暑期学校课程

《机器学习与分类算法》

夏静波
华中农业大学信息学院
http://xiajingbo.weebly.com

——本课件由朱强老师和程文文提供帮助。

机器学习之算法分类

1.机器学习概念

机器学习 (Machine Learning) 是近 20 多年兴起的一门多领域交叉学科，涉及概率论、统计学、逼近论、凸分析、算法复杂度理论等多门学科。机器学习理论主要是设计和分析一些让计算机可以自动学习的算法，这些算法能从数据中自动分析获得规律，并利用规律对未知数据进行预测。

问： Why machine learning?

机器学习之算法分类

2.算法分类

机器学习的算法繁多，不同的算法有各自的特点且适合的数据集也有区别，我们可以按照不同的角度将其分类。按照学习方式可以划分为监督式学习、非监督式学习、半监督式学习，强化学习。常见的算法包括：贝叶斯分类算法、SVM、决策树学习、随机森林、人工神经网络、K-近邻、深度学习等。

问： Why classification?

机器学习之算法分类

问： Merely classification?

问： What about GWAS (Genome wide associate analysis?)

问： What about Microarray analysis?

问： When classification?

机器学习之算法分类

问： Merely classification?

问： What about GWAS (Genome wide associate analysis?)

问： What about Microarray analysis?

问： When classification?

机器学习之算法分类

问： Merely classification?

问： What about GWAS (Genome wide associate analysis?)

问： What about Microarray analysis?

问： When classification?

机器学习之算法分类

问： Merely classification?

问： What about GWAS (Genome wide associate analysis?)

问： What about Microarray analysis?

问： When classification?

一、朴素贝叶斯分类

1.贝叶斯分类特点

贝叶斯分类算法是统计学的一种分类方法，它是一类利用概率统计知识进行分类的算法。在许多场合，朴素贝叶斯(Naïve Bayes，NB)分类算法可以与决策树和神经网络分类算法相媲美，该算法能运用到大型数据库中，而且方法简单、分类准确率高、速度快。

2.朴素贝叶斯的核心公式

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

3.朴素贝叶斯分类流程图

第一阶段——准备工作阶段，主要工作是根据具体情况确定特征属性，并对每个特征属性进行适当划分，然后由人工对一部分待分类项进行分类，形成训练样本集合。

第二阶段——分类器训练阶段，这个阶段的任务就是生成分类器，主要工作是计算每个类别在训练样本中的出现频率及每个特征属性划分对每个类别的条件概率估计，并将结果记录。

第三阶段——应用阶段。这个阶段的任务是使用分类器对待分类项进行分类，其输入是分类器和待分类项，输出是待分类项与类别的映射关系。

24.2 Naive Bayes

The Naive Bayes classifier is a classical demonstration of how generative assumptions and parameter estimations simplify the learning process. Consider the problem of predicting a label $y \in \{0,1\}$ on the basis of a vector of features $\mathbf{x} = (x_1, \ldots, x_d)$, where we assume that each x_i is in $\{0,1\}$. Recall that the Bayes optimal classifier is

$$h_{\text{Bayes}}(\mathbf{x}) = \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y | X = \mathbf{x}].$$

To describe the probability function $\mathcal{P}[Y = y | X = \mathbf{x}]$ we need 2^d parameters, each of which corresponds to $\mathcal{P}[Y = 1 | X = \mathbf{x}]$ for a certain value of $\mathbf{x} \in \{0,1\}^d$. This implies that the number of examples we need grows exponentially with the number of features.

In the Naive Bayes approach we make the (rather naive) generative assumption that given the label, the features are independent of each other. That is,

