Number of Speakers Estimation using Spectrogram Factorization

Posholi S. Nyamane*
Electrical and Computer Engineering
Carnegie Mellon University
Kigali, Rwanda
pnyamane@andrew.cmu.edu

Dereje Shenkut*
Electrical and Computer Engineering
Carnegie Mellon University
Kigali, Rwanda
dshenkut@andrew.cmu.edu

Daniel Wesego*
Electrical and Computer Engineering
Carnegie Mellon University
Kigali, Rwanda
dwesego@andrew.cmu.edu

Abstract

Number of concurrent speakers estimation in single channel mixed speech arises in a number of audio processing applications or tasks such as auditory scene classification, speech recognition, audio enhancement, human-computer interaction applications, and others. In this study we solve the problem of estimating the number of speakers in a single channel mixed speech through mixed speech spectrogram factorization. We specifically implemented a non-negative matrix factorization feature extraction followed by SVM multi-label classifier to solve this problem on LibriCount dataset.

1 Literature Research

The problem of estimating the number of speakers in single channel mixtures has been solved using various approaches. For example, in [3] a density-based clustering method is proposed. [1] formulated an algorithm that is based on the modulation characteristics of speech in a way that a single speech utterance typically has a distinct modulation pattern with a peak around 4-5 Hz. Another study, [5] , solves this problem using mixed speech with up to 7 speakers through NMF (non-negative matrix factorization) algorithm. NMF calculates a decomposition of the spectrogram into nonnegative factors and has been successfully applied to audio source separation and thus it has the potential to be robust to noise disturbances when used for feature calculation [5]. We replicate the study by [5] to perform number of speakers estimation using a larger dataset than was used in [5], with up to 10 concurrent speakers.

2 Method

2.1 Spectrogram

Since we are dealing with audio datasets, it is best to represent the audio signal in terms of its spectrogram using short-time Fourier transform. Each mixture of audio will $m(t)$ be converted to its spectrogram $\mathbf{S}$ with m rows and n columns:

$$\mathbf{S}_{mxn} = |\text{STFT(m(t)}|^2 \tag{1}$$

2.2 Non-negative Matrix Factorization

Non-negative Matrix Factorization (NMF) is a rank reduction technique which decomposes a matrix of non-negative data $\mathbf{S} \in \mathbb{R}_{+}^{m \times r}$ into a product of two non-negative matrices, $\mathbf{W} \in \mathbb{R}_{+}^{m \times r}$ and $\mathbf{H} \in \mathbb{R}_{+}^{r \times n}$ where r < min(m,n) [4]. The desired decomposition minimizes the reconstruction error **D(S, WH)**, where **D** is some distance or divergence measure[4].
The factorization task is to find non-negative matrices $\mathbf{W} \in \mathbb{R}^{+}$ and $\mathbf{H} \in \mathbb{R}^{+}$such that their product is close to matrix **S**:

$$\mathbf{S} \approx \mathbf{W}^{m \times r} \cdot \mathbf{H}^{r \times n} \tag{2}$$

where r denotes the rank of the approximation. Non-negative matrix factorization is an optimization problem defined as follows:

$$\underset{\mathbf{W},\,\mathbf{H}}{\text{minimize}}\; D(\mathbf{S}|\mathbf{WH}) = \sum_{i,j} \mathbf{S}_{ij} \cdot log \frac{\mathbf{S}_{ij}}{(\mathbf{WH})_{ij}} - \mathbf{S}_{ij} + \mathbf{WH}_{ij} \tag{3}$$

To optimize the divergence function D, a multiplicative update rules are applied until convergence[5]. Estimation algorithm of the matrices **W** and **H** can be summarized in the following form:

$$\mathbf{W} = \mathbf{W} \circ \frac{\frac{\mathbf{S}}{\mathbf{WH}} \cdot \mathbf{H}^T}{\mathbf{1} \cdot \mathbf{H}^T} \tag{4}$$

$$\mathbf{H} = \mathbf{H} \circ \frac{\mathbf{W}^T \cdot \frac{\mathbf{S}}{\mathbf{WH}}}{\mathbf{W}^T \cdot \mathbf{1}} \tag{5}$$

where **1** is matrix of ones and ∘ is the symbol of element-wise product. We use the **H** matrix as a feature vector by unfolding it.

