Estimation of Direction of Arrival (DOA) for First Order Ambisonic Audio Files using Artificial Neural Networks

Technical Report

Pedro Pablo Lucas Bravo
pedropl@uio.no

1 The Application

The aim of this project is to develop a solution to estimate the *Direction of Arrival (DOA)* from a sound event encoded as a *First Order Ambisonic (FOA)* audio file (4-channel). DOA is usually presented as two angles: *elevation* ϕ, and *azimuth* θ as shown in Fig. 1. The sound event is the blue sound source as depicted in the illustration, and the estimated direction is represented as the red source. The goal is to reduce the error between both sources regarding ϕ and θ to get the best possible estimation.

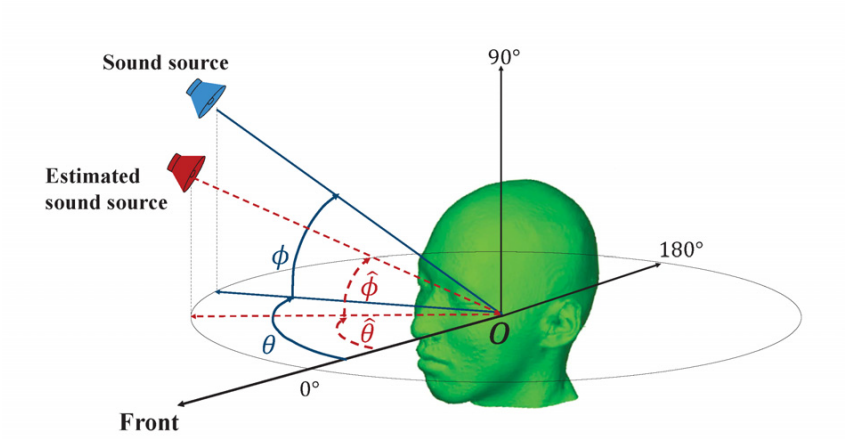

Figure 1: DOA estimation illustration. (Bui, Morikawa, and Unoki 2018)

There are deterministic methods to estimate DOA such as *time difference* and *level difference* as mentioned by Wierzbicki, Małecki, and Wiciak (2013), who also used an energy-based method in a controlled environment. There are other strategies based on complex signal processing operations such as *Multiple Signal Classification (MUSIC)* proposed by Schmidt (1986), who also compared it with *beamforming (BF)*, *maximum likelihood (ML)*, and *maximum entropy (ME)*. Nevertheless, their accuracy is degraded when the complexity of the signal increases in terms of noise, reverberation, interference, polyphony, and non-stationarity of sound sources (Nguyen et al. 2021). In this case, the use of Machine Learning (ML) techniques can increase the prediction of DOA (Hirvonen 2015).

The characteristic of this problem demands the use of *ML* strategies capable of predicting multiple outputs. Particularly, it needs to be solved with a *regression* technique since the values to estimate (ϕ and θ) are continuous numbers in the real domain. Moreover, the scope of this

project is limited to a set of techniques from which the best suitable is an *Artificial Neural Network (ANN)*, since it deals with regression problems with more than one output.

This task requires to choose the right features and architecture for the ANN. Previous works, that introduce more level of complexity by converting this problem in a *Sound Event Localization and Detection (SELD)* for FOA files, use mel spectrograms and interchannel phase differences, as well as a detailed emphasis on the architecture of *Convolutional Neural Networks (CNN)*, which is part of a *Deep Learning* approach (Nguyen et al. 2021).

A solution to this problem is relevant for several applications like in robotics for drone exploration (Choi and Chang 2020), or speech and music for feedback systems that support audio material (e.g. visuals for a spatialized music performance).

2 The Data Set

The data set was taken from the *Sound Event Localization and Detection Challenge (2019)* (DCASE 2019). It consists of two parts: A development set of 400 First Order Ambisonic (FOA) files, one-minute long sampled at 48 kHz; and an evaluation set of 100 files with the same characteristics. Each file has several sound events (11 classes across the data set) in that one-minute time frame. For achieving spatialization, the sound events were convolved with spatial room impulse responses (IRs) from five different locations, ambience noise from those locations was added to have an average SNR of 30dB. Moreover, half of the files in each set are overlapped with one more sound event. The total of sound events in the development set is 15798 and the evaluation set has 3974.