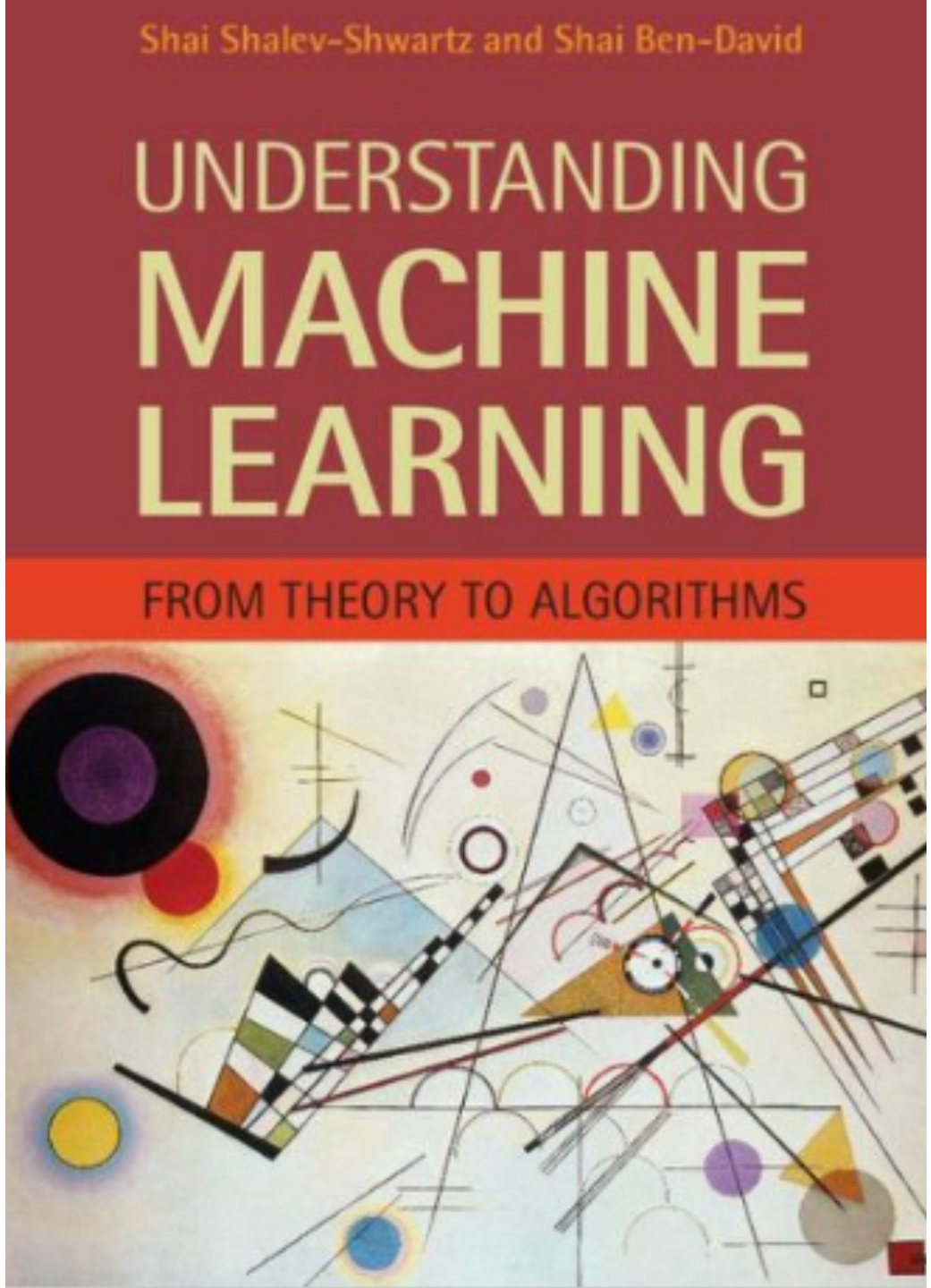
$$\mathcal{P}[X = \mathbf{x} | Y = y] = \prod_{i=1}^{d} \mathcal{P}[X_i = x_i | Y = y].$$

With this assumption and using Bayes' rule, the Bayes optimal classifier can be further simplified:

$$\begin{aligned} h_{\text{Bayes}}(\mathbf{x}) &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y | X = \mathbf{x}] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y] \mathcal{P}[X = \mathbf{x} | Y = y] \\ &= \operatorname*{argmax}_{y \in \{0,1\}} \mathcal{P}[Y = y] \prod_{i=1}^{d} \mathcal{P}[X_i = x_i | Y = y]. \end{aligned} \tag{24.7}$$

That is, now the number of parameters we need to estimate is only $2d + 1$. Here, the generative assumption we made reduced significantly the number of parameters we need to learn.

When we also estimate the parameters using the maximum likelihood principle, the resulting classifier is called the *Naive Bayes* classifier.



问: Is it readable?

问: How about now?

二、支持向量机（SVM）算法

1.SVM特点

支持向量机(support vector machine)是一种分类算法，通过寻求结构化风险最小来提高学习机泛化能力，能够在统计样本量较少的情况下，获得良好的统计规律。它是一种二类分类模型，其基本模型定义为特征空间上的间隔最大的线性分类器，即支持向量机的学习策略便是间隔最大化，最终可转化为一个凸二次规划问题的求解。

2.SVM具体原理：

- 在n维空间中找到一个分类超平面，将空间上的点分类。如下图是线性分类的例子。（图1）
- 一般而言，一个点距离超平面的远近可以表示为分类预测的确信或准确程度。SVM就是要最大化这个间隔值。而在虚线上的点便叫做支持向量Supprot Verctor。（图2、图3）

