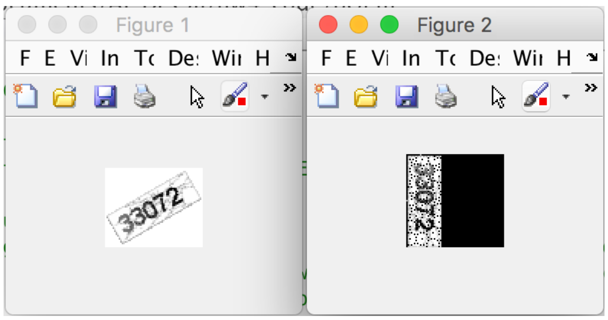
1. Principal Component Analysis

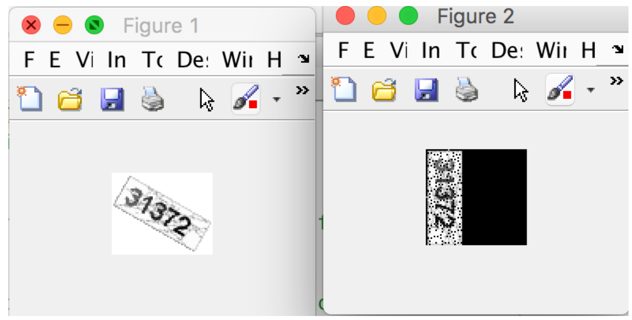
(a)

First I sort eigenvectors from large to small. the first eigenvector corresponds to the largest eigenvalue. Here are the results:

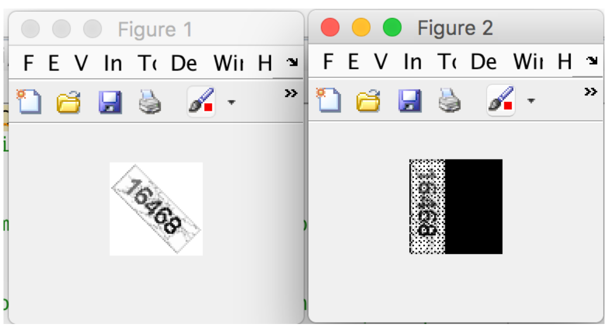
1.gif:



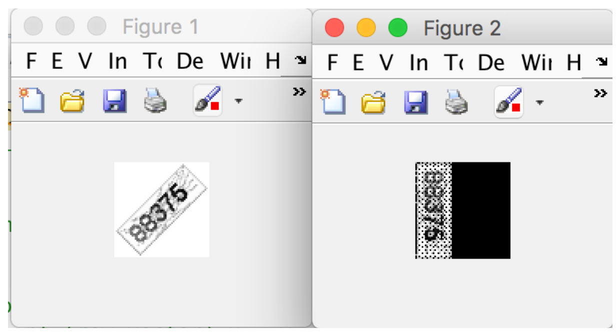
2.gif:



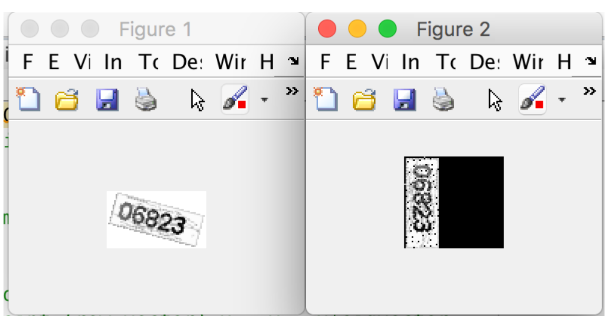
3.gif:



4.gif:

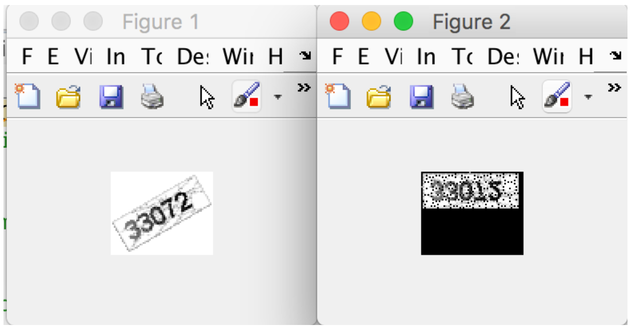


5.gif:

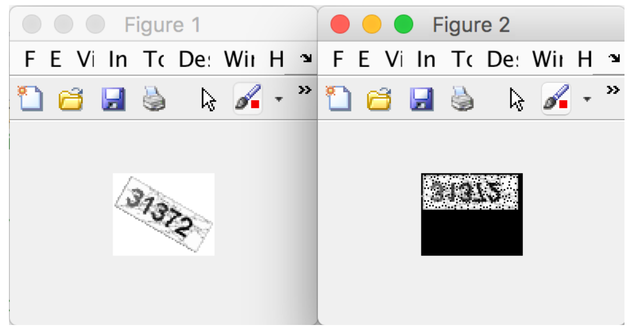


Then I sort eigenvectors from small to large. the first eigenvector corresponds to the smallest eigenvalue. Here are the results:

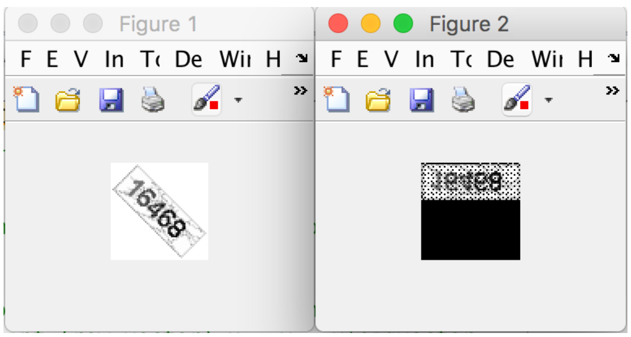
1.gif:



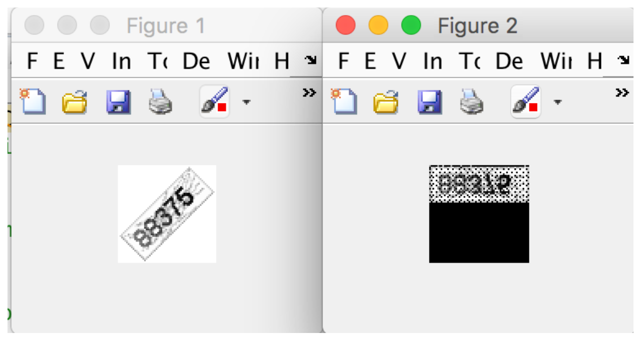
2.gif:



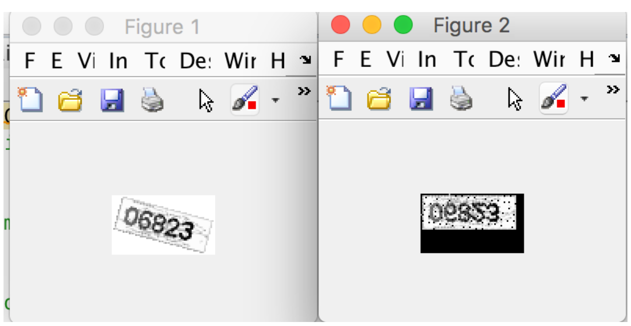
3.gif:



4.gif:



5.gif:



I have some questions and I will ask you later:

1.when I use the eigenvector corresponding to smallest eigenvalue as the first coordinate, the numbers become symmetric to this coordinate, why?

2. I will centralize the data prior to processing. After projecting the data to new coordinate, how can I decentralize the data?