Each recording is associated with a CSV metadata file that contains a description about all the events on that minute. A description per event is composed of: the class of the sound, start and end time in the file, elevation, azimuth, and distance. The spherical coordinate system (elevation ϕ, and azimuth θ) is right-handed with the front at (0°, 0°), left at (0°, 90°) and top at (90°, 0°).

3 The Machine Learning Technique

As previously mentioned, this task is a regression problem for multiple continuous outputs which can be solved with an *Artificial Neural Network (ANN)*. Although there are other regression techniques, they provide a solution for just one output and therefore they cannot be used in this context as independent solutions since the targets (elevation ϕ, and azimuth θ) are closely related and need to be treated together (Nguyen et al. 2021).

The solution is limited to use of one type of ANN known as *Full-connected Feed-forward Multi-layer Perceptron regressor*. A detailed description is given in the following subsections.

3.1 The Feature Extraction Rationale

To achieve the best possible solution given the limitations explained above, the feature extraction technique considered the *Time Difference of Arrival (TDoA)* in psychoacustics as a departing point. Based on this, the *lag* was calculated between every pair of channel in a FOA file as in (MathWorks 2021), as well as the difference in RMS from those pairs. The average was considered in order to have a feature vector of 12 elements (6 for lags and 6 for RMS diferrences). However, the algorithm did not succeed in an acceptable way given values of *Coefficient of Determination (R^2)* below zero.

As mentioned in section 1, previous works use the information from spectrogram, thus it was considered as a second alternative by taking several approaches (interpolation, normalization with the maximum, normalization for norm as one, whitening per frequency bin) to arrange the feature row to feed the ANN. In this case, R^2 continued being below zero and the network presented overfitting for some experiments.

The FOA audio file was normalized and the output was converted from spherical coordinates (ϕ, θ) to cartesian coordinates (x,y,z) since they has the advantage of continuity for improving the performance (Kapka and Lewandowski 2019), and the loss function for the ANN is based on

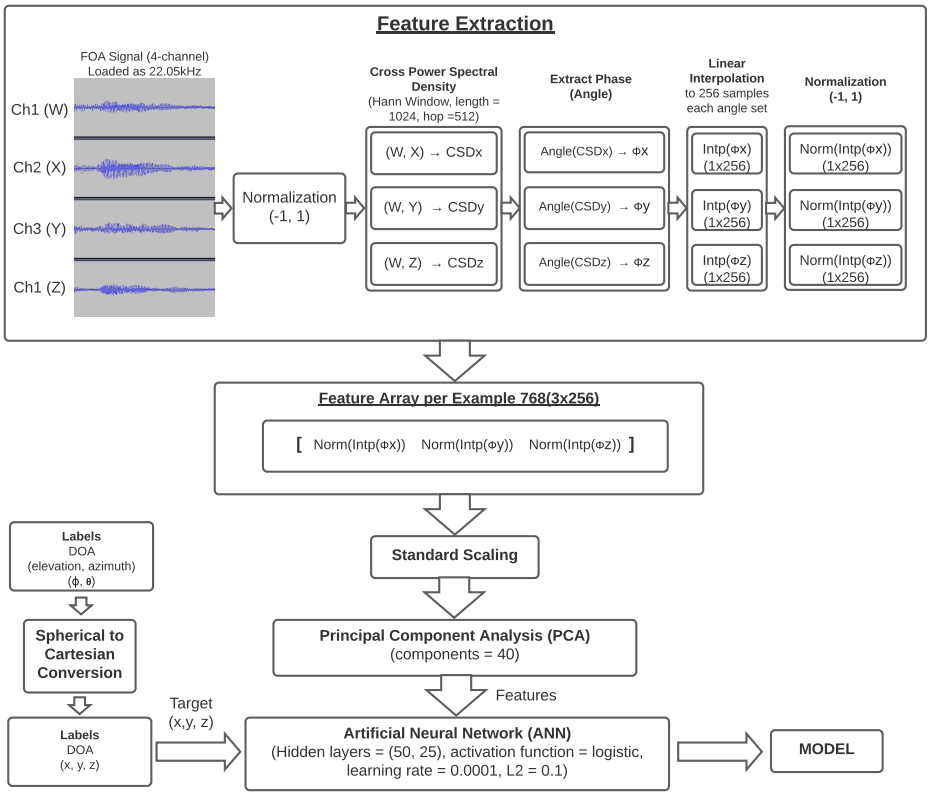

Figure 2: Architecture for DOA estimation solution.