This feature vector is then used to train different machine learning algorithms like SVM to identify the number of speakers. Figure 1 below shows the complete pipeline of our system.

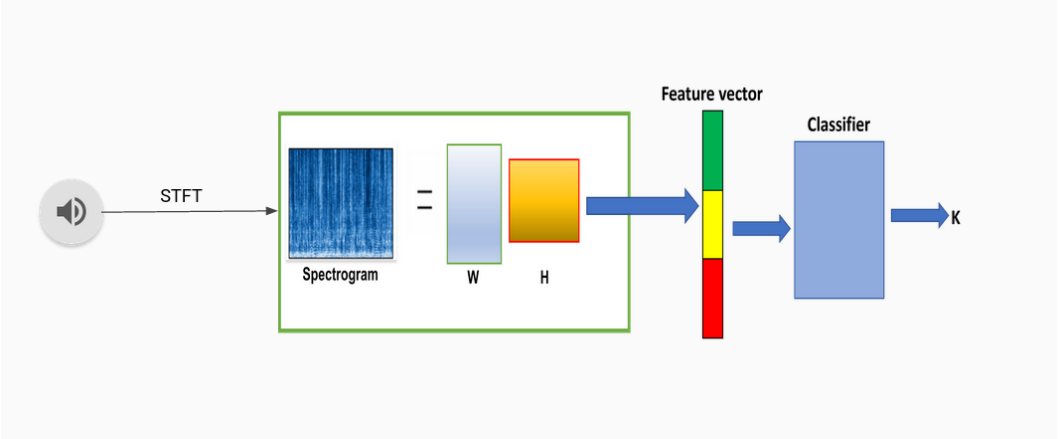


Figure 1: Overall Pipeline – the unrolled features from H matrix of NMF are fed to SVM classifier

3 Dataset

LibriCount dataset is used for both training and validation of the above algorithm. LibriCount is a synthetic dataset for speaker count estimation. The dataset contains a simulated cocktail party environment of 0 to 10 speakers [2], mixed with 0 dB SNR from random utterances of different number of speakers (NoS) from the LibriSpeech CleanTest dataset. All recordings are of 5s durations, and all speakers are active for the most part of the recording. The inputs/features are the audio wave files(16bits, 16kHz, mono) which are mixed speech time domain signals and the json format annotation files that contain the labels (number of speakers in the corresponding wave file) and other details such as the genders of the speakers and the time ranges during which each speaker is speaking. For our use case, the dataset is complete; requiring minimum pre-processing to convert the wave

files to magnitude spectrograms and to extract the labels from the annotation files. Currently, there is no benchmark on this dataset. Table 1 shows the size of the training and the validation set. We removed the data with 0 number of speakers since we it doesn't make sense to classify an empty audio.

Table 1: Dataset

Type	Size
Training	(4160, dimension of H)
Validation	(1040, dimension of H)

4 Evaluation Metrics

The performance of the NMF algorithm in [5] followed by SVM is compared with the actual number of speakers. The onfusion matrix for estimating our model's performance.

Confusion matrix also known as an error matrix,is a specific table layout that allows visualization of the performance of an algorithm, typically a supervised learning. Each row of the matrix represents the instances in an actual class while each column represents the instances in a predicted class. The method with confusion matrix values sorted on the diagonal of the indicates relatively more accurate classification.

We have also calculated the accuracy with a tolerance of ± 1 NoS. This percentage should tell us how close the predicted class is to the true value.

