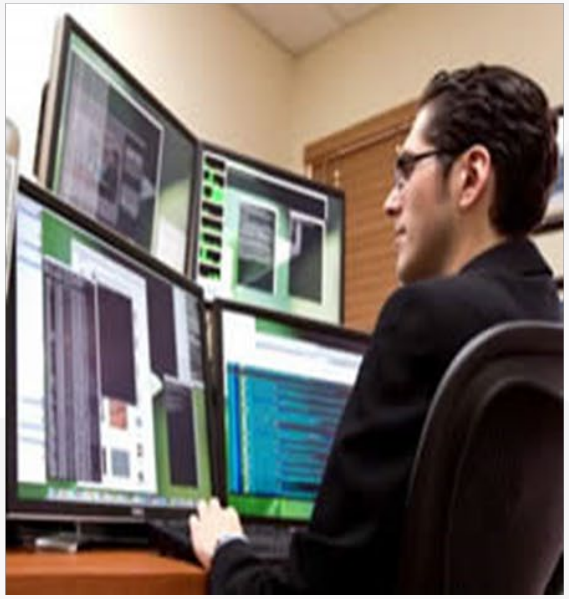
第四章 特征**选择**和**提取**

第四章 特征选择和提取

- 特征选择和提取是模式识别中的一个关键问题
  - •前面讨论分类器设计的时候，一直假定已给出了特征向量维数确定的样本集，其中各样本的每一维都是该样本的一个特征；
  - •这些特征的选择是很重要的，它强烈地影响到分类器的设计及其性能；
  - •假若对不同的类别，这些特征的差别很大，则比较容易设计出具有较好性能的分类器。

例子

- 判断一个人是否是软件工程师？
  - [肤色，体重，身高，工资，编程，受教育程度，性别，….]
- 判断一个人是否是篮球运动员？
  - [肤色，体重，身高，工资，编程，受教育程度，性别，….]



第四章 特征选择和提取

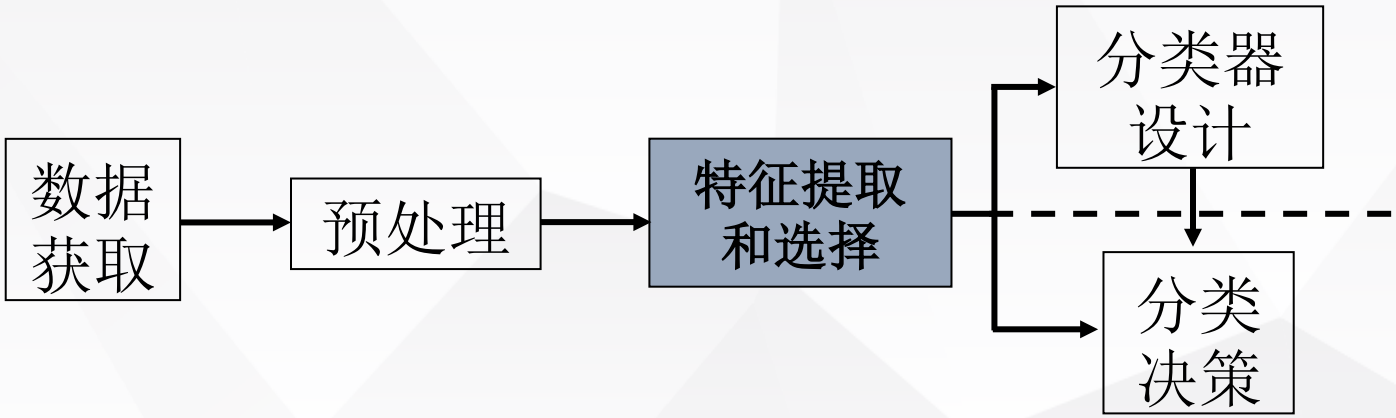
- 特征选择和提取是构造模式识别系统时的一个重要课题
  - •在很多实际问题中，往往不容易找到那些最重要的特征，或受客观条件的限制，不能对它们进行有效的测量；
  - •因此在测量时，由于人们心理上的作用，只要条件许可总希望把特征取得多一些；
  - •另外，由于客观上的需要，为了突出某些有用信息，抑制无用信息，有意加上一些比值、指数或对数等组合计算特征；
  - •如果将数目很多的测量值不做分析，全部直接用作分类特征，不但耗时，而且会影响到分类的效果，产生“特征维数灾难”问题。

第四章 特征选择和提取

- 为了设计出效果好的分类器，通常需要对原始的测量值集合进行分析，经过选择或变换处理，组成有效的识别特征；
- 在保证一定分类精度的前提下，减少特征维数，即进行“降维”处理，使分类器实现快速、准确和高效的分类。
- 为达到上述目的，关键是所提供的识别特征应具有很好的可分性，使分类器容易判别。为此，需对特征进行选择。**直观标准?**
  - 应去掉模棱两可、不易判别的特征；
  - 所提供的特征不要重复，即去掉那些相关性强且没有增加更多分类信息的特征。

■说明

•实际上，特征选择和提取这一任务应在设计分类器之前进行；



•从通常的模式识别教学经验看，在讨论分类器设计之后讲述特征选择和提取，更有利于加深对该问题的理解。

第四章 特征选择和提取

- 所谓特征选择，就是从n个度量值集合$\{x_1, x_2, \ldots, x_n\}$中，按某一准则选取出供分类用的子集，作为降维（$m$维，$m<n$）的分类特征；
- 所谓特征提取，就是使$(x_1, x_2, \ldots, x_n)$通过某种变换，产生m个特征$(y_1, y_2, \ldots, y_m)$ $(m<n)$，作为新的分类特征（或称为二次特征）；
- 其目的都是为了在尽可能**保留识别信息的前提下，降低特征空间的维数**，已达到有效的分类。

以细胞自动识别为例

- 通过图像输入得到一批包括正常细胞和异常细胞的图像，我们的任务是根据这些图像区分哪些细胞是正常的，哪些细胞是异常的；
- 首先找出一组能代表细胞性质的特征，为此可计算

细胞总面积

总光密度

胞核面积

核浆比

细胞形状

核内纹理

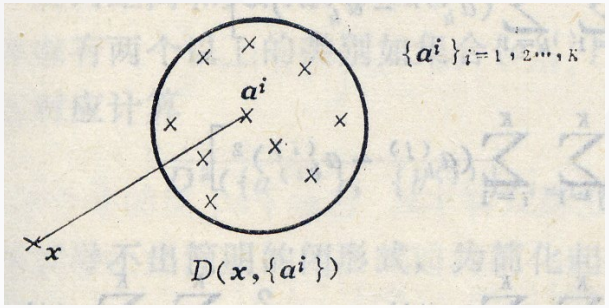
……

- 以细胞自动识别为例
  - •这样产生出来的原始特征可能很多（几十甚至几百个），或者说原始特征空间维数很高，需要降低（或称压缩）维数以便分类；
  - •一种方式是**从原始特征中挑选出**一些最有代表性的特征，称之为**特征选择**；
  - •另一种方式是用映射（或称变换）的方法**把原始特征变换为较少的特征**，称之为**特征提取。**