the *Mean Squared Error (MSE)*, which can be considered as a scaled version of the distance in 3D space, and thus the optimization implies the reduction in the distance between two points (Adavanne et al. 2019). Nevertheless the results were still poor, but these operations (normalization and cartesian conversion) were kept for the rest of experiments.

From Cao et al. (2019), the idea of using an *intensity vector* from channel 2,3,4 (X,Y,Z) regarding channel 1 (W) was implemented with an important improvement in the z component ($R^2 \approx 0.3$). From this same work, the approach related with *interchannel phase difference* was used for a new iteration of the solution considering the calculation of the *Cross Power Spectral Density* between the pair channels (W, X), (W, Y), and (W, Z), and taking only the phase. After a normalization (taking the max value) and an interpolation to keep the feature row as the same size, the total of features per example were 768 (3 correlations of 256 elements each). The results gave a total of $R^2 \approx 0.58$ as the average of the three outputs (x,y,z) which was suitable enough for a modest prediction considering the characteristics of the data presented in section 2. Cao et al. (ibid.) explain that this technique is used in estimation of TDoA and Yalta, Nakadai, and Ogata (2017) state that power and phase are important for multi-channel files because of the effects of noise and reflections, however, this work does not use power since it degrades the performance of the solution, presumably a reason could be the overlapping of sounds (there are sounds that contains up to two overlapped sources).

3.2 The Artificial Neural Network Architecture

For this project, a *Full-connected Feed-forward Multi-layer Perceptron regressor* is used. Since there were limitations of time and tools for going deeper into a more sophisticated ANN, the method to find the best hyper-parameters was limited to this type of network. The full architecture of the solution is presented in Fig. 2.

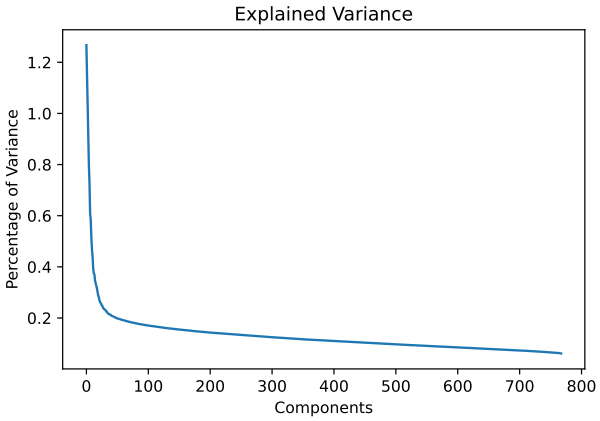

Figure 3: Explained Variance for PCA.

The training chain for this network considered the following steps:

1. A *whitening transformation* as a medium for feature normalization.
2. A *Principal Component Analysis (PCA)* process to reduce the feature dimension from 768 to 40 components.
3. The ANN composed of the following hyper-parameters: 2 hidden layers with 50 and 25 neurons respectively, an iteration limit of 500, a logistic activation function, a constant learning rate of 0.0001, a L2 regularization parameter of 0.1 and the Adam optimization algorithm as solver.

The parameters mentioned above were obtained through experimentation by using the repeated k-fold cross-validation technique with $k = 5$ and 5 repetitions. Details about the evaluation are provided in section 5. The experimentation is related with the feature extraction exploration process presented in section 3.1. A grid-search strategy was applied to every feature extraction case by using: the number of components for PCA, the number of hidden layers and neurons per layer, the activation function, and the regularization parameter. It is important to remark that the inclusion of more hidden layers and neurons were considered, but results on R^2 did not favored them.

