```
{\rtf1\ansi\ansicpg1252\cocoartf1038\cocoasubrtf360
{\fonttbl\f0\fmodern\fcharset0 Courier;}
{\colortbl;\red255\green255\blue255;}
\margl1440\margr1440\vieww12700\viewh11720\viewkind0
\deftab720
\pard\pardeftab720\ql\qnatural

\f0\fs28 \cf0 \
Chapter 9.  Clustering, Classification and Data Mining\
\
# Color-magnitude diagram for low-redshift COMBO-17 galaxies\
\
COMBO_loz=read.table('http://astrostatistics.psu.edu/MSMA/datasets/COMBO17_lo
wz.dat', \
  header=T, fill=T)\
dim(COMBO) ; names(COMBO)\
par(mfrow=c(1,2))\
plot(COMBO_loz, pch=20, cex=0.5, xlim=c(-22,-7), ylim=c(-2,2.5), \
  xlab=expression(M[B]~~(mag)), ylab=expression(M[280] - M[B]~~(mag)), \
  main='')\
\
# Two-dimensional kernel-density estimator\
\
library(MASS)\
COMBO_loz_sm <- kde2d(COMBO_loz[,1], COMBO_loz[,2], h=c(1.6,0.4), \
  lims = c(-22,-7,-2,2.5), n=500)\
image(COMBO_loz_sm, col=grey(13:0/15), xlab=expression(M[B]~~(mag)), \
  ylab=expression(M[280] - M[B]~~(mag)), xlim=c(-22,-7), ylim=c(-2,2.5), \
  xaxp=c(-20,-10,2))\
par(mfrow=c(1,1))\
\
# Standardize variables\
\
Mag_std <- scale(COMBO_loz[,1]) \
Color_std <- scale(COMBO_loz[,2])\
COMBO_std <- cbind(Mag_std,Color_std)\
\
# Hierarchical clustering\
\
COMBO_dist <- dist(COMBO_std)\
COMBO_hc <- hclust(COMBO_dist, method='complete')\
COMBO_coph <- cophenetic(COMBO_hc)\
cor(COMBO_dist, COMBO_coph)\
\
# Cutting the tree at k=5 clusters\
\
```

```
plclust(COMBO_hc, label=F)\
COMBO_hc5a <- rect.hclust(COMBO_hc, k=5, border='black') \
COMBO_hc5b <- cutree(COMBO_hc, k=5) \
plot(COMBO_loz, pch=(COMBO_hc5b+19), cex=0.7, xlab=expression(M[B]~~(mag)), \
  ylab=expression(M[280] - M[B]~~(mag)), main='')\
\
# Density-based clustering algorithm\
\
install.packages('fpc') ; library(fpc)\
COMBO_dbs <-  dbscan(COMBO_std, eps=0.1, MinPts=5, method='raw')\
print.dbscan(COMBO_dbs) ; COMBO_dbs$cluster\
plot(COMBO_loz[COMBO_dbs$cluster==0,], pch=20, cex=0.7, xlab='M_B (mag)',\
  ylab='M_280 - M_B (mag)')\
points(COMBO_loz[COMBO_dbs$cluster==2,], pch=2, cex=1.0)\
points(COMBO_loz[COMBO_dbs$cluster==1 | COMBO_dbs$cluster==3,], pch=1,
cex=1.0)\
\
\
#   ***********  CLASSIFICATION OF SLOAN POINT SOURCES **********\
#   ************************************************************\
# R script for constructing SDSS test and training datasets is given \
# in Appendix C.\
\
\
# SDSS point sources test dataset, N=17,000 (mag<21, point sources, hi-qual)\
\
SDSS <- read.csv('http://astrostatistics.psu.edu/MSMA/datasets/SDSS_test.csv',
h=T)\
dim(SDSS) ; summary(SDSS)\