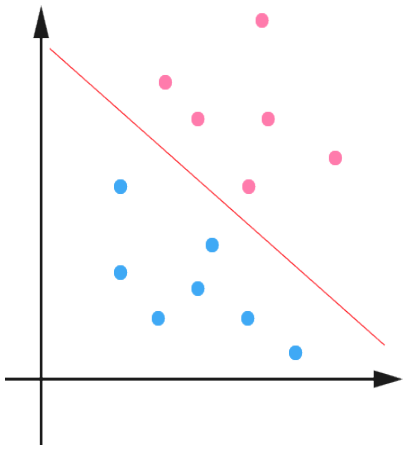


图1

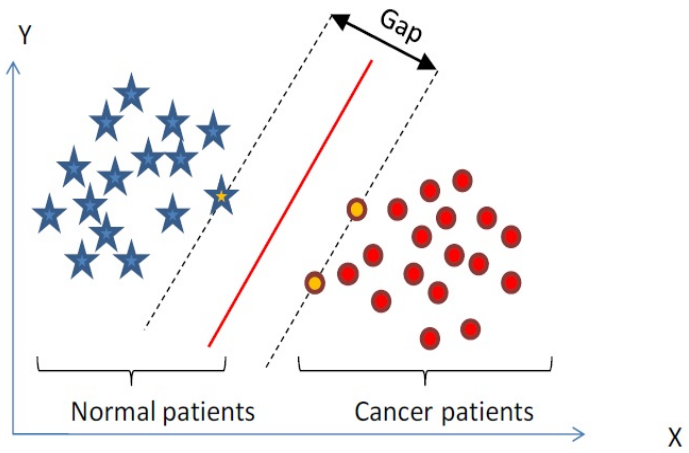


图2

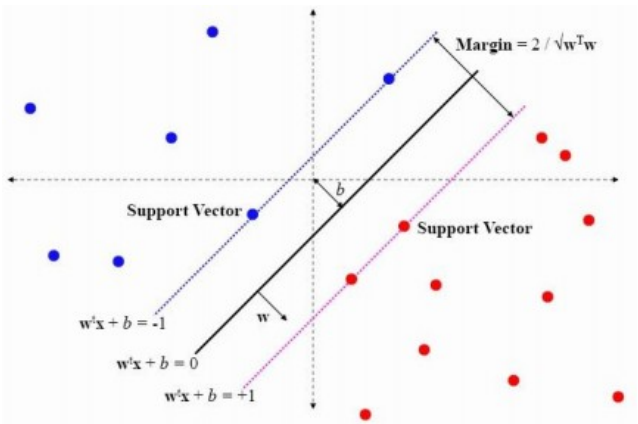


图3

问: Is it high time to go on?

- 实际中，我们会经常遇到线性不可分的样例，此时，我们的常用做法是把样例特征映射到高维空间中去。（图4）

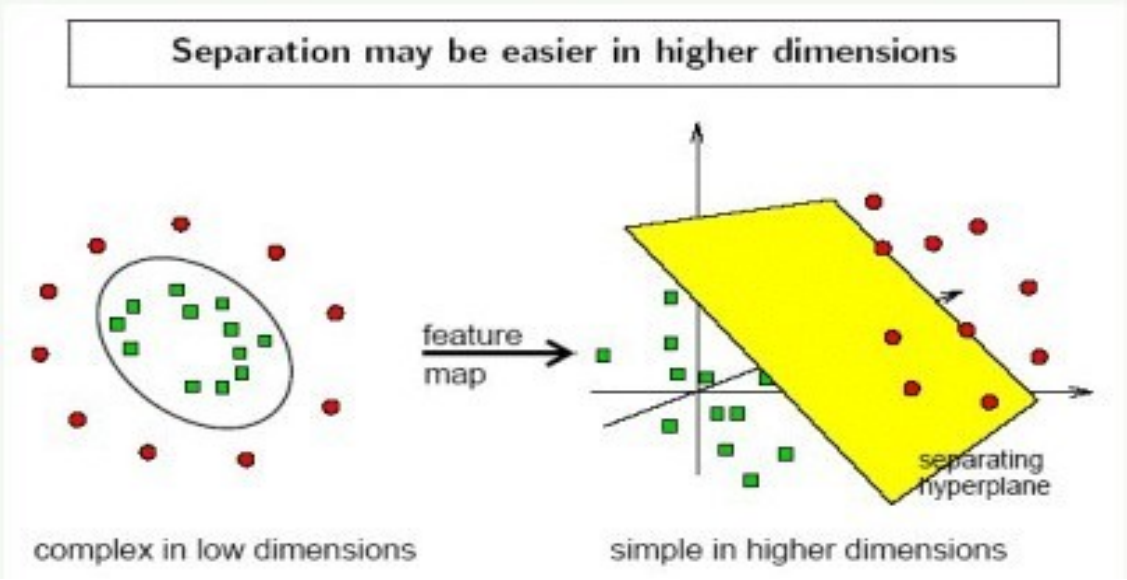


图4

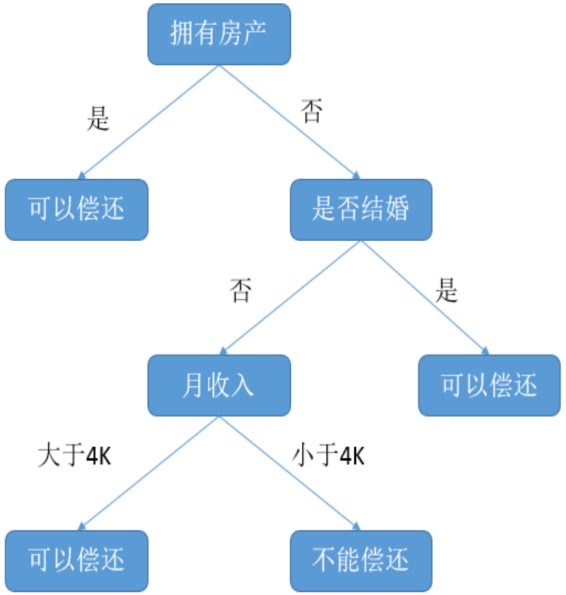
三、决策树学习

1.决策树算法原理

决策树（Decision Tree）是一种简单但是广泛使用的分类器。通过训练数据构建决策树，可以高效的对未知的数据进行分类。决策数有两大优点：1）决策树模型可以读性好，具有描述性，有助于人工分析；2）效率高，决策树只需要一次构建，反复使用，每一次预测的最大计算次数不超过决策树的深度。

2.决策树案例：

右图是一棵结构简单的决策树，用于预测贷款用户是否具有偿还贷款的能力。贷款用户主要具备三个属性：是否拥有房产，是否结婚，平均月收入。每一个内部节点都表示一个属性条件判断，叶子节点表示贷款用户是否具有偿还能力。



