Universidade do Minho
Escola de Engenharia
Mestrado Integrado em Engenharia Informática

Unidade Curricular de Computação Natural

Ano Letivo de 2017/2018

Relatório dos Exercícios de SVM

Bruno Pereira (a75135), João Coelho (a74859), Luís Fernandes (a74748), Maria Ana de Brito (a73580)

Abril, 2018

CN

Índice

Índice de Figuras

Índice de Tabelas

1. Introdução

Este trabalho prático consiste na realização de três exercícios relacionados com a temática de *Support Vector Machine* (SVM). Para cada um deles é explicado o que foi feito em relação ao *dataset* utilizado (*HVAC24hS16-11-2016—0*) e quais as observações utilizadas como *input* e *output* de treino, bem como *input* de teste. Através dos resultados obtidos das previsões executadas, tecem-se conclusões acerca do processo realizado.

Desta maneira, este relatório divide-se em 7 partes. Primeiramente, faz-se uma contextualização do tema a ser trabalhado e de seguida apresenta-se o problema de investigação abordado. Cada um dos três exercícios representa uma secção, onde se justifica a sua resolução. Por fim, é efetuada uma avaliação crítica do trabalho realizado pelo grupo e são, ainda, apresentadas as referências bibliográficas utilizadas durante a sua elaboração.

Áreas de Aplicação: SVM, R, *HVAC*.
Palavras-chave: *dataset*, MAPE, SVM, *Kernel*, *Epsilon*

2. Contextualização

Os exercícios sugeridos no enunciado do trabalho prático remetem à utilização de *Support Vector Machine* de forma a realizar previsões sobre os dados de teste fornecidos no *dataset*. Este remete a dados sobre climatização, sendo que a sigla *HVAC* corresponde a *Heating, Ventilating and Air Conditioning*, isto é, aquecimento, ventilação e ar condicionado. Estes três elementos estão intimamente relacionados com essa tecnologia.

Por outro lado, SVM faz parte da Aprendizagem Supervisionada e pode ser usado para fazer previsões ou para classificar instâncias. Um modelo é construído e pode ser representado por um conjunto de pontos no espaço separados por uma linha, isto é, um hiperplano ou vetor.

Na resolução dos exercícios propostos vamos aplicar este conceito usando a linguagem R e os conceitos lecionados nas aulas teórico-práticas.

3. Apresentação do Caso de Estudo

Os dados do *dataset* fornecido estão inseridos na área de *HVAC* (*Heating, Ventilating and Air Conditioning)*, isto é, na área de climatização de meios de forma a que permaneçam num intervalo de temperatura confortáveis para os seres que estão lá inseridos. Assim, esta tecnologia destina-se principalmente ao conforto ambiental interior de edifícios ou veículos. Para cada uma das datas da observação foram retirados valores do interior desse espaço de hora em hora.

Assim, recorrendo aos SVM pretendemos fazer previsões de valores tendo por base o treino com outras observações em dias distintos.

4. Exercício 1

Primeiramente utilizamos o *dataset* original que nos foi fornecido, *HVAC24hS16-11-2016—0*, para determinar qual o resultado do teste. Assim, com 10 registos no *training input* e no *training output* e com um registo no teste *input* distinto dos de treino, executamos o algoritmo e obtivemos o seguinte resultado:

```
> ex1_original
       1
507.6597
```

Figura 1 - Resultado do primeiro exercício com *dataset* original

De seguida, tal como o enunciado do exercício sugere, retirámos os três primeiros *inputs* e *outputs* de treino do *dataset*. Além disso, a quarta observação foi utilizada como *input* de teste, logo esta também foi retirada dos dados de treino.

Deste modo, ficamos com seis registos como *input* e *output* de treino, como era pretendido. O resultado obtido com esta execução do SVM está representado na seguinte figura:

```
> runSVM
       1
524.2618
```

Figura 2 - Resultado do primeiro exercício com *dataset* alterado

Como o *input* de teste era uma observação do conjunto de dados original, sabemos qual o valor correto do resultado através do *train output*. Assim, tendo em conta que este valor é de 525.1277, o resultado obtido (524.2618) desvia-se 0.8659 do correto. Pode-se concluir que a previsão realizada foi satisfatória.

5. Exercício 2

Neste exercício realizamos várias execuções do SVM com diferentes números de observações no conjunto de dados de treino. Como o *dataset* fornecido contém 10 registos diferentes de *input*, repetimos o processo de treino 4 vezes para 6, 7, 8 e 9 observações no *training input* e *output*.

