# INFOB3ML Programming Assignment 3: Speech Emotion Recognition with Bag-of-Audio-Words Representation

Heysem Kaya <h.kaya@uu.nl>

September 12, 2023

---

## 1 Introduction

In this programming assignment, you will be implementing Bag-of-Audio-Words (BoAW) method for acoustic low-level descriptor (LLD) representation over utterances. As the name implies, an LLD is a low-level feature extracted from the signal to quantify the local characteristics. In our case, the acoustic LLDs are extracted from 25ms portions of the speech signal, with shifts of 10ms. You do not need to and will not be involved with acoustic feature extraction from the speech signal - that direction is beyond the scope of this project. Acoustic features are already extracted from the speech recordings and are provided to you in separate .csv files per actor.

Your task is to implement the BoAW approach and the component methods, namely Principal Component Analysis, K-Means algorithm, followed by classification using existing python libraries.

In a nutshell, you will be implementing the below functions (use of an existing library/package is not allowed):

1. Principal Component Analysis (PCA)
2. K-Means algorithm
3. BoAW representation method

## 2 The Bag-of-Audio-Words Approach

In the BoAW approach, we first train a K-Means model using the combined LLDs from all training set utterances. Since K-Means assumes no correlations in the data, and partly for reducing dimensionality, we first apply PCA to LLDs, which is also learned from the combined LLDs from all training set utterances. The BoAW representation amounts to **assigning each LLD feature vector to the nearest K-Means cluster and computing a K-dimensional histogram of the accumulated instances around each mean**. Further normalization (e.g. normalizing each histogram to unity) is possible and found effective.

Optional: While you may use the BoAW representation directly, you may also reduce the dimensionality via PCA before training a classifier.

The processing pipeline BoAW representation is illustrated in Figure 1. First, we collect LLDs from all training set files. This is followed by learning a PCA model. The PCA model parameters (including parameters used for e.g. preprocessing) are stored. The PCA reduced dataset is then fed to K-Means to learn K-Means model, namely a set of $K$ means. Then LLDs from each utterance (including those from the validation and test sets) are separately PCA projected and assigned to the nearest cluster among the $K$ means. The $K$-dimensional histogram obtained this way is a fixed-length, supra-segmental representation of the corresponding utterance. These representations can then be used to learn utterance-level labels, in our case emotions.

## 3 Emotion Classification

In the dataset, you will find a metadata, which includes the emotion name/ID, speaker id and the corresponding fold name/ID. You may prefer to use these or go along with the metadata extracted from the filenames in the provided scripts. In the ipython notebook, you will be given a template to guide your implementation. You are strongly advised to keep the function signatures the same. You are also given the necessary scripts for data ingestion (i.e., loading the files into an appropriate data structure). At each step, there will be commented instructions to ease your solution.

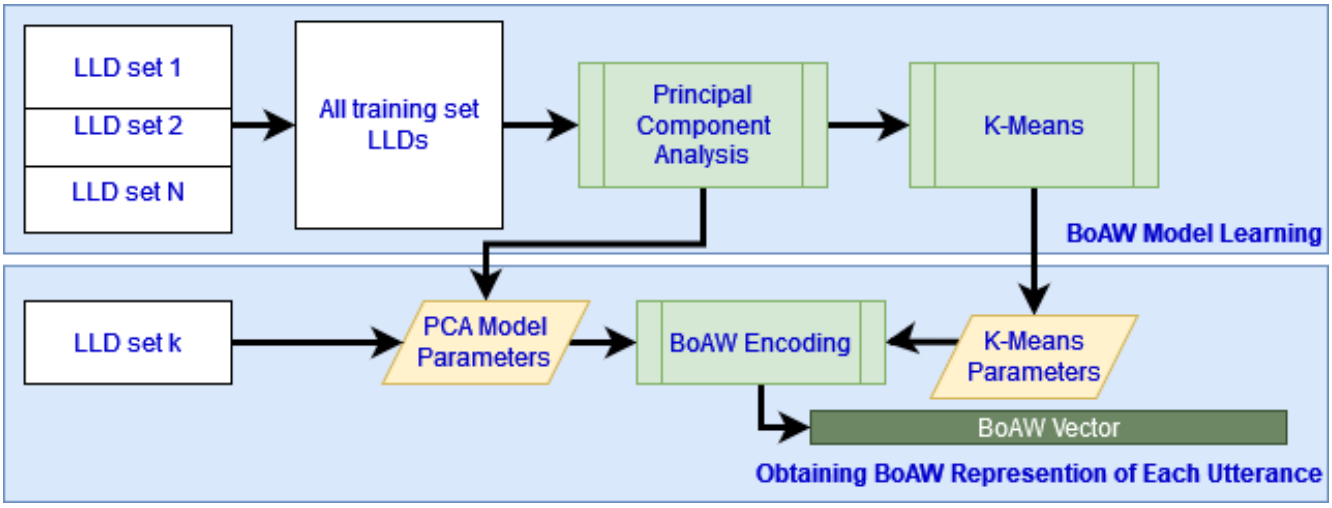


**Figure 1: Bag-of-Audio-Words representation generation pipeline.**

You will experiment with different combinations of PCA dimensions (p_pca) and K-Means components ($K$) optimizing the combinations using a built-in Support Vector Machine (SVM) classifier. You are given data of 16 speakers, where for the classification stage, the data of

- the first 8 speakers will be used as training set
- the speakers 9-12 will be used as validation set and
- the speakers 13-16 will be used as the test set.

Hence, we will ensure a speaker-disjoint training, validation and testing protocol.

The *training set* will be used to learn PCA and K-Means model, as well as to train the classifiers following the BoAW representation. Hyper-parameters of the classifiers as well as the number of principal components (p_pca) and cluster components ($K$) will be optimized on the validation set. The top model(s) (resulting from different combinations of p_pca, $K$ and classifier hyper-parameters) will be applied to the test set.

You are expected to try at least two combinations of PCA dimensions (p_pca) and $K$ to obtain different BoAW representations, and then optimize an SVM model (e.g. its complexity hyper-parameter and kernel type) on the validation set. The best setting on the validation set will then be applied to the test set. This can be done by simply applying the best performing model to the test set. An alternative is retraining the combination of the training and the validation set using the optimized setting and then applying the resulting model to the test set.