Exercício 11 - PCA - SVM

Rodrigo Machado Fonseca - 2017002253

February 6, 2022

1 Introdução

Neste será implementado do método de Análise de Componentes Principais (PCA) e de uma Máquina de Vetores de Suporte (SVM) para realizar a classificação do conjunto de dados Breast Cancer.

2 PCA

A técnica de PCA baseia-se em reduzir a dimensionalidade de um conjunto de dados, preservando o máximo de "variabilidade" (ou seja, informações estatísticas) possível. Diminuir a dimensionalidade implica reduzir a complexidade do sistema, o que acarreta, menos gasto computacional para resolver determinados problemas. Além disso, reduzir a dimensionalidade implica a possibilidade de uma análise visual, o que não seria possível em um espaço de alta dimensão.

O algoritmo do PCA baseia-se nos seguintes passos:

- Calcula a média dos dados
- Subtrai a média dos dados
- Calcula a matriz de covariância
- Encontre os auto-valores e auto-vetores

3 Experimento

A priori, será carregado a base *Breast Cancer* e nela será aplicado o método PCA. Abaixo segue a figura da variância das componentes ordenadas:

Em sequência, iremos definir o conjunto x de treinamento.

```
> x <- PC[, 1:2]
```

Por fim, separaremos o conjunto de dados em 10 *folds* para fazermos o treinamento.

```
> set.seed(123)
> index <- sample(1:nrow(x), length(1:nrow(x)))
> step <- length(index) %/% 10
> step_vec <- seq(step, step*10, step)
> step_vec[10] <- length(index)
> flds <- list()
> j = 1
> for(i in step_vec){
+   if(j != 10){
+     flds[[j]] <- index[(i-67):i]
+   }
+   else{
+     flds[[j]] <- index[step_vec[j-1]:i]
+   }
+   j <- j+1
+ }
```

4 Resultados

A seguir, está o treinamento para cada fold criado.

```
> j = 1
> accuracy <- matrix(nrow = 10, ncol = 1)
> accuracy_fold <- matrix(nrow = 10, ncol = 2)
> for(i in flds){
+   x_train <- x[-i,]
+   y_train <- y[-i, ]
```

```
> rm(list = ls())
> library(mlbench)
> library(caret)
> library('kernlab')
> data(BreastCancer)
> db <- na.omit(BreastCancer)
> db$Label[db$Class == "benign"] <- -1
> db$Label[db$Class == "malignant"] <- 1
> x <- data.matrix(db[,2:10])
> y <- data.matrix(db[,12])
> # PCA
> trans <- prcomp(x)
> PC <- predict(trans, x)
> screeplot(trans, type = "l", npcs = 9, main = "Variâncias das componentes orde
```

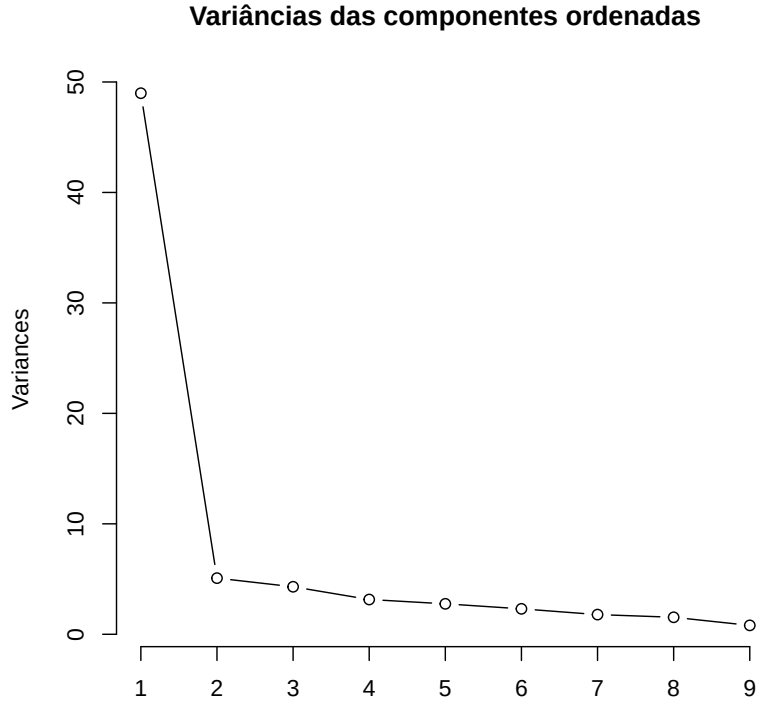

Figure 1: Gráfico da variância das componentes ordenadas

```
+   x_test <- x[i,]
+   y_test <- y[i, ]
+   svm <- ksvm(x_train,y_train,type='C-bsvc',kernel='rbfdot',kpar=list(sigma=0.
+   y_hat <- predict(svm, x_test)*1
+   accuracy[j] <- sum((y_hat == y_test)*1)/length(y_hat)
+
+   accuracy_fold[j, 1] <- sum((y_hat[y_hat==-1] == y_test[y_hat==-1])*1)/length
+   accuracy_fold[j, 2] <-  sum((y_hat[y_hat==-1] == y_test[y_hat==-1])*1)/lengt
+   j <- j +1
+ }

> print(accuracy_fold)

           [,1]      [,2]
 [1,] 1.0000000 1.0000000
 [2,] 1.0000000 1.0000000
 [3,] 1.0000000 1.0000000
 [4,] 0.9523810 0.9523810
 [5,] 1.0000000 1.0000000
 [6,] 0.9795918 0.9795918
 [7,] 1.0000000 1.0000000
 [8,] 1.0000000 1.0000000
 [9,] 1.0000000 1.0000000
[10,] 1.0000000 1.0000000
```

Em sequência, abaixo é apresentada a média da acurácia para cada classe considerando os 10 folds:

```
> print(apply(accuracy_fold, 2, mean))

[1] 0.9931973 0.9931973
```

Por fim, mostramos também a acurácia média total, considerando ambas as classes:

```
> print(mean(accuracy))

[1] 0.9736928
```

5 Discussão

Neste exercício escolheu apenas 2 componentes principais, pois essa na figura 1 as duas primeiras componentes apresentam a maior variabilidade. Com o experimento foi possível demonstrar que essa trativa funciona, pois obtivemos uma acurácia média de $\approx 99\%$.

Pode-se afirmar que os parâmetros do kernel da SVM foram escolhidos corretamente, já que obtivemos desempenho muito alto na classificação, com valores muito próximos a 100%.

Ao final do experimento, novamente, pode-se validar os conceito de PCA e SVM. Além disso, conseguimos reduzir a complexidade do problema *Brest Cancer* e obtermos uma acurácia de 97%.