Deste modo, ignoram-se sucessivamente as últimas *n* observações, isto é, se estivermos a utilizar 8 observações de treino, ignoramos os últimos dois registos. Tendo em conta que neste exercício nunca se utilizam as 10 observações do *dataset* original, escolhemos a última linha, ou seja, a última observação, como *test input*.

Assim, o resultado correto do *test input* é 725.9484, podendo, então, calcular-se o valor do erro percentual médio (MAPE) através da seguinte fórmula:

$$MAPE = \frac{valor\ real - valor\ previsto}{valor\ real}$$

Pode-se verificar todo este processo, feito em R, no ficheiro *SVM.R.*

Os resultados obtidos estão representados na tabela abaixo:

Tabela 1 - Resultados obtidos no exercício 2

Número de observações	Resultado obtido	MAPE
6	491.1537	0.3234
7	497.0896	0.3153
8	493.7768	0.3198
9	494.6759	0.3186
Média	494.174	0.3193

Constatamos que a média do valor do erro é alta (cerca de 32%), uma vez que as observações usadas como treino possuem como *output* valores muito distintos do valor correto da última observação. Esta com o *output* de cerca de 725 pode ser considerada um *outlier* em relação aos outros *outputs,* cuja média ronda os 503.5566. Por esta razão, verificamos que as execuções do SVM para as 6, 7, 8 e 9 observações produziram um MAPE elevado, pois os resultados previstos não se aproximaram do valor verdadeiro, dado que a sua média, tal como consta na tabela, é de 494.174.

6. Exercício 3

O exercício 3 requer que sejam efetuados testes com a métrica MAPE e variando o número de inputs do *dataset* (tal como no exercício 2), mas também alterando o *epsilon* e diferentes funções de *kernel*. Antes de avançar para a análise dos resultados do exercício 3, será necessário verificar os conceitos de *epsilon* e funções de *kernel*.

As funções de *kernel* caracterizam-se por retornar o produto interno entre dois pontos no espaço considerado. Assim, alternando estas funções é possível obter formas diferentes de calcular a distância entre pontos, podendo variar as divisões do espaço. As funções disponibilizadas pelo erro são:

- *linear*;
- *polynomial*;
- *radial basis;*
- *sigmoid*;

em que a única alteração entre elas é a maneira como calculam a distância entre pontos.

Relativamente ao *epsilon*, o valor deste define uma margem de tolerância em que não existirão penalizações quando ocorrem erros. Resumindo, quanto mais alto o valor de *epsilon*, mais erros são admitidos na solução, ou seja, um *epsilon* que tenda para 0, terá que cada erro será muito penalizado, fazendo com que a solução crie tantos *SVs* como número de instâncias do *dataset*.

O código R elaborado para resolver este exercício encontra-se nos anexos, bem como as alterações que este sofreu.

Assim, o próximo passo consistirá em efetuar testes para os diferentes componentes mencionados no início. De seguida, apresentam-se as tabelas referentes ao MAPE para estes testes.

Tabela 2 - Resultados do MAPE para 6 observações

		epsilon			
		0.1	0.5	0.9	Média
kernel	linear	0.356331	0.320181	0.365108	0.347207
	polynomial	0.363606	0.336841	0.300434	0.333627
	radial	0.321574	0.31527	0.301898	0.312914
	sigmoid	0.353509	0.324272	0.296645	0.324809
	Média	0.348755	0.324141	0.316021	0.329639

Tabela 3 - Resultados do MAPE para 7 observações

		epsilon			
		0.1	0.5	0.9	Média
kernel	linear	0.343067	0.313078	0.297424	0.317857
	polynomial	0.323192	0.317619	0.298127	0.312979
	radial	0.315191	0.31367	0.302796	0.310552
	sigmoid	0.345363	0.316837	0.296589	0.319596
	Média	0.331703	0.315301	0.298734	0.315246

Tabela 4 - Resultados do MAPE para 8 observações

		epsilon			
		0.1	0.5	0.9	Média
kernel	linear	0.317421	0.315015	0.299424	0.31062
	polynomial	0.295167	0.29949	0.296914	0.29719
	radial	0.318436	0.31439	0.30478	0.312535
	sigmoid	0.328166	0.3218	0.298899	0.316288
	Média	0.314797	0.312673	0.300004	0.309158

Tabela 5 - Resultados do MAPE para 9 observações

		epsilon			
		0.1	0.5	0.9	Média
kernel	linear	0.301406	0.30235	0.298509	0.300755
	polynomial	0.302316	0.299233	0.298779	0.300109
	radial	0.313616	0.313839	0.305958	0.311138
	sigmoid	0.313364	0.308796	0.298387	0.306849
	Média	0.307675	0.306054	0.300408	0.304713

De modo a analisar corretamente os resultados será necessário, para diferentes observações, considerar um parâmetro e examiná-lo minuciosamente.

