Exercício 8 - KNN

Reconhecimento de Padrões - UFMG

Guilherme Capanema de Barros (28 de Outubro de 2017)

1 Introdução

O KNN (*K nearest neighbours*) é um método de classificação determinístico que calcula a distância entre um ponto no espaço e os seus K vizinhos mais próximos.

O algoritmo pode ser implementado de diversas formas. Neste trabalho, a métrica de distância utilizada foi a **distância euclidiana**, e a classificação foi do tipo **maioria simples**: se a maioria dos K vizinhos mais próximos de um ponto for de uma determinada classe, o ponto pertencerá a essa classe.

2 Implementação

O KNN foi implementado na função abaixo. Ela recebe uma amostra a ser classificada, todos os dados de treinamento, os rótulos dos dados de treinamento e o parâmetro K.

```
1  euc.dist <- function(x1, x2) sqrt(sum((x1 - x2) ^ 2))
2
3  myKNN <- function(amostra, dados, rotulos, K) {
4
5      distancias = apply(dados, 1, euc.dist, amostra)
6      vizinhos = order(distancias)[1:K]
7
8      if (sum(rotulos[vizinhos]==1) > sum(rotulos[vizinhos
           ]==-1)) {
9          classe = 1
10     } else {
11         classe = -1
12     }
13     return(classe)
14 }
```

3 Experimentos

O ajuste do parâmetro K pode gerar uma classificação adequada, *underfitting* (se K for muito alto) ou *overfitting* (se K for muito baixo).

3.1 Underfitting

Para demonstrar uma situação de *underfitting*, foi gerada uma base de dados sintética em espiral. O parâmetro definido foi $K = 20$. Os resultados estão exibidos na Fig. 1

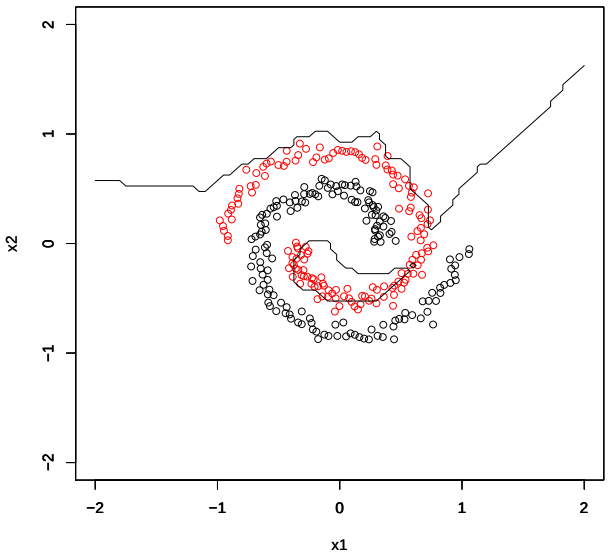

Figura 1: *Underfitting* para $K = 20$

Abaixo, o código usado para gerar essa visualização:

```
1  library(mlbench)
2  source('./myKNN.r')
3  # Gera os dados sint ticos
4
5  data <- mlbench.spirals(300,1,0.05)
6
7  classe1 <- data$x[data$classes==1,]
8  classe2 <- data$x[data$classes==2,]
9  dados <- data$x
10 dados[dados==2] <- -1
11 rotulos <- data$classes
12
13 # Plotando os dados
14 xlim <- c(-2,2)
15 ylim <- xlim
16
17 plot(classe1[,1], classe1[,2], xlim=xlim, ylim=ylim, xlab
       ='x1', ylab='x2')
18 par(new=T)
```

```
19 plot(classe2[,1], classe2[,2], xlim=xlim, ylim=ylim, col=
       'red', xlab='x1', ylab='x2')
20
21 # Classificador
22
23 # Plotando a separacao
24 K = 20
25 intervalo = seq(from=xlim[1],to=xlim[2],by=0.05)
26 grid = expand.grid(intervalo,intervalo)
27
28 results = apply(grid, 1, myKNN, dados, rotulos, K)
29 z = matrix(results, ncol=length(intervalo), nrow=length(
       intervalo))
30
31
32 par(new=T)
33 contour(intervalo, intervalo, z, xlim=xlim, ylim=ylim,
       drawlabels=F, nlevels=1)
```

3.2 Overfitting

Para demonstrar uma situação de *overfitting*, foi gerada uma base de dados sintética de duas classes. O parâmetro definido foi $K = 2$. Os resultados estão exibidos na Fig. 2

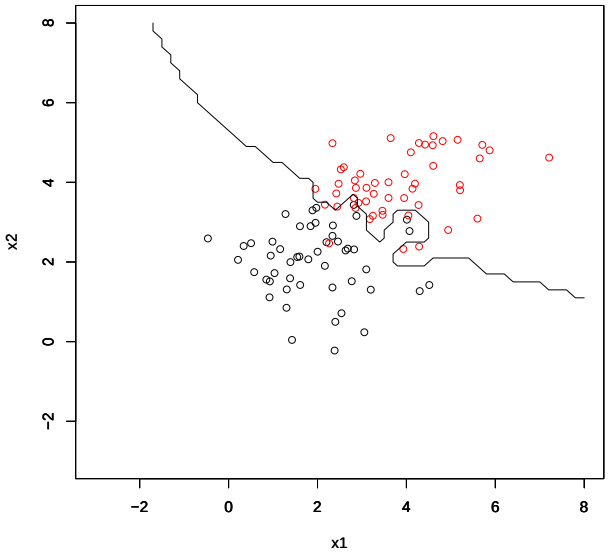

Figura 2: *Overfitting* para $K = 2$

Abaixo, o código usado para gerar essa visualização:

```
1  source('./myKNN.r')
2  # Gera os dados sint ticos
3  media1 = 2
4  sigma1 = 1
5  xClasse1 = rnorm(n=50, media1, sigma1)
6  yClasse1 = rnorm(n=50, media1, sigma1)
7  classe1 = cbind(xClasse1, yClasse1)
8  rotulos1 = rep(-1, nrow(classe1))
9
10 media2 = 4
11 sigma2 = 1
12 xClasse2 = rnorm(n=50, media2, sigma2)
13 yClasse2 = rnorm(n=50, media2, sigma2)
14 classe2 = cbind(xClasse2, yClasse2)
15 rotulos2 = rep(1, nrow(classe2))
16
17 dados = rbind(classe1, classe2)
18 rotulos = c(rotulos1, rotulos2)
19
20 # Plotando os dados
21 xlim=c(-3,8)
22 ylim=c(-3,8)
23
24 plot(classe1[,1], classe1[,2], xlim=xlim, ylim=ylim, xlab
       ='x1', ylab='x2')
25 par(new=T)
26 plot(classe2[,1], classe2[,2], xlim=xlim, ylim=ylim, col=
       'red', xlab='x1', ylab='x2')
27
28 # Classificador
29
30 # Plotando a separacao
31 K = 2
32 intervalo = seq(from=-3,to=8,by=0.2)
33 grid = expand.grid(intervalo,intervalo)
34
35 results = apply(grid, 1, myKNN, dados, rotulos, K)
36 z = matrix(results, ncol=length(intervalo), nrow=length(
       intervalo))
37
38
39 par(new=T)
40 contour(intervalo, intervalo, z, xlim=xlim, ylim=ylim,
       drawlabels=F, nlevels=1)
```

4 Conclusão

Apesar da simplicidade, o KNN demonstra resultados razoáveis nas bases de dados testadas. O ajuste de K pode ser feito, por exemplo, usando a validação

cruzada *10-fold*.

Para bases de dados desbalanceadas, é necessário cuidado no ajuste de K.