SDSS_test <- data.frame(cbind((SDSS[,1]-SDSS[,2]), (SDSS[,2]-SDSS[,3]), \
        (SDSS[,3]-SDSS[,4]), (SDSS[,4]-SDSS[,5])))\
names(SDSS_test) <- c('u_g', 'g_r', 'r_i', 'i_z')\
str(SDSS_test) \
\
par(mfrow=c(1,3))\
plot(SDSS_test[,1], SDSS_test[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8),pch=20, \
        cex=0.6, cex.lab=1.5, cex.axis=1.5, main='', xlab='u-g (mag)', ylab='g-r (mag)') \
plot(SDSS_test[,2], SDSS_test[,3], xlim=c(-0.7,1.8), ylim=c(-0.7,1.8), pch=20, \
        cex=0.6, cex.lab=1.5, cex.axis=1.5, main='', xlab='g-r (mag)', ylab='r-i (mag)') \
plot(SDSS_test[,3], SDSS_test[,4], xlim=c(-0.7,1.8), ylim=c(-1.1,1.3), pch=20, \
        cex=0.6, cex.lab=1.5, cex.axis=1.5, main='', xlab='r-i (mag)', ylab='i-z (mag)') \
par(mfrow=c(1,1))\
\
# Quasar training set, N=2000 (Class 1)\
 \
```

```
qso1 <- read.table('http://astrostatistics.psu.edu/MSMA/datasets/SDSS_QSO.dat',
h=T)  \
dim(qso1) ; summary(qso1)\
bad_phot_qso <- which(qso1[,c(3,5,7,9,11)] > 21.0 | qso1[,3]==0)\
qso2 <- qso1[1:2000,-bad_phot_qso,]\
qso3 <- cbind((qso2[,3]-qso2[,5]), (qso2[,5]-qso2[,7]), (qso2[,7]-qso2[,9]), (qso2[,9]-
qso2[,11]))\
qso_train <- data.frame(cbind(qso3, rep(1, length(qso3[,1]))))\
names(qso_train) <- c('u_g', 'g_r', 'r_i', 'i_z', 'Class')\
dim(qso_train) ; summary(qso_train) \
\
# Star training set, N=5000 (Class 2)\
\
temp2 <- read.csv('http://astrostatistics.psu.edu/MSMA/datasets/SDSS_stars.csv',
h=T)\
dim(temp2) ; summary(temp2) \
star <- cbind((temp2[,1]-temp2[,2]), (temp2[,2]-temp2[,3]), (temp2[,3]-temp2[,4]), \
        (temp2[,4]-temp2[,5]))\
star_train <- data.frame(cbind(star, rep(2, length(star[,1]))))\
names(star_train) <- c('u_g','g_r','r_i','i_z','Class')\
dim(star_train) ; summary(star_train) \
\
# White dwarf training set, N=2000 (Class 3)\
\
temp3 <- read.csv('http://astrostatistics.psu.edu/MSMA/datasets/SDSS_wd.csv',
h=T)\
dim(temp3) ; summary(temp3)\
temp3 <- na.omit(temp3)\
wd <- cbind((temp3[1:2000,2]-temp3[1:2000,3]), (temp3[1:2000,3]-
temp3[1:2000,4]),\
        (temp3[1:2000,4]-temp3[1:2000,5]), (temp3[1:2000,5]-temp3[1:2000,6]))\
wd_train <- data.frame(cbind(wd, rep(3, length(wd[,1]))))\
names(wd_train) <- c('u_g', 'g_r', 'r_i', 'i_z', 'Class')\
dim(wd_train) ; summary(wd_train) \
\
# Combine and plot the training set (9000 objects)\
\
SDSS_train <- data.frame(rbind(qso_train, star_train, wd_train))\
names(SDSS_train) <- c('u_g', 'g_r', 'r_i', 'i_z', 'Class')\
str(SDSS_train)\
\
par(mfrow=c(1,3))\
plot(SDSS_train[,1], SDSS_train[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8), pch=20, \