3.构建决策树及其剪枝步骤：

- 导入需要的函数库。当然如果本地开发环境没有相应的库的话，还需要通过 install.packages 函数对库进行安装。
- 查看本次构建决策树的数据源。
- 通过 rpart 函数构建决策树，以研究癌复发与病人年龄、肿瘤等级、癌细胞比例，癌细胞分裂状况等之间的关系。
- 查看决策树的具体信息。 绘制构建完成的决策树图。
- 通过 prune 函数对该决策树进行适当的剪枝，防止过拟合，使得树能够较好地反映数据内在的规律并在实际应用中有意义。
- 绘制剪枝完后的决策树图。

四、随机森林(Random Forest)

1.随机森林原理

随机森林是用随机的方式建立一个森林，森林里面有很多的决策树组成，随机森林的每一棵决策树之间是没有关联的。在得到森林之后，当有一个新的输入样本进入的时候，就让森林中的每一棵决策树分别进行一下判断，看看这个样本应该属于哪一类，然后看看哪一类被选择最多，就预测这个样本为那一类。

2.随机森林的优点

- 能够处理很高维度（feature很多）的数据，并且不用做特征选择
- 在训练完后，它能够给出哪些feature比较重要
- 在创建随机森林的时候，对generlization error使用的是无偏估计
- 在训练过程中，能够检测到feature间的互相影响
- 训练速度快、实现比较简单

五、k折交叉验证 k-fold cross validation

1. K折交叉验证基本原理

- 将全部训练集 S分成 k个不相交的子集，假设 S中的训练样例个数为 m，那么每一个子 集有 m/k 个训练样例，，相应的子集称作 {s1,s2,⋯,sk}。
- 每次从分好的子集中里面，拿出一个作为测试集，其它k-1个作为训练集
- 根据训练集训练出模型或者假设函数。
- 把这个模型放到测试集上，得到分类率。
- 计算k次求得的分类率的平均值，作为该模型或者假设函数的真实分类率。

这个方法充分利用了所有样本。但计算比较繁琐，需要训练k次，测试k次。

案例及其程序实现

例题：(Breast Cancer 二分类问题）

- Wisconsin Breast Cancer Database 数据集是收集威斯康森州的699例病人乳腺癌的诊断情况
- 分为两类：恶性（malignant）有241例，良性（benign）有458例；
- 测试了一些相关数据：
  - Sample code number , Clump Thickness , Uniformity of Cell Size , Uniformity of Cell Shape , Marginal Adhesion , Single Epithelial Cell Size, Bare Nuclei, Bland Chromatin, Normal Nucleoli, Mitoses.

##朴素贝叶斯方法

调用以下命令：

```
library(e1071)  #调用支持向量机包e1071
library(mlbench)  #调mlbench包，内含BreastCancer数据集
library(pROC) #auc函数包
data("BreastCancer")
head(BreastCancer)  #显示数据的前几行
dim(BreastCancer)  #数据的维度
BC=BreastCancer[,-1]  #删除第一列ID号
BC=na.omit(BC)  #去除数据中的缺失值
dim(BC)
# naiveBayes朴素贝叶斯函数建立预测模型
baye.mod=naiveBayes(Class~.,BC)
summary(baye.mod)
fit=predict(baye.mod,newdata  = BC[,-10])
table(fit,BC$Class)  #查看预测结果
## fit        benign malignant
    benign      431        3
    malignant    13       236
table(BC$Class)  #BC数据中benign个数为444，malignant个数为239
##benign malignant
   444      239
##朴素贝叶斯预测的准确率ACC=(TP+TN)/n
(431+236)/683 ##0.9765739
head(fit)
index=sample(1:683,floor(683*0.8)) #把数据的80%当做训练集，20%当做测试集
train=BC[index,]  #80%数据作为训练集
test=BC[-index,]  #20%数据作为测试集
dim(train)  #查看训练集维数
dim(test)  #查看测试集维数
mode=naiveBayes(Class~.,data=train)  #朴素贝叶斯建立训练模型
pre=predict(mode,newdata=test[,-10])  #利用模型预测测试集，去掉第十列类别项
table(pre,test$Class)
## pre        benign malignant
  benign       88        0
  malignant     3       46
table(test$Class)  #测试集中有92个benign，56个malignant
 ##benign malignant
       91       46
##预测准确率ACC=(88+46)/(91+46)=0.9781022
pred.train=predict(mode,newdata=train[,-10])#将训练数据集回带查看预测结果
table(pred.train,train$Class)
## pred.train  benign malignant
      benign      343        3
     malignant    10       190
##预测准确率ACC=(343+190)/(343+190+13)= 0.9761905
```

案例及其程序实现

例题：(Breast Cancer 二分类问题)

问: What are TP, TN, FP, FN, ACCU?

		True class	
		p	n
Hypothesized class	Y	True Positives	False Positives
	N	False Negatives	True Negatives
Column totals:		P	N

$$\text{fp rate} = \frac{FP}{N} \qquad \text{tp rate} = \frac{TP}{P}$$

$$\text{precision} = \frac{TP}{TP+FP} \qquad \text{recall} = \frac{TP}{P}$$

$$\text{accuracy} = \frac{TP+TN}{P+N}$$

$$\text{F-measure} = \frac{2}{1/\text{precision}+1/\text{recall}}$$

Fig. 1. Confusion matrix and common performance metrics calculated from it.

案例及其程序实现

例题：(Breast Cancer 二分类问题）

问: Is the result reliable?

