KNN - K Nearest Neighbor

ENG121 - Reconhecimento de Padrões

Autor: Matheus Castro Silva

RESUMO

O presente trabalho tem como objetivo testar, o modelo de KNN - Nearest Neighbor, para a base de dados sintética produzida em sala de aula.

Palavras Chaves **- algoritmos, KNN, classificação, reconhecimento de padrões, nearest.**

I. INTRODUÇÃO

O presente trabalho busca introduzir o teste do classificador *K Nearest Neighbor* - KNN, para uma base de dados Sintética produzida em sala de aula. Realizar uma comparação entre o modelo do KNN para várias entradas de *k* e verificar as condições de underfitting e overfitting.

II. DEFINIÇÃO DO PROBLEMA

Utilizando um conjunto de base de dados sintética feito em sala de aula identifique os casos de overfitting e underfitting variando o parâmetros K presente no algoritmo *K Nearest Neighbor* - KNN.

III. MODELO KNN

O KNN foi proposto por Fukunaga e Narendra em 1975. É um dos classificadores mais simples de ser implementado, de fácil compreensão e ainda hoje pode obter bons resultados dependendo de sua aplicação. Antes de iniciar, caso você não tenha afinidade com o problema de classificação, sugiro que leia nosso post sobre classificadores. Agora, sem mais delongas, vamos ao que interessa.

A ideia principal do KNN é determinar o rótulo de classificação de uma amostra baseado nas amostras vizinhas advindas de um conjunto de treinamento. Nada melhor do que um exemplo para explicar o funcionamento do algoritmo como o da Figura 1, na qual temos um problema de classificação com dois rótulos de classe e com k = 7. No exemplo, são aferidas as distâncias de uma nova amostra, representada por uma estrela, às demais amostras de treinamento, representadas pelas bolinhas azuis e amarelas. A variável *k* representa a quantidade de vizinhos mais próximos que serão utilizados para averiguar de qual classe a nova amostra pertence. Com isso, das sete amostras de treinamento mais próximas da nova amostra, 4 são do rótulo A e 3 do rótulo B. Portanto, como existem mais vizinhos do rótulo A, a nova amostra receberá o mesmo rótulo deles, ou seja, A.

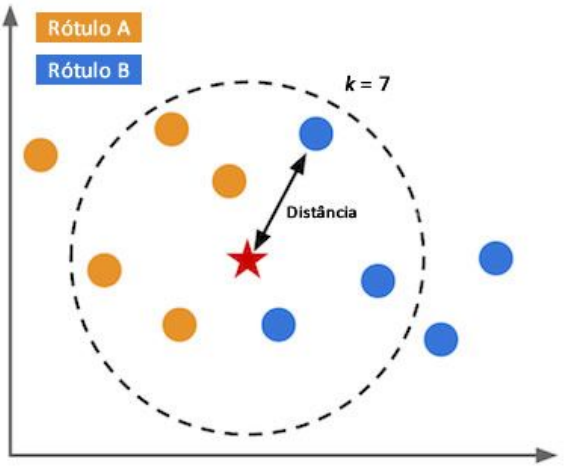

Figura 1: exemplo de classificação do KNN com dois rótulos de classe e k = 7

Dois pontos chaves que devem ser determinados para aplicação do KNN são: a métrica de distância e o valor de k. Para métrica de distância a mais utilizada é a distância Euclidiana, descrita por:

$$P = \sqrt{(P_1 - Q_1)^2 + \ldots + (P_n - Q_n)^2} = \sqrt{\sum_{i=1}^{n} (P_i - Q_i)^2}$$

onde $P = (P_i, \ldots, P_n)$ *e* $Q = (Q_i, \ldots, Q_n)$ são dois pontos n-dimensionais. No exemplo da Figura 1, essa distância seria calculada entre as bolinhas (azuis e laranjas) e a estrela (a nova entrada). Como o exemplo é 2D, cada uma cada ponto teria seu valor em x e em y. Para problemas com dimensões maiores a abordagem é a exatamente a mesma.