(b)

Here are my thoughts:

Given an SVD of X, the following two relations hold:

$$X^T X = V\Sigma^T U^T U\Sigma \mathrm{V}^\mathrm{T} = \mathrm{V}(\Sigma^T\Sigma)\mathrm{V}^\mathrm{T}$$

$$XX^T = U\Sigma V^T V\Sigma^T U^T = U(\Sigma\Sigma^T)U^T$$

X = fix(rand(10,3)*50)
S=X-repmat(mean(X),10,1)
S is X that has been centralized.

p*lambda*p'=v*(s*s')*v'/(N-1)
lambda=(s*s')/(N-1)

Let X denote the d x n data matrix with column xi as the image vector with mean subtracted. Then,

$$covariance(X) = \frac{XX^T}{n}$$

Let the singular value decomposition (SVD) of X be:

$$X = U\Sigma V^T$$

Then the eigenvalue decomposition for XX^T is:

$XX^T = U\Sigma\Sigma^T U^T = U\Lambda U^T$, where Λ=diag (eigenvalues of XX^T)

Thus we can see easily that:

The eigenfaces = U, the left singular vectors of X
The ith eigenvalue of $XX^T = \frac{1}{n}($ ith singular value of $X)^2$

Using SVD on data matrix X, we don't need to calculate the actual covariance matrix to get eigenfaces.

Next, I will show my answers to the question (b)

(i)original face:

centralized face:

eigenface:

(ii)

```
[u,s,v]=mySVD(X);
dims=[8,16,32,64,128];
for i=1:size(dims,2)
    s_re=s(1:dims(i),1:dims(i));
    u_re=u(:,1:dims(i));
    v_re=v(:,1:dims(i));
    X_train=X*v_re;
    X_test=(fea_Test-repmat(mean(fea_Test),size(fea_Test,1),1))*v_re;
    y=knn(X_test',X_train',gnd_Train',1);
    fprintf('Knn k=1, The error rate of reduced dimensionality=%d is %f.\n',...
    dims(i),length(find(y' ~= gnd_Test))/length(y));
end
```

```
>> pca_exp1
Knn k=1, The error rate of reduced dimensionality=8 is 0.255000.
Knn k=1, The error rate of reduced dimensionality=16 is 0.175000.
Knn k=1, The error rate of reduced dimensionality=32 is 0.155000.
Knn k=1, The error rate of reduced dimensionality=64 is 0.135000.
Knn k=1, The error rate of reduced dimensionality=128 is 0.135000.
```

(iii)
dims=8:

dims=16:

dims=32:

dims=64:

dims=128:

original:

2. Course Feedback

扯淡的问题我就不回答了，直接写写心得吧。

这是我在浙大上过的最好的课程，也是让我收获最大的一门课。我觉得很幸福也很幸运能在大三上学习这门课。很感谢服珏吉学长耐心地辅导。助教哥哥不仅能力超强，而且认真的批改作业还能抽时间给我细心讲解问题。实验室的Wiki、服务器等都是学长一个人在管理，真的是让我非常佩服。

我在这门课不仅仅是学到了知识，可能影响我更多的是结识了实验室优秀的师兄、学长。他们对我的震撼是很深远的，有时候我就是需要一个目标，需要一个方向，然后就能激发自己的斗志。

