Algoritmo k-Nearest Neighbors: Distância Euclidiana vs Distância de Manhattan

Lucas Fontes Buzuti
Departamento de Engenharia Elétrica
Centro Universitário FEI
São Bernardo do Campo-SP, Brasil
lucas.buzuti@outlook.com

***Resumo*—Esse artigo tem uma finalidade acadêmica na compreensão e implementação do algoritmo k-Nearest Neighbors (k-NN). O algoritmos foi analisado em duas distâncias sendo elas: a Distância Euclidiana e a Distância de Manhattan. O conjunto de dados utilizado nesse trabalho foi o *Breast Cancer Wisconsin (Diagnostic) Data Set*. Na implementação foi utilizada a programação orientada a objeto na linguagem *C++* em programação paralela (a biblioteca utilizada para programação paralela foi a OpenMP), tendo em foco a otimização e a velocidade na execução de algoritmos.**

***Index Terms*—algoritmo, k-Nearest Neighbors, otimização, OpenMP, C++**

I. Introdução

Esse artigo tem em seu objetivo a compreensão e implementação do algoritmo k-Nearest Neighbors (k-NN) [3], uma vez que a compreensão é uma análise comparativa do algoritmo em duas distância: a Distância Euclidiana e a Distância de Manhattan. Na implementação visa-se como alvo a utilização da linguagem *C++* em programação paralela com a biblioteca OpenMP, visto que essa linguagem é focada na otimização e na velocidade da execução de algoritmos.

O k-NN é um algoritmo de aprendizado supervisionado, em que observa-se alguns pares de exemplos de entrada e saída, de tal forma a aprender uma função que mapeia a entrada para saída. Em outras palavras, insere ao sistema a resposta correta durante o processo de treinamento. O aprendizado supervisionado é eficiente, pois o sistema pode trabalhar diretamente com as informações corretas.

II. Teoria

O k-NN é um tipo de aprendizado baseado em instância, uma vez que é uma família de algoritmos de aprendizado que, em vez de executar generalizações explicitas, compara novas instâncias de problemas com instâncias vistas em treinamento que foram alocadas em memória, e também é um método não paramétrico usado para classificação e regressão [2].

Entre os algoritmos de classificação o k-NN pode ser considerado o mais simples, sendo que classifica os objetos com base em exemplos de treinamento que estão mais próximos do espaço de características.

Para treinar o k-NN é necessário um conjunto de treinamento, no qual são vetores em um espaço de característica multidimensional, cada um com seu rótulo de classe correspondente. Em sua fase de treinamento os vetores de características das amostras de treinamento são apenas armazenados junto com seus rótulos de classe correspondente.

Na fase de classificação, define o valor impar de K e um vetor de característica não rotulado, onde K representa o número de vizinhos mais próximos que serão considerados pelo algoritmo para classificar a nova amostra. Em outras palavras, o novo vetor de característica é classificado a partir do rótulo mais frequente entre as K amostras mais próximas a esse novo vetor. Para fazer a classificação, uma métrica de distância precisa ser definida [1]. Esse trabalho utilizou-se da Distância Euclidiana e da Distância de Manhattan, demostradas nas Equações 1 e 2, respectivamente.

$$d(p,q) = d(q,p) = \sqrt{\sum_{i=1}^{n}(q_n - p_n)^2}, \tag{1}$$

$$d(p,q) = d(q,p) = \sum_{i=1}^{n}|q_n - p_n|^2, \tag{2}$$

onde p e q são vetores de pontos.

III. Proposta e Implementação

Esse artigo propõe utilizar a linguagem *C++* em programação paralela para a implementação[1] do algoritmo k-NN. Além da implementação, uma análise comparativa entre dois tipos de métrica é proposta. Esta implementação foi feita em um computador com sistema operacional *Linux* e com o compilador *G++*, a versão da linguagem utilizada foi a *C++11*. Para desenvolver o algoritmo, foi utilizada a programação paralela junto com orientação a objeto (POO), onde visa a construção de classes e métodos.

Foi codificada uma classe denominada *kNN* do Algoritmo 1. Para avaliar a classe, utilizou-se o conjunto de dados denominado *Breast Cancer Wisconsin (Diagnostic) Data Set*[2] disponível publicamente. Esse conjunto contêm 569 instâncias e 32 atributos dividido em: textura do tumor, perímetro, areá etc, e duas classes: maligno e benigno.