Em relação ao valor k, não existe um valor único para a constante, a mesma varia de acordo com a base de dados. É recomendável sempre utilizar valores ímpares/primos, mas o valor ótimo varia de base para base. Dependendo do seu problema você pode utilizar um algoritmo de otimização (PSO, GA, DE…) para encontrar o melhor valor para o seu problema. Todavia, você pode deixar o desempenho geral do modelo bem lento na etapa de seleção de k. Outra maneira e simplesmente testar um conjunto de valores e encontrar k empiricamente.

Resumidamente, a grande vantagem do KNN é sua abordagem simples de ser compreendida e implementada.

Todavia, calcular distância é tarefa custosa e caso o problema possua grande número de amostras o algoritmo pode consumir muito tempo computacional. Além disso, o método é sensível à escolha do k. Na sequência é apresentado um pseudocódigo do algoritmo:

```
1  inicialização:
2      Preparar conjunto de dados de entrada e saída
3      Informar o valor de k;
4  para cada nova amostra faça
5      Calcular distância para todas as amostras
6      Determinar o conjunto das k's distâncias mais próximas
7      O rótulo com mais representantes no conjunto dos k's
8      vizinhos será o escolhido
9  fim para
10 retornar: conjunto de rótulos de classificação
```

Figura 2: Pseudocódigo do algoritmo KNN.

IV. TREINAMENTO E TESTE

a. KNN

A base de dados deve ser dividida em dois conjuntos, um de treinamento e outro para os testes, sendo 70% para treinamento e 30% para teste.

V. RESULTADOS DE SEPARAÇÃO

Utilizando o conjunto de amostras referente às duas classes abaixo, foi feito a superfície de separação utilizando KNN.

- **K = 1**

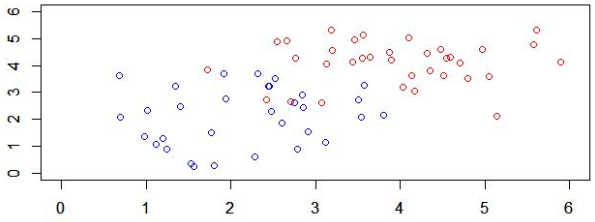

Figura 3: Ilustração das amostras de treinamento no espaço 2D referente a duas classes.

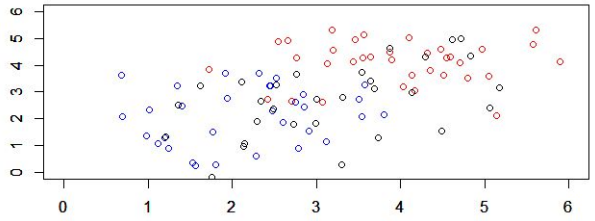

Figura 4: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes.

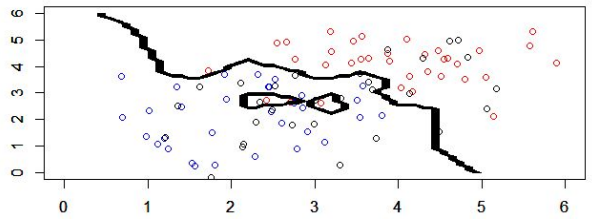

Figura 5: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes e a curva de classificação.

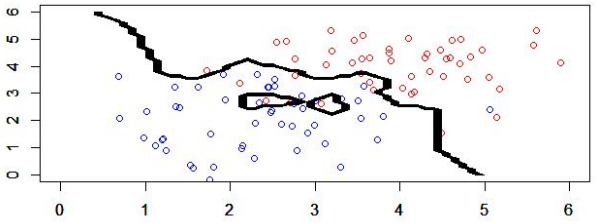

Figura 6: Ilustração do resultado da classificação das amostras de teste em 2D.

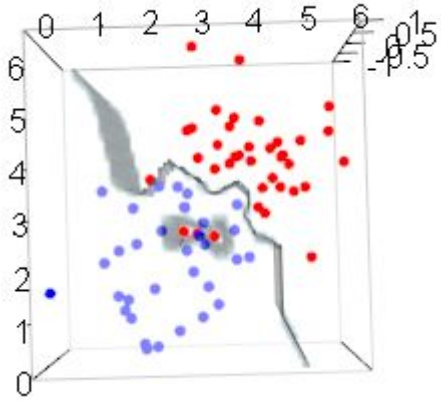

Figura 7: Ilustração do resultado da classificação das amostras de teste em 3D.