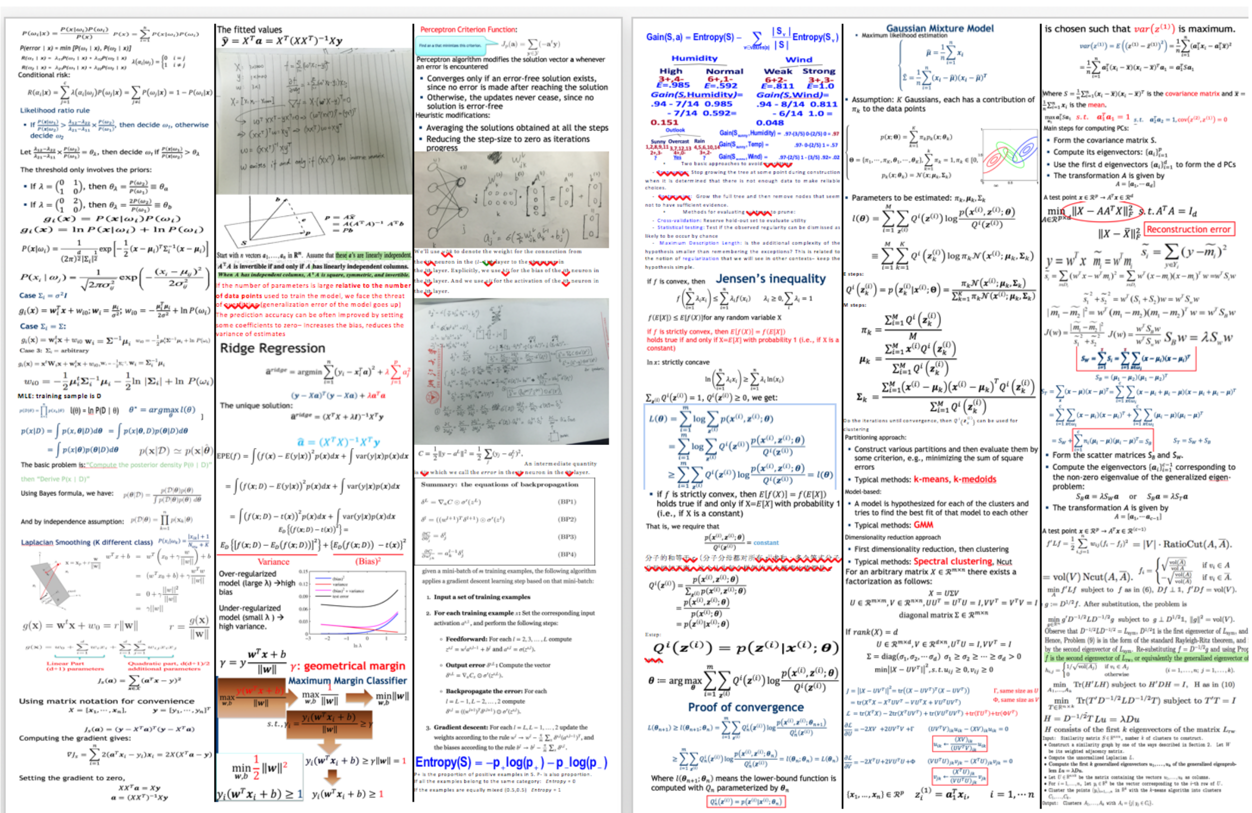
最后总结一下这门课的学习。我这个学期大部分精力都在学习这门课，课余时间把每个算法的wiki，blog，paper都搜索阅读过。

Backpropagation、SVM、Spectral Clustering、EM、PCA这些模型我都读paper并且自己在纸上推导了一遍。（SVM的Lagrange Duality 和 KKT条件那部分目前还看不懂）

线性代数的书又阅读复习了两遍，加深对矩阵理解。知乎上machine learning的帖子挨个读了一遍。我很重视这

些算法的intuitive understanding。有时候博客的语言会让我对算法的大局理解上上一个台阶。

这是我最后考试前花了几十个小时通读了一遍所学内容自己总结的笔记，很多推导都是我自己理解后写在纸上拍成照片记录下来的：



最后是遗憾：

上这门课前数学准备不足，比如Positive definite、投影矩阵、SVD这些内容我到学期末才抽时间过了一遍，然后恍然大悟原来上课讲的是这么回事。以后数学的基础我要继续加强。

期末考试感觉考得不好，也是很多细节的推导还不到位，我下面要把Bayes的概率部分还有模特卡罗马尔可夫这些概率相关的知识好好学学。