模式识别-作业3

贾成君 2014011552 自46

1. **证：**

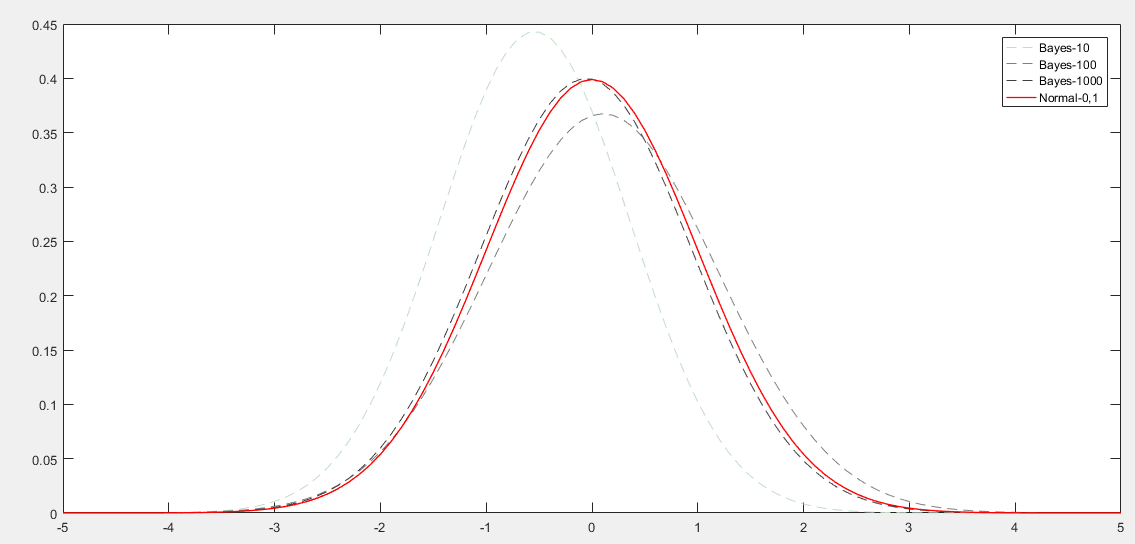


显然，由二次函数的性质可知当时，R最小；证毕。

1. **解：**

最大似然估计使用的公式为 （注意，与MATLAB内置的拟合函数fitdist使用的公式不同——对于方差，MATLAB使用的是无偏估计）。

1. 对符合正态分布的样本绘制出估计的图形如下：

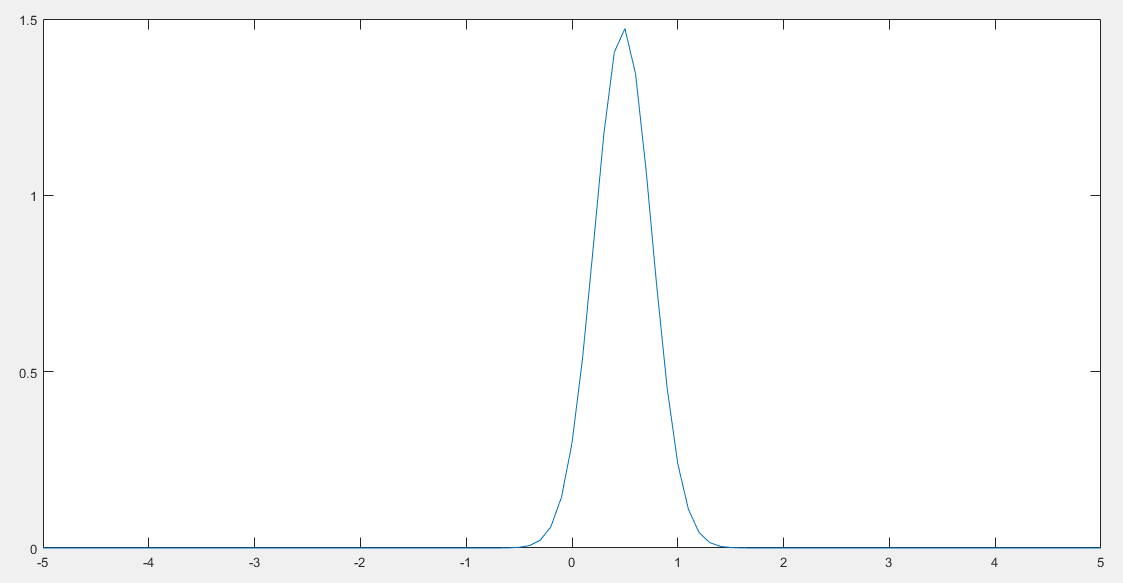


估计的结果为

|  |  |  |  |
| --- | --- | --- | --- |
| 样本数量 | 10 | 100 | 1000 |
| 均值 | -0.5418 | 0.1042 | -0.0550 |
| 标准差 | 0.9004 | 1.0858 | 0.9977 |

