Breast Cancer Winconsin

Kernel SVM Case Study

MAY 14, 2017

DOLORES KE DING

1. Introduction

In this case study, I'll be using kernel support vector machine to train a model to classify a breast cancer dataset from Wisconsin. This breast cancer databases was obtained from the University of Wisconsin Hospitals, Madison from Dr. William H. Wolberg.

When it comes to disease classification, high accuracy would be the priority concerning because we wouldn't want to miss any chance of detecting a severe disease and take care of it as soon as possible. Therefore, in this case study, I would put more emphasize on classification accuracy and the sensitivity of the prediction outcome.

We know that support vector machine works well with high-dimension and small-size dataset. Therefore, it will be a great fit for this breast cancer dataset. Plus, this dataset is a binary classification case, and support vector machine works in here.

Section 1 is a brief introduction of the case study. Section 2 is the description of the dataset and the feature explanation in this dataset. Section 3 is the data cleaning process. In section 4, some exploratory analysis is conducted to get a rough sense of the dataset. Experimental process and results, conclusions and insights generated from the experimental results are presented in section 5. And section 6 provides the limitations and conclusion of this case study. Appendix is the coding part.

2. Data Description

2.1 Dataset

The dataset I'm using is an disease dataset which is collected from the University of Wisconsin Hospitals. The link of this dataset is http://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Original%29

The dataset has 699 observations, and 11 attributes.

Detailed information of the data attributes is presented later.

2.2 Attributes

There are 11 attributes in total.

#	Attribute	Domain
1.	Sample code number	id number
2.	Clump Thickness	1 - 10
3.	Uniformity of Cell Size	1 - 10
4.	Uniformity of Cell Shape	1 - 10
5.	Marginal Adhesion	1 - 10
6.	Single Epithelial Cell Size	1 - 10
7.	Bare Nuclei	1 - 10
8.	Bland Chromatin	1 - 10
9.	Normal Nucleoli	1 - 10
10.	Mitoses	1 - 10
11.	Class:	(2 for benign, 4 for malignant)

the last attribute, Class, will be the dependent variable.

2.3 Classification Labels

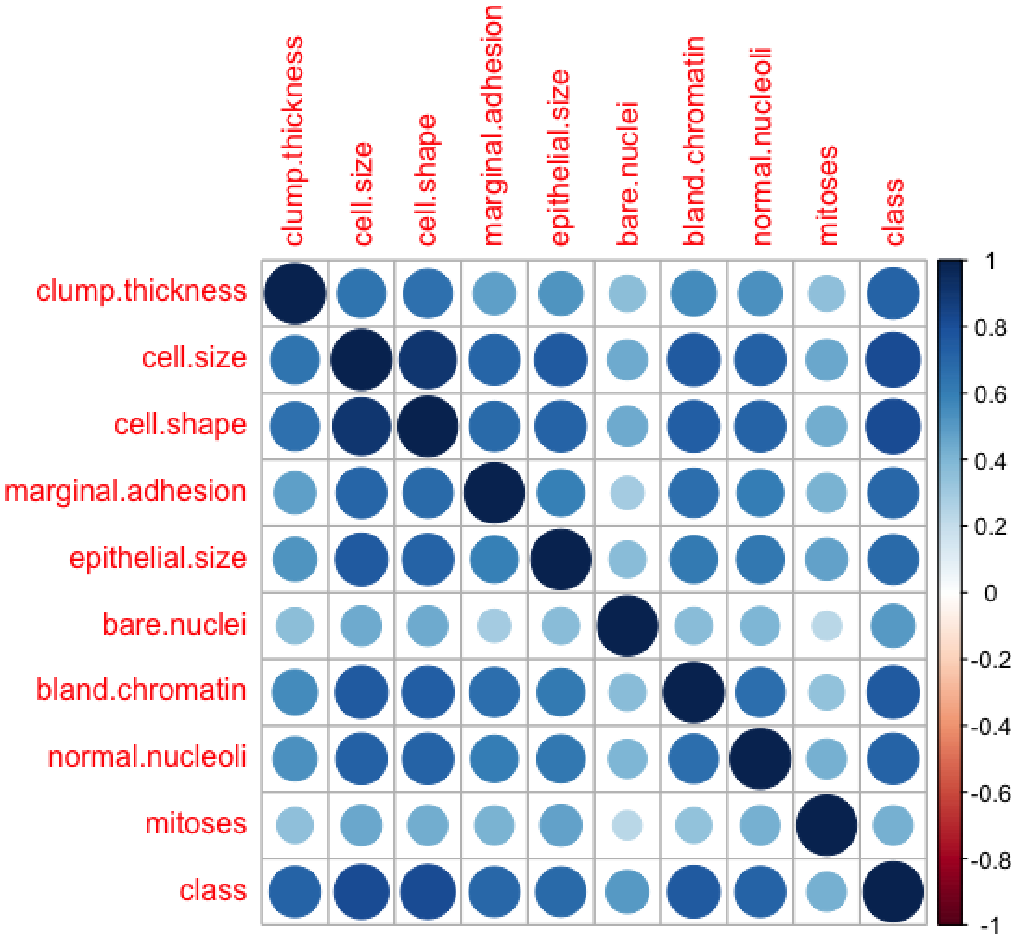
The dependent variable in this study is the Class. 2 for benign, 458 out of 699, which is 65.5% of all observations are benign. 4 for malignant, 241 out of 699 are malignant.

