

AMD (MLDM – Machine Learning and Data Mining) – Module of Practice

Paulo Trigo Silva

1. [before-proceeding] – a spreadsheet (Excel) “crash-course”

Consider the file: “a00_excel_crash_course.xlsx”.

- Open the “a00_excel_crash_course.xlsx” and explore each of the sections (“Arithmetic”, “Statistical Summaries”, “Random Numbers” and “Flow Control”).
- Move to the “Exercises” part of the sheet and fill the “gray-cells” in order to implement the formulation that is described in each column.

2. How to “learn a codebook”, for 1Epoch, using the LVQ method?

Consider the file: “LVQ_algorithm_exercise.xlsx”. *Note:* the data is the same as in slides.

- Open the “LVQ_algorithm_exercise.xlsx” and explore the “LVQ_1Epoch.xlsx” sheet. Analyze the “Dataset” and the “Codebook Vectors”; identify each point plotted in the graphs.
- Move to the “Training” part of the sheet and analyze the data that is already filled.
- Fill the first row (of the “Training” part) with the formulations indicated in each column. Whenever the first row is correctly implemented the “CHECK (oneEpoch)” value of “notOk” will turn into “ok” (for that first row).
- Follow the previous reasoning and fill the cells of the first 4 rows. Whenever the corresponding indication turns into “ok” fill the next 4 rows (so, by now all first 8 rows are correctly implemented!).
- Extend the previous reasoning to implement the training for an whole epoch (the first epoch). Make sure that all rows get the “ok” mark!
- Make sure that the “CHECK (codebook-afterOneEpoch)” is all marked with the “ok” tag. Now you have a solid understanding of the “mechanics” behind the implementation of the LVQ method.

3. How to use the “learned codebook” to make “predictions”?

Consider the file: “LVQ_algorithm_exercise.xlsx”.

- Open the “LVQ_algorithm_exercise.xlsx” and explore the “LVQ_1Epoch.xlsx” sheet. Analyze the “Dataset” and the “Codebook Vectors”; identify each point plotted in the graphs.
- Move to the “Prediction” part of the sheet and analyze the data that is already filled.
- Fill the first 4 rows (of the “Prediction” part) with the formulations indicated in each column. The “CHECK (Prediction 1Epoch)” is already “ok” (for those 4 rows) but the BMU is not correct; there can be only one BMU for those 4 rows! Make sure that a single BMU appears for those 4 rows.
- Follow the previous reasoning and fill the cells of the next 4 rows. Also make sure that those 4 rows only have a single BMU.
- Extend the previous reasoning to implement the prediction for each dataset instance. Make sure that all tags become “ok” and that each dataset instance is only associated with a single BMU.

AMD (MLDM – Machine Learning and Data Mining) – Module of Practice

Paulo Trigo Silva

- f) Make sure that the “CHECK (Accuracy Prediction 1Epoch)” is all marked with the “ok” tag. *Note:* recall that the predictions used the training dataset instances (i.e., the instances that were taken to train the codebook); therefore, these predictions cannot represent a model validation. We are using them in order to better understand the prediction process (not as a validation process).

4. How to “learn a codebook” using the “full-power” of LVQ method?

Consider the file: “LVQ_algorithm.py”.

- a) Open the “LVQ_algorithm.py” and analyze the “train_LVQ” function; this is the top-level for the learning method (the one that invokes the lower-level functions).
- b) Take a look at “The Learning Parameters” section, notice that “_maxEpoch=1” and execute the algorithm (e.g., in the command-line write: `python LVQ_algorithm.py`). Compare the result with the one from the Spreadsheet. The results are different! Lets fix them in the next items!

Note: The file “_info_1MaxEpoch&100MaxEpoch.txt” holds the expected results for 1Epoch (same as in the Spreadsheet) and 100Epoch learning.

- c) Move to the “get_euclidean_distance” function and fix-it focusing into the <HERE> region!
Hint: you may use the Spreadsheet to test your implementation.
- d) Move to the “apply_repulsion_rule” function and fix-it focusing into the <HERE> region!
- e) Move to the “get_learning_rate” function and fix-it focusing into the <HERE> region!
- f) Now you should have your 1Epoch learning correctly implemented.
- g) Set “_maxEpoch=100”, execute and validate using “_info_1MaxEpoch&100MaxEpoch.txt”.

5. ... explore the “full-power” of the LVQ method

- a) Move to “The Learning Parameters” section of the code and experiment with different setting. Compare results considering the “totalError” metric computed by the algorithm.
- b) Set a fixed value for the “_alpha” (e.g. 0.7) and change increment the “_maxEpoch” from 1 to 100; change the “main” function in order to implement such a loop and store the gathered values of the “totalError” metric. Open a Spreadsheet and plot a graph that relates the number of learning epochs with the total error; draw some conclusions!
- c) Use a randomly generated codebook, i.e., set “_codebookRandom=True” and explore different “_dimCodebook” values. Choose a method to graphically exhibit the impact of augmenting the “_dimCodebook” when using a randomly generated codebook; draw some conclusions!
- d) ... explore the LVQ freely, e.g., using different datasets, or by varying the learning parameters and share your thoughts!