→ Cross-validation.

```
##均衡5折验证
d=1:nrow(BC);dd=list();Z=5
nn=levels(BC$Class);KL=length(nn)  #nn表示benign和malignant两种水平；KL结果为nn的长度 2
for(i in 1:KL){ dd[[i]]=d[BC$Class==nn[i]]}
kk=NULL
for(i in 1:KL){kk=c(kk,round(length(dd[[i]])/Z))}
kk   #有89个benign , 48个malignant
#[1] 89 48
yy=list(NULL,NULL,NULL,NULL,NULL)
for(i in 1:KL){xx=list();uu=dd[[i]];
for(j in 1:(Z-1)){xx[[j]]=sample(uu,kk[i])
uu=setdiff(uu,xx[[j]])}
xx[[Z]]=uu
for(k in 1:Z) yy[[i]][[k]]=xx[[k]]}
mm=list(NULL,NULL,NULL,NULL,NULL)
for(i in 1:Z){
  for(j in 1:KL){
    mm[[i]]=c(mm[[i]],yy[[j]][[i]])
  }
}
#构建朴素贝叶斯训练模型
sv.tab=list(NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:5){m=mm[[i]];
baye.mod=naiveBayes(Class~.,BC[-m,])
pred=predict(baye.mod,BC[m,])
prob=c(prob,predict(baye.mod,BC[m,],type="raw")[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)
}
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]])]
 ##计算预测的准确率
AUC=auc(dat.class,prob,levels=c("benign","malignant"))
 tab=sv.tab[[1]]+sv.tab[[2]]+sv.tab[[3]]+sv.tab[[4]]+sv.tab[[5]]
TP=tab[2,2];TN=tab[1,1]
FP=tab[2,1]
FN=tab[1,2]
ACC=(TP+TN)/nrow(BC)  #准确率
Recall=TP/(TP+FN)  #召回率
Specificity=TN/(TN+FP) #特异性或真阴性率
Precision=TP/(TP+FP)  #精度
MCC=(TP*TN-FP*FN)/(sqrt(TP+FN)*sqrt(TN+FP)*sqrt(TP+FP)*sqrt(TN+FN))
```

案例及其程序实现

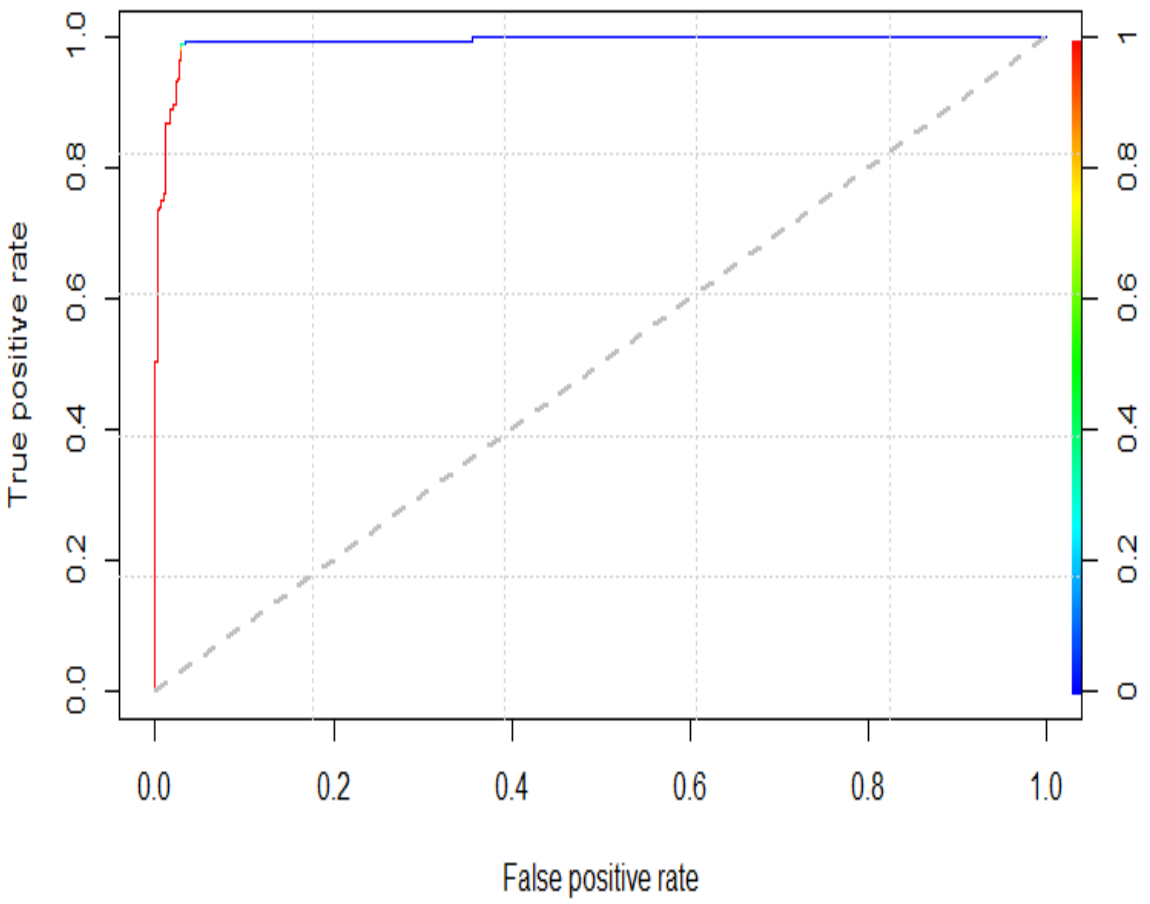
例题：(Breast Cancer 二分类问题）

问: Any other metric for evaluation?

