


# 非负矩阵分解（NMF）的Matlab实例与说明

原创 Shian150629 2021-06-12 15:52:04 325 收藏 8 编辑 版权

分类专栏: 编程随记 文章标签: 人脸识别 matlab 机器学习

 编程随记 专栏收录该内容

0 订阅 14 篇文章

原理啥的到处都有，就直接跳过了。这里主要是NMF的基础实验。下一篇是NMF的高光谱实验总结。

## 1. matlab示例解说

这一节的图片来自官方文档。

```
load fisheriris
```

Compute a nonnegative rank-two approximation of the measurements of the four variables in Fisher's iris data.

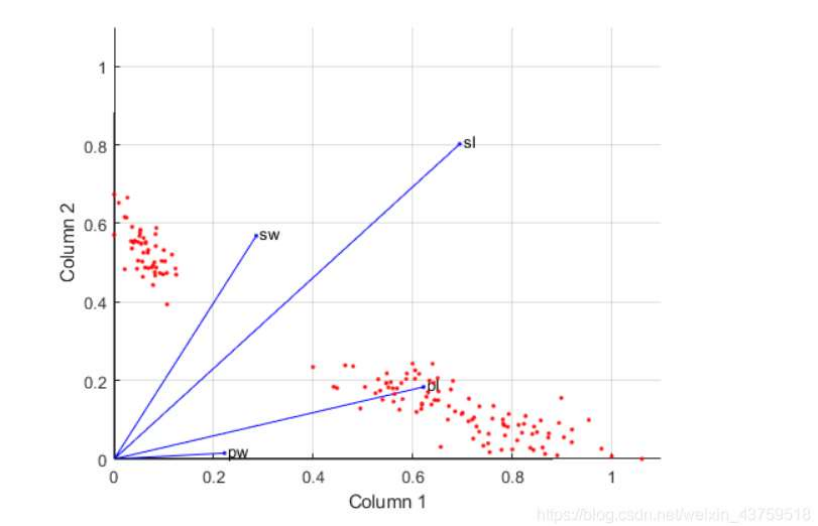
```
rng(1) % For reproducibility
[W,H] = nnmf(meas,2);
H
```

H = 2×4


0.6945	0.2856	0.6220	0.2218
0.8020	0.5683	0.1834	0.0149

这里第一和第三变量在第一行的值0.6945和0.62220对W的第一列有相当强的权重。第一个第二变量在第二行的值0.8020和0.5683对W的第二列有相当大的权重。

```
biplot(H,'Scores',W,'VarLabels',{'sl','sw','pl','pw'});
axis([0 1.1 0 1.1])
xlabel('Column 1')
ylabel('Column 2')
```



其中，H的转置是4乘2大小的。也就是有4是二维坐标轴X和Y轴】，转化到二维图上

 Shian150629

是这四个。

## 2. NMF代码与Lenna图

这个函数是拷别人的，来自[这里](#)。

```
1 function [W,H,errs,loss] = nmf_euc(V, r)
2 % 输入检查
3 % 不含负值元素
4 % H为系数矩阵， 一列不能全为0
5
6 if min(min(V)) < 0
7     error('Matrix entries can not be negative');
8 end
9 if min(sum(V,2)) == 0
10    error('Not all entries in a row can be zero');
11 end
12
13 % V是一个含有n个样本的矩阵， 每一列对应一个样本， 每个样本为m维
14 [m,n] = size(V);
15
16 % 随机初始化W和H矩阵
17 W = rand(m,r);
18 H = rand(r,n);
19
20 % 设置最大迭代次数， 保证分解结果是收敛的
21 niter = 10000;
22
23 myeps = 1e-10;
24
25 errs = zeros(niter,1);
26
27 for t = 1:niter
28
29     W = W .* ( (V*H') ./ max(W*(H*H'), myeps) );
30 %   W = normalize_W(W,1);
31     H = H .* ( (W'*V) ./ max((W'*W)*H, myeps) );
32
33     loss = sum((V-W*H).^2);
34     errs(t) = sum(sum(loss));
35 end
```