可见，使用的样本越多，估计出的结果越接近真实值。

1. 对符合均匀分布的样本估计出的概率密度分布如下：

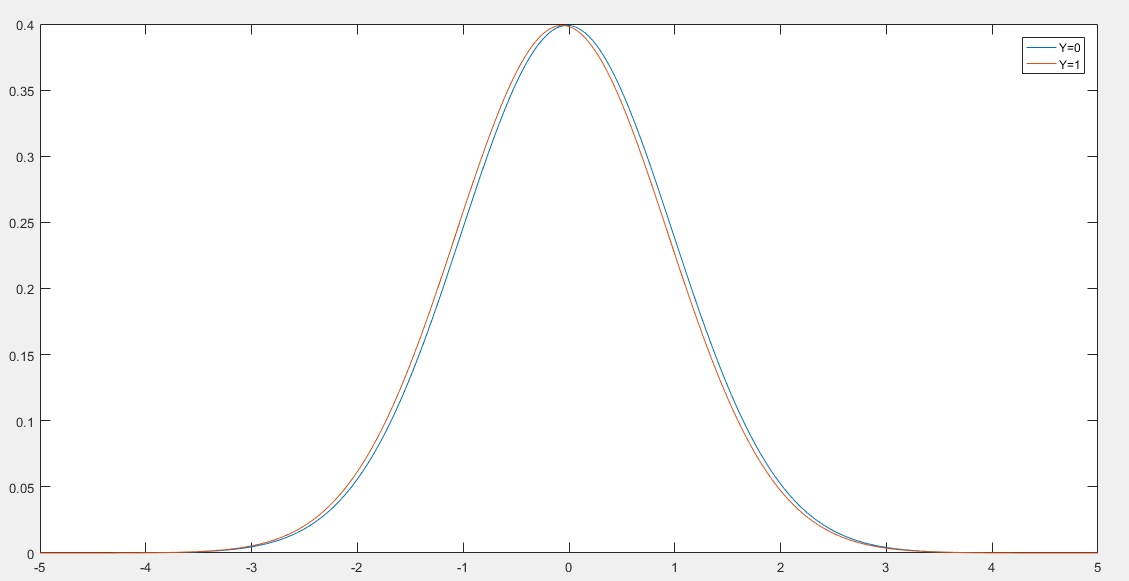


此时对应的均值和方差为0.4840，0.2704。

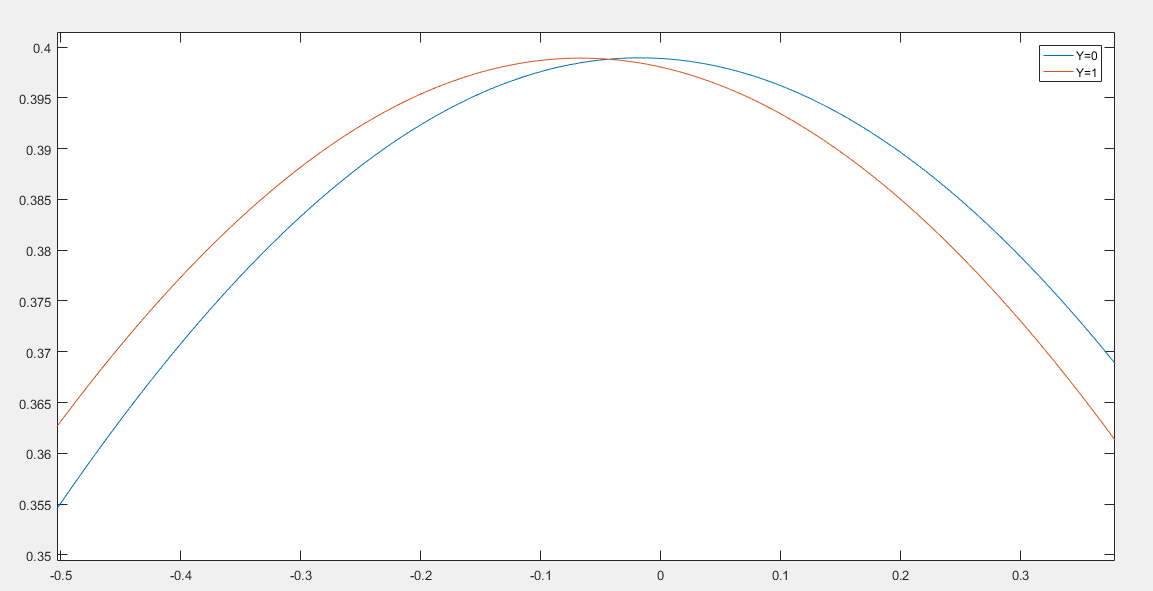
1. 讨论参数估计受到的影响

可以看出：样本量越大，使用最大似然估计得到的参数越准确；如果模型选择不当，计算出的参数所形成的概率密度函数虽然能够一定程度上近似，但是近似的效果并不是很理想。

1. **解：**
2. 使用高斯窗对训练样本进行非参数估计（Y=0为良性，Y=1为恶性），结果如下：



局部放大后，结果为：



可见，两者的概率密度非常接近。

1. 使用最小错误率的贝叶斯决策，需要知道两类样本出现的概率（先验概率），这里有两种估计方式：从已知条件中不知道其概率，假设两类样本等概率；从训练样本中估计两类样本的先验概率。计算出的结果分别如下：

|  |  |  |
| --- | --- | --- |
| 先验概率 | P(w\_0)=P(w\_1)=0.5 | 样本估计P(w\_0),P(w\_1) |
| 测试样本数 | 210 | 210 |
| 错误数 | 13 | 77 |
| 假阳性(误认为恶性) | 4 | 0 |
| 错误率 | 6.19% | 36.67% |

1. 使用最小风险决策，计算过程同（2），则

|  |  |  |
| --- | --- | --- |
| 先验概率 | P(w\_0)=P(w\_1)=0.5 | 样本估计P(w\_0),P(w\_1) |
| 测试样本数 | 210 | 210 |
| 错误数 | 133 | 133 |
| 假阳性(误认为恶性) | 133 | 133 |
| 错误率 | 63.33% | 63.33% |

而p(w\_0)使用样本进行估计，得到的先验概率为66.46%，与错误率相同，我们可以认为：该估计把所有样本都估计为恶性肿瘤。

1. 我们把数据结果统计如下：

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 先验概率 | 相等 | | 样本估计 | |
| 贝叶斯决策 | 最小错误率 | 最小风险 | 最小错误率 | 最小风险 |
| 错误数 | 13,5,12 | 133,143,124 | 77,67,86 | 133,143,124 |
| 假阳性 | 4,2,2 | 133,143,124 | 0,0,0 | 133,143,124 |

显然可见，使用最小风险进行预测，错误率上升，但是假阴性错误率迅速下降——在该模型中，错误率下降为0。可见最小风险的预测结果会使得结果可能偏向减少某一类错误（以增加另一类错误为代价）。

