

Federal University of Santa Maria Department of Applied Computing Course: Machine Learning – PPGCC

Individual Assignment on Case-Based Reasoning (CBR) in the Context of Machine Learning (Lazy Learning)

This assignment involves downloading an API for the development of CBR systems, studying the documentation of the chosen API, and building a prototype of a CBR application.

In this case, the system must include at least the steps of case retrieval – query – and solution reuse, properly implemented and tested for an application problem.

While executing each step in building the system to solve the chosen application problem, record a video explaining what you are doing/building (slides are not required).

In the video, explain which CBR features/functions are available in the API used to implement the chosen application. For example: - the format used to represent the cases, - the types and value domains of all modeled attributes, - the similarity functions created (local and global similarity functions), - the case retrieval functionality, and - the solution reuse functionality implemented.

The implemented system must also be tested using at least the leave-one-out-and-test method (or an equivalent method). The test results must be presented in the video.

To develop this assignment, use ONE of the following CBR APIs/tools:

(a) myCBR (simple to use and also educational, but not as complete as jCOLIBRI)
<http://mycbr-project.org/download.html>

(b) CBR API in Python Lenz, M., Malburg, L., Bergmann, R. (2024). CBRkit: An Intuitive Case-Based Reasoning Toolkit for Python. In: Recio-Garcia, J.A., Orozco-del-Castillo, M.G., Bridge, D. (eds) Case-Based Reasoning Research and Development. ICCBR 2024. Lecture Notes in Computer Science(), vol 14775. Springer, Cham. https://doi.org/10.1007/978-3-031-63646-2_19
GitHub: <https://github.com/wi2trier/cbrkit> Video: <https://iccbr2024.org/accepted-papers>

(c) Implementations of K-NN algorithms in Python or another language: K-NN implementations can be reused, but some CBR functionalities must be implemented from scratch (see ready-made features in the systems above). In Python or another language, command-line functions are sufficient; no graphical interfaces need to be implemented.

(d) jCOLIBRI (most complete system, but a prototype with educational purposes)
<http://gaia.fdi.ucm.es/research/colibri/jcolibri> <https://sourceforge.net/projects/jcolibri-cbr/>

Note: adjust the Windows PATH variable to run Java programs. jCOLIBRI was developed in Eclipse, but the Java code also works in other IDEs (e.g., import Eclipse code into NetBeans).

Version 2.1 of jCOLIBRI contains application examples that are easy to run. (Important: the version of jCOLIBRI with examples is NOT the latest available on the website. The examples do not work in the latest version provided there.) <https://sourceforge.net/projects/jcolibri-cbr/files/jCOLIBRI-CBR/>

Submission

Send the video (and the dataset used in the assignment) by email to the professor in .MP4 format (or upload to a cloud folder for download).

Test the video audio quality. Use screen-recording software, e.g., OBS (www.techtudo.com.br/dicas-e-tutoriais/2020/03/obs-studio-tutorial-de-como-fazer-download-configurar-e-usar-o-software-esports.ghml).

File naming format: videoAplicaçãoCBR.mp4

Note: In the course's shared cloud folders, several videos from past students are available. Along with the tool documentation you must study, and tutorials with step-by-step examples of building CBR applications in these tools, use the videos and materials provided to better develop your work.

Professor: Luís Alvaro Email: luisalvaro@inf.ufsm.br