这一段是调用该函数与matlab自带的nnmf函数进行的实验。文件lenna图在后面，是一个彩图，真实大小是 $512 \times 512 \times 3$ 。所以使用了reshape函数。我知道lenna是拼错的，该是lenna，但懒得改。

```
1 clear
2 clc
3
4 V=double(imread('lenna.jpg'));
5 imshow(mat2gray(V));
6
7
8 V = reshape(V,512,1536);
9 [W,H] = nnmf(V,50);
10 img_V=W*H;
11 img_V = reshape(img_V,512,512,3);
12 figure;
13 imshow(mat2gray(img_V));
14
15
16 V = reshape(V,512,1536);
17 [W,H,errs,loss] = nmf_euc(V, 100);
18
19 img_V=W*H;
20 img_V = reshape(img_V,512,512,3);
21 figure;
22 imshow(mat2gray(img_V));
```

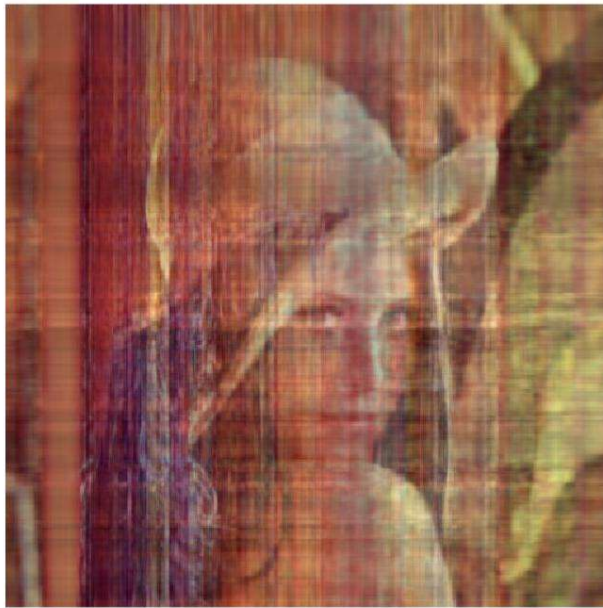


Shian150629

分解出来W大小是 $512 \times 50$ ，H是 $50 \times 1536$ 大小。分解之后再乘回去即可复原。可以发现matlab自带的nnmf函数复原结果不好。使用的Lenna图如下，直接保存就可以用了：



nnmf分解再复原



对不起，Lenna女士。然后是nmf\_euc分解再复原的。这图边界那么大是因为我直接保



Shian150629

存图片然后复制过来的，其实该复制图窗再处理，但是懒orz



可以发现比官方的要好很多。

### 3. NMF与人脸识别

使用的数据集来自[这里的](#)Yale database 32乘32的data file。使用下面的代码打乱顺序并储存

```
1 leng = size(gnd,1);
2 lis = randperm(leng);
3 gnd = gnd(lis,:);
4 fea = fea(lis,:);
5 save('yaleFace.mat','gnd','fea');
```

此后，对数据进行NMF分解

```
1 clear
2 clc
3
4 load('yaleFace.mat')
5 fea = fea';
6 readFace = reshape(fea,32,32,165);
7 for i = 1:165
8     imshow(readFace(:,:,i)./255)
9 end
10
11 rng(1) % For reproducibility
12 k = 16;
13 [W,H] = nnmf(fea,k);
14 C = reshape(W,32,32,k);
15 for i = 1:k
16     imshow(C(:,:,i)./255)
17 end
```

在这里对fea转置，所以实际上fea是1024乘165的。 $1024 = 32 \times 32$ ，是维度。而165是样本数。在随机数种子 `rng(1)` 之前可以把原图给过一遍。此后，降维到 $k=16$ ，那么W是 $1024 \times 16$ 大小的基矩阵，H是 $16 \times 165$ 的系数矩阵。

好了重点来了。我代码这么写的思路是：是 `[W,H] = nnmf(fea,k);` 的k值。之后resh



Shian150629

## 4. 参考资料

- 文档1
- 文档2

[https://blog.csdn.net/weixin\\_43759518/article/details/117848483?spm=1001.2014.3001.5502](https://blog.csdn.net/weixin_43759518/article/details/117848483?spm=1001.2014.3001.5502)