Relativamente ao *epsilon*, independentemente do número de observações consideradas, quando o *epsilon* apresenta valores mais elevados, o erro é inferior. Esta situação ocorre, pois, há uma maior folga na tolerância de erros e não será construído um modelo que se adapte totalmente ao *dataset* (*overfitting*), mas sim um modelo capaz de suportar novos dados. Os valores do erro, para o epsilon, tendem a diminuir à medida que se aumenta o número de observações, sendo que, no entanto, o valor mais baixo registado ocorre quando foram feitas 7 observações.

Com as funções de *kernel*, alterando as mesmas, não existe um padrão tão óbvio, tal como aconteceu com o *epsilon*, sendo que para 6 e 7 observações a função de *kernel* que melhor representa é a *radial basis*, enquanto que com 8 e 9 observações é a *polynomial* que se adequa melhor. Visto que estas funções determinam como é calculada a distância entre pontos, podemos, por exemplo, concluir que os valores referentes a 9 observações são identificados corretamente por uma função polinomial. Contrariamente ao que aconteceu com o *epsilon*, é com 9 observações que acontecem os melhores resultados.

Concluindo, e considerando os resultados (médias finais), os valores mais baixos de erro foram obtidos com 9 observações, no entanto o valor do erro mais baixo que foi registado com 8 observações, *epsilon* = 0.1 e a função de *kernel* utilizada era polinomial.

De modo a ser possível efetuar a comparação com o exercício 2, de seguida apresenta-se uma tabela com os melhores resultados para cada uma das observações consideradas.

Tabela 6 - Resumo dos resultados do exercício 3

Número de observações	Resultado	MAPE
6	510.5995555	0.296644856
7	510.6401511	0.296588935
8	511.6721497	0.295167348
9	509.3348511	0.298386996
Média	510.5616768	0.296697034

Todos estes valores, exceto o que envolve as 8 observações (polinomial e *epsilon*=0.1), são correspondentes a um *epsilon*= 0.9 e função de *kernel* utilizada é *sigmoid*, levando à conclusão de que apesar dos resultados gerais apontarem para, geralmente, uma superioridade das funções de *kernel radial basis* e *polynomial*, temos que os melhores resultados são obtidos para quando a função de *kernel* é a *sigmoid*.

7. Avaliação Crítica dos Exercícios

O primeiro exercício consistia em fazer uma previsão através de um processo de treino com apenas seis observações. O resultado obtido foi satisfatório, visto que a diferença em comparação com o resultado verdadeiro não foi substancial.

Por outro lado, no segundo exercício foram feitas várias experiências usando um número diferente de observações para o conjunto de treino. Neste caso, verificou-se que o *MAPE* apresentava um valor elevado, que já foi justificado na secção 5.

A diferença entre o sucesso do resultado do primeiro exercício e do insucesso dos resultados do segundo exercício deve-se ao facto de que no primeiro exercício a observação usada para *test input* possuía valores semelhantes às observações do conjunto de treino. Deste modo, a previsão feita pelo algoritmo de SVM foi bastante correta, pois os dados de *training* aproximavam-se dos dados de teste. Isto não ocorreu no segundo exercício, uma vez que foi usada uma observação com valores bastante diferentes dos restantes registos para *input test*. Assim, o treino realizado com esses dados não permitiu produzir uma previsão correta acerca do resultado de teste.

Analisando os resultados do exercício 3 e comparando-os com o exercício 2, vemos que há uma clara melhoria em termos do erro. No exercício 2, os valores de erro rondavam os 32%, enquanto que no exercício 3, o erro não supera os 30%, existindo uma melhoria de cerca de 2%. Esta mudança no erro, faz com que o valor do resultado seja alterado, sendo que no caso do exercício 3 a média dos resultados seja, aproximadamente, 510, enquanto que no exercício 2 ande por volta dos 494. Como o erro do exercício 3 é inferior, podemos concluir que o resultado por si apresentado se aproxima mais do valor real do que o valor de resultado exibido pelo segundo exercício.

