

Machine Learning Seminar 4.

Supervised, mixed and geometry morphometry methods.

March 19, 2020

1 R skills

```
## I used feedback from the previous seminar to improve Misclass()
## update.packages("shipunov")
```

2 Supervised methods

```
## first, we define training and (fake) testing groups:
iris.train <- iris[seq(1, nrow(iris), 5), ]
iris.unknown <- iris[-seq(1, nrow(iris), 5), ]
```

1. Linear Discriminate Analysis (LDA)

```
library(MASS)
iris.lda <- lda(Species ~ ., data=iris.train)
iris.ldap <- predict(iris.lda, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.ldap$class)
```

2. Multivariate ANOVA

```
summary(manova(as.matrix(iris[, -5]) ~ iris$Species), test="Wilks")
library(vegan)
adonis(as.matrix(iris[, 1:4]) ~ iris$Species)
```

3. Recursive partitioning

```
library(rpart)
iris.rpart <- rpart(Species ~ ., data=iris.train)
iris.rpartp <- predict(iris.rpart, iris.unknown[, -5], type="class")
Misclass(iris.unknown$Species, iris.rpartp)
plot(iris.rpart); text(iris.rpart, xpd=TRUE)
```

4. Bagging: Random Forest

```
library(ranger)
iris.rg <- ranger(Species ~ ., data=iris.train, importance="impurity")
iris.rgp <- predict(iris.rg, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.rgp$predictions)
iris.rg$variable.importance
```

5. Boosting: extreme gradient boosting

```
library(gbm)
iris.gbm <- gbm(Species ~ ., data=rbind(iris.train, iris.train)) # make bigger
iris.gbmp1 <- predict(iris.gbm, iris.unknown[, -5], n.trees=iris.gbm$n.trees)
iris.gbmp2 <- apply(iris.gbmp1, 1,
  function(.x) colnames(iris.gbmp1)[which.max(.x)]) # membership trick
library(shipunov)
Misclass(iris.unknown$Species, iris.gbmp2)
```

6. Naïve Bayes and how to plot the learning

```
library(e1071)
iris.nb <- naiveBayes(Species ~ ., data=iris.train)
iris.nbp <- predict(iris.nb, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.nbp)

iris.p <- prcomp(iris[, -5], scale=TRUE)$x[, 1:2]
sel <- 1:nrow(iris) %in% seq(1, nrow(iris), 5)
plot(iris.p, col=iris$Species, pch=ifelse(sel, 19, 1), main="naiveBayes")
iris.nb2 <- naiveBayes(Species ~ ., data=cbind(iris[5], iris.p)[sel, ])
Gradd(iris.nb2, iris.p[sel, ], what="lines")
```

7. Nearest neighbor methods

```
library(class)
iris.knn.pred <- knn(train=iris.train[, -5],
  test=iris.unknown[, -5], cl=iris.train[, 5], k=5)
library(shipunov)
Misclass(iris.knn.pred, iris.unknown[, 5])

iris.bootknn <- BootKNN(iris[, -5], iris[, 5])
st <- apply(iris.bootknn, 1, function(.x) var(as.numeric(as.factor(.x))))
plot(prcomp(iris[, -5])$x, col=iris$Species, pch=ifelse(st == 0, 19, 1))

cl1 <- iris$Species
sam <- c(rep(0, 4), 1) > 0
cl1[!sam] <- NA

for (d in (5:14)/100) {
  iris.pred <- DNN(dst=dist(iris[, -5]), cl=cl1, d=d)
  cl1[is.na(cl1)] <- iris.pred
}
```

```

table(cl1, useNA="ifany")
Misclass(iris$Species, cl1)

cl2 <- iris$Species
iris.d <- Gower.dist(iris[, -5])
iris.proximity <- t(DNN(dst=iris.d, cl=cl2, k=5, details=TRUE, self=TRUE))/5
head(iris.proximity)

```

8. Support Vector Machines

```

library(e1071)
iris.svm <- svm(Species ~ ., data=iris.train)
iris.svm.p <- predict(iris.svm, iris.unknown[, -5])
library(shipunov)
Misclass(iris.unknown$Species, iris.svm.p)

```

9. Neural Networks

```

library(nnet)
iris.nn <- nnet(Species ~ ., data=iris.train, size=4)
iris.nnp <- predict(iris.nn, iris.unknown[, -5], type="class")
library(shipunov)
Misclass(iris.unknown$Species, iris.nnp)
library(NeuralNetTools)
oldpar <- par(mar=c(0, 2, 0, 1), xpd=TRUE)
plotnet(iris.nn)
par(oldpar)

```

3 Mixed methods

Cluster sharpening

```

library(kpeaks)
nn <- max(findk(iris[, -5])$pcounts)
library(ksharp)
iris.km <- kmeans(iris[, -5], centers=nn)
Misclass(iris$Species, iris.km$cluster, best=TRUE)
names(iris.km$cluster) <- row.names(iris)
iris.ksharp <- ksharp(iris.km, data=iris[, -5])
library(shipunov)
Misclass(iris$Species, iris.ksharp$cluster, ignore=0, best=TRUE)

iris.kx <- list(cluster=as.numeric(iris$Species))
names(iris.kx$cluster) <- row.names(iris)
iris.ksharpx <- ksharp(iris.kx, data=iris[, -5])
plot(prcomp(iris[, -5])$x, col=iris$Species,
     pch=ifelse(iris.ksharpx$cluster == 0, 1, 19))

```

4 Geometric morphometrics

Package geomorph example

```
library(geomorph)
## # download: http://ashipunov.me/shipunov/school/biol_240/061.jpg
## # download: http://ashipunov.me/shipunov/school/biol_240/046.jpg
## # download: http://ashipunov.me/shipunov/school/biol_240/650.jpg
## # check that all these files present in working directory
## digitize2d(filelist=c("061.jpg", "406.jpg", "650.jpg"),
## nlandmarks=4, tpsfile="046.tps")
## readland.tps("046.tps")

am <- read.table("http://ashipunov.me/data/bigaln2.txt", as.is=TRUE, head=TRUE)
ag <- readland.tps("http://ashipunov.me/data/bigaln.tps", specID="imageID")
ag.gpa <- gpagen(ag) # Generalized Procrustes Analysis and projection
ag.pca <- gm.prcomp(ag.gpa$coords)$x # PCA in tangent space
veins <- as.numeric(cut(am$N.ZHILOK, 2))
plot(ag.pca, col=veins, main="Shape vs. number of lateral veins")

v1 <- mshape(ag.gpa$coords[, , veins == 1])
v2 <- mshape(ag.gpa$coords[, , veins == 2])
all <- mshape(ag.gpa$coords)
ag.links <- matrix(c(1, rep(c(2:7, 12:8), each=2), 1), ncol=2, byrow=TRUE)
ag.links # matrix of links
old.par <- par(mfrow=c(1, 2), mar=rep(0, 4)) # side by side
plotRefToTarget(v1, all, links=ag.links)
plotRefToTarget(v2, all, links=ag.links)
par(old.par)
```