3. Data Cleaning

After importing the dataset, I found that all the variables do not have labels, so firstly, adding column names as showed in the dataset file. Then, I found that the first attribute, code id number, does not have any effect on the data analysis, so I removed this column. Then, I checked if there’s any missing value and got the observation removed. All the rest variables looks tidy and clean.

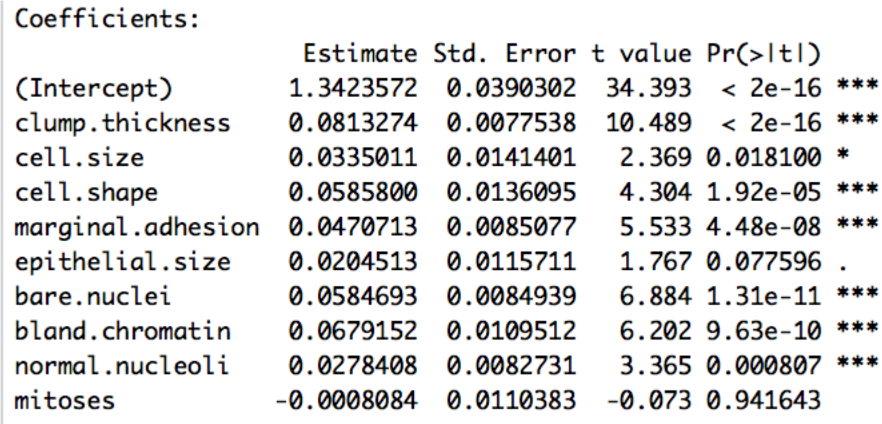
Then I checked the structure of the dataset and found that the class variable is integer. Therefore, I converted them to factor for future classification. Moreover, I also converted the bare.nuclei variable from factor to numeric for correlation analysis, and later converted it back to build support vector machine.

4. Data Analysis



Firstly, I plotted the correlation matrix between all the ten variables. Apparently, a lot of the variables are highly correlated. Class is correlated with every other attribute, although in a relatively low level with mitoses.

Then, I fitted a simple linear regression model to further look at the correlations.



```
Coefficients:
                   Estimate Std. Error t value Pr(>|t|)    
(Intercept)       1.3423572  0.0390302  34.393  < 2e-16 ***
clump.thickness   0.0813274  0.0077538  10.489  < 2e-16 ***
cell.size         0.0335011  0.0141401   2.369 0.018100 *  
cell.shape        0.0585800  0.0136095   4.304 1.92e-05 ***
marginal.adhesion 0.0470713  0.0085077   5.533 4.48e-08 ***
epithelial.size   0.0204513  0.0115711   1.767 0.077596 .  
bare.nuclei       0.0584693  0.0084939   6.884 1.31e-11 ***
bland.chromatin   0.0679152  0.0109512   6.202 9.63e-10 ***
normal.nucleoli   0.0278408  0.0082731   3.365 0.000807 ***
mitoses          -0.0008084  0.0110383  -0.073 0.941643    
```

The coefficients analysis matches with the correlation matrix. Mitoses is not significantly correlated with class.

5. Experimental Results and Analysis

In this case study, I conducted several support vector machine classifiers with different kernels and tuned some of them.

To evaluate the classifiers, I used 10-fold cross validations through caret.package in R to create confusion matrix, calculate accuracy, error rate, and so on.

In this study, I created 4 support vector machine classifiers with 4 different kernels, including linear, polynomial, radial, and sigmoid. I applied 10-fold cross validations on them and created the confusion matrix.