A razão das melhorias apresentadas no exercício 3 deve-se unicamente à alteração dos parâmetros epsilon e funções de kernel. Nos exercícios anteriores, os valores apresentados por estes parâmetros eram os valores dados por omissão pelo R, no entanto, no terceiro exercício, com a alteração dos parâmetros, as SVMs já terão capacidades de se adaptar melhor ao dataset, visto que a sua tolerância aos erros muda, bem como o cálculo da distância entre os pontos e as divisões não estão limitadas aos valores de omissão do R.

Referências

- Artificiencia.com, (2017). *Máquina de vetores de suporte.* [online] Available at: http://artificiencia.com/aprenda/maquina-de-vetores-de-suporte/
- Ibm.com. *Como o SVM Funciona*. [online] Available at: https://www.ibm.com/support/knowledgecenter/pt-br/SS3RA7_17.1.0/modeler_mainhelp_client_ddita/clementine/svm_howwork.html
- Webarcondicionador.com.br, (2015). *Você sabe o que significa HVAC, HVAC-R, AVAC e AVAC-R?*. [online] Availabe at: http://www.webarcondicionado.com.br/voce-sabe-o-que-significa-hvac-hvac-r-avac-e-avac-r

Lista de Siglas e Acrónimos

SVM *Support Vector Machine*

MAPE *Mean Absolute Percentage Error*

HVAC *Heating, Ventilating and Air Conditioning*

Anexos

Em anexo encontram-se os conteúdos dos ficheiros R que foram usados na elaboração deste trabalho prático.

O primeiro anexo corresponde ao ficheiro *SVM*.R que apresenta a função *calculaMAPE*, onde se realizam os cálculos necessários do erro percentual e se filtram quais as observações pretendidas para o conjunto de dados de treino.

O segundo anexo, *runSVM.R*, é um *script* onde se encontram as chamadas das funções com os devidos parâmetros para cada um dos exercícios propostos.

I. Anexo 1 – SVM.R

```
runSVM <- function(file_name){
  library(e1071)
  library(xlsx)
  trainingInput <- read.xlsx(file_name, 1, header=T)
  ncols.trainingInput <- ncol(trainingInput)
  trainingInput <- trainingInput[ , 2 : ncols.trainingInput]

  trainingOutput<- read.xlsx(file_name, 2, header=T)
  trainingOutput <- matrix(c(trainingOutput[ , 2]), ncol=1)

  testData <- read.xlsx(file_name, 3, header=T)
  ncols.testData <- ncol(testData)
  testData <- testData[, 2 : ncols.testData]

  colnames(trainingInput) <- paste0("x", 1:ncol(trainingInput))
  colnames(trainingOutput) <- paste0("y", 1:ncol(trainingOutput))

  trainingdata <- cbind(trainingInput,trainingOutput)

  m <- svm(trainingInput,trainingOutput)
  pred <- predict(m, testData)

  return (pred)
}

calculaMAPE <- function(file_name, num_obs, epsilon = 0.1, kernel =
"linear") {
  library(e1071)
  library(xlsx)

  trainingInput <- read.xlsx(file_name, 1, header=T)
  ncols.trainingInput <- ncol(trainingInput)
  trainingInput <- trainingInput[ , 2 : ncols.trainingInput]
```

```
  trainingOutput<- read.xlsx(file_name, 2, header=T)
  trainingOutput <- matrix(c(trainingOutput[ , 2]), ncol=1)
  real_value <- trainingOutput[10]

  testData <- read.xlsx(file_name, 3, header=T)
  ncols.testData <- ncol(testData)
  testData <- testData[, 2 : ncols.testData]

  colnames(trainingInput) <- paste0("x", 1:ncol(trainingInput))
  colnames(trainingOutput) <- paste0("y", 1:ncol(trainingOutput))

  trainingInput <- trainingInput[1:num_obs, ]
  trainingOutput <- trainingOutput[1:num_obs, ]

  m <- svm(trainingInput,trainingOutput, epsilon = epsilon, kernel =
kernel)
  pred <- predict(m, testData)

  mape <- (real_value - pred)/real_value
  print(paste("Erro Médio: ", mape))
  print(paste("Resultado: ", pred))

  #write.xlsx(pred, file=file_name, sheetName= paste("Test Result",
x), col.names=F, row.names=F, append=T)

  return(pred)
}
```

