模式识别第二次作业实验报告

2013012245 白可

概要：
这次实验，对于比较流行的四种传统方法进行了实验，收获很大，先将最后测得的数据附在这里，然后依次对于每种方法进行讨论。分隔符前面的/是训练集正确率，后面是测试集正确率（随机取最大）。在文章末尾，我附上了一二的代码说明

训练样本数	特征数	1	2	3	4
10+10	10	0.92/0.83	0.95/0.902	1/0.909	1/0.908
	2	1/0.810	0.90/0.87	0.95/0.93	1/0.994
469+485	10	0.905/0.841	0.905/0.899	0.912/0.881	0.921/0.875
	2	0.907/0.875	0.915/0.884	0.911/0.878	0.912/0.891

实验一：bayes

这次我又重新写了 bayes，力图比上回的代码更加简洁，更具有可重用性。其中一个值得思考的问题是：

训练样本数	特征数	B
10+10	10	0.92/0.83
	2	0.9/0.805
469+485	10	0.95/0.84
	2	0.903/0.869

观察这个表格，在训练样本数较少和较多时，他们的正确率似乎没有很大的提升。

我们是否就应该说没有很好的优化吗？答案是否定的。统计上可以证明，因为大的训练样本数得到的均值的方差更小，就是数据会更加准确。我测试了若干组随机数据，得到的结果，测试集的准确率大约在 0.7——0.8 范围内波动。

因此，当测试集样本较小时，我们应该采取 n_fold 方式，这样增加自己的训练量，提高答案的可信度。

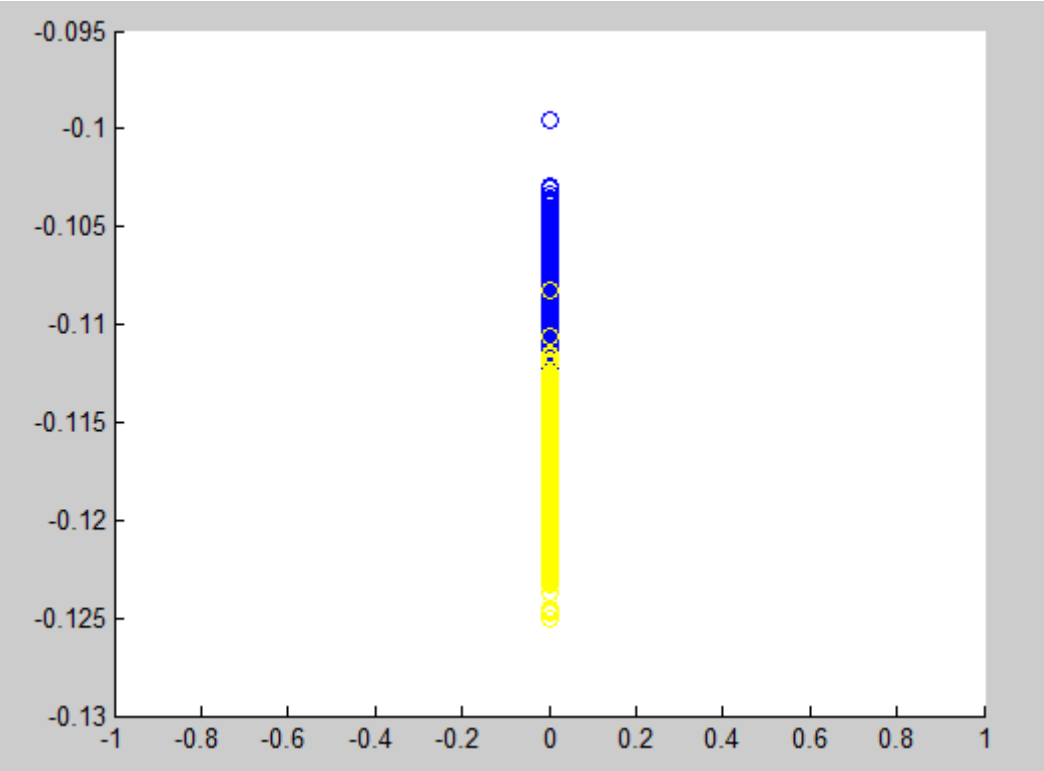
实验二：fisher 变换

此处进行的是一维 fisher 变换。

一维 fisher 变换较为普遍且在大量参考书中都有介绍，此处不再对其全部流程做推理。它的主要思想是：实现类内方差小，类间距大。因此定义了一个函数 J，通过最大化 J 来实现这一分类目的。经过数学推导，最后对于我们的计算有帮助的式子为：