→ ROC and AUC

→ http://alexkong.net/2013/06/introduction-to-auc-and-roc/

```
AUC
#0.9933
ACC
#[1] 0.9751098
Recall
#[1] 0.9832636
Specificity
#[1] 0.9707207
Precision
#[1] 0.9475806
MCC
#[1] 0.9461526
#画ROC曲线
library(gplots)
library(ROCR)
preed=prediction(prob,dat.class)
perf=performance(preed,"auc");as.numeric(perf@y.val
ues)
#[1] 0.9932621
perf=performance(preed,"tpr","fpr")
plot(perf,colorize=T)
grid(5,5,lwd=1)
points(c(0,1),c(0,1),type="l",lty=2,lwd=2,col="grey")
```



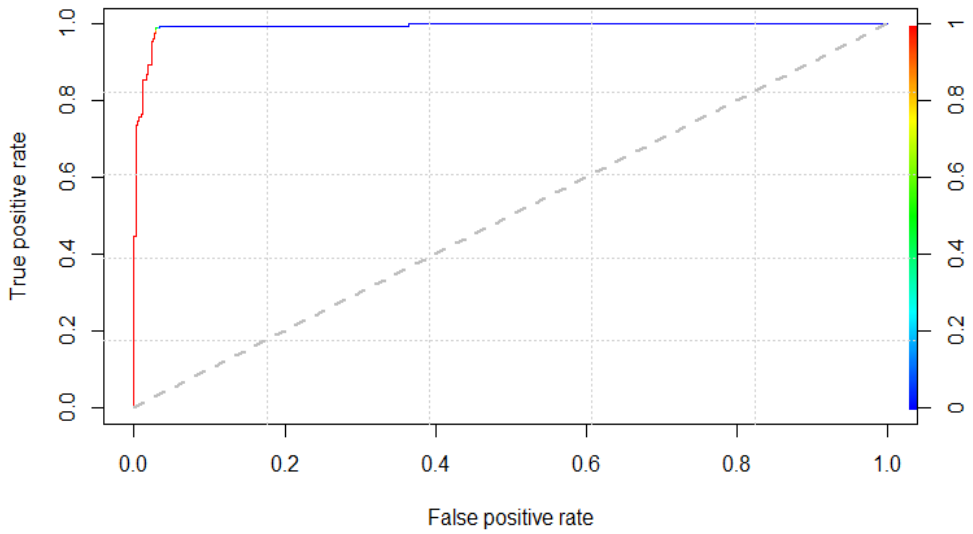
均衡5折贝叶斯ROC曲线

##均衡10折

```
d=1:nrow(BC);dd=list();Z=10
nn=levels(BC$Class);KL=length(nn)
for(i in 1:KL){ dd[[i]]=d[BC$Class==nn[i]]}
kk=NULL
for(i in 1:KL){kk=c(kk,round(length(dd[[i]])/Z))}
kk
yy=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)
for(i in 1:KL){xx=list();uu=dd[[i]];
for(j in 1:(Z-1)){xx[[j]]=sample(uu,kk[i])
uu=setdiff(uu,xx[[j]])}
xx[[Z]]=uu
for(k in 1:Z) yy[[i]][[k]]=xx[[k]]}
mm=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)
for(i in 1:Z){
  for(j in 1:KL){
    mm[[i]]=c(mm[[i]],yy[[j]][[i]])}}
sv.tab=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:10){m=mm[[i]];
baye.mod=naiveBayes(Class~.,BC[-m,])
pred=predict(baye.mod,BC[m,])
prob=c(prob,predict(baye.mod,BC[m,],type="raw")[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)}
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]],mm[[6]],mm[[7]],mm[[8]],mm[[9]],mm[[10]])]
```

```
##计算准确率
AUC<-auc(dat.class,prob,levels=c("benign","malignant"))
tab=sv.tab[[1]]+sv.tab[[2]]+sv.tab[[3]]+sv.tab[[4]]+sv.tab[[5]]+sv.tab[[6]]+sv.tab[[7]]+sv.tab[[8]]+sv.tab[[9]]+sv.tab[[10]]
TP=tab[2,2];TN=tab[1,1]
FP=tab[2,1]
FN=tab[1,2]
ACC=(TP+TN)/nrow(BC)
Recall=TP/(TP+FN)
Specificity=TN/(TN+FP)
Precision=TP/(TP+FP)
MCC=(TP*TN-FP*FN)/(sqrt(TP+FN)*sqrt(TN+FP)*sqrt(TP+FP)*sqrt(TN+FN))
AUC  #Area under the curve: 0.9934
ACC #[1] 0.9765739
Recall #[1] 0.9874477
Specificity#[1] 0.9707207
Precision#[1] 0.9477912
MCC#[1] 0.9494842
```

```
#画图
library(gplots)
library(ROCR)
preed=prediction(prob,dat.class)
perf=performance(preed,"auc");as.numeric(perf@y.values)
perf=performance(preed,"tpr","fpr")
plot(perf,colorize=T)
grid(5,5,lwd=1)
points(c(0,1),c(0,1),type="l",lty=2,lwd=2,col="grey")
```