**附录：代码也可以下载于https://github.com/ChengjunJia/PatternRecognition**

第2题代码：

|  |
| --- |
| figure**;**  **[**mu10**,**sigma10**]** **=** plotpdf**(**'Normal'**,**10**);**  hold on**;**  **[**mu100**,**sigma100**]** **=** plotpdf**(**'Normal'**,**100**);**  **[**mu1000**,**sigma1000**]** **=** **plotpdf(**'Normal'**,**1000**);**  plotnormal**(**0**,**1**,**1**);**  legend**(**'Bayes-10'**,**'Bayes-100'**,**'Bayes-1000'**,**'Normal-0,1'**);**  Figure**,**  **[**MuU, sigmaU**]** **=** plotpdf**(**'Uniform'**,**100**);**  **function** **[**mu**,**sigma**]** **=** plotpdf**(**RandModule**,**size**)**  variable1 **=** 0**;**variable2 **=** 1**;**  X **=** random**(**RandModule**,**variable1**,**variable2**,[**size**,**1**]);**  **[**mu sigma**]** **=** BayesianEstimationNormalDistribute**(**X**);**  plotnormal**(**mu**,**sigma**,**0.5**);**  **end**  **function** **[]** **=** plotnormal**(**mu**,**sigma**,**linewidth**)**  pd **=** makedist**(**'Normal'**,**'mu'**,**mu**,**'sigma'**,**sigma**);**  x\_values **=** **-**5**:**0.1**:**5**;**  y\_values **=** pdf**(**pd**,**x\_values**);**  plot**(**x\_values**,**y\_values**,**'LineWidth'**,**linewidth**);**  **end**  **function** **[**mu**,**sigma**]** **=** BayesianEstimationNormalDistribute**(** X **)**  **if** size**(**X**,**2**)** **~=** 1  disp**(**'Matrix Size is Worry! Should be n\*1'**);**  **return;**  **end**  mu **=** mean**(**X**);**  sigma\_2 **=** var**(**X**)/(**size**(**X**,**1**)-**1**)\***size**(**X**,**1**);**  sigma **=** sqrt**(**sigma\_2**);**  **end** |