$$\omega \propto S_w^{-1}(\overrightarrow{m_1} - \overrightarrow{m_2})$$

然后将所有数据使用ω 进行投影得到一个实数。对不同类别的所得到的点进行标记，最后观察不同类别点的分布。



书上将其投影到条斜线上，然而实际上，他们都只是变换到了一维上面，因此用一条线就可以表达出他们之间的关系。根据图像也可以看出，他们被较好地区分开来。

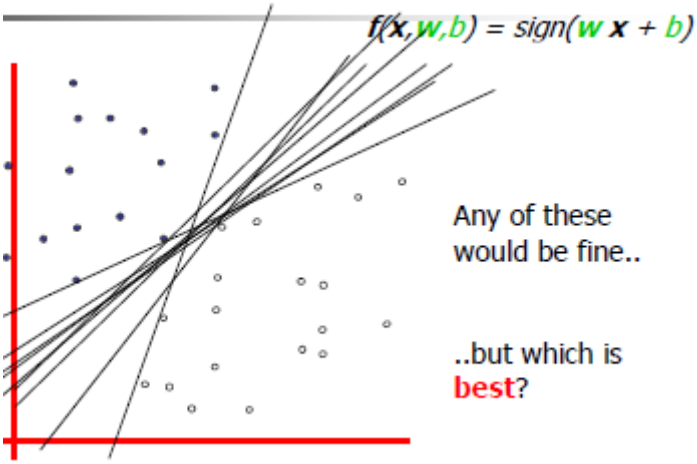
在我的模型中，训练样本数和特征维数对于结果没有起到很大的作用。而正确率最高的（训练样本数为 10）由于极大的随机性。他虽然并不能够说明问题。但是它却反映了，在这种状态下，我们确实可以有比较好的投影效果。这启示我们，不一定要把所有的数据都拿去训练。可以随机划分为几组后进行训练。然后挑取正确率最高的一组。去用测试集进行检测。从而提高分类器的效果。

实验 3：SVM

这个实验的数据预处理是从同学“张一铭”那里拿到的。

SVM 做为一个非常“人工”的机器学习方法，在很多问题上可以得到很好的效果。它有很多的运用，主要适用于去学习“多项式模型”，两层“感知机模型”。通过非线性变换（kernel）后，得到一个线性可分的数据。

我们在课堂上学到的是一种非常初级的模型。



希望找到最优分类面

$$r=\frac{g(x^{(s)})}{\|w_0\|}=\begin{cases}\dfrac{1}{\|w_0\|} & if \quad y^{(s)}=+1\\ -\dfrac{1}{\|w_0\|} & if \quad y^{(s)}=-1\end{cases}$$

即：我们希望找到一个最佳分类面使得 r 达到最大，我就就需要最小化 w, 再加上约束

$$y_i(w^T x_i+b)\geq 1 \qquad for \qquad i=1,2,\cdots,N$$

利用拉格朗日乘子法便可以得到想要的结果。但是 SVM 存在的一个较为严重的问题就是“过拟合”，我们可以看到，当我们拿我们的“训练集”去测试的时候，正确率惊人地达到了 100%。我们看到在最后的结果中，“训练集”的正确率反而很低。