5 Experiments

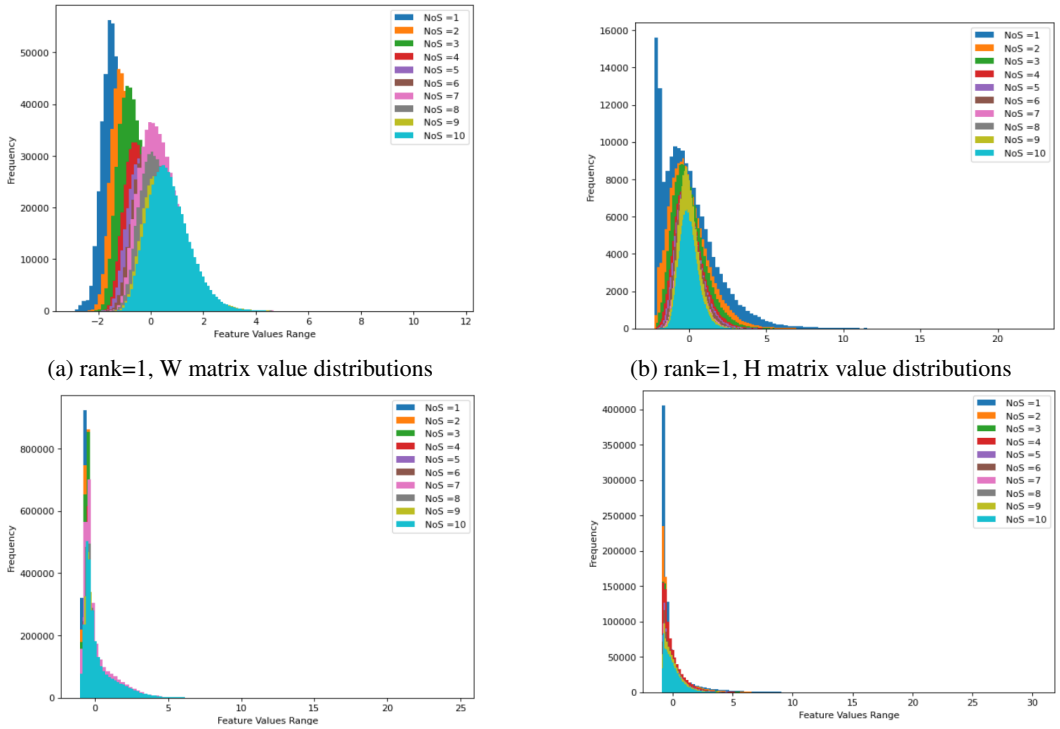
We extracted the W (basis) and H matrix using Non-Negative Matrix Factorization method. The NMF technique was performed on each sample of the dataset to extract the H matrix. We performed Equation 4 on each sample of the dataset to get the H matrix which will be used to train our model. A model will be trained on the training dataset and evaluated on the validation dataset. Assuming the basis would be similar for all speech combinations, we used the unrolled version of the H matrix as a feature for our model. The model was trained on the extracted feature with the labels from the dataset to predict the number of speakers.

6 Results and Discussion

We tried different models including logistic regression, adaboost, and SVM. The SVM model was the one that gave us the best result. We experimented with different SVM kernels including polynomial, sigmoid, linear, and the rbf kernel. The rbf kernel gave us the best result.

We also tried different ranks to decompose the signal using NMF. The features with fewer number of ranks seemed to perform better than the features with high number of ranks. In fact, when we increased the number of rows in the H matrix, we got a really poor performance. We observed that increasing the rank resulted in different classes having similar feature values as shown in the feature values distributions in Figure 2 below, thus making hard for the classifier to separate them. As seen in the Figure, the classes overlap less when rank =1 than when rank =5. The confusion matrix are shown in 4. The first two confusion matrices show that the diagonals are brighter showing a good performance but the last confusion matrix shows a bad performance. Table 2 shows the result of the 4 best selected ranks.

The result for different classifiers is shown in figure 3. SVM outperforms Adaboost and Logistic regressions.



