W and H. Unlike the unary constraint of VQ, these non-negativity constraints permit the combination of multiple basis images to represent a face. But only additive combinations are allowed, because the non-zero elements of W and H are all positive. In contrast to PCA, no subtractions can occur. For these reasons, the non-negativity constraints are compatible with the intuitive notion of combining parts to form a whole, which is how NMF learns a parts-based representation.

4. Convergence of NMF

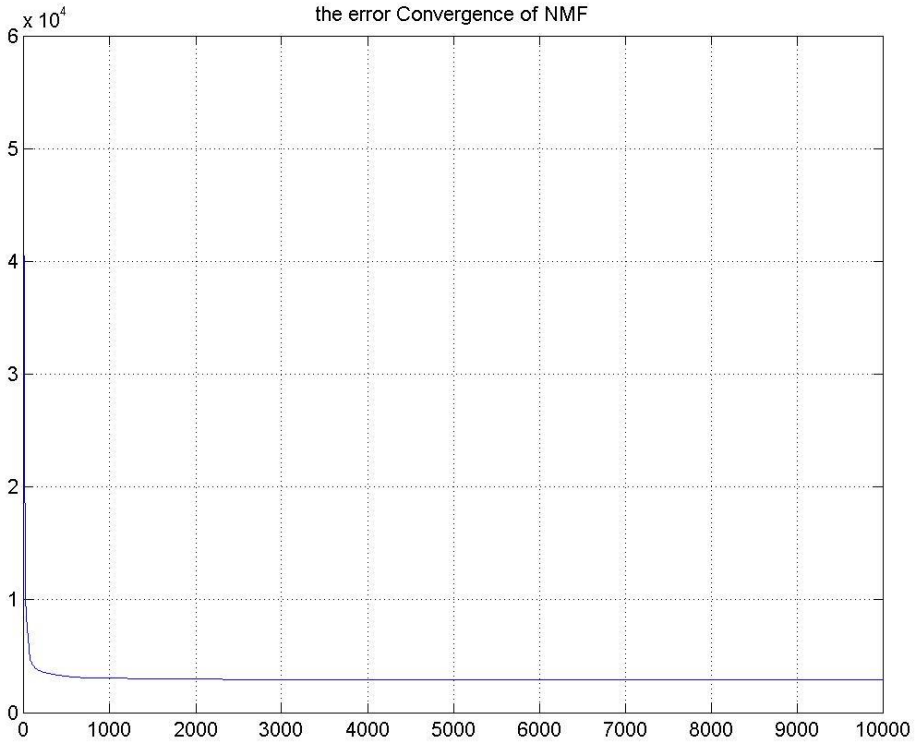

Figure 6: Error ($||V - W \times H||$) convergence of NMF

From the figure, we can see that：

- It decreases quickly at the first 500 iteration times,
- It doesn't converge to zero. So, the error between the original image and the restricted image always exists.

5. Problems & Solutions

I found that after 10000 times' iterations, the error of NMF that $||V - W \times H||$ was still large, which is about 10^3.

So, I write a function of NMF to test some easy real positive matrix, by testing that I found W will converge quickly, but it will converge to a settled error, which is always bigger than 1.

As a result, I get 2 tricks:

- The approximation quality of NMF is not very good.
- Since W converges quickly and always converges to an invariable matrix, while the $||V - W \times H||$ is still large, when it converges to the invariable matrix, we can choose $||W^{(N)} - W^{(N-1)}||$ to be the boundary instead of $||V - W \times H||$ to reduce the computation time.

By using $||W^{(N)} - W^{(N-1)}||$ to be the boundary with tol=0.01, we just need to do 2570 iterations, instead of 10000 and the result is :

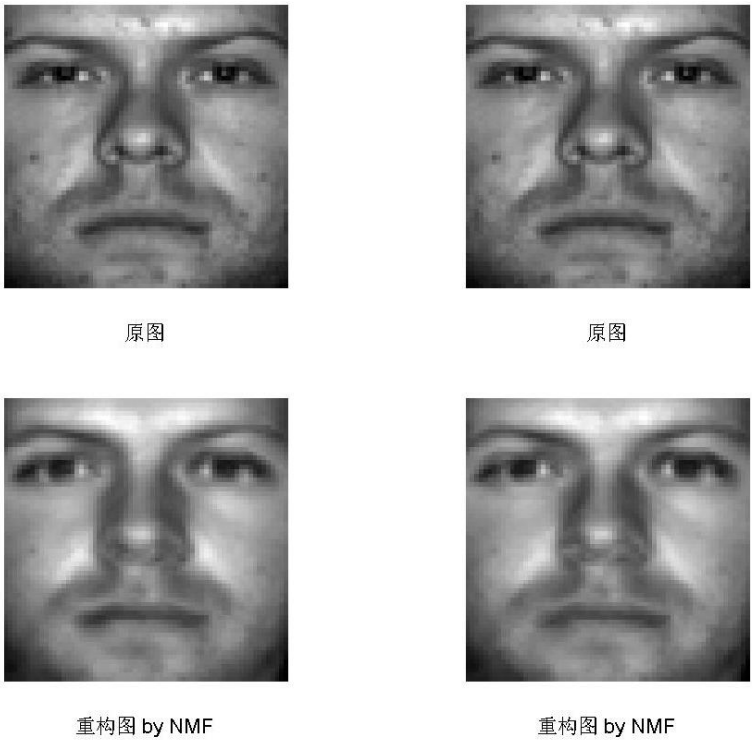

Figure 6: The left result is get by the boundary $||W^{(N)} - W^{(N-1)}||$ and tol=0.01 with 2750 iterations, the right result is get by the boundary $||\mathrm{V} - \mathrm{W} \times \mathrm{H}||$ and tol=1 with 10000 iterations.

It's easily to see the result is almost the same, while with the boundary $||W^{(N)} - W^{(N-1)}||$, we need less iterations, which reduces much computation time.

The difference in codes is little as below:

Ps: The Main Code:

```
% iterations for non-negative matrix factorization
N=10000;
tol=0.01; % it need to be changed with the change of the boundary
err=zeros(N,1);
for i=1:N
    W0=W;
    W=W.*(V*H')./(W*(H*H'));
    H=H.*(W'*V)./(W'*W*H);

%   err(i)=norm(V-W*H,2); % when the boundary is ||V-W*H||
    err(i)=norm(W-W0,2);    % when the boundary is ||W(N)-W(N-1)||
    if(err(i)<tol)
        disp(['Iteration times are:' num2str(i)]);
        display(err(i));
        figure;plot(1:i,err(1:i));
        grid on;title('the error Convergence of NMF');
        break;
    end
end
if(err(i)>=tol)
    disp('N is not enough.');
    display(err(i));
    figure;plot(1:N,err);
    grid on; title('the error Convergence of NMF');
end

% restructed image by NMF
nmf=W*H;

%%%%%%%%%%%%%%%
```