比较其他实验，这很有可能是过拟合的结果。训练数据难免存在误差。

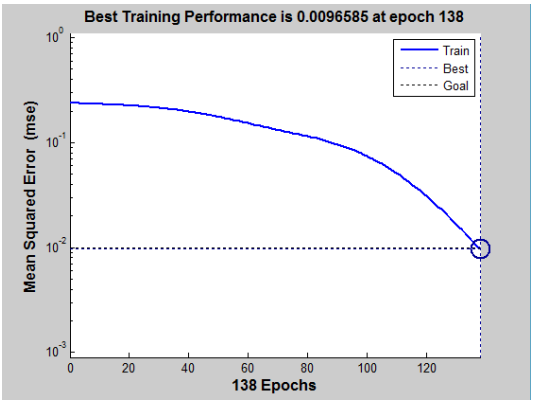
因此在很多线性 SVM 中还引入了“松弛因子”的概念，也叫做“soft margin classification”通过最小化这个函数，来解决“过拟合”的问题。

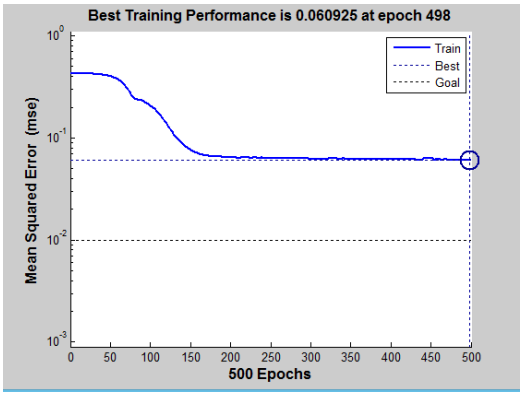
$$\frac{1}{2}\mathbf{w}.\mathbf{w}+C\sum_{k=1}^{R}\xi_k$$

在未来实际的应用中，我们也应该时刻考虑到这一点。

实验 4：MLP

这个网络我在《人工神经网络》这门课上已经搭过一次。由于时间比较紧，这次直接采用了同学的代码。正确率前面已经说过，现在对结果进行一个简要的探讨。





20 组数据 –全部特征-测试集　　全部特征-全部数据-测试集

根据这两组线，我们可以看到，特征更多时，它收敛的更快。但是为什么“特征变多”，但是正确率会更低。这要考虑到它的步长，某一个步长对应调整的是一大组特征，如果后面用的是“欧拉距离”作为 cost 函数的话，某一两个差距比较大的特征就可能对权值矩阵造成比较大的影响。这也告诉我们，一味地增加数据的维数是不可取的。

MLP 有一条结论，当数据维数较低时，足够多的中间隐层可以足够好的拟合出数据的结果。在这里得到了印证，当数据的维数较低时，的确我们看到最后的错误率较小。

当然，另外一个原因可能是该函数陷入了局部极小值。这个有两个方法解决，1.“增大步长”或将步长调整为动态。2. 在权值调整过程中加入其他约束因素。例如 weight-decay factor，这些因素都有利于最后较高的正确率。

对比以上这四种模型：前两种受样本量和特征的影响较小。后两种较大。这主要是由于他们的本质不同，在每个例子中已经详细分析过。前两种找的是“均值”“阈值”，而后两种类似于“拟合”“回归”。因此他们出现了不同的性质。

程序说明：

由于 Matlab 是一种脚本语言。因此我在测试的时候，经常会先将数据预处理好或“读入”，放在 workspace 中，然后函数直接调用，所以可能操作上有一些困难。如果我哪里没有说清楚，请助教直接问我~！谢谢！

LAB1

数据预处理：

```
A = importdata('dataset3.txt');
 datatrans
A = importdata('dataset4.txt');
 testtrans
```

（目的是得到 label ， dataset (总的数据集)，womenset(女性) ，menset(男性)，testdata(训练数据)，testlable(训练标签)）

然后进入 Bayes 函数中，在上面选择需要的“数据类数”（2 或 10）提交的作业中，此时测试的情况为：全部数据+两个特征。

1.如果要切换为：随机数据+两个特征。则将

```
%  nw = randint(1,10,[1,size(womenset,1)]);
%  nm = randint(1,10,[1,size(menset,1)]);
%  nlabel = label(([nw,nm+size(womenset,1)]))
```

