Implementação do algoritmo de classificação K-NN

Emanuel Giroldo Mazzer[1], **Leonardo Pontes Baiser**[1]

[1]DACOM – Universidade Tecnológica Federal do Paraná (UTFPR)
Caixa Postal 271 – 87301-899 – Campo Mourão – PR – Brazil

{emanuelgiroldo,lpbaiser}@gmail.com

Resumo. *Relata o procedimento para implementação do algoritmo de classificação K-NN (K-nearest neighbors algo- rithm) e resultados de testes empregados perante uma base de treinamento.*

Abstract. *Reported the procedure for implementing the K-NN classification algorithm (K-nearest neighbors algo- rithm) and test results used before a training base.*

1. Introdução

O algoritmo de classificação baseado nos vizinhos mais próximos (Nearest Neighbors - NN) é um método muito utilizado para reconhecimento de padrões, o seu conceito de funcionamento é descobrir o vizinho mais próximo de uma dada instância, no grupo de métodos de classificação está presente o algoritmo K-NN (K-Nearest Neighbors), neste algoritmo são encontrados os **k** vizinhos mais próximos do padrão de consulta, ao invés de apenas o vizinho mais próximo. Neste artigo apresentaremos o algoritmo K-NN implementado em *python* para classificação de um conjunto de treinamento e teste que representam meses do ano.

2. O algoritmo

Para a implementação, utilizou-se conhecimentos sobre o algoritmo K-NN, distância euclidiana e normalização dos dados utilizando Min-Max.

Data: Base de dados de treinamento e teste
Result: Taxa de acerto
Se faz a leitura dos dados de treinamento e testes.
Normaliza todos os dados utilizando Min-Max.
Faz a chamada da função **classificador** com os parâmetros: matriz de treinamento, matriz de teste, valor de k.
A função **classificador** percorre toda a matriz de teste e para cada instância da matriz teste é chamada a função **get_k_menores_distancias** que encontra as k menores distâncias, dentro da função **get_k_menores_distancias** é calculado a distância euclidiana entre duas instancias uma de treino e outra de teste.
Após o término do cálculo das menores distâncias o vetor de distâncias segue para a função **classifica_voto_majoritario** que define qual classe é a mais votada dentre as classes existentes no vetor. Cada classe classificada é adicionada em um vetor que é retornada pela função **classificador**
E fim o vetor de classificados é utilizado para obtermos a taxa de acerto, juntamente com a matriz de teste. A resposta do algoritmo é a taxa de acerto obtida para cada conjunto de instâncias e a matriz de confusão.

Algorithm 1: Implementação do Algoritmo K-NN

Para calcular a distância entre duas instâncias, utilizou-se a distância euclidiana, define-se por distância euclidiana d ao comprimento de um segmento de reta que une dois pontos, o que matematicamente pode ser escrito através da expressão:

$$d(x_i, x_j) = \sqrt{\sum_{i=1}^{n} (x_i - x_j)^2} \tag{1}$$

3. Resultados

Executou-se os seguintes testes:

1. Para 25% do conjunto de treinamento com k = 3
2. Para 50% do conjunto de treinamento com k = 3
3. Para 100% do conjunto de treinamento com k = 3
4. Para 25% do conjunto de treinamento com k = 5
5. Para 50% do conjunto de treinamento com k = 5
6. Para 100% do conjunto de treinamento com k = 5
7. Para 25% do conjunto de treinamento com k = 9
8. Para 50% do conjunto de treinamento com k = 9
9. Para 100% do conjunto de treinamento com k = 9

Os resultados observados são os que seguem:

1. Taxa de acerto o 1º teste: 18.916666666666668%
2. Taxa de acerto o 2º teste: 29.833333333333336%
3. Taxa de acerto o 3º teste: 51.916666666666664%
4. Taxa de acerto o 4º teste: 16.166666666666664%
5. Taxa de acerto o 5º teste: 25.083333333333336%

6. Taxa de acerto o 6º teste: 41.25%
7. Taxa de acerto o 7º teste: 13.583333333333334%
8. Taxa de acerto o 8º teste: 18.166666666666668%
9. Taxa de acerto o 9º teste: 28.666666666666668%

4. Considerações Finais

Considera-se, deste modo, que a medida que aumentamos o valor de k a taxa de acerto diminui gradativamente em todos os casos de teste, de fato podemos afirmar que estas instâncias estão a uma curta distância entre si, logo explica-se o motivo da taxa diminuir quando acrescido o valor e de k. Nos testes submetidos para avaliar o impacto de mais ou menos instâncias no conjunto de treinamento podemos afirmar que não obteve-se melhores resultados quando o conjunto de treinamento fora reduzido para 20% e 50% do seu tamanho real, como prova mostra os resultados das classificações com estes conjuntos na seção anterior.
Sobretudo, ressalta-se o melhor resultado obtido entre todos os teste realizados, com k = 3 e conjunto de treinamento 100% obteve-se a taxa próxima de 52% de acerto, está taxa mostra-se relativamente baixa por dois motivos, a escolha inadequada dos dados para treinamento ou má configuração do algoritmo de classificação, sendo assim podem ser necessários ajustes finos no algoritmo afim de obter melhores resultados.

5. Referências

GARCÍA, Salvador, LUGUENO, Julián, HERRERA, Francisco **Data Preprocessing in Data Mining.**
SINWAR, Deepak e KAUSHIK, Rahul **Study of Euclidean and Manhattan Distance Metrics using Simple K-Means Clustering.**