svm

datainput=function(max,goal){  
 data=c()  
 index1=c()  
 index2=c()  
 for (i in 0:9){  
 aa=paste(goal,i,"\_",0:num[i+1],".txt",sep="")  
 bb=rep(i,num[i+1]+1)  
 cc=0:num[i+1]  
 index1=c(index1,bb)  
 index2=c(index2,cc)  
 data=c(data,aa)  
 }  
#############製作所有目錄，並記錄檔案編號到index1 ,index2  
 xx=matrix(0,ncol=32\*32+2,nrow=length(data))  
 for (i in 1:length(data)){  
 x=readLines(data[i])  
 x1=strsplit(x,split="")  
 x1=unlist(x1)  
 x1=as.numeric(x1)  
 xx[i,3:1026]=x1  
 }  
 xx[,1]=index1  
 xx[,2]=index2  
 return(xx)  
}  
##############xx 為第一行是編號0~9 第二行為第二個編號 第三行到第1026行是將該個資料的data 以數值存入  
  
  
#########讀入資料  
num=c(188,197,194,198,185,186,194,200,179,203)  
train=datainput(max=num,goal="C:\\Users\\TSR\\Desktop\\文字辨識\\trainingDigits\\")  
num=c(86,96,91,84,113,107,86,95,90,88)  
test=datainput(max=num,goal="C:\\Users\\TSR\\Desktop\\文字辨識\\testDigits\\")  
######建立一個函數，將32\*32筆資料，分成8組，分別另為x1~x8，並記錄每組內1的個數  
trans=function(max,data,newname,sp){  
 newname=matrix(0,ncol=sp+1,nrow=max)  
 newname[,1]=data[,1]  
 for (j in 1:max){  
 for (i in 1:sp){  
 newname[j,i+1]=sum(data[j,(32\*32/sp\*(i-1)+2):(32\*32/sp\*i+1)])  
 }  
 }  
 newname=as.data.frame(newname)  
 names(newname)=c("index1",paste("x",1:sp,sep=""))  
 return(newname)  
}  
#######將資料進行轉換  
newtrain=trans(dim(train)[1],train,newtrain,32)  
newtest=trans(dim(test)[1],test,newtest,32)  
  
###將0~9換成factor類型，否則後面會有問題  
newtrain[,1]=as.factor(newtrain[,1])  
newtest[,1]=as.factor(newtest[,1])  
library(e1071)  
#####建立預測模型 以每列加總值當作x變數  
model=svm(index1~.,data=newtrain)  
####預測  
predict=predict(model,newtest[,2:dim(newtest)[2]])  
  
######以每列加總當作x變數下的正確率  
sum(predict==newtest[,1])/length(test[,1])

## [1] 0.8816068

### 以每列加總當作x變數下的正確率

################將原始資料丟入svm判別  
newtrain=as.data.frame(train[,-2])  
newtest=as.data.frame(test[,-2])  
  
newtrain[,1]=as.factor(newtrain[,1])  
newtest[,1]=as.factor(newtest[,1])  
colnames(newtrain)=c("index",paste("x",1:1024,sep=""))  
colnames(newtest)=c("index",paste("x",1:1024,sep=""))  
library(e1071)  
#####建立預測模型  
model=svm(index~.,data=newtrain)

## Warning in svm.default(x, y, scale = scale, ..., na.action = na.action):  
## Variable(s) 'x1' and 'x2' and 'x3' and 'x4' and 'x5' and 'x6' and 'x7' and  
## 'x33' and 'x34' and 'x35' and 'x36' and 'x37' and 'x38' and 'x65' and 'x66'  
## and 'x67' and 'x68' and 'x69' and 'x97' and 'x98' and 'x99' and 'x100' and  
## 'x101' and 'x129' and 'x130' and 'x131' and 'x132' and 'x161' and 'x162'  
## and 'x163' and 'x164' and 'x193' and 'x194' and 'x195' and 'x196' and  
## 'x225' and 'x226' and 'x227' and 'x228' and 'x256' and 'x257' and 'x258'  
## and 'x259' and 'x288' and 'x289' and 'x290' and 'x291' and 'x319' and  
## 'x320' and 'x321' and 'x322' and 'x351' and 'x352' and 'x353' and 'x354'  
## and 'x355' and 'x382' and 'x383' and 'x384' and 'x385' and 'x386' and  
## 'x387' and 'x414' and 'x415' and 'x416' and 'x417' and 'x418' and 'x419'  
## and 'x445' and 'x446' and 'x447' and 'x448' and 'x449' and 'x450' and  
## 'x451' and 'x452' and 'x477' and 'x478' and 'x479' and 'x480' and 'x481'  
## and 'x482' and 'x483' and 'x484' and 'x509' and 'x510' and 'x511' and  
## 'x512' and 'x513' and 'x514' and 'x515' and 'x516' and 'x517' and 'x540'  
## and 'x541' and 'x542' and 'x543' and 'x544' and 'x545' and 'x546' and  
## 'x547' and 'x548' and 'x572' and 'x573' and 'x574' and 'x575' and 'x576'  
## and 'x577' and 'x578' and 'x579' and 'x580' and 'x605' and 'x606' and  
## 'x607' and 'x608' and 'x609' and 'x610' and 'x611' and 'x637' and 'x638'  
## and 'x639' and 'x640' and 'x641' and 'x642' and 'x643' and 'x670' and  
## 'x671' and 'x672' and 'x673' and 'x674' and 'x702' and 'x703' and 'x704'  
## and 'x705' and 'x706' and 'x735' and 'x736' and 'x737' and 'x738' and  
## 'x767' and 'x768' and 'x769' and 'x800' and 'x801' and 'x832' and 'x833'  
## and 'x864' and 'x865' and 'x866' and 'x896' and 'x897' and 'x898' and  
## 'x899' and 'x900' and 'x929' and 'x930' and 'x931' and 'x932' and 'x933'  
## and 'x961' and 'x962' and 'x963' and 'x964' and 'x965' and 'x993' and  
## 'x994' and 'x995' and 'x996' and 'x997' and 'x998' constant. Cannot scale  
## data.

model

##   
## Call:  
## svm(formula = index ~ ., data = newtrain)  
##   
##   
## Parameters:  
## SVM-Type: C-classification   
## SVM-Kernel: radial   
## cost: 1   
## gamma: 0.0009765625   
##   
## Number of Support Vectors: 1141

####預測  
predict=predict(model,newtest[,2:dim(newtest)[2]])  
######以每筆資料當作x變數的正確率  
sum(predict==newtest[,1])/length(test[,1])

## [1] 0.9735729

### 以每筆資料當作x變數的正確率