Liver disease classification using support vector machine and model based clustering

STATS/CSE 790 Assignment 5

2024-03-19

Pao Zhu Vivian Hsu (400547994)

Introduction

In this study, we use Support Vector Machines (SVMs) and model based clustering to predict whether a patient has liver disease or not. The data in this study comes from the UCI Machine Learning Repository (Ramana & Venkateswarlu, 2012). It contains 583 rows of patient data and 11 variables of measurements. The response variable is a binary categorical variable that indicates whether the patient is diagnosed with liver disease or not. The potential predictor variables include age, gender, total Bilirubin, direct Bilirubin, total proteins, albumin, A/G ratio, SGPT, SGOT and Alkphos.

Methods

We start the study by checking the shape of the data and handling any missing values through imputation. We then visualized the data in a pairs plot to check for any patterns in the data. Figure 1 shows a subset of this plot, where blue indicates the presence of liver disease and red indicates the absence of it. We can see that the observations have a non-linear boundary between them, which makes the data suitable for SVM or model based clustering.

As such, we then split the data into two equal parts to form training and testing sets and applied these two different techniques to the data: SVM and model based clustering.

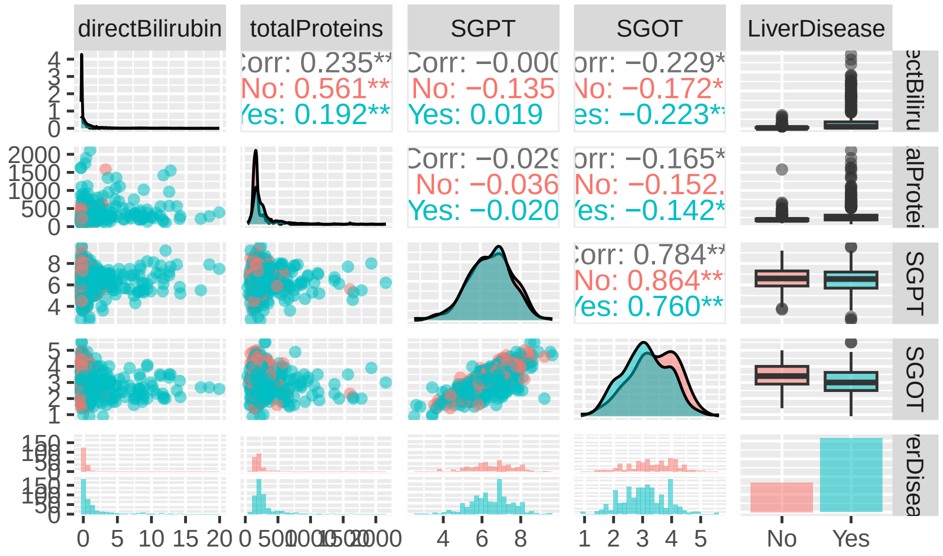

Figure 1: Pairs plot of a selection of the variables

The first technique was SVM, a supervised learning method that uses non-linear decision boundaries to classify data. The boundaries are defined using a kernel function which is used to enlarge the feature space and measure the similarity between observations. For our study, we used linear and radial kernels to model the data. The linear kernel is defined as $K(x_i, {x_i}') = \sum_{j=1}^{p} x_{ij}, {x_i}'_j$ and the radial kernel is defined as $K(x_i, {x_i}') = exp(-\gamma \sum_{j=1}^{p} (x_{ij} - {x_i}'_j)^2)$ where γ is a positive constant and i represents the ith observation (King-Yu, 2024b).

We used cross validation to tune the cost parameter C and gamma constant γ. The cost parameter measures the number of observations that can be on the wrong side of the hyperplane and the tolerance towards margin violations. In other words, it puts a limitation on the sum of slack variables ϵ_i where ϵ_i indicates whether the ith observation is on the correct side of the margin ($\epsilon_i = 0$) or violates the margin ($\epsilon_i > 0$). Thus, C controls the bias and variance in the model which we seek to optimize (King-Yu, 2024a). For the linear kernel, we tried 11 different values of the cost parameter and found 0.1 to be the optimal value. For the radial kernel, we tried 10 different values of the cost parameter and found 6 to be the optimal value. Likewise, we tried 10 different values of gamma and found 5 to be the optimal value.