前面注释去掉，同时改变 D{1},D{2}的取值范围，同时将 nlabel 注释掉。

2.如果要切换为全部数据+全部特征，则改变 demn 为 10，并将代码中出现的所有 3:2:5 变成 ：即可

3. 如果要切换为随机数据+两个特征，则同时进行 1 和 2 步骤。

后面的主要流程之前讲过。先把训练集送到 findpara 中得到均值与方差。然后根据这套均值与方差对训练集和测试集进行检测。其中 f(I) 是归一化因子，p(i)是先验概率。

```
ans =

    45   156
    55   170
    53   171
    50   165
    45   168
    53   158
    53   166
    56   173
    53   172
    46   164


ans =

    59   171
    52   165
    62   175
    65   181
    63   173
    60   175
    75   183
    80   170
    65   176
    65   170
```

0.884/0.95 下图为 0.811/1

```
ans =

  256.0000   26.0100   51.0000   19.9200  160.0000  171.0000   40.0000  173.5300  166.6700  166.9800
  289.0000   42.2500   65.0000   22.4900  170.0000  167.0000   65.0000  178.4200  166.7100  154.9700
  249.6400   27.0400   52.0000   20.8300  158.0000  158.0000   70.0000  161.4700  179.2300  149.9800
  265.6900   22.0900   47.0000   17.6900  163.0000  173.0000   50.0000  173.9800  175.4400  192.2100
  249.6400   20.2500   45.0000   18.0300  158.0000  180.0000   85.0000  165.8400  168.2700  157.3200
  262.4400   24.0100   49.0000   18.6700  162.0000  175.0000   57.0000  182.3100  176.5200  201.2100
  256.0000   25.0000   50.0000   19.5300  160.0000  162.0000   59.0000  175.1400  169.3900  181.9400
  289.0000   31.3600   56.0000   19.3800  170.0000  174.0000   50.0000  170.0500  162.6100  179.3400
  256.0000   22.0900   47.0000   18.3600  160.0000  160.0000   49.0000  180.2600  175.9000  197.1300
  299.2900   43.5600   66.0000   22.0500  173.0000  180.0000   62.0000  185.0200  171.5100  172.1000


ans =

  331.2400   56.2500   75.0000   22.6400  182.0000  173.0000   54.0000  157.6100  168.9000  154.4600
  316.8400   38.4400   62.0000   19.5700  178.0000  183.0000   50.0000  162.0600  166.6600  179.4900
  324.0000   56.2500   75.0000   23.1500  180.0000  181.0000   49.0000  179.4800  162.1600  184.2600
  309.7600   49.0000   70.0000   22.6000  176.0000  182.0000   52.0000  164.6100  170.9500  174.0100
  295.8400   51.8400   72.0000   24.3400  172.0000  178.0000   78.0000  154.6100  178.3000  164.2500
  316.8400   49.0000   70.0000   22.0900  178.0000  165.0000   57.5000  171.2200  169.8100  172.2300
  309.7600   42.2500   65.0000   20.9800  176.0000  170.0000   53.0000  194.2000  173.6400  153.0500
  289.0000   51.8400   72.0000   24.9100  170.0000  175.0000   63.0000  169.3700  170.3000  187.0500
  289.0000   46.2400   68.0000   23.5300  170.0000  177.0000   72.0000  148.2100  172.1300  159.5300
  299.2900   49.0000   70.0000   23.3900  173.0000  180.0000   59.0000  168.5200  169.5000  161.9100
```

LAB2

数据预处理与 LAB1 相同，因此我把他们放在了一个文件夹下。

Fisher 函数，计算权值矩阵的步骤很清晰，cen_av_wom 代表减去均值后的矩阵，w 是计算得到的权值矩阵。接下来两个循环时用来画投影后的散点图的。后面两个循环各自分别计算了训练集和测试集“男生正确率”，“女生正确率”，“总的正确率”。