And then, I tuned the parameters for the polynomial, radial, and linear kernel, trying to find out if tuning parameters would improve the performance of the classifiers. The confusion matrixes of all the 7 support vector machines are presented in the table below.

linear svm		malignant	benign
Predicted	Malignant	42	5
	benign	4	99

poly svm		malignant	benign
Predicted	Malignant	32	0
	benign	14	104

radial svm		malignant	benign
Predicted	Malignant	43	7
	benign	3	97

sigmoid svm		malignant	benign
Predicted	Malignant	43	5
	benign	3	99

Tune linear		malignant	benign
Predicted	Malignant	42	6
	benign	4	98

Tune poly		malignant	benign
Predicted	Malignant	43	6
	benign	3	98

Tune radial		malignant	benign
Predicted	Malignant	45	9
	benign	1	95

table. 1

Below is the table presenting the number of support vectors, accuracy rate, 95% confidence interval, and sensitivity of the 7 support vector machines

	# support vectors	accuracy	95% CI	sensitivity
linear	43	94.00%	(0.8892, 0.9722)	91.30%
poly	169	90.67%	(0.8484, 0.948)	69.57%
radial	67	93.33%	(0.8808, 0.9676)	93.48%
sigmoid	64	94.67%	(0.8976, 0.9767)	93.48%
linear tuned	43	94.00%	(0.8892, 0.9722)	91.30%
poly tuned	57	94.00%	(0.8892, 0.9722)	93.48%
radial tuned	262	93.33%	(0.8808, 0.9676)	97.83%

table. 2

Since I’m trying to classify breast cancer, any chance of possible disease shouldn’t be neglected. Before building support vector machines, I set “malignant” as positive class, which means “malignant” is cancer positive.

And during this classification, I'm trying to improve the performance and accuracy of detecting possible cancer. Therefore, it requires high sensitivity, which is the percentage of true positive out of the true positive plus false negative, meaning how many cancer cases you get right among all the ones you predicted as malignant cancer.

From table. 2 we can see that linear support vector machine has a great accuracy but comparing to radial and sigmoid, it's sensitivity is not the best. Polynomial has a decent accuracy but the sensitivity the very low. By looking at table.1, we can see that it classified all the benign class right, but only 32 malignant cases out of 46 is classified correctly. Hence the sensitivity has bad performance. Then I conducted a tuned polynomial support vector machine, with coef0 of 1 and degree of 2, the performance of the new tuned polynomial support vector machine was improved drastically. The accuracy doesn't jump to much but the sensitivity increased from 70% to 93.5%, and from the confusion matrix we can see that it got 43 out of 46 cases correctly. This indicates that support vector machine using polynomial with tuned parameters can significantly improve the performance of detecting breast cancer. This shows that after tuning parameters, the performance of the support vector machines will get better or at least stay the same.

I also conducted tuned support vector machine with linear kernel, and the accuracy and sensitivity stayed the same. I think it's because when building the simple linear support vector machine, I didn't specifically set the cost and it was default as 1. When building the tuned support vector machine using a range of cost, the results also return that cost of 1 has the best performance. So, tuning parameters is not making much difference to the linear support vector machine.

After all the research, I found that different kernel support vector machine has their own advantage on different kinds of data. For this specific breast cancer dataset, I would say that radial support vector machine after tuning parameters have the best performance in detecting breast cancer, reaching the sensitivity of 97.83%, while also has a great accuracy of general classification of 93.33%. This radial support vector machine after tuning parameters classifier can detect almost all of the malignant breast cancer, correctly classified 44 out of 45 cases.

6. Conclusion

There’re some limitations in this study. As the data description mentioned, the observations of this dataset are collected from 1989 to 1991, therefore, the dataset is not up-to-date, and some features of the disease may have changed a little during those year, or new features that has an impact on breast cancer have been discovered till now. Moreover, considering the environmental pollution nowadays, human being’s health has also been effected from that. An old-time dataset may not be able to completely represent the current situation.

The analysis shows that support vector machines can classify this dataset with great predicting performance and support vector machine with tuned parameters generally improve the performance of original kernel support vector machine. And for the breast cancer dataset, radial has better performance than other kernels. In this analysis, tuned radial support vector machine improved the sensitivity by 4.35%, while the accuracy stays the same. But if we want high accuracy here, a support vector machine with kernel sigmoid would be the best, with the highest accuracy of 94.67%. In a word, different support vector machine works for different needs and different datasets.