"Porcentagem: 76.6666666666667 %"

- **K = 5**

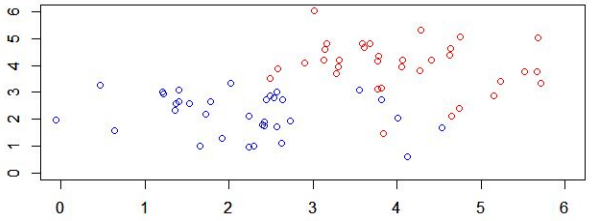

Figura 8: Ilustração das amostras de treinamento no espaço 2D referente a duas classes.

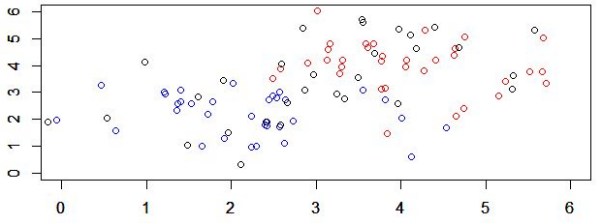

Figura 9: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes.

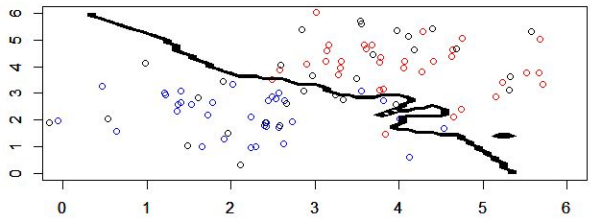

Figura 10: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes e a curva de classificação.

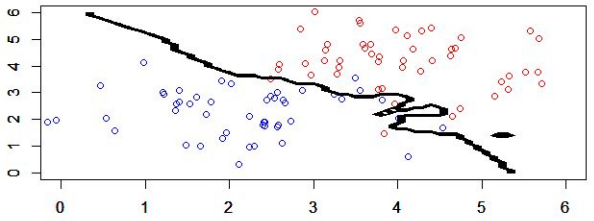

Figura 11: Ilustração do resultado da classificação das amostras de teste em 2D.

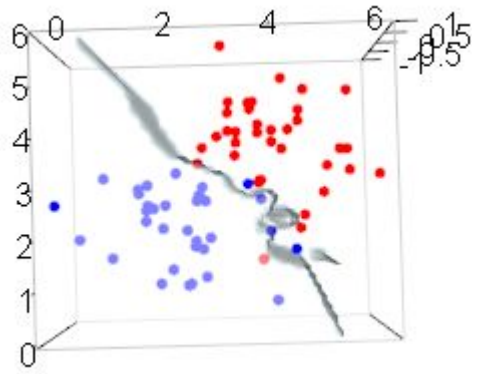

Figura 12: Ilustração do resultado da classificação das amostras de teste em 3D.

"Porcentagem: 93.3333333333333 %"

- K = **10**

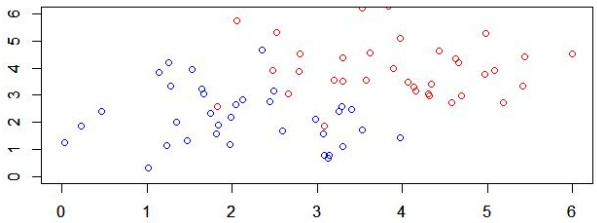

Figura 13: Ilustração das amostras de treinamento no espaço 2D referente a duas classes.

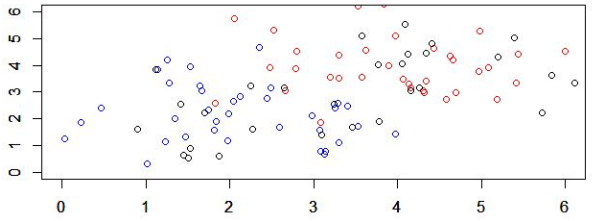

Figura 14: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes.