Fisher2 函数，计算全部样本，3、5 特征正确率。

Fisher3 函数，计算 20 样本，3、5 特征正确率

将 Fisher3 中的 3:2:5 换成:，则可以计算 20 样本，全部特征正确率。

```
womenset3 =

    52   162
    55   160
    50   163
    47   162
    50   165
    50   161
    51   166
    63   171
    48   163
    53   165


menset3 =

    65   178
    60   178
    65   176
    78   184
    70   178
    75   175
    64   171
    58   170
    72   178
    68   163
```

左图正确率 0.9/0.878，下图正确率 1/0.72

```
womenset3 =

  259.2100   28.0900   53.0000   20.4500  161.0000  168.0000   45.0000  159.8600  178.9000  185.8100
  278.8900   25.0000   50.0000   17.9300  167.0000  176.0000   48.0000  164.8500  155.0300  186.3400
  265.6900   30.2500   55.0000   20.7000  163.0000  177.0000   60.0000  175.9700  174.8500  166.6100
  262.4400   30.2500   55.0000   20.9600  162.0000  182.0000   60.0000  171.4800  171.6500  156.2800
  246.4900   27.0400   52.0000   21.1000  157.0000  180.0000   52.0000  182.3300  167.5300  189.7200
  272.2500   32.4900   57.0000   20.9400  165.0000  164.0000   85.0000  169.7100  176.1300  148.8500
  272.2500   32.4900   57.0000   20.9400  165.0000  164.0000   85.0000  169.7100  176.1300  148.8500
  272.2500   23.0400   48.0000   17.6300  165.0000  180.0000   89.0000  181.7800  175.3600  156.0000
  243.3600   16.0000   40.0000   16.4400  156.0000  165.0000   62.0000  175.2200  164.7900  180.4000
  252.8100   20.2500   45.0000   17.8000  159.0000  184.0000   62.0000  159.5200  165.5700  150.2600


menset3 =

  292.4100   25.0000   50.0000   17.1000  171.0000  165.0000   49.0000  161.3200  170.2400  202.2700
  327.6100   49.0000   70.0000   21.3700  181.0000  195.0000   46.0000  190.4000  169.1100  165.4500
  327.6100   42.2500   65.0000   19.8400  181.0000  173.0000   57.0000  157.0700  171.0500  171.1800
  316.8400   44.8900   67.0000   21.1500  178.0000  179.0000   65.0000  183.0500  175.9400  153.1500
  331.2400   60.8400   78.0000   23.5500  182.0000  160.0000   80.0000  167.3000  167.3100  148.0100
  289.0000   33.6400   58.0000   20.0700  170.0000  170.0000   47.0000  174.9300  168.1900  187.0800
  316.8400   36.0000   60.0000   18.9400  178.0000  173.0000   70.0000  172.2300  165.1300  160.9800
  338.5600   49.0000   70.0000   20.6800  184.0000  158.0000   73.0000  165.0800  171.4800  162.4000
  285.6100   46.2400   68.0000   23.8100  169.0000  168.0000   60.0000  158.4200  176.6100  169.3800
  285.6100   39.6900   63.0000   22.0600  169.0000  159.0000   53.0000  168.2400  172.3200  154.5000
```

Lab3

SVM 使用了现有的包。http://blog.csdn.net/lqhbupt/article/details/8596349
（台湾大学：林智仁）
我的代码：

```
[label,data]=libsvmread('heart_scale');
model = svmtrain(label,data, '-t 0');
[predict_label, accuracy, dec_values] =svmpredict(label, data, model); % test the
trainingdata
```

svmtrain 函数相关参数说明

svmtrain 函数返回的 model 可以用来对测试数据集进行预测。这是一个结构体变量，主要包括了以下几个域。[Parameters, nr_class, totalSV, rho, Label, ProbA, ProbB, nSV,sv_coef, SVs]。

-Parameters: parameters
-nr_class: number of classes; = 2 for regression/one-class svm
-totalSV: total #SV
-rho: -b of the decision function(s) wx+b
-Label: label of each class; empty for regression/one-class SVM
-ProbA: pairwise probability information; empty if -b 0 or in one-classSVM
-ProbB: pairwise probability information; empty if -b 0 or in one-classSVM
-nSV: number of SVs for each class; empty for regression/one-class SVM
-sv_coef: coefficients for SVs in decision functions
-SVs: support vectors