7. Appendix

```
rm(list=ls())

library(e1071)
library(caret)
library(kernlab)
library(corrplot)

setwd("~/Desktop/CSC 529/case study 2")
bcancer <- read.table("breast-cancer-wisconsin.data", sep = ",", header = FALSE)


#data cleaning
#add variables names
colnames(bcancer)[1] <- "code.number"
colnames(bcancer)[2] <- "clump.thickness"
colnames(bcancer)[3] <- "cell.size"
colnames(bcancer)[4] <- "cell.shape"
colnames(bcancer)[5] <- "marginal.adhesion"
colnames(bcancer)[6] <- "epithelial.size"
colnames(bcancer)[7] <- "bare.nuclei"
colnames(bcancer)[8] <- "bland.chromatin"
colnames(bcancer)[9] <- "normal.nucleoli"
colnames(bcancer)[10] <- "mitoses"
colnames(bcancer)[11] <- "class"

#remove id column
bcancer$code.number <- NULL

#remove missing values
na.omit(bcancer)

#convert factor value to numeric for linear regression
bcancer$bare.nuclei <- as.numeric(bcancer$bare.nuclei)
str(bcancer)

#exploratory analysis
```

```
str(bcancer)
summary(bcancer)
prop.table(table(bcancer$class))

#correlation plot
M <- cor(bcancer)
corrplot(M, method="circle")


#fit linear regression model
fit <- glm(class~., data=bcancer)
summary(fit)

#change classification lable to factor
bcancer$class[bcancer$class=="2"] <- "benign"
bcancer$class[bcancer$class=="4"] <- "malignant"
bcancer$class <- factor(bcancer$class)
bcancer$bare.nuclei <- factor(bcancer$bare.nuclei)

bcancer$class <- factor(bcancer$class, levels=c("malignant","benign"))


#prepare training and test dataset
ind <- sample(2, nrow(bcancer), replace=TRUE, prob=c(0.8, 0.2))
train <- bcancer[ind==1,]
test <- bcancer[ind==2,]


#build the original svm model
svm_model <- svm(class ~ ., data=train, cross=10)
summary(svm_model)

pred <- predict(svm_model, test)
t <- table(pred, test$class)
confusionMatrix(t)


#svm linear using cv
svm_model.1 <- svm(class ~ ., kernel="linear", data=train, cross=10)
```

```
summary(svm_model.1)
pred.1 <- predict(svm_model.1, test)
t1 <- table(pred.1,test$class)
confusionMatrix(t1)

#svm polynomial using cv
svm_model.2 <- svm(class ~ ., kernel="polynomial", data=train, cross=10)
summary(svm_model.2)
pred.2 <- predict(svm_model.2,test)
t2 <- table(pred.2,test$class)
confusionMatrix(t2)

#svm radial using cv
svm_model.3 <- svm(class ~ ., kernel='radial', data=train, cross=10)
summary(svm_model.3)
pred.3 <- predict(svm_model.3,test)
t3 <- table(pred.3,test$class)
confusionMatrix(t3)

#svm sigmoid using cv
svm_model.4 <- svm(class ~ ., kernel="sigmoid", data=train, cross=10)
summary(svm_model.4)
pred.4 <- predict(svm_model.4,test)
t4 <- table(pred.4,test$class)
confusionMatrix(t4)

#tuned kernel svm polynomial using cv
svm_model.poly.tuned <- tune.svm(class ~ ., kernel = "polynomial", data =
train, coef0 = (-1:4), degree = (1:4),
tunecontrol=tune.control(sampling="cross", cross=10))
summary(svm_model.poly.tuned)
plot(svm_model.poly.tuned,xlab="degree", ylab="coef0")

svm_model.2.tuned <- svm(class ~ ., data=train, kernel = "polynomial", coef0
= 1, degree = 2, tunecontrol=tune.control(sampling="cross", cross=10))
summary(svm_model.2.tuned)
pred.2.tuned <- predict(svm_model.2.tuned, test)
t2.tuned <- table(pred.2.tuned, test$class)
confusionMatrix(t2.tuned)
```

```
#tuned radial using cv
svm_model.radial.tuned <- tune.svm(class ~ ., kernel = "radial", data = train,
cost=(1:4), gamma=(1:4), tunecontrol=tune.control(sampling="cross",
cross=10))
summary(svm_model.radial.tuned)
plot(svm_model.radial.tuned,xlab="gamma", ylab="cost")

svm_model.3.tuned <- svm(class ~ ., data=train, kernel = "radial", cost=2,
gamma=1, tunecontrol=tune.control(sampling="cross", cross=10))
summary(svm_model.3.tuned)
pred.3.tuned <- predict(svm_model.3.tuned, test)
t3.tuned <- table(pred.3.tuned, test$class)
confusionMatrix(t3.tuned)


#tuned linear using cv
svm_model.linear.tuned <- tune.svm(class ~ ., kernel = "linear", data = train,
cost=(1:4), tunecontrol=tune.control(sampling="cross", cross=10))
summary(svm_model.linear.tuned)
plot(svm_model.linear.tuned)

svm_model.1.tuned <- svm(class ~ ., data=train, kernel = "linear", cost=1,
tunecontrol=tune.control(sampling="cross", cross=10))
summary(svm_model.1.tuned)
pred.1.tuned <- predict(svm_model.1.tuned, test)
t1.tuned <- table(pred.1.tuned, test$class)
confusionMatrix(t1.tuned)
```