Project CEED 2024-25

M.Sc. in Informatics and Computing Engineering (M.EIC), 2nd Year

João M. P. Cardoso
Dep. de Engenharia Informática, Faculdade de Engenharia (FEUP),
Universidade do Porto, Porto, Portugal
Email: jmpc@fe.up.pt

U.PORTO

FEUP FACULDADE DE ENGENHARIA
UNIVERSIDADE DO PORTO

DEI DEPARTAMENTO DE
ENGENHARIA INFORMÁTICA

Outline

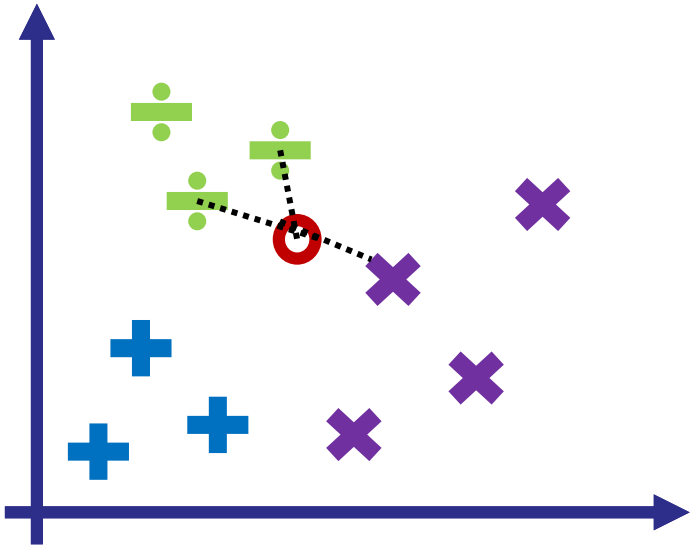
- kNN Machine Learning Algorithm
- kNN for Classification
- kNN Code Provided
- kNN Code Structure
- Project Goals
- Bibliography

kNN Machine Learning Algorithm

- Supervised machine learning algorithm
  - Used for regression and for classification
  - Simple and based on the K nearest neighborhoods
  - Scalability problems as it requires the calculation of the distances of the instance to classify to every instance in the training set (knowledge base)
- Lazy training algorithm
  - Training set is stored and no model is built from the training set
  - => Neglected overhead for online/incremental learning
- There are optimization schemes for kNN that:
  - Represent the knowledge base in data structures (e.g., KD-tree) that make the classification/regression more efficient
  - Provide implementations of approximate kNN (i.e., kNN that may not give results based on the true k NNs)
  - For both see, e.g., Cunningham and Delany, ACM CSUR, 2021)

kNN Algorithm for Classification

- Giving an instance to classify, the algorithm infers/outputs a class for that instance
- Example with vectors with two features (2D space), K=3, 10 instances in the training dataset, and 3 classes:



Instance to classify

Training instance of class A

Training instance of class B

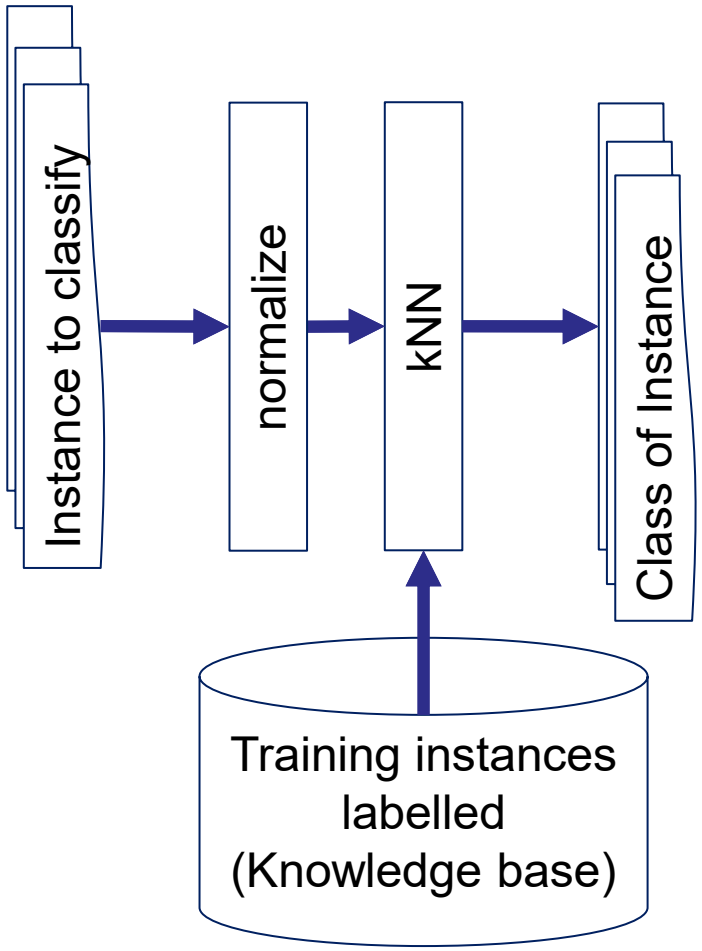
Training instance of class C

Answer: Instance is of class B

kNN Code Provided

- C code of a possible implementation of the typical kNN
- Distance calculations use the Euclidean distance
- The implementation is targeting a HAR (Human-Activity Recognition) system for embedded devices and provides data for testing
- Code includes a couple of scenarios, including the training and the testing datasets
- The implementation does not consider the noise reduction and extraction of features typically needed in this kind of HAR systems

kNN Code Structure



- Main includes the outer loop that input the instance to classify to kNN, and kNN returns the inferred class of the instance
- Each instance is represented as a *Point* struct which includes the vector of features
- Prototype of the knn function: *knn_classifyinstance(Point new_point, int k, int num_classes, Point *known_points, int num_points, int num_features);*

Project Goals

- Accelerate the kNN implementation by code optimizations/transformations and by using hardware accelerators
- In the end:
  - Report about the work done and analysis of the intermediate and final results
  - Presentation and discussion of the project

Bibliography

- Source publications:
  - Evelyn Fix and Joseph L Hodges Jr., “Discriminatory analysis-nonparametric discrimination: consistency properties,” Technical report, DTIC Document, 1951. https://apps.dtic.mil/dtic/tr/fulltext/u2/a800276.pdf
  - Thomas M Cover and Peter E Hart, “Nearest neighbor pattern classification,” in IEEE Transactions on Information Theory, 13(1):21-27, 1967. https://doi.org/10.1109/TIT.1967.1053964
- Brief summaries about kNN:
  - “k-nearest neighbors algorithm,” https://en.m.wikipedia.org/wiki/K-nearest_neighbors_algorithm
  - “What is the k-nearest neighbors algorithm?,” IBM, URL: https://www.ibm.com/topics/knn, [accessed in October 2023]
- About kNN
  - Pádraig Cunningham and Sarah Jane Delany. 2021. K-Nearest Neighbour Classifiers - A Tutorial. ACM Comput. Surv. 54, 6, Article 128 (July 2022), 25 pages. https://doi.org/10.1145/3459665
- Relevance of kNN:
  - Wu, X., Kumar, V., Ross Quinlan, J. et al., “Top 10 algorithms in data mining,” in Knowlege Information Systems, 14, 1–37 (2008). https://doi.org/10.1007/s10115-007-0114-2