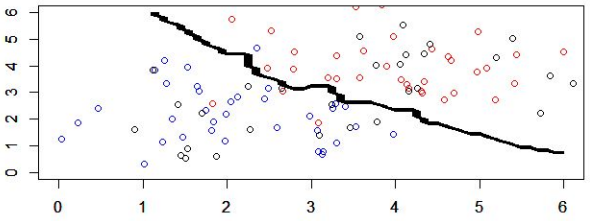

Figura 15: Ilustração das amostras de treinamento e teste no espaço 2D referente a duas classes e a curva de classificação.

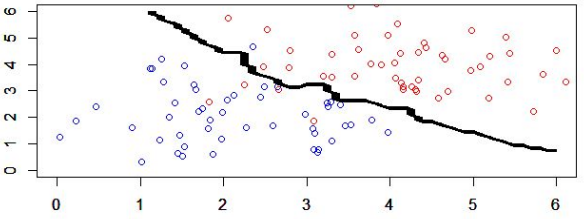

Figura 16: Ilustração do resultado da classificação das amostras de teste em 2D.

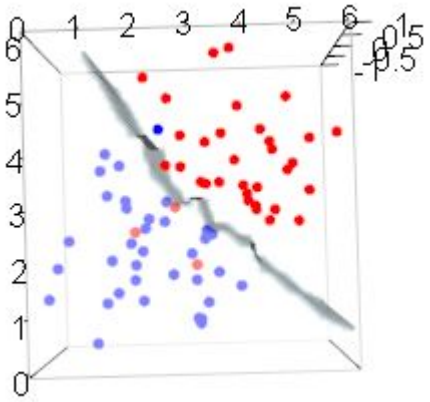

Figura 17: Ilustração do resultado da classificação das amostras de teste em 3D.

VI. COMPARAÇÃO DOS ERROS

a. KNN

Utilizando também o passo a passo do tópico de implementação III neste trabalho os resultados obtidos, considerando a variação das iterações foram:

Iterações	% Acerto
k = 1	76.66
k = 5	93.33
k = 10	96.66

O percentual de acerto perante os testes ficou entre 76% à 97%, considerando a técnica KNN um método bastante eficiente porém que ainda tem pontos a serem levados em consideração. Pois a variação do parâmetro K é visivelmente um dos fatores para a melhora do algoritmo, deixando-o com overfitting ou underfitting. Também este método pode ser custoso em alguns casos que há um grande número de amostras e necessita-se calcular a distância euclidiana de todas estas amostras, porém o resultado foi bastante satisfatório para os testes propostos.

VII. CONCLUSÕES

O método de vizinhos mais próximos (KNN) foi bastante satisfatório para encontrar a separação das amostras, mas também foi visto que o algoritmo nem sempre terá 100% de acerto, podendo apresentar resultados piores em alguns momentos, dependendo diretamente dos parâmetros da técnica.

Para obter uma melhora no algoritmo seria interessante aumentar o tamanho da base de dados de entrada para o modelo de treinamento e também realizar testes para encontrar o valor ideal para o parâmetro K, que para alguns casos pode ser muito custoso de obter o valor de K ideal para base de dados muito grandes.

Como foi visto, o método satisfez as expectativas do trabalho. Apresentou pontos positivos e negativos da técnica, e a escolha para utilização do método deve-se diretamente ao parâmetros K e o tamanho da base de dados disponível.

VIII. REFERÊNCIAS

[1] R tutorial, http://www.r-tutor.com/r-introduction, acessado 30/08/2017.

[2] Antônio de Pádua Braga - Notas de Aulas de Redes Neurais Artificiais e de Reconhecimento de Padrões, acessado 22/09/2017.

[3] KNN - K NEAREST NEIGHBOR http://www.computacaointeligente.com.br/algoritmos/knn-k-vizinhos-mais-proximos/ acessado 30/10/2017.