十折划分ROC曲线图

```
##均衡20折
d=1:nrow(BC);dd=list();Z=20
nn=levels(BC$Class);KL=length(nn)
for(i in 1:KL){ dd[[i]]=d[BC$Class==nn[i]]}
kk=NULL
for(i in 1:KL){kk=c(kk,round(length(dd[[i]])/Z))}
kk
yy=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)
for(i in 1:KL){xx=list();uu=dd[[i]];
for(j in 1:(Z-1)){xx[[j]]=sample(uu,kk[i])
uu=setdiff(uu,xx[[j]])}
xx[[Z]]=uu
for(k in 1:Z) yy[[i]][[k]]=xx[[k]]
}
mm=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL)
for(i in 1:Z){
  for(j in 1:KL){
    mm[[i]]=c(mm[[i]],yy[[j]][[i]])
  }
}
```

```
####计算20折预测准确率
sv.tab=list(NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:20){m=mm[[i]];
baye.mod=naiveBayes(Class~.,BC[-m,])
pred=predict(baye.mod,BC[m,])
prob=c(prob,predict(baye.mod,BC[m,],type="raw")[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)}
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]],mm[[6]],mm[[7]],mm[[8]],mm[[9]],mm[[10]],mm[[11]],mm[[12]],mm[[13]],mm[[14]],mm[[15]],mm[[16]],mm[[17]],mm[[18]],mm[[19]],mm[[20]])]
AUC<-auc(dat.class,prob,levels=c("benign","malignant"))
tab=sv.tab[[1]]+sv.tab[[2]]+sv.tab[[3]]+sv.tab[[4]]+sv.tab[[5]]+sv.tab[[6]]+sv.tab[[7]]+sv.tab[[8]]+sv.tab[[9]]+sv.tab[[10]]+sv.tab[[11]]+sv.tab[[12]]+sv.tab[[13]]+sv.tab[[14]]+sv.tab[[15]]+sv.tab[[16]]+sv.tab[[17]]+sv.tab[[18]]+sv.tab[[19]]+sv.tab[[20]]
TP=tab[2,2];TN=tab[1,1]
FP=tab[2,1]
FN=tab[1,2]
ACC=(TP+TN)/nrow(BC)
Recall=TP/(TP+FN)
Specificity=TN/(TN+FP)
Precision=TP/(TP+FP)
MCC=(TP*TN-FP*FN)/(sqrt(TP+FN)*sqrt(TN+FP)*sqrt(TP+FP)*sqrt(TN+FN))
```

AUC

#Area under the curve: 0.9936

ACC

#[1] 0.9751098

Recall

#[1] 0.9832636

Specificity

#[1] 0.9707207

Precision

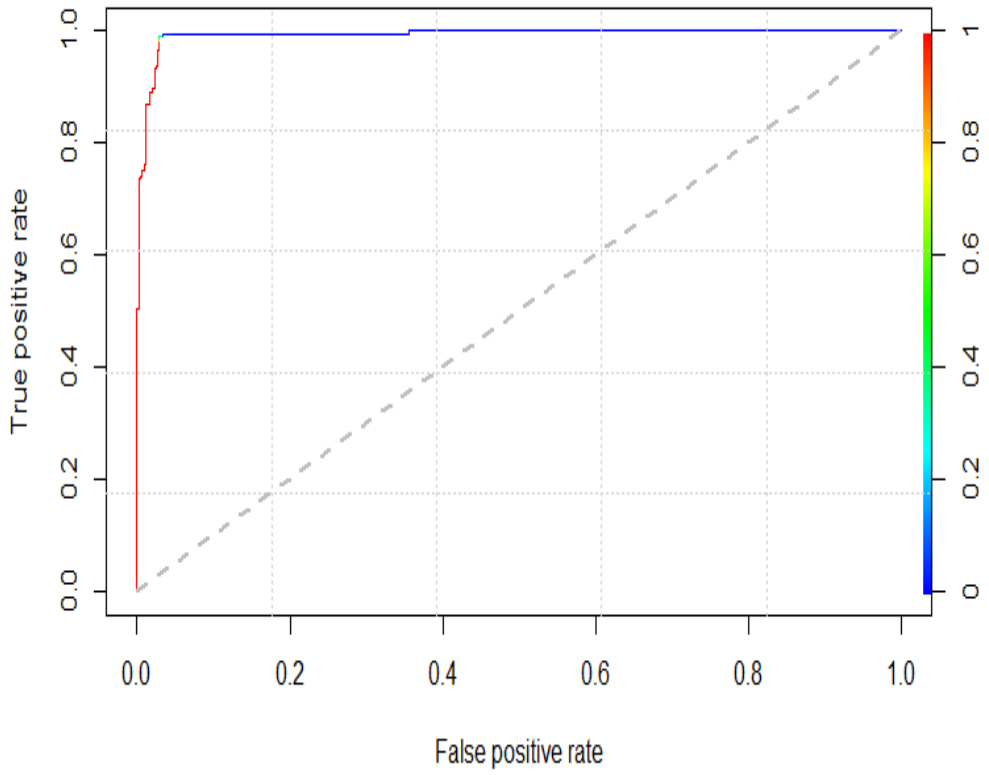
#[1] 0.9475806

MCC

#[1] 0.9461526

##画ROC曲线

```
library(gplots)
library(ROCR)
perf=performance(preed,"auc");as.numeric(perf@y.values)
perf=performance(preed,"tpr","fpr")
plot(perf,colorize=T)
grid(5,5,lwd=1)
points(c(0,1),c(0,1),type="l",lty=2,lwd=2,col="grey")
```