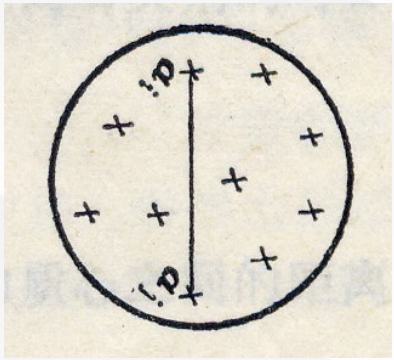
4.1 模式类别可分性的测度

■距离和散布矩阵

- 点到点之间的距离
- 点到点集之间的距离



- 类内距离



4.1 模式类别可分性的测度

- 距离和散布矩阵
  - 类内散布矩阵
  - 类间距离和类间散布矩阵
  - 多类模式集散布矩阵

4.2 特征选择

- 设有n个可用作分类的测量值，为了在不降低（或尽量不降低）分类精度的前提下，减小特征空间的维数以减少计算量，需从中**直接**选出m个作为分类的特征。
- 问题：在n个测量值中选出哪一些作为分类特征，使其具有最小的分类错误?

4.2 特征选择

- 从n个测量值中选出m个特征，一共有C_n^m中可能的选法。
  - 一种“穷举”办法：对每种选法都用训练样本试分类一下，测出其正确分类率，然后做出性能最好的选择，此时需要试探的特征子集的种类达到$C_n^m = \frac{n!}{m!(n-m)!}$种，非常耗时。
  - 需寻找一种简便的可分性准则，间接判断每一种子集的优劣。

    对于独立特征的选择准则

    一般特征的散布矩阵准则

4.2 特征选择

- 对于独立特征的选择准则
  - 类别可分性准则应具有这样的特点，即不同类别模式特征的均值向量之间的距离应最大，而属于同一类的模式特征，其方差之和应最小。
  - 假设各原始特征测量值是统计独立的，此时，只需对训练样本的n个测量值独立地进行分析，从中选出m个最好的作为分类特征即可。

例：对于ω_i和ω_j两类训练样本的特征选择

4.2 特征选择

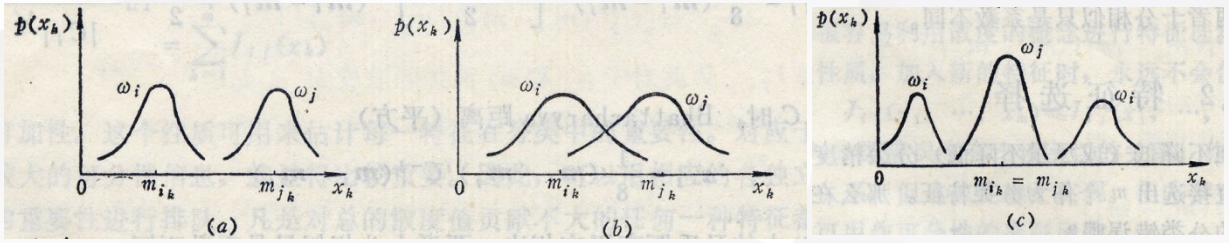
- 讨论：上述基于距离测度的可分性准则，其适用范围与模式特征的分布有关。
  - •三种不同模式分布的情况

    (a) 中特征x_k的分布有很好的可分性，通过它足以分离ω_i和ω_j两种类别；

    (b) 中的特征分布有很大的重叠，单靠x_k达不到较好的分类，需要增加其它特征；

    (c) 中的ω_i类特征x_k的分布有两个最大值，虽然它与ω_j的分布没有重叠，但计算G_k约等于0，此时再利用G_k作为可分性准则已不合适。

  - •因此，假若类概率密度函数不是或不近似正态分布，均值和方差就不足以用来估计类别的可分性，此时该准则函数不完全适用。



4.2 特征选择

一般特征的散布矩阵准则

$$J_1 = \det(S_w^{-1}S_b) = \prod_i \lambda_i$$

$$J_2 = \mathrm{tr}(S_w^{-1}S_b) = \sum_i \lambda_i$$

- 类内、类间的散布矩阵S_w和S_b
- 类间离散度越大且类内离散度越小，可分性越好。
- 散布矩阵准则J_1和J_2形式
- 使J_1或J_2最大的子集可作为所选择的分类特征。

注：这里计算的散布矩阵不受模式分布形式的限制，但需要有足够数量的模式样本才能获得有效的结果

作业

- 设有如下三类模式样本集ω_1，ω_2和ω_3，其先验概率相等，求$\boldsymbol{S}_w$和$\boldsymbol{S}_b$

ω_1：$\{(1\ 0)^T, (2\ 0)^T, (1\ 1)^T\}$

ω_2：$\{(-1\ 0)^T, (0\ 1)^T, (-1\ 1)^T\}$

ω_3：$\{(-1\ -1)^T, (0\ -1)^T, (0\ -2)^T\}$

4.3 离散K-L变换

- 全称：Karhunen-Loeve变换（卡洛南-洛伊变换）
- 前面讨论的特征选择是在一定准则下，从n个特征中选出k个来反映原有模式。
- 这种简单删掉某n-k个特征的做法并不十分理想，因为一般来说，原来的n个数据各自在不同程度上反映了识别对象的某些特征，简单地删去某些特征可能会丢失较多的有用信息。
- 如果将原来的特征做正交变换，获得的每个数据都是原来n个数据的线性组合，然后从新的数据中选出少数几个，使其尽可能多地反映各类模式之间的差异，而这些特征间又尽可能相互独立，则比单纯的选择方法更灵活、更有效。
- K-L变换就是一种适用于任意概率密度函数的正交变换。

4.3 离散K-L变换

4.3.1 离散的有限K-L展开

■展开式的形式

- 如果对M种模式类别$\{\omega_i\}_{i=1,\ldots,M}$做离散正交展开，则对每一模式可分别写成：$\boldsymbol{x}_i=\boldsymbol{\Phi}\boldsymbol{a}_i$，其中矩阵$\boldsymbol{\Phi}$取决于所选用的正交函数。
- 对各个模式类别，正交函数都是相同的，但其展开系数向量$\boldsymbol{a}_i$则因类别的不同模式分布而异。

