MLPR – 2023 – Biometric identity verification

The goal of the project is the development of a classifier to perform biometric verification, i.e. to verify whether an identity claim is correct based on biometric characteristics of a subject. This project considers a speaker verification task, but similar considerations would hold for a face or fingerprint verification task.

A speaker verification system should verify whether a spoken utterance of an unknown person corresponds to a claimed identity. This involves comparing the unknown person utterance with those belonging to the claimed identity, to assess whether the unknown utterance is similar enough to the latters. For simplicity, in this project it is assumed that the claimed identity utterances are just one. The problem can then be cast as a binary problem over pairs of utterances, i.e., given the unknown utterance and the claimed identity utterance, the classifier should decide whether the pair belongs to the *same speaker* (label 1) or *different speaker* (label 0) class.

In this project, the utterances are represented in terms of speaker embeddings, i.e. low-dimensional representations of speech obtained by mapping images to a duration-independent, low-dimensional manifold (typically few hundred dimensions). To keep the model tractable, the dataset consists of synthetic data, and embeddings have significantly lower dimension than in real use-cases. The embeddings are 5-dimensional, continuous-valued vectors, and features have no specific physical interpretation.

The datasets consist of embedding *pairs*, i.e., each sample consists of a pair of embeddings, for a total dimension of 10. Pairs have been already created. File `Train.txt` contains the embedding pairs that can be employed to build the classification model (training set). The evaluation embedding pairs (evaluation set) are provided in file `Test.txt`. Each row of each file correspond to a sample (pair of embeddings). The features and corresponding label of each sample are separated by commas, with the first 10 columns of each file corresponding to the sample components, whereas the last column contains the corresponding label.

The datasets are slightly imbalanced, with the different speaker class having slightly more samples. Pairs belong to either the male or female gender (gender label is not provided, and cross-gender pairs are not present in the dataset). The target application working point is defined by the triple $(\pi_T = 0.1, C_{fn} = 1, C_{fp} = 1)$.

NOTE: In real speaker verification models the pairs are usually obtained by creating combinations out of a fixed set of utterances. This implies that different pairs may share one of the embedding sides, or may refer to the same speaker, i.e. embeddings are not independent. This would significantly complicate the generation of suitable validation sets, since the validation set speakers should not overlap with the training set speakers. To avoid this complication, the dataset has already been generated in such a way that all pairs can be considered independent, i.e. they can be considered as belonging to different speakers. In practice, this implies that the standard methods to create validation sets can be safely employed.

NOTE: Typically, when working with embedding pairs, pre-processing (dimensionality reduction, whitening, ...) is done at the embedding level, i.e. before forming the pairs. Again, to simplify the task you can analyze the standard pre-processing ignoring that fact that samples represent pairs.

Model training and model selection

The report should provide an analysis of the dataset and of suitable models for the task, and the methodology employed to select a candidate solution among different possible alternatives (e.g. different classification models, different values of the hyperparameters, different pre-processing strategies).
The models must be trained over the training set only. When needed, validation data can be extracted from the training set (for example, to compare competing models, to select suitable values for the hyperparameters of each model, or to train score calibration models). Models should be trained to optimize the target application, but performance of the models for alternative applications should also be analyzed and discussed. At the end of this stage, a candidate solution should be provided for the classification task.

Evaluation of the candidate model

The proposed solution must be evaluated on the evaluation set. The evaluation samples should be treated as independent, i.e. the value or the score of an evaluation sample should have no influence on the score of a different sample. The evaluation should report the performance in terms of the primary metric, but also analyze how the model would perform for alternative applications (i.e. alternative working points).

Post-evaluation analysis

The choices made during model training should be analyzed to verify that the selected model is indeed competitive for the task. This requires comparing on the evaluation set alternative models that were analyzed and trained, but discarded, in the first phase, to verify whether the proposed solution is indeed optimal or close to optimal with respect to possible alternatives (e.g., the chosen models is effective, the chosen hyperparameters are optimal, etc.).