(a) rank=1, W matrix value distributions

(b) rank=1, H matrix value distributions

(c) rank=5, W matrix value distributions

(d) rank=5, H matrix value distributions

Figure 2: Feature Values Distributions – the distinguishing

Table 2: Result

Rank	Accuracy with ± 1 tolerance
rank=1	73.4%
rank=2	72.9%
rank=5	72.1%
rank=6	69.5%

7 Conclusion

In our proposed method, we improve the baseline paper by training on a larger dataset with higher number of speakers. The method showed a competitive accuracy for estimating the number of speakers with ± 1 tolerance. Future scope of our work includes working on better methods of feature extraction that better distinguishes between each classes to solve the problem of degrading accuracy as the number of speakers become higher.

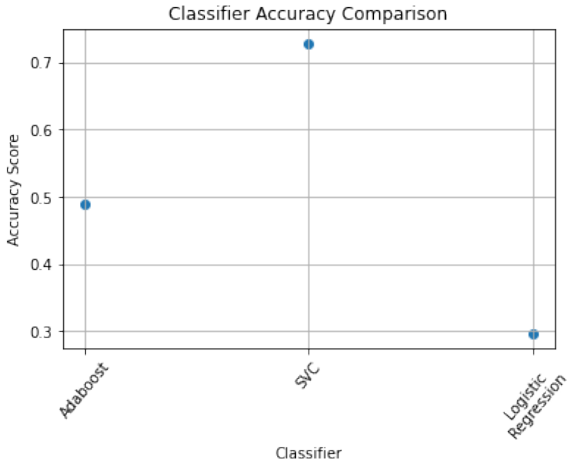
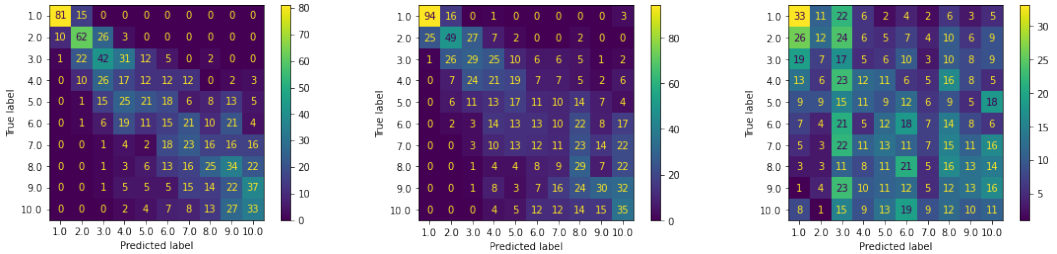
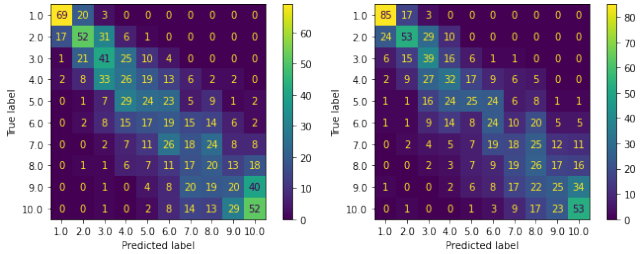


Figure 3: Comparison of different classifiers



(a) rank=6 (b) rank=15 (c) rank=200



(d) rank=2 (e) rank=5

Figure 4: Confusion Matrix for the 3 models

References

[1] T. Arai. Estimating number of speakers by the modulation characteristics of speech. *Dept. of Electrical and Electronics Eng., Sophia University, Tokyo, JAPAN.*

[2] Stöter Fabian-Robert, Chakrabarty Soumitro, Habets Emanuël, and Edler Bernd. Libricount, a dataset for speaker count estimation, Apr. 2018.

[3] Z. Yang L. Yang J. Yang, Y. Guo and S. Xie. Estimating number of speakers via density-based clustering and classification decision. 7:176541–176551, 2019.

[4] Cyril Joder and Bjoern Schuller. Exploring nonnegative matrix factorization for audio classification: Application to speaker recognition. In *Speech Communication; 10. ITG Symposium*, pages 1–4, 2012.

[5] T. Maka and M. Lazoryszczak. Detecting the number of speakers in speech mixtures by human and machine. *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA)*, page 239–244, 2018.