■K-L展开式的性质

- K-L展开式的根本性质是将随机向量x展开为另一组正交向量φ_j的线性和，且其展开式系数a_j（即系数向量$\boldsymbol{a}$的各个分量）具有不同的性质。
- 在此条件下，正交向量集$\{\varphi_j\}$的确定
- K-L展开式系数的计算步骤

4.3 离散K-L变换

4.3.2 按K-L展开式选择特征

- K-L展开式用于特征选择相当于一种线性变换。
- 若从n个特征向量中取出m个组成变换矩阵$\boldsymbol{\Phi}$，即

$$\boldsymbol{\Phi} = (\boldsymbol{\varphi}_1\ \boldsymbol{\varphi}_2\ \dots\ \boldsymbol{\varphi}_m),\quad m<\mathrm{n}$$

此时，$\boldsymbol{\Phi}$是一个n*m维矩阵，$\boldsymbol{x}$是n维向量，经过$\boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$变换，即得到降维为$m$的新向量。

- 问题：选取变换矩阵$\boldsymbol{\Phi}$，使得降维后的新向量在最小均方差条件下接近原来的向量$\boldsymbol{x}$

4.3.2 按K-L展开式选择特征

■结论

•从K-L展开式的性质和按最小均方差的准则来选择特征，应使$E[a_j]=0$。由于$E[\boldsymbol{a}]=E[\boldsymbol{\Phi}^T\boldsymbol{x}]=\boldsymbol{\Phi}^T E[\boldsymbol{x}]$，故应使$E[\boldsymbol{x}]=0$。基于这一条件，在将整体模式进行K-L变换之前，**应先将其均值作为新坐标轴的原点，采用协方差矩阵C或自相关矩阵R来计算特征值**。如果$E[\boldsymbol{x}]\neq 0$，则只能得到“次最佳”的结果。

4.3.2 按K-L展开式选择特征

$$\Phi^T = \begin{pmatrix} \phi_1^T \\ \phi_2^T \\ \vdots \\ \phi_m^T \end{pmatrix}$$

■结论

•将K-L展开式系数a_j（亦即变换后的特征）用y_j表示，写成向量形式：$\boldsymbol{y}= \boldsymbol{\Phi}^{\mathrm{T}}\boldsymbol{x}$。此时变换矩阵$\boldsymbol{\Phi}$用$m$个特征向量组成。为使误差最小，不采用的特征向量，其对应的特征值应尽可能小。因此，将特征值按大小次序标号，即

$$\lambda_1 > \lambda_2 > \ldots > \lambda_m > \ldots > \lambda_n >= 0$$

若首先采用前面的m个特征向量，便可使变换误差最小。此时的变换矩阵为

4.3.2 按K-L展开式选择特征

■结论

•K-L变换是在均方误差最小的意义下获得数据压缩（降维）的最佳变换，且不受模式分布的限制。对于一种类别的模式特征提取，它不存在特征分类问题，只是实现用低维的m个特征来表示原来高维的n个特征，使其误差最小，亦即使其整个模式分布结构尽可能保持不变。

4.3.2 按K-L展开式选择特征

■结论

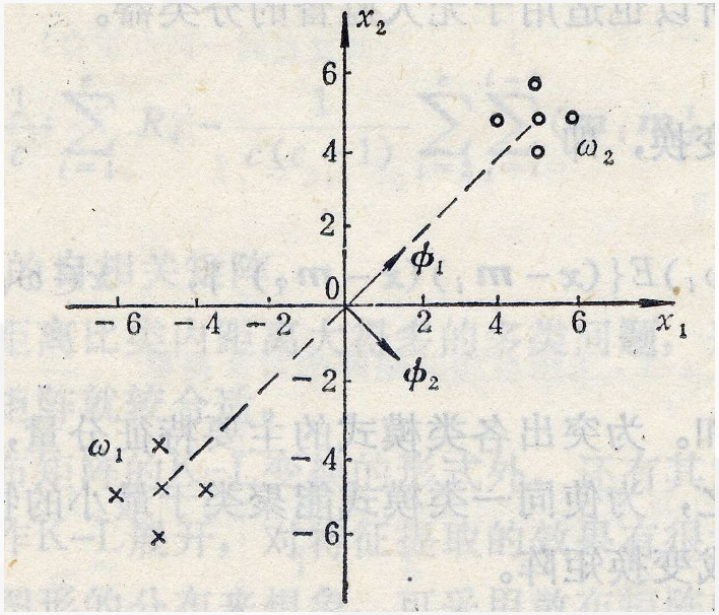
•通过K-L变换能获得互不相关的新特征。若采用较大特征值对应的特征向量组成变换矩阵，则能对应地保留原模式中方差最大的特征成分，所以K-L变换起到了减小相关性、突出差异性的效果。在此情况下， K-L变换也称为**主成分变换（PCA变换）**。

•需要指出的是，采用K-L变换作为模式分类的特征提取时，要特别注意保留不同类别的模式分类鉴别信息，仅单纯考虑尽可能代表原来模式的主成分，有时并不一定有利于分类的鉴别。

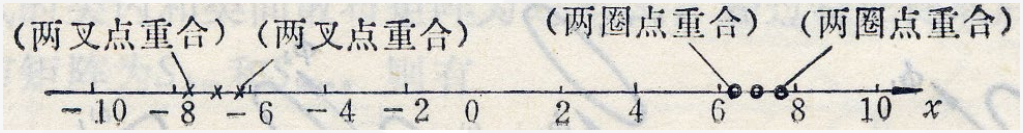
4.3.2 按K-L展开式选择特征

■ [K-L变换实例]