        col=SDSS_train[,5], cex=0.6, cex.lab=1.6, cex.axis=1.6, main='', xlab='u-g
(mag)',\
        ylab='g-r (mag)')\
```

```
legend(-0.5, 1.7, c('QSO','MS + RG','WD'), pch=20, col=c('black','red','green'), \
        cex=1.6)\
plot(SDSS_train[,2], SDSS_train[,3], xlim=c(-0.7,1.8), ylim=c(-0.7,1.8), pch=20, \
        col=SDSS_train[,5], cex=0.6, cex.lab=1.6, cex.axis=1.6, main='', xlab='g-r
(mag)',\
        ylab='r-i (mag)') \
plot(SDSS_train[,3], SDSS_train[,4], xlim=c(-0.7,1.8), ylim=c(-1.1,1.3), pch=20, \
        col=SDSS_train[,5], cex=0.6, cex.lab=1.6, cex.axis=1.6, main='', xlab='r-i (mag)',\
        ylab='i-z (mag)') \
par(mfrow=c(1,1))\
\
\
# Unsupervised k-means partitioning\
\
SDSS.kmean <- kmeans(SDSS_test,6)\
print(SDSS.kmean$centers)\
plot(SDSS_test[,1], SDSS_test[,2], pch=20, cex=0.3, col=gray(SDSS.kmean$cluster/7), \
  xlab='u-g (mag)', ylab='g-r (mag)', xlim=c(-0.5,3), ylim=c(-0.6,1.5)) \
\
# Linear discriminant analysis\
\
library(MASS)\
SDSS_lda <- lda(SDSS_train[,1:4], as.factor(SDSS_train[,5]))\
SDSS_train_lda <- predict(SDSS_lda, SDSS_train[,1:4])\
SDSS_test_lda <- predict(SDSS_lda, SDSS_test[,1:4])\
\
par(mfrow=c(2,1))\
plot(SDSS_test[,1],SDSS_test[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8), pch=20, \
  col=SDSS_test_lda$class, cex=0.5, main='', xlab='u-g (mag)', ylab='g-r (mag)') \
\
# k-nn classification\
\
install.packages('class') ; library(class)\
SDSS_knn4 <- knn(SDSS_train[,1:4], SDSS_test, \
  as.factor(SDSS_train[,5]), k=4, prob=T)\
plot(SDSS_test[,1], SDSS_test[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8), pch=20, \
  col=SDSS_knn4, cex=0.5, main='', xlab='u-g (mag)', ylab='g-r (mag)') \
\
# Validation of k-nn classification\
\
SDSS_train_lda <- lda(SDSS_train[,1:4], as.factor(SDSS_train[,5]))\
SDSS_train_knn4 <- knn(SDSS_train[,1:4], SDSS_train[,1:4], SDSS_train[,5],k=4)\
\
plot(jitter(as.numeric(SDSS_train_lda$class), factor=0.5), jitter(as.numeric\
  (SDSS_train[,5]), factor=0.5), pch=20, cex=0.5, xlab='LDA class',\
  ylab='True class', xaxp=c(1,3,2),yaxp=c(1,3,2))\
```

```
plot(jitter(as.numeric(SDSS_train_knn4), factor=0.5), jitter(as.numeric\
  (SDSS_train[,5]), factor=0.5), pch=20, cex=0.5, xlab='k-nn class',\
  ylab='True class', xaxp=c(1,3,2),yaxp=c(1,3,2))\
par(mfrow=c(1,1))\
\
# Single layer neutral network\
\
library(nnet)\
options(size=100, maxit=1000) \
SDSS_nnet <- multinom(as.factor(SDSS_train[,5]) ~ SDSS_train[,1] + SDSS_train[,2] + \
  SDSS_train[,3] + SDSS_train[,4], data=SDSS_train)\