第3题代码：

|  |
| --- |
| %HW3  allerr **=** zeros**(**50**,**4**);**  load**(**'Breast\_Cancer\_Wisconsin\_data.txt'**);**  InData **=** Breast\_Cancer\_Wisconsin\_data**;**  XAll **=** InData**(:,**2**:**10**);**  YAll **=** **(**InData**(:,**11**)-**2**)** **./**2**;**%0:Benign/gentle,1:Malignant  nAll **=** size**(**XAll**,**1**);**  nTest **=** uint32**(**0.7**\***nAll**);**  Choice **=** randperm**(**nAll**);**  XTrain **=** XAll**(**Choice**(**1**:**nTest**),:);**  YTrain **=** YAll**(**Choice**(**1**:**nTest**),:);**  XTest **=** XAll**(**Choice**(** **(**nTest**+**1**):**nAll**),:);**  YTest **=** YAll**(**Choice**(** **(**nTest**+**1**):**nAll**),:);**  %% Fisher  Fisherw **=** Fisher**(**XTrain**,**YTrain**);**  XFisherTrain **=** XTrain**\***Fisherw**;**  XSamplew0 **=** XFisherTrain**(** find**(**YTrain**==**0**)** **);**%Samples belong to Y=0  XSamplew1 **=** XFisherTrain**(** find**(**YTrain**==**1**)** **);**%Samples belong to Y=1  P0 **=** size**(**XSamplew0**)/**size**(**XFisherTrain**);**  P1 **=** size**(**XSamplew1**)/**size**(**XFisherTrain**);**  XFisherTest **=** XTest**\***Fisherw**;**  PFisherTestw0 **=** XFisherTest**;**  PFisherTestw1 **=** XFisherTest**;**  **for** m **=** 1**:**size**(**XFisherTest**)**  PFisherTestw0**(**m**)** **=** KernalP**(**XFisherTest**(**m**),**XSamplew0**);**  PFisherTestw1**(**m**)** **=** KernalP**(**XFisherTest**(**m**),**XSamplew1**);**  **end**  disp**([**'Test Sample:'**,**num2str**(**size**(**YTest**,**1**))** **]),**  YMW **=** log**(** PFisherTestw0 **./** PFisherTestw1 **)** **<** log**(**1**);**  disp**(**'P(w\_0)=P(w\_1):'**),**  disp**(** **[**' All:'**,** num2str**(**sum**(**abs**(**YMW**-**YTest**)))** **]** **),**  disp**(** **[**' False Postive:'**,** num2str**(**sum**(**YMW**-**YTest**>**0**))** **]** **),**  YMW2 **=** log**(** PFisherTestw0 **./** PFisherTestw1 **)** **<** log**(**size**(**XSamplew1**,**1**)/**size**(**XSamplew0**,**1**));**  disp**(**'P(w\_0),P(w\_1) From the Sample:'**),**  disp**(** **[**' All:'**,** num2str**(**sum**(**abs**(**YMW2**-**YTest**))** **)]** **),**  disp**(** **[**' False Postive:'**,** num2str**(**sum**(**YMW2**-**YTest**>**0**))** **]** **),**  YLL **=** log**(** PFisherTestw0 **./** PFisherTestw1 **)** **<** log**(**10**/**1**);**  disp**(**'P(w\_0)=P(w\_1),Least Loss:'**),**  disp**(** **[**' All:'**,** num2str**(**sum**(**abs**(**YLL**-**YTest**)))** **]** **),**  disp**(** **[**' False Postive:'**,** num2str**(**sum**(**YLL**-**YTest**>**0**))** **]** **)**  YLL2 **=** log**(** PFisherTestw0 **./** PFisherTestw1 **)** **<** log**(**10**/**1**\***size**(**XSamplew1**,**1**)/**size**(**XSamplew0**,**1**));**  disp**(**'P(w\_0),P(w\_1)From the Sample,Least Loss:'**),**  disp**(** **[**' All:'**,** num2str**(**sum**(**abs**(**YLL2**-**YTest**)))** **]** **),**  disp**(** **[**' False Postive:'**,** num2str**(**sum**(**YLL2**-**YTest**>**0**))** **]** **),**  %Calculate the plot of two lines(possibilty of the x|w0)  test **=** **-**5**:**0.001**:**5**;**  py0 **=** test**;** py1 **=** test**;**  **for** m **=** 1**:**size**(**test**,**2**)**  py0**(**m**)** **=** KernalP**(**test**(**m**),**XSamplew0**);**  py1**(**m**)** **=** KernalP**(**test**(**m**),**XSamplew1**);**  **end**  figure**,**plot**(**test**,**py0**),**  hold on**;**plot**(**test**,**py1**),**  legend**(**'Y=0'**,**'Y=1'**);**  %Calculate the plot of two lines(possibilty of the x|w0)  test **=** **-**1**:**0.0001**:**1**;**  py0 **=** test**;** py1 **=** test**;**  **for** m **=** 1**:**size**(**test**,**2**)**  py0**(**m**)** **=** KernalP**(**test**(**m**),**XSamplew0**);**  py1**(**m**)** **=** KernalP**(**test**(**m**),**XSamplew1**);**  **end**  figure**,**plot**(**test**,**py0**),**  hold on**;**plot**(**test**,**py1**),**  legend**(**'Y=0'**,**'Y=1'**);**  %%Kernal Density  **function** **[**poss**]** **=** KernalP**(**x**,**XSample**)**  sigma **=** 1**;**  poss **=** mean**(** 1**/(**sqrt**(**2**\***pi**)\***sigma**)\***exp**(-(**x**-**XSample**)** **.^**2**/(**2**\***sigma**^**2**))** **);**  **end**  %% Fisher Function to Calculate the w (with inv(S\_w) )  **function** **[**w**]** **=** Fisher**(**xT**,**yT**)**  % ybar=w'\*x+w0, ybar>0 ==> x<-w1,y=0  %  logic0 **=** **(**yT **==** 0 **);**  logic1 **=** **(**yT **==** 1 **);**  x0 **=** xT**(**logic0**,:);**  x1 **=** xT**(**logic1**,:);**  m0 **=** mean**(**x0**);**  m1 **=** mean**(**x1**);**  nFeature **=** size**(**m0**',**1**);**  Sw **=** zeros**(**nFeature**,**nFeature**);**  **for** m **=** 1**:**size**(**x0**,**1**)**  temp **=** x0**(**m**,:)** **-**m0**;**  Sw **=** Sw**+**temp**'** **\*** temp**;**  **end**  **for** m **=** 1**:**size**(**x1**,**1**)**  temp **=** x1**(**m**,:)-**m1**;**  Sw **=** Sw**+**temp**'** **\*** temp**;**  **end**  w **=** Sw**\(**m0**-**m1**)';**  **end** |