LINFO2275 – Data mining and decision making

Gestures Recognition

Project 2 guidelines

Professor: Marco Saerens

Teaching Assistants: Sylvain Courtain and Pierre Leleux

1 Objective

The objective of this project is to put into practice some of the techniques introduced in the data mining and decision making lectures. This is done through the study of a practical case which requires the use of a scientific programming language, or libraries, related to the statistical processing of data. More precisely, this work aims at applying data mining/statistical learning techniques for supervised classification of three-dimensional recorded gestures.

The realization of the project will be carried out in groups of maximum 3 students – the same as the previous project.

2 Problem statement

You are a young data scientist hired by an IT company. You first work is to implement a hand gesture recognition system for a smart user interface. A three-dimensional tracking of the position vector $\mathbf{r}(t) = [x(t), y(t), z(t)]^{\mathrm{T}}$ of the hand is recorded as a sequence of the three coordinates in function of the time t. Thus, the user is executing a sketch or a symbol with his hand, which is recorded and recognised by the system. Sketches and symbols are rapidly executed freehand drawings (Huang et al., 2019). Here, the sketches are drawings of the numbers from 0 to 9 (10 classes or categories).

You are asked to develop a sketch recognition system identifying the category of the sequence, and implementing/testing it with Python.

3 Data sets

In order to train the recognition system, a publicly available gestures dataset, gathered by and described in (Huang et al., 2019), is used. Ten users participated and were each asked to sketch ten times the numbers $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$ (called "domain 1" data in Huang et al., 2019). We therefore have ten repetitions of these ten numbers by ten users, that is, 1000 sequences in total. The same data collection contains other drawing types; if you have the time, feel free to investigate these other data sets (as a bonus). The data are available on Moodle.

4 Implementation

The project is very open. You should first implement a baseline technique like the dynamic time warping (Rabiner et al., 1993) or the edit-distance (or the longest common subsequence) after having performed a vector quantization (clustering) of the signals. Then, you simply rely on a nearest-neighbor technique. You must implement yourself this baseline in Python 3 and obtain baseline results on the dataset.

Then, you are free to use and test at least one other method found in the literature, like the $1 recognizer (see the references below), or any other technique you are interested in. For this second part, you may use existing implementations in Python and, of course, refer to them in the report.

One possibility would be to compute a kernel matrix thanks to the pairwise distances (or similarities) between samples of the training set and then train a logistic regression or a SVM on this kernel matrix. Then, apply this model on the test set for out-of-training-sample validation.

5 Validation tests

You will evaluate the classification accuracy of your model by cross-validation. For the cross-validation, you have to use a leave-one-user-out procedure by using 90% of the data from 9 users for fitting the model, and then testing the model on the remaining 10% data provided by the remaining user who was removed from the training set (there are ten users in total). The same procedure is repeated 10 times by holding all the data of each user in turn for testing (first remove user 1 from the training set and use this user 1 for testing, then remove user 2, etc). Therefore, the evaluation is user-independent because it quantifies the ability of identifying the gestures of a new user who does not appear in the training set. For each compared method, you should record the different accuracies for the different users and then compute the average accuracy and its standard deviation.

Overall, you should compare the different investigated methods according to their average accuracy and standard deviation. It is also interesting to provide a confusion matrix showing the sketches (classes) that are easily confused by the best model.

6 Report

Please do not forget to mention your affiliation on the cover page, together with your name (SINF, INFO, MAP, STAT, BIR, DATS, etc) and group number. You are asked to write a report (PDF) in English of maximum 7 pages (everything included). This report must have a professional look & feel, like a scientific or a technical report. Therefore, your report can integrate plots, but no screenshot (image capture) of equations and/or screenshot of code outputs. Do not forget to provide references for your sources of information[1] and be consistent in your notations.

Your report must at least contain:

- a short introduction;
- a brief theoretical explanation (with main equations in Latex) of the investigated methods;
- a short description of your implementation[2];

[1] Some papers are available on Moodle.

[2] Do not go into details about your code itself. If you still wish to provide details about the code and the functions of your implementation, do not hesitate to integrate a ReadMe file in your zip, but your report should not focus on the code.

- a description and a discussion of the comparison results.
- a conclusion.

Your project will be evaluated on the following aspects:

- The quality of your report and code;
- The amount of experimental work (including potential additional experiments);
- The relevance and description of the investigated methods.

This report must be uploaded before May 16, 2022, 23:55, together with the code (all files zipped together) on Moodle in the section "Assignments". Do not forget to comment your code. This second project accounts for 5 points in the final grade of the course. Your project can be handed behind schedule, but your group will get a penalty of -0.5 for each day late, as for the first project.

Some references

- Garcia-Diez, Fouss, Shimbo & Saerens, M. (2011). "A sum-over-paths extension of edit distances accounting for all sequence alignments". Pattern Recognition, 44 (6), 1172-1182.
- Huang, Jaiswal & Rai (2019). "Gesture-based system for next generation natural and intuitive interfaces". Artificial Intelligence for Engineering Design, Analysis and Manufacturing, 33 (1), pp. 54-68.
- Rabiner & Juang (1993). "Fundamentals of speech recognition". Prentice-Hall.
- Wobbrock, Wilson & Li (2007). "Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes". In Proceedings of the 20th annual ACM symposium on User Interface Software and Technology, pp. 159-168.

Good work !