After cross-validation, the models were then fit using the test set and results were recorded.

The second technique we applied to the data is model based clustering, which involves .

The second method was FA, which involves replacing p observed variables by $q < p$ latent factors.

The model can be written as $X_i = \mu + \Lambda U_i + \epsilon$ for $i = 1, ..., n$ factors (**lecture11?**). For each FA, a hypothesis test is performed where the null hypothesis states that the number of factors in the model is sufficient. We performed FA using 1 to 9 factors and selected the optimal factor size as 9 because it failed to reject the null hypothesis (i.e. had a p-value larger than a significance level of $\alpha = 0.05$) while the other factor sizes rejected the null hypothesis.

After applying PCA and FA to the data, we then performed a clustering analysis. First, we performed clustering on the full data set. Then we applied clustering using only the PCs from our PCA.

Results

Table 1 below summarizes the SVM results using linear and radial kernels. The adjusted Rand index (ARI) is very poor for both kernels despite cross-validation. The ARI is slightly higher for the radial kernel compared to the linear kernel indicating that a radial kernel may be a better fit. We see a similar pattern for the misclassification error rate where the performance is better for the radial kernel compared to the linear kernel.

Table 1: SVM performance comparison

Method	Adjusted Rand Index	Misclassification Error Rate
SVM Linear Kernel	0.00010	0.99315
SVM Radial Kernel	0.00184	0.99658

Table 2 below summarizes the model based clustering results using a full and reduced model. The adjusted Rand index (ARI) is very poor for both kernels despite cross-validation. The ARI is slightly higher for the radial kernel compared to the linear kernel indicating that a radial kernel may be a better fit. We see a similar pattern for the misclassification error rate where the performance is better for the radial kernel compared to the linear kernel.

Table 2: Model based clustering performance comparison

Method	Adjusted Rand Index	Misclassification Error Rate
Full Model	-0.023	0.455

Method	Adjusted Rand Index	Misclassification Error Rate
Reduced Model	-0.011	0.777

Conclusion

Overall, our study shows that applying SVM to liver disease

Overall, our study shows that applying PCA prior to clustering for diabetes classification is not very effective. A few ways to improve the accuracy of these models include feature engineering, performing more extensive data cleansing prior to modelling, and incorporating subject-matter expertise when interpreting results.

References

King-Yu, S. (2024a). *Statistical learning lecture 13: Support vector machines i.*

King-Yu, S. (2024b). *Statistical learning lecture 14: Support vector machines II.*

Ramana, B., & Venkateswarlu, N. (2012). *ILPD (indian liver patient dataset).* UC Irvine Machine Learning Repository. https://doi.org/10.24432/C5D02C