•原始模式分布



•特征提取



作业

- 设有如下两类样本集，其出现的概率相等：

$$\omega_1: \{(0\ 0\ 0)^T, (1\ 0\ 0)^T,$$
$$(1\ 0\ 1)^T, (1\ 1\ 0)^T\}$$

$$\omega_2: \{(0\ 0\ 1)^T, (0\ 1\ 0)^T,$$
$$(0\ 1\ 1)^T, (1\ 1\ 1)^T\}$$

用K-L变换，分别把特征空间维数降到二维和一维，并画出样本在该空间中的位置。

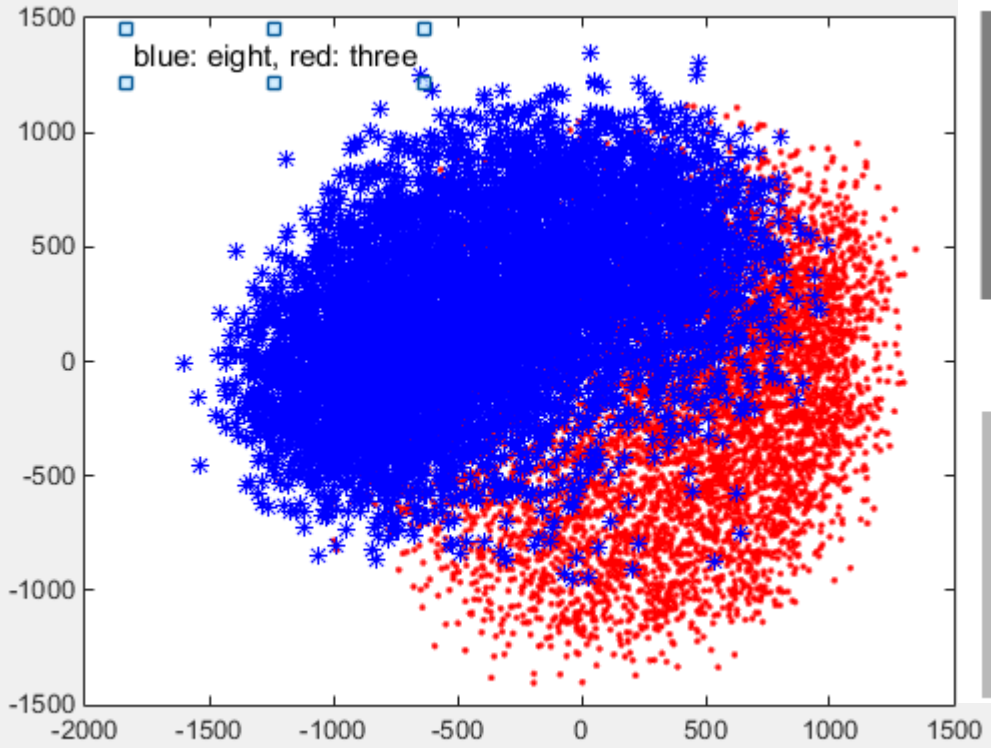
例子-手写数字

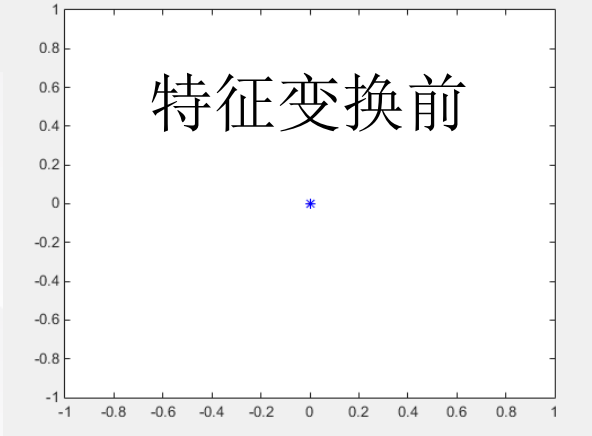
```
load mnist_all;
thr_train = double(train3');
eig_train = double(train8');
m = mean([thr_train, eig_train],2);
thr_No = size(thr_train,2);
eig_No = size(eig_train,2);

thr_train_zero = thr_train-repmat(m,1,thr_No);
thr_C = thr_train_zero*thr_train_zero';
eig_train_zero = eig_train-repmat(m,1,eig_No);
eig_C = eig_train_zero*eig_train_zero';
total_C = 0.5/thr_No*thr_C+0.5/eig_No*eig_C;
```

```
[V,D]=eigs(total_C,10);
new_thr_train =V'*thr_train_zero;
new_eig_train =V'*eig_train_zero;
```

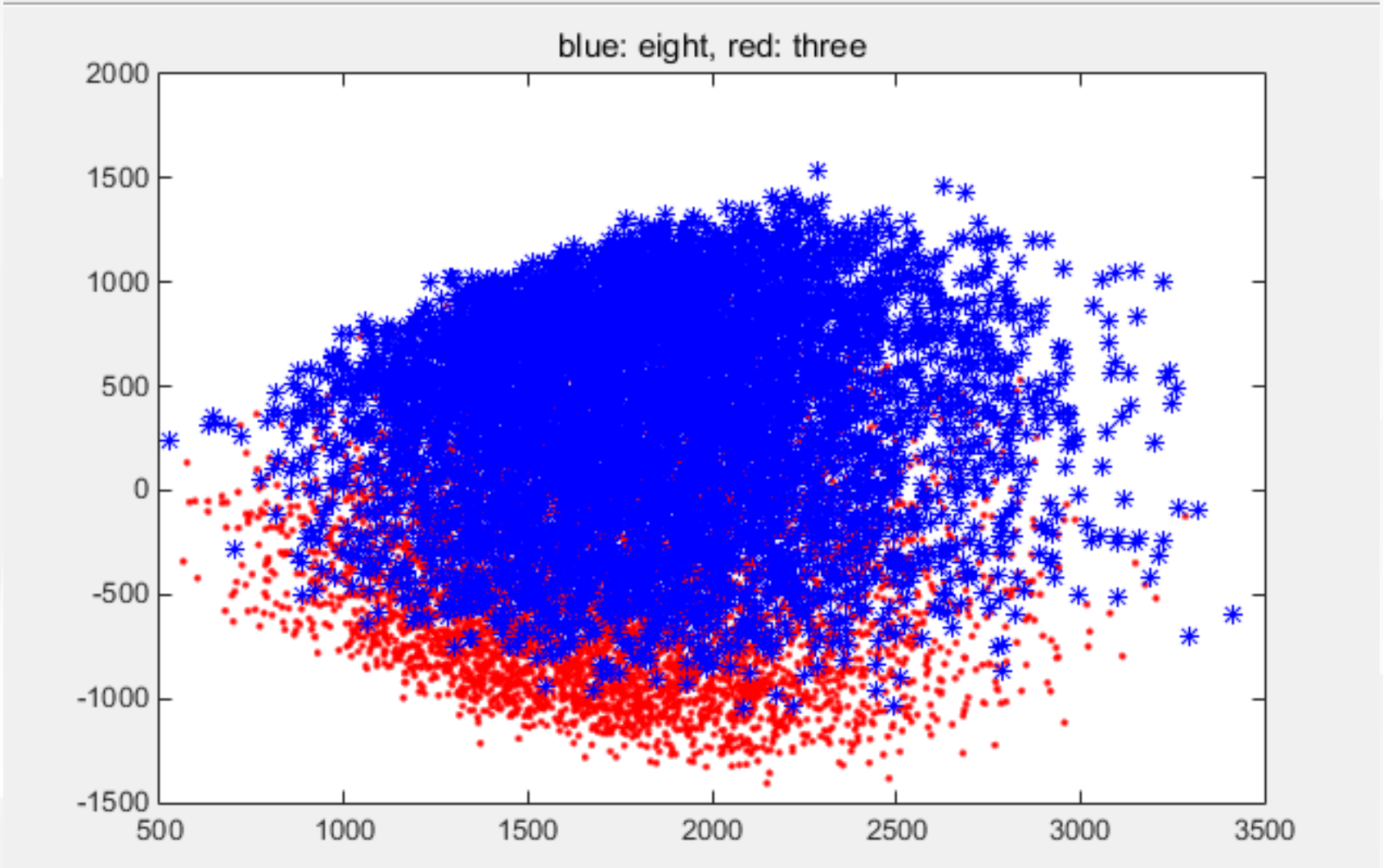
```
figure(1);
plot(new_thr_train(1,:),new_thr_train(2,:),'r.');
hold on;
plot(new_eig_train(1,:),new_eig_train(2,:),'b*');
title('blue: eight, red: three')
figure(2)
plot(thr_train(1,:),thr_train(2,:),'r.');
hold on;
plot(eig_train(1,:),eig_train(2,:),'b*');
```