如果没有指定'-b 1'选项则 ProbA 和 ProbB 为空矩阵。此外，当指定'-v'选项时，返回的 model 是一个数值，是 cross-validation 的准确率。

其中 model.paramter 是一个 5X1 的向量，参数意义为：

model.Parameters 参数意义从上到下依次为：

-s svm 类型：SVM 设置类型(默认 0)

-t 核函数类型：核函数设置类型(默认 2)

-d degree：核函数中的 degree 设置(针对多项式核函数)(默认 3)

-g r(gama)：核函数中的 gamma 函数设置(针对多项式/rbf/sigmoid 核函数) (默认类别数目的倒数)

-r coef0：核函数中的 coef0 设置(针对多项式/sigmoid 核函数)((默认 0)

7）svmpredict 函数参数说明

svmpredict 函数返回三个值，predict_label，是训练集预测得到的 label 向量。第二个输出是 accuracy，是一个 3 维的向量，从上到下分别是：分类准率（分类问题中用到的参数指标）；平均平方误差（MSE (mean squared error)） （回归问题中用到的参数指标）；平方相关系数（r2 (squared correlation coefficient)）（回归问题中用到的参数指标）。第三个输出是个矩阵，包含着决策值或者是概率估计（当'-b 1'被指定时）。当训练数据有 k 类时，决策值矩阵是一个 n 行 k*(k-1)/2 列的矩阵（n 为测试数据集个数，k 为类别数），而每一行的输出是 k*(k-1)/2 个二分类器的结果。当'-b 1'被指定时，概率估计矩阵是一个 n 行 k 类的矩阵（n 为测试数据集个数，k 为类别数），每一行的输出是该测试数据属于每个类的概率。

Lab4

使用同班同学“张一铭”的代码。

[pn,minp,maxp]=premnmx(p) 将矩阵归一一化的函数（即把各元素归化到［－1，1］）

输入入参数：

p：作归一一化的矩阵

输出参数：

pn：归一一化之后的矩阵

minp，maxp：矩阵的每一一行的最小值、最大值

net=newff(A,B,{C},’trainFun’); 创建前馈网络的函数

输入入参数：

A：一一个大小为n乘以2的矩阵，每行行的两个元素分别为每维特征值的最小值和最大值

B：k维行向量，其元素为网络中各层节点数

C：k维字符串向量，每一一分量为对应层神经元的激活函数

本次实验中采用一一个隐层的神经冗网网络，隐层和输出层的激活函数都为Sigmoid函数，符号为’logsig’

‘trainFun’：学习规则采用用的训练算法

本次实验中选取算法为’traingdx’，即梯度下降自自适应训练函数（能够自自动调整BP学习的步长）

输出参数：

net：一一个新创建的神经网络，有如下几个参数需要设置：

net.trainparam.goal:神经网络训练误差目标

net.trainparam.show: 显示中间结果的周期

net.trainparam.epochs:最大迭代次数

net.trainParam.lr:学习率

[net,tr,Y1,E]=train(net,X,Y); 训练函数

输入入参数：

net：训练前的网络

X：网络实际输入入

Y：网络目标输出

输出参数：

net：训练后的网络

tr：训练跟踪信息
Y1:网络实际输出
E：误差矩阵
pn=tramnx(p,min,max)；归一一化函数
输入入参数：
p：待归一一的参数
min：一一个最小小值矩阵（用用来计算归一一化）
max：一一个最大大值矩阵（用用来计算归一一化）
输出参数：
pn：作归一一处理后的矩阵p
Y=sim(net,X) 测试分类函数
输入入参数：
net：神经冗网网络
X：实际输入入矩阵，行行数为输入入层节点个数，列数为样本数，即一一列为一一个输入入
Y：输出矩阵，为列向量，本次实验中为行行数为 1 的一一个输出值