Appendix

```
# ----- PACKAGE & DATA SETUP ----- #
library(tidyverse)
library(GGally)
library(e1071)
library(mclust)
library(vscc)
library(kableExtra)

# Load data
liver_raw <-
  read_csv("/Users/Vivian/Documents/Workspaces/R_Workspace/stats790/A3/Indian Liver Patient
           col_names = FALSE)
colnames(liver_raw) <- c("age", "gender", "totalBilirubin", "directBilirubin",
                         "totalProteins", "albumin","agRatio", "SGPT", "SGOT",
                         "alkphos", "diagnosis")

# ----- DATA TRANSFORMATION ----- #
# Check for missing data
sum(sapply(liver_raw, function(col){ifelse(is.na(col), 1, 0)}))
sum(sapply(liver_raw, function(col){ifelse(sum(col == ""), 1, 0)}))

# Data transformation
liver <- liver_raw %>%
  mutate(diagnosis = ifelse(diagnosis == 1, 1, 0), # One hot encoding
         gender = ifelse(gender == "Male", 0, 1), # One hot encoding
         alkphos = ifelse(is.na(alkphos), mean(alkphos, na.rm=TRUE), alkphos) # Impute missi
         )

# Check for missing data
sum(sapply(liver, function(col){ifelse(is.na(col), 1, 0)}))
```

```r
sum(sapply(liver, function(col){ifelse(sum(col == ""), 1, 0)}))

# Full pairs plot
liverV <- within(liver, LiverDisease <- ifelse(diagnosis==1,"Yes", "No"))
ggpairs(data=liverV[,-c(11)], aes(colour=LiverDisease, alpha=0.4))

# Smaller pairs plot
liverV <- within(liver, LiverDisease <- ifelse(diagnosis==1,"Yes", "No"))
ggpairs(data=liverV[,-c(11,1,2,3,6,7,10)], aes(colour=LiverDisease, alpha=0.4))

# Split data into train and test
set.seed(1)
train.ind <- sample(1:nrow(liver), nrow(liver) / 2)
liver.train <- liver[train.ind,]
liver.test <- liver[-train.ind,]
liver.test.labs <- liver[-train.ind, "diagnosis"]

svmfit <- svm(diagnosis ~ ., data = liver.train, kernel = "linear",
    cost = 0.1, scale = FALSE)

#clearly a non-linear boundary
plot(svmfit, liver.train)

summary(svmfit)
# Cross-validation to choose the best gamma and cost
set.seed(1)
tune.out <- tune(svm, diagnosis ~ ., data = liver.train,
    kernel = "linear",
    ranges = list(
      cost = c(0.05, 0.08, 0.09, 0.1, 0.11, 0.13, 0.15, 0.5, 1, 10, 100)
    )
```

```
  )

summary(tune.out)
# Prediction
pred <- predict(tune.out$best.model, newdata = liver.test)

tabSvmLinear <- table(pred, liver.test.labs$diagnosis)

svmfit <- svm(diagnosis ~ ., data = liver.train, kernel = "radial", gamma = 1, cost = 1)

#clearly a non-linear boundary
plot(svmfit, liver.train)

summary(svmfit)
# Cross-validation to choose the best gamma and cost
set.seed(1)
tune.out <- tune(svm, diagnosis ~ ., data = liver.train,
    kernel = "radial",
    ranges = list(
      cost = c(1, 5, 6, 7, 8, 9, 10, 15, 20, 30),
      gamma = c(1, 2, 3, 4, 5, 6, 7, 8, 9, 10)
    )
  )

summary(tune.out)
# Prediction
pred <- predict(tune.out$best.model, newdata = liver.test)

tabSvmRadial <- table(pred, liver.test.labs$diagnosis)

# RESULTS
```

```
method <- c("SVM Linear Kernel", "SVM Radial Kernel")
crand <- c(classAgreement(tabSvmLinear)$crand,
           classAgreement(tabSvmRadial)$crand)
misclass <- c(1-classAgreement(tabSvmLinear)$diag,
              1-classAgreement(tabSvmRadial)$diag)
summary <- data.frame("Method" = method,
                      "Adjusted Rand Index" = round(crand,5),
                      "Misclassification Error Rate" = round(misclass,5),
                      check.names = FALSE)
kable(summary)

# MCLUST
x <- scale(liver[,-11])
modclust <- Mclust(x, 2)
summary(modclust)

tabFull <- table(class=liver$diagnosis,
                 predictions=factor(as.character(map(modclust$z)-1)))

# VSCC
x <- scale(liver[,-11])
modvscc <- vscc(x)
tabReduced <- table(class=liver$diagnosis,
                    predictions=modvscc$bestmodel$classification)

# Show selected variables
head(modvscc$topselected)
plot(modvscc)

# RESULTS
method <- c("Full Model", "Reduced Model")
```

```r
crand <- c(classAgreement(tabFull)$crand,
           classAgreement(tabReduced)$crand)
misclass <- c(1-classAgreement(tabFull)$diag,
              1-classAgreement(tabReduced)$diag)
summary <- data.frame("Method" = method,
                      "Adjusted Rand Index" = round(crand,3),
                      "Misclassification Error Rate" = round(misclass,3),
                      check.names = FALSE)
kable(summary)
```