SDSS_train_nnet <- predict(SDSS_nnet,SDSS_train[,1:4])\
plot(jitter(as.numeric(SDSS_train_nnet), factor=0.5), jitter(as.numeric\
  (SDSS_train[,5]), factor=0.5), pch=20, cex=0.5, xlab='nnet class',\
  ylab='True class', xaxp=c(1,3,2),yaxp=c(1,3,2))\
\
# Classification And Regression Tree model, prediction and validation\
\
library('rpart')\
SDSS_rpart_mod <- rpart(SDSS_train[,5] ~., data=SDSS_train[,1:4])\
SDSS_rpart_test_pred <- predict(SDSS_rpart_mod, SDSS_test)\
SDSS_rpart_train_pred <- predict(SDSS_rpart_mod, SDSS_train)\
summary(SDSS_rpart_mod) ; str(SDSS_rpart_mod)\
\
plot(SDSS_test[,1], SDSS_test[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8), pch=20, \
  col=round(SDSS_rpart_test_pred), cex=0.5,\
  main='', xlab='u-g (mag)', ylab='g-r (mag)')\
plot(jitter(SDSS_rpart_train_pred, factor=5), jitter(SDSS_train[,5]), pch=20, \
  cex=0.5, xlab='CART class', ylab='True class',yaxp=c(1,3,2))\
\
plot(SDSS_rpart_mod, branch=0.5, margin=0.05) \
text(SDSS_rpart_mod, digits=3, use.n=T, cex=0.8)\
plotcp(SDSS_rpart_mod, lwd=2, cex.axis=1.3, cex.lab=1.3)\
\
\
# Support Vector Machine model, prediction and validation\
\
install.packages('e1071') ; library(e1071)\
SDSS_svm_mod <- svm(SDSS_train[,5] ~.,data=SDSS_train[,1:4],cost=100, gamma=1)\
summary(SDSS_svm_mod) ; str(SDSS_svm_mod) \
SDSS_svm_test_pred <- predict(SDSS_svm_mod, SDSS_test)\
SDSS_svm_train_pred <- predict(SDSS_svm_mod, SDSS_train)\
\
plot(SDSS_test[,1], SDSS_test[,2], xlim=c(-0.7,3), ylim=c(-0.7,1.8), pch=20, \
  col=round(SDSS_svm_test_pred), cex=0.5, main='',\
  xlab='u-g (mag)', ylab='g-r (mag)')\
```

```
plot(SDSS_svm_train_pred, jitter(SDSS_train[,5]), pch=20, cex=0.5, \
  xlab='SVM class', ylab='True class', yaxp=c(1,3,2))\
\
# Final SVM classification of the test set\
\
par(mfrow=c(1,3))\
plot(SDSS_test[,1], SDSS_test[,2], xlim=c(-0.7,3), col=round(SDSS_svm_test_pred), \
  ylim=c(-0.7,1.8), pch=20, cex=0.5, main='', xlab='u-g (mag)',ylab='g-r (mag)') \
plot(SDSS_test[,2], SDSS_test[,3], xlim=c(-0.7,1.8), col=round(SDSS_svm_test_pred),\
  ylim=c(-0.7,1.8), pch=20, cex=0.5, main='', xlab='g-r (mag)',ylab='r-i (mag)') \
plot(SDSS_test[,3], SDSS_test[,4], xlim=c(-0.7,1.8), col=round(SDSS_svm_test_pred),\
  ylim=c(-1.1,1.3), pch=20, cex=0.5, main='', xlab='r-i (mag)',ylab='i-z (mag)') \
par(mfrow=c(1,1))\
\
SDSS_test_svm_out <- cbind(SDSS[,6], SDSS[,7], SDSS_test, SDSS_svm_test_pred)\
names(SDSS_test_svm_out)[c(1,2,7)] <- c('R.A.', 'Dec', 'SVM Class')\
write.table(format(SDSS_test_svm_out),
file='SDSS_test_svm.out',sep='\\\\t',quote=F)\
\
\
}
```