可视化结果

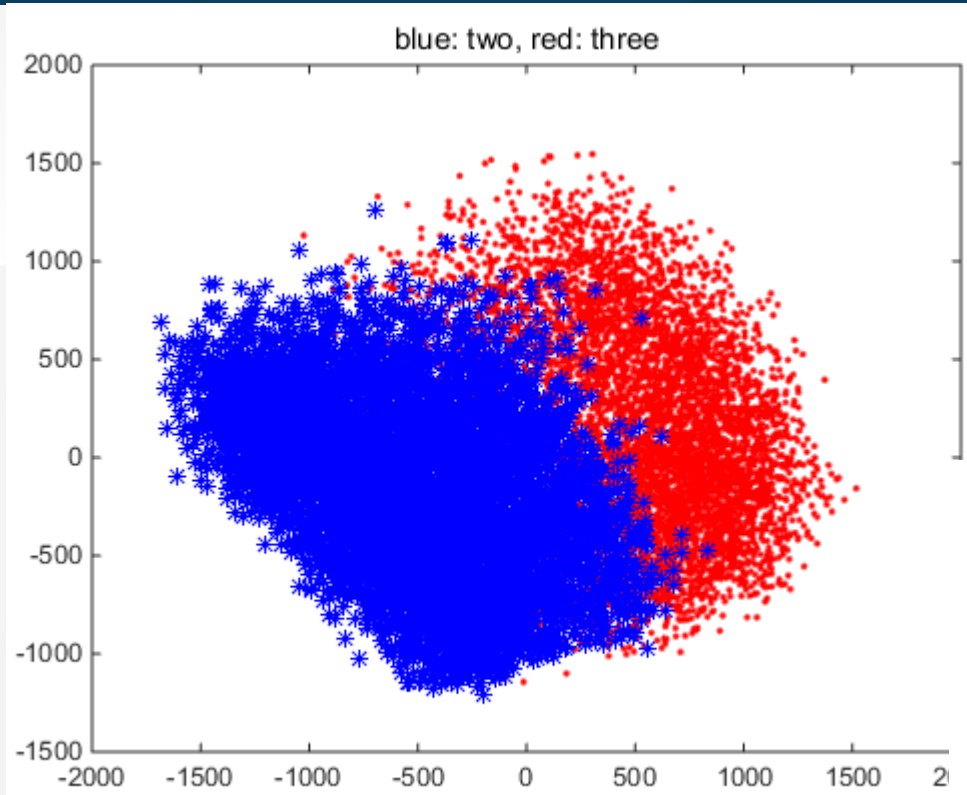

blue: eight, red: three
1500
1000
500
0
-500
-1000
-1500
-2000
-1500
-1000
-500
0
500
1000
1500


特征变换前
1
0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
-1
-1
-0.8
-0.6
-0.4
-0.2
0
0.2
0.4
0.6
0.8
1

用自相关矩阵-KL变换

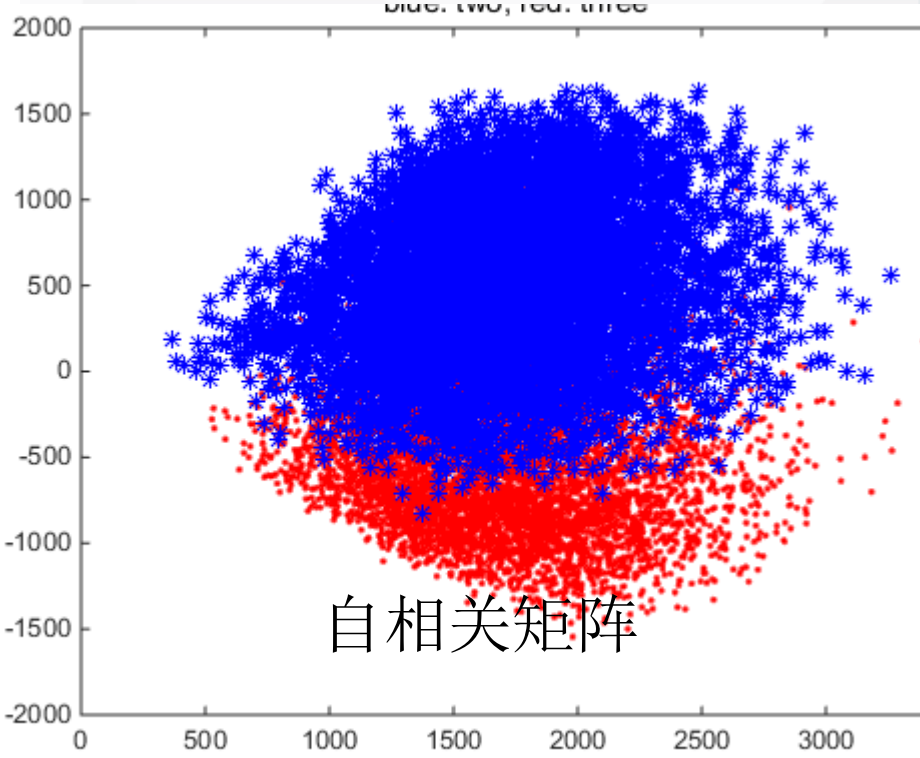

blue: eight, red: three
2000
1500
1000
500
0
-500
-1000
-1500
500
1000
1500
2000
2500
3000
3500