[1]https://github.com/buzutilucas/scientific-programming/tree/master/Ex09
[2]https://archive.ics.uci.edu/ml/datasets/

Algorithm 1 k-Nearest Neighbor

X: training data
Y: class labels of **X**
x: unknown sample

for $i = 1$ **to** length of **X do**
    Compute distance $d(\mathbf{X}_i, x)$
end for
Compute set *I* containing indices for the *k* smallest distances $d(\mathbf{X}_i, x)$.
return majority label for $\{\mathbf{Y}_i \; where \; i \in I\}$

IV. Experimentos e Resultado

A analise entre as duas distâncias estabelecida nesse trabalho através do algoritmo k-NN, foi expressa via matriz de confusão e as métricas acurácia, recall, precisão e f-score. Para a obtenção da matriz de confusão, o algoritmo foi treinado com 529 instâncias e com valor de K sendo 11. As 40 instâncias restantes do conjunto de dados foi utilizadas para testar o algoritmo, assim determinando a matriz de confusão e abstraindo dela as métricas para o algoritmo ser avaliado. A Tabela I ilustra a matriz de confusão do k-NN com a Distância Euclidiana e a Tabela II a matriz de confusão do k-NN com a Distância de Manhattan .

Tabela I
Matriz de confusão com a Distância Euclidiana.

		Predicted	
		benign	malignant
Actual	benign	20	0
	malignant	1	19

Tabela II
Matriz de confusão com a Distância Manhattan.

		Predicted	
		benign	malignant
Actual	benign	20	0
	malignant	20	0

A partir das matrizes de confusão pode-de obter as métrica de cada modelos e comparar as suas eficiências. Na Tabela III são ilustradas as métricas correspondente a cada modelo.

Tabela III
Métrica de cada modelo k-NN.

Model k-NN	**Accuracy**	**Recall**	**Precision**	**F-score**
Euclidean Distance	0.975	0.952	1.0	0.975
Manhattan Distance	0.5	0.5	1.0	0.667

Analisando as métricas obtida de cada modelo note-se que, embora o modelo com a Distância de Manhattan apresenta uma acurácia baixa, obteve uma precisão máxima indicando que o modelo teve uma proporção de identificações positivas realmente correta. Em outras palavras, o modelo obteve eficiência em seu trabalho. Entretanto, o modelo não responde bem ao recall, uma vez que o recall identifica se os positivos foram identificado corretamente, sendo positivos entendido como a classe que se quer prever. Consegue-se visualizar o desbalanceamento desse modelo no f-score.

O modelo com a Distância Euclidiana é totalmente o inverso do modelo com a Distância de Manhattan, visto que o f-score seu é alto indicando que há um balanço entre a precisão e o recall. É claro que existe outras distâncias que pode ser superior que a Distância Euclidiana e além disso, há outros tipos de classificadores mais eficientes que o k-NN para executar uma classificação tão complexa que é identificar se um tumor é maligno ou benigno.

V. Trabalhos Correlatos

Mesmo o algoritmo k-NN sendo um dos ou o primeiro algoritmo de classificação e já tendo classificadores mais modernos e melhores, o k-NN ainda é um algoritmo de pesquisas [4] [5].

VI. Conclusão

Nesse artigo pode-se implementar e avaliar dois tipos de distâncias no algoritmo k-Nearest Neighbors (k-NN). A partir das métricas acurácia, recall, precisão e f-score obtidas via matriz de confusão de cada modelo, pode-se concluir que o modelo com a Distância Euclidiana obteve o melhor resultado entre as métricas, visto que a Distância Euclidiana traça uma reta do ponto A para o ponto B, já a Distância de Manhattan traça uma trajetória entra A e B mediante as coordenadas de um espaço de dimensão, por isso que a Distância Euclidiana se mostrou eficiente. O k-NN é um classificador bastante flexível e em alguns casos apresenta ótimos resultados, entretanto, quando o problema é complexo não apresenta resultados ótimos, assim tendo que utilizar algoritmos mais complexos e robustos.

Referências

[1] Bishop, Christopher M. Pattern recognition and machine learning. springer, p.67-124, 2006.
[2] N. S. Altman (1992) An Introduction to Kernel and Nearest-Neighbor Nonparametric Regression, The American Statistician, 46:3, 175-185, DOI: 10.1080/00031305.1992.10475879
[3] Fukunaga, K.; Narendra, P. M. A branch and bound algorithm for computing k-nearest neighbors. IEEE Transactions on Computers, v. 100, n. 7, p. 750–753, 1975.
[4] Hyvönen, Ville, Teemu Pitkänen, Sotiris K. Tasoulis, Liang Wang, Teemu Roos and Jukka Corander. “Fast k-NN search.” ArXiv abs/1509.06957 (2015): n. pag.
[5] Wang, Nannan, et al. ”Anchored neighborhood index for face sketch synthesis.”IEEE Transactions on Circuits and Systems for Video Technology 28.9 (2017): 2154-2163.