II. Anexo 2 – runSVM.R

```
file <- "C:\\Users\\PC\\Desktop\\SVM\\HVAC24hS16-11-2016--0.xls"

# Exercício 1
ex1_original <- runSVM(file)

# Exercício 2
ex2 <- calculaMAPE(file, 6)
ex2 <- calculaMAPE(file, 7)
ex2 <- calculaMAPE(file, 8)
ex2 <- calculaMAPE(file, 9)

# Exercício 3
# Testes para treino com 6 instruções
ex3 <- calculaMAPE(file, 6, epsilon = 0.1, kernel = "linear")
ex3 <- calculaMAPE(file, 6, epsilon = 0.5, kernel = "linear")
ex3 <- calculaMAPE(file, 6, epsilon = 0, kernel = "linear")

ex3 <- calculaMAPE(file, 6, epsilon = 0.1, kernel = "polynomial")
ex3 <- calculaMAPE(file, 6, epsilon = 0.5, kernel = "polynomial")
ex3 <- calculaMAPE(file, 6, epsilon = 0.9, kernel = "polynomial")

ex3 <- calculaMAPE(file, 6, epsilon = 0.1, kernel = "radial")
ex3 <- calculaMAPE(file, 6, epsilon = 0.5, kernel = "radial")
ex3 <- calculaMAPE(file, 6, epsilon = 0.9, kernel = "radial")

ex3 <- calculaMAPE(file, 6, epsilon = 0.1, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 6, epsilon = 0.5, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 6, epsilon = 0.9, kernel = "sigmoid")

# Testes para treino com 7 instruções
ex3 <- calculaMAPE(file, 7, epsilon = 0.1, kernel = "linear")
ex3 <- calculaMAPE(file, 7, epsilon = 0.5, kernel = "linear")
ex3 <- calculaMAPE(file, 7, epsilon = 0.9, kernel = "linear")
```

```
ex3 <- calculaMAPE(file, 7, epsilon = 0.1, kernel = "polynomial")
ex3 <- calculaMAPE(file, 7, epsilon = 0.5, kernel = "polynomial")
ex3 <- calculaMAPE(file, 7, epsilon = 0.9, kernel = "polynomial")

ex3 <- calculaMAPE(file, 7, epsilon = 0.1, kernel = "radial")
ex3 <- calculaMAPE(file, 7, epsilon = 0.5, kernel = "radial")
ex3 <- calculaMAPE(file, 7, epsilon = 0.9, kernel = "radial")

ex3 <- calculaMAPE(file, 7, epsilon = 0.1, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 7, epsilon = 0.5, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 7, epsilon = 0.9, kernel = "sigmoid")

# Testes para treino com 8 instruções
ex3 <- calculaMAPE(file, 8, epsilon = 0.1, kernel = "linear")
ex3 <- calculaMAPE(file, 8, epsilon = 0.5, kernel = "linear")
ex3 <- calculaMAPE(file, 8, epsilon = 0.9, kernel = "linear")

ex3 <- calculaMAPE(file, 8, epsilon = 0.1, kernel = "polynomial")
ex3 <- calculaMAPE(file, 8, epsilon = 0.5, kernel = "polynomial")
ex3 <- calculaMAPE(file, 8, epsilon = 0.9, kernel = "polynomial")

ex3 <- calculaMAPE(file, 8, epsilon = 0.1, kernel = "radial")
ex3 <- calculaMAPE(file, 8, epsilon = 0.5, kernel = "radial")
ex3 <- calculaMAPE(file, 8, epsilon = 0.9, kernel = "radial")

ex3 <- calculaMAPE(file, 8, epsilon = 0.1, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 8, epsilon = 0.5, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 8, epsilon = 0.9, kernel = "sigmoid")

# Testes para treino com 9 instruções
ex3 <- calculaMAPE(file, 9, epsilon = 0.1, kernel = "linear")
ex3 <- calculaMAPE(file, 9, epsilon = 0.5, kernel = "linear")
ex3 <- calculaMAPE(file, 9, epsilon = 0.9, kernel = "linear")

ex3 <- calculaMAPE(file, 9, epsilon = 0.1, kernel = "polynomial")
ex3 <- calculaMAPE(file, 9, epsilon = 0.5, kernel = "polynomial")
ex3 <- calculaMAPE(file, 9, epsilon = 0.9, kernel = "polynomial")

ex3 <- calculaMAPE(file, 9, epsilon = 0.1, kernel = "radial")
ex3 <- calculaMAPE(file, 9, epsilon = 0.5, kernel = "radial")
```

```
ex3 <- calculaMAPE(file, 9, epsilon = 0.9, kernel = "radial")

ex3 <- calculaMAPE(file, 9, epsilon = 0.1, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 9, epsilon = 0.5, kernel = "sigmoid")
ex3 <- calculaMAPE(file, 9, epsilon = 0.9, kernel = "sigmoid")
```