20折ROC曲线

#SVM模型

```
#导入数据
library(pROC)
library(e1071)
library(mlbench)
data("BreastCancer")
BC=BreastCancer[,-1]
BC=na.omit(BC)
dim(BC)
#均衡五折划分 代码见幻灯片11
#构建SVM模型并计算准确率
sv.tab=list(NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:5){m=mm[[i]];
svm.mod=svm(Class~.,gamma=0.25,cost=16,data=BC[-m,],probability=T)
pred=predict(svm.mod,BC[m,],probability = T)
prob=c(prob,attr(pred,"probabilities")[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)
 }
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]])]
AUC=auc(dat.class,prob,levels=c("benign","malignant"))
tab=sv.tab[[1]]+sv.tab[[2]]+sv.tab[[3]]+sv.tab[[4]]+sv.tab[[5]]
TP=tab[2,2];TN=tab[1,1]
FP=tab[2,1]
FN=tab[1,2]
ACC=(TP+TN)/nrow(BC)
Recall=TP/(TP+FN)
Specificity=TN/(TN+FP)
Precision=TP/(TP+FP)
MCC=(TP*TN-
FP*FN)/(sqrt(TP+FN)*sqrt(TN+FP)*sqrt(TP+FP)*sqrt(TN+FN))
AUC
#Area under the curve: 0.9935
ACC
#[1] 0.9751098
Recall
#[1] 0.9748954
Specificity
#[1] 0.9752252
Precision
#[1] 0.954918
MCC
#[1] 0.9456752
#画ROC曲线 代码见幻灯片12
```

```
#决策树C5.0
#加载包并导入数据
install.packages("C50")
library(pROC);library(e1071);library(C50)
data("BreastCancer")
BC=BreastCancer[,-1]
BC=na.omit(BC)
#均衡五折划分 代码见幻灯片11
#C5.0决策树算法训练模型
sv.tab=list(NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:5){m=mm[[i]];
svm.mod=C5.0(Class~.,data=BC[-m,],trials=100)
pred=predict(svm.mod,BC[m,],type="class")
pr=predict(svm.mod,BC[m,],type="prob")
prob=c(prob,pr[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)
}
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]])]
```

```
##计算预测准确率和画图代码见幻灯片11和12
 AUC
#Area under the curve: 0.9926
ACC
#[1] 0.9633968
Recall
#[1] 0.9539749
Specificity
#[1] 0.9684685
Precision
#[1] 0.9421488
MCC
#[1] 0.9198207
#画ROC图像 代码见幻灯片12
```

#随机森林

```
#安装加载包并导入数据
install.packages("randomForest")
library(pROC);library(e1071);library(C50);library(randomForest)
data("BreastCancer")
BC=BreastCancer[,-1]
BC=na.omit(BC)
#均衡五折划分 代码见幻灯片11
#训练随机森林
sv.tab=list(NULL,NULL,NULL,NULL,NULL);prob=c()
for(i in 1:5){m=mm[[i]];
svm.mod=randomForest(Class~.,data=BC[-m,])
pred=predict(svm.mod,BC[m,])
pr=predict(svm.mod,BC[m,],type="prob")
prob=c(prob,pr[,2])
sv.tab[[i]]=table(pred,BC[m,]$Class)
}
dat.class=BC$Class[c(mm[[1]],mm[[2]],mm[[3]],mm[[4]],mm[[5]])]
```

```
##计算预测准确率和画图代码见幻灯片4和5
#输出结果为
AUC
#Area under the curve: 0.9928
ACC
#[1] 0.9751098
Recall
#[1] 0.9832636
Specificity
#[1] 0.9707207
Precision
#[1] 0.9475806
MCC
#[1] 0.9461526
#画ROC图像 代码见幻灯片12
```

各算法预测的准确率参数汇总表

	AUC	ACC	Recall	Specificity	Precision	MCC
Bayes5折	0.9933	0.9751	0.9833	0.9707	0.9476	0.9462
Bayes10折	0.9934	0.9766	0.9874	0.9707	0.9478	0.9495
Bayes20折	0.9936	0.9751	0.9833	0.9707	0.9476	0.9462
SVM5折	0.9935	0.9751	0.9749	0.9752	0.9550	0.9457
决策树5折	0.9926	0.9634	0.9539	0.9685	0.9421	0.9198
随机森林5折	0.9928	0.9751	0.9833	0.9707	0.9476	0.9462

注：对朴素贝叶斯分别进行5折,10折,20折划分并计算准确率参数，SVM、决策树、随机森林算法仅练习了5折划分方法。

问: Any drawback?

Attribute Information:

1. Sample code number: id number **
2. Clump Thickness: 1 - 10 物理特征
3. Uniformity of Cell Size: 1 - 10 物理特征
4. Uniformity of Cell Shape: 1 - 10 物理特征
5. Marginal Adhesion: 1 - 10 物理特征
6. Single Epithelial Cell Size: 1 - 10 物理特征
7. Bare Nuclei: 1 - 10
8. Bland Chromatin: 1 – 10 染色质
9. Normal Nucleoli: 1 - 10 核仁
10. Mitoses: 1 - 10 有丝分裂
11. Class: (2 for benign, 4 for malignant)

暑期课堂结语

学S
想T
问A
读R
试T