Universidad Politécnica de Yucatán

Computational Robotics

Teacher: Victor Ortiz

Student: Fernando Loera

Quarter and grade: Quarter 9°A

Date: 09/10/2023

Implementation for KNN

Introduction

KNN is a supervised machine learning algorithm which is mostly used when classifying data point but it can also perform regression as well. The main objective of the KNN is to classify or predict a new data point based on its similarity with the other data points to a particular category or average value of its K nearest neighbors in the feature space.

Development

Intuition behind KNN

KNN selects any functional form from the trainin daa to map the input data to the target data. It is a nonparametric algorithm for the fact that it does not take any specific assumption about the mapping function.

In the context of classification, KNN starts by identifying the K data points in the training dataset that have the highest similarity or closeness to the new data point in terms of feature values. This proximity is evaluated using distance metrics such as Euclidean distance or Manhattan distance. Once the K nearest neighbors have been determined, KNN counts the class labels associated with these neighbors and assigns to the new data point the class label that appears most frequently among them. The decision is guided by the notion that data points with similar characteristics tend to belong to the same class.

For regression tasks, the KNN algorithm takes a slightly different approach because it takes in consideration the class labels, it focuses on predicting a continuous target value for the new data point. The regression variant of KNN is especially useful for tasks where the outcome is a continuous variable, such as predicting housing or stock prices.

Algorithm Pseudocode

These are the steps to follow when making a KNN Algorithm Pseudocode.

1. Load the training and test data
2. Choose the value of K
3. For test data is necessary to find the Euclidian distances in a list, choose the first k points and assign a class to the test point based on the majority of clases present in the chosen points.

Loss Function (The algorithm does not have loss function because the model d not learn a parametric function)

It is a matematical function which is a crucial component used to quantify the error or discrepancy between the predicted output and the current value. The goal is to

measure the models performance and guides the optimization process and minimize this loss.

Main purpose:

- Model Evaluation: The loss function provides a numerical value that indicates how far off the model's predictions are from the true values.
- Optimization: In many machine learning algorithms, the training process involves finding the model's parameters that minimize the loss function.

Optimization function identification

It is the process of finding the best parameters for the model, in order to minimize a chosen objective function. Te goa lis to adjust the parameters so that the model can make accurate predictions or a better classification on the given data. To use it appropiately is necessary to have a good understanding of the problema that is faced, chaacteristics,and the type pf data that is used, so we can choose the best loss function that better solves these factors

NOTE: The algorithm does not have loss function and optimization function because the model does not learn parametrics functions.

Conclusion

KNN is a versatile and intuitive machine learning algorithm that plays a valuable role in various applications. Its strenghts lies in its simplicity and ease of implementation, making it a popular choice for both classification and regression tasks.

Loss functions and optimization functions differs from traditional supervised learning algorithms. KNN does not involve a formal training or optimization process like gradient descent. Instead, it relies on the identification of appropriate distance metrics and the selection of K. The key challenge lies in choosing the right distance metric that aligns with the problem's characteristics and goals.

References

[1] Joos Korstanje, “The k-Nearest Neighbors (kNN) Algorithm in Python.” [Online] Accessed: October 7th, 2023.

Available:

https://realpython.com/knn-python/

[2] Roberto Diaz, “Algoritmo KNN – Como funciona y ejemplos en Python.” [Online]

Accessed: October 8th, 2023.

Available: https://www.themachinelearners.com/algoritmo-knn/

[3] Shankar, “Understanding Loss Function in Deep Learning.” June 2023. [Online]

Accessed: October 8th, 2023.

Available:

https://www.analyticsvidhya.com/blog/2022/06/understanding-loss-function-in-deep-learning/#:~:text=a%20loss%20function%3F-,A.,well%20it%20fits%20the%20data.