结果



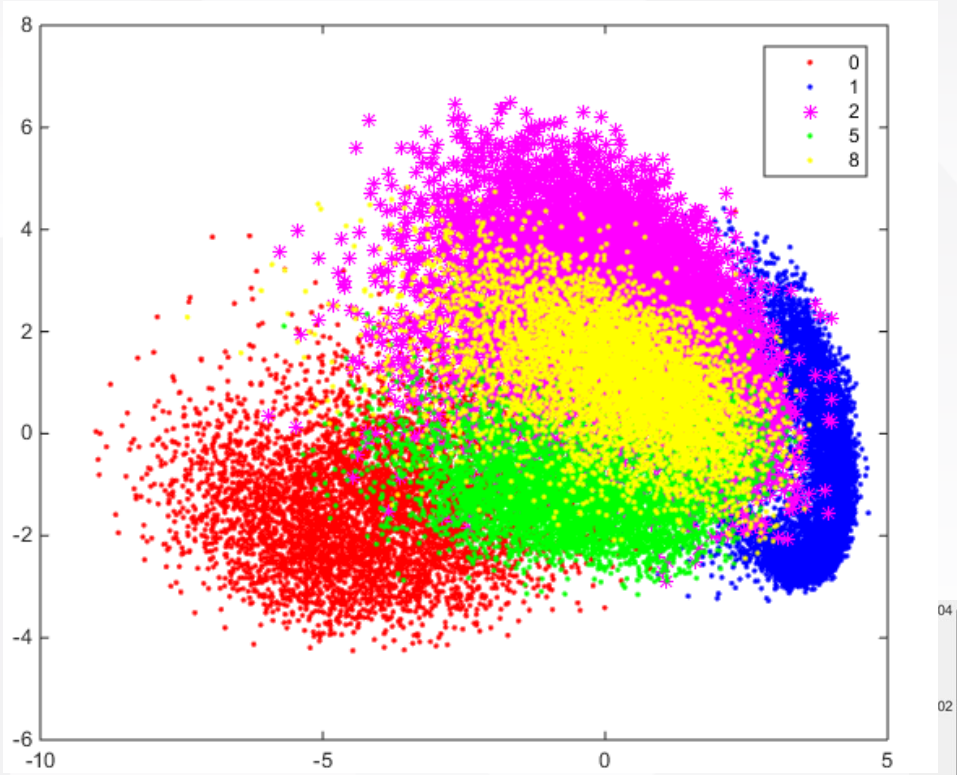
平均重构均方误差：
1.6634e+06

平均重构均方误差：
1.6743e+06



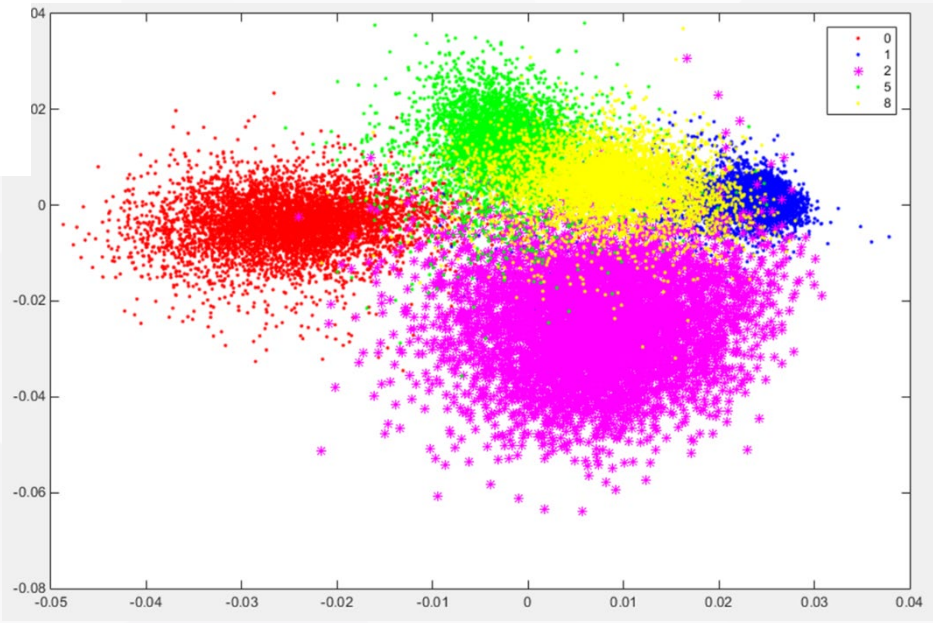
自相关矩阵

多类

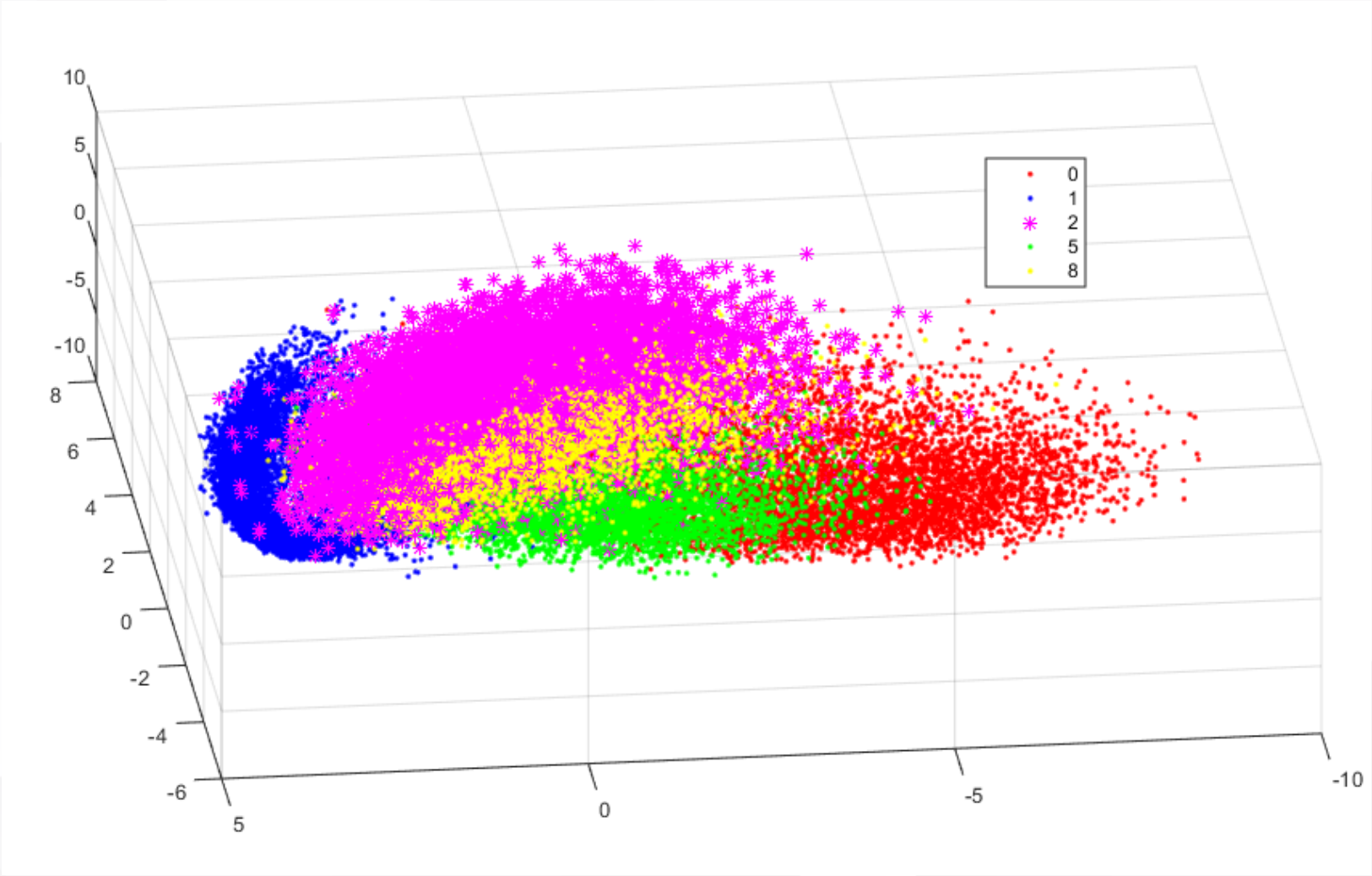


K-L变换

Fisher



多类



```
clear all;
load KL_trainData;%数字0 1 2 5 8降维后的数据
load mnist_all;
d=4;

X=[new_train0(1:d,:)';new_train1(1:d,:)';new_train2(1:d,:)';new_train3(1:d,:)';new_train4(1:d,:)'];
n0= size(new_train0,2);
n1=size(new_train1,2);
n2=size(new_train2,2);
n3=size(new_train3,2);
n4=size(new_train4,2);
Y=[zeros(n0,1);ones(n1,1);ones(n2,1)*2;ones(n3,1)*3;ones(n4,1)*4];
tc = fitctree(X,Y);
resuberror = resubLoss(tc)%衡量分类误差test0_y=W'*(double(test0'/256));

test0= double(test0'/256)-repmat(m_total,1,size(test0,1));
test0_y=V'*test0;
test1= double(test1'/256)-repmat(m_total,1,size(test1,1));
test1_y=V'*test1;
test2= double(test2'/256)-repmat(m_total,1,size(test2,1));
test2_y=V'*test2;
test5= double(test5'/256)-repmat(m_total,1,size(test5,1));
test3_y=V'*test5;
test8=double(test8'/256)-repmat(m_total,1,size(test8,1));
test4_y=V'*test8;

pre_Ynew_1 = predict(tc,test1_y(1:d,:)');
pre_Ynew_0 = predict(tc,test0_y(1:d,:)');
pre_Ynew_2 = predict(tc,test2_y(1:d,:)');
pre_Ynew_3 = predict(tc,test3_y(1:d,:)');
pre_Ynew_4 = predict(tc,test4_y(1:d,:)');
accuracy_0 = sum(pre_Ynew_0==0)/size(pre_Ynew_0,1)
accuracy_1 = sum(pre_Ynew_1==1)/size(pre_Ynew_1,1)
accuracy_2 = sum(pre_Ynew_2==2)/size(pre_Ynew_2,1)
```