Regarding the selection of the number of components in PCA, the grid-search showed the best results for 256 components in a set of values greater and less than this number. However, the *explained variance* for this process was used to find a suitable number of components by preserving a proper percentage of variance. The *elbow* method was applied and, as you can note in Fig. 3, the elbow leads to an approximate value of 40 components, which was used for this application. The difference in R^2 between 256 and 40 components was about 0.01.

The resulting strategy gave the loss curved shown in Fig. 4. It is evident that the 500 iterations were not needed for the convergence and even an earlier stop is possible for similar results.

4 The Implementation

The strategy presented in section 3 was implemented in *Python* through the *Jupyter Notebook* framework. The used packages and specific functionalities within this project are listed in Table 1.

The source code is arranged mainly in three notebooks: *feature extraction*, *training*, and *inference*. The features are saved in two additional data sets (train, and test) to reduce the computational time for training and inference. The model is also saved and retrieved in the *inference* notebook to improve calculation time. This configuration allows independent runs

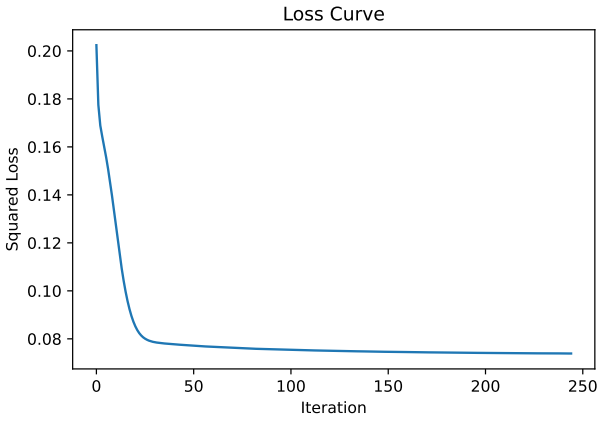

Figure 4: Loss curve for the selected architecture after training.

Package	Functionalities
`numpy`	Liner algebra, and collections' management (arrays and matrices).
`pandas`	Load, manage, and save metadata through CSV files.
`librosa`	Load, and normalize audio files.
`scipy`	Linear interpolation, and cross power spectral density.
`os`	File management to retrieve metadata files descriptors.
`sklearn`	For the ML process: standard scaling, PCA, ANN regressor, cross-validation, evaluation metrics
`matplotlib`	2D and 3D plots for: explained variance in PCA, loss function, and 3D vectors (arrows) to compare estimation.
`joblib`	Save and load ML models.
`soundfile`	For writing extracted examples of sound events to wav. files.

Table 1: Packages used in *Python* and their purpose in the solution.

per each component. A detailed description for the source code is provided in each notebook, as well as references to snipped of code taken externally. The original contributions regarding this implementation are: the code organization, the discrimination of sound events, the feature extraction, the angle error calculation, and the graphical demonstration of specific cases in the evaluation. The source code is available on GitHub. [1]

5 The Evaluation: Results

This problem is framed as a ML supervised regression, which can be evaluated through the error between the actual and the estimated output. Considering the conversion of (ϕ, θ) to cartesian coordinates (x, y, z) as a unit vector, the performance metrics for every stage of the solution for the average of the three outputs are shown in Table 2. *Train* means the metrics between the training set and its prediction, *Validation* is taken from the average values from cross-validation, and *Testing* considers the actual and predicted values in the training set. Note that the angle error for *validation* is not given since this metric is manually calculated and the cross-validation (k-fold) in Python does not provide such specific measurement as score.

Table 3 shows the R^2 for every output for *training* and *testing*, *validation* is not provided because the cross-validation process gave a total R^2, as illustrated in Table 2.

[1] `https://github.com/pedro-lucas-bravo/mct_machine_learning_project`

Stage	Mean Squared Error	Mean Absolute Error	Median Absolute Error	R^2	Angle Error
Train	0.125	0.261	0.203	0.588	31.698°
Validation	0.127	0.265	0.207	0.579	-
Testing	0.115	0.252	0.196	0.614	30.235°

Table 2: Performance metrics.

Stage	x	y	z
Train	0.646	0.647	0.472
Testing	0.669	0.682	0.489

Table 3: *Coefficient of Determination (R^2)* for the DOA unit vector.

The worst, average, and best cases from the testing process show an angle error of 178.34°, 30.24