降至 4 维

train_error = 0.0382

accuracy_0 = 0.8949

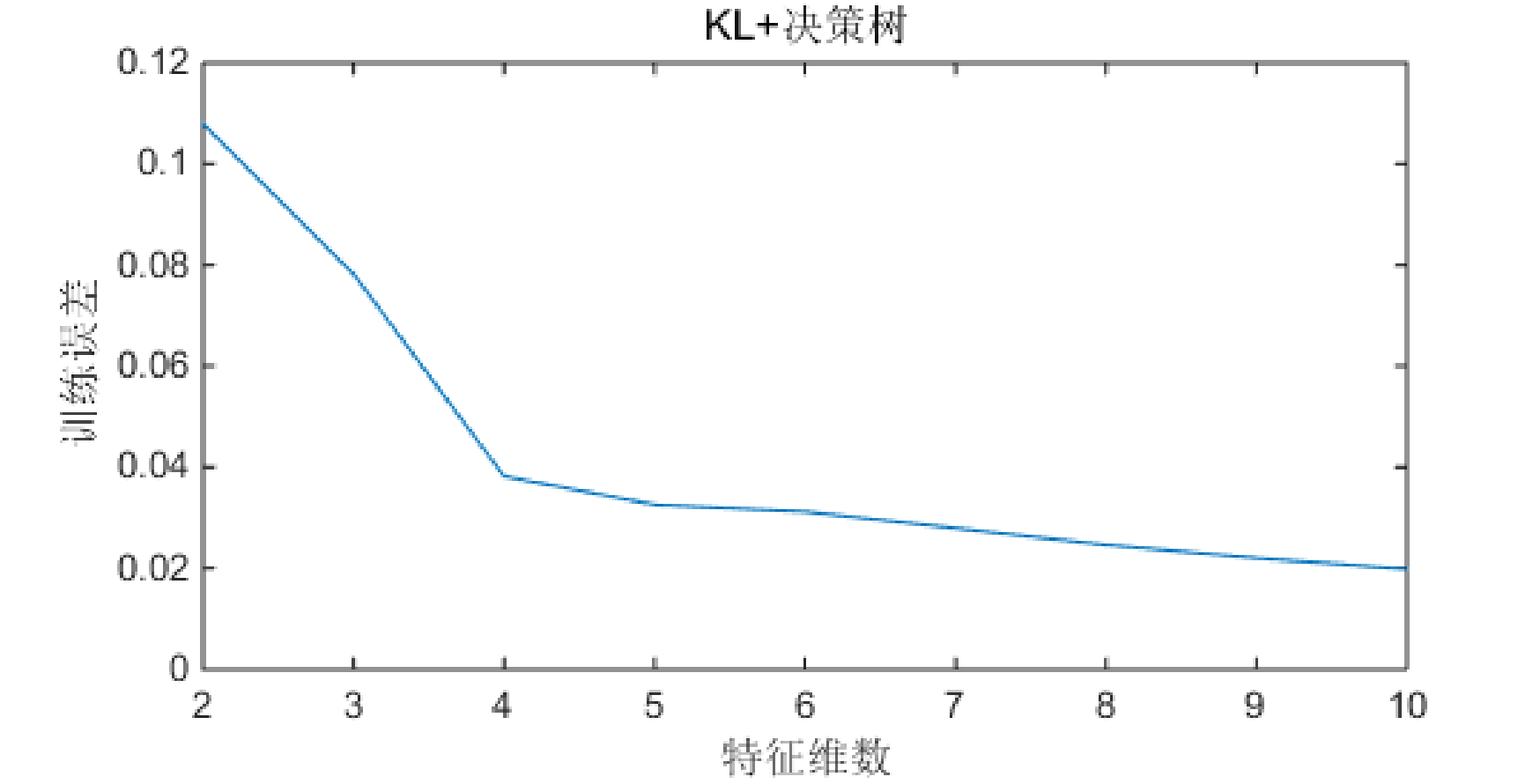
accuracy_1 = 0.9656

accuracy_2 = 0.8905

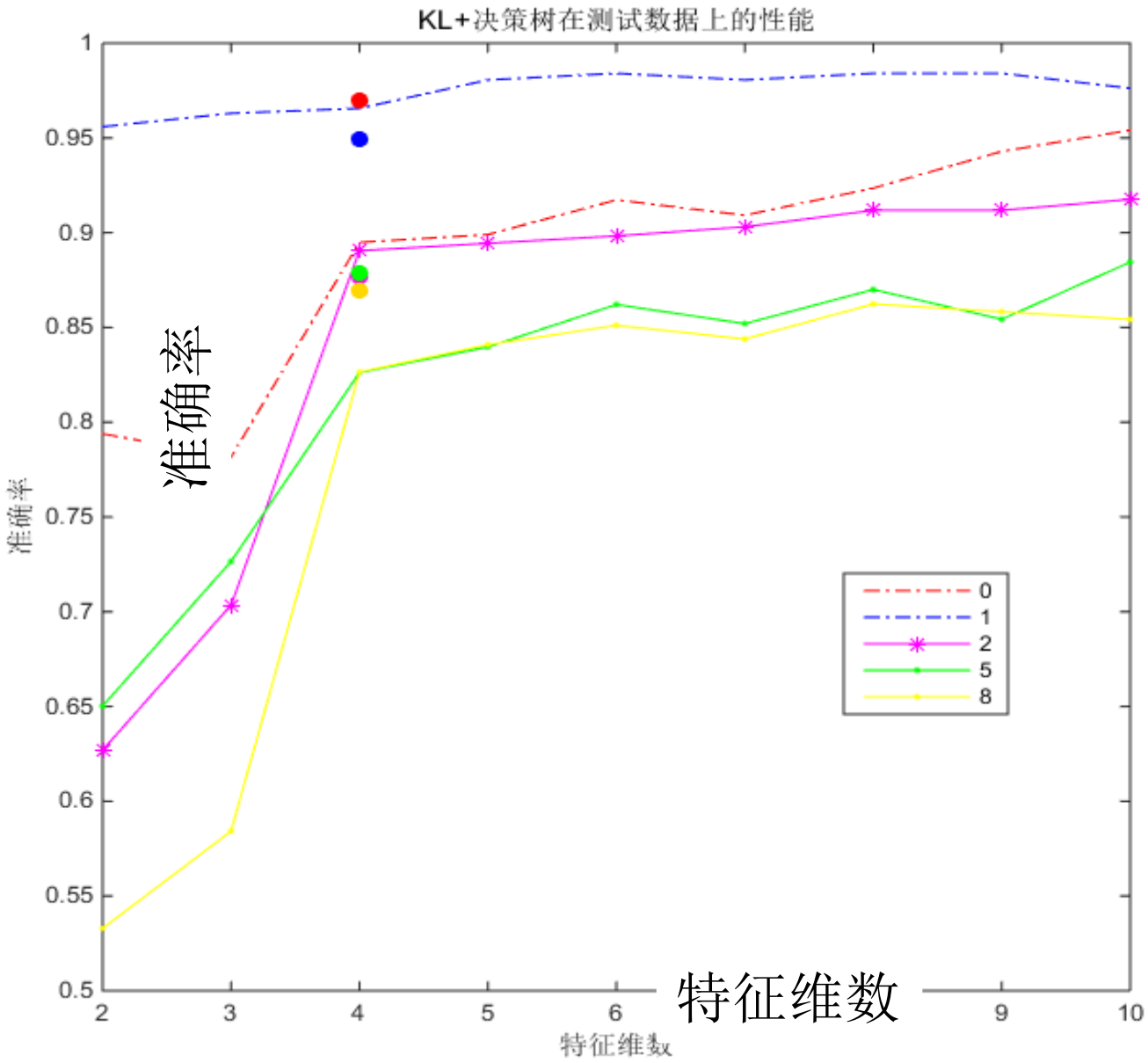
accuracy_3 = 0.8262

accuracy_4 = 0.8265

特征维数 VS 训练误差



KL 参数敏感性


KL+决策树在测试数据上的性能
准确率
特征维数
1
0.95
0.9
0.85
0.8
0.75
0.7
0.65
0.6
0.55
0.5
2
3
4
5
6
9
10
0
1
2
5
8

对一类数字单独进行KL变换

```
clear all
load mnist_all;
thr_train = double(train3');
eig_train = double(train2');
m = mean([thr_train, eig_train],2);
thr_No = size(thr_train,2);
eig_No = size(eig_train,2);

thr_train_zero = thr_train-repmat(m,1,thr_No);
thr_C = thr_train_zero*thr_train_zero';
eig_train_zero = eig_train-repmat(m,1,eig_No);
eig_C = eig_train_zero*eig_train_zero';
total_C = thr_No/(thr_No+eig_No)*thr_C+eig_No/(thr_No+eig_No)*eig_C;

[V,D]=eigs(total_C,10);
new_thr_train =V'*thr_train_zero;
new_eig_train =V'*eig_train_zero;

figure(1);
plot(new_thr_train(1,:),new_thr_train(2,:),'r.');
hold on;
plot(new_eig_train(1,:),new_eig_train(2,:),'b*');
title('blue: two, red: three')
figure(2)
plot(thr_train(1,:),thr_train(2,:),'r.');
hold on;
plot(eig_train(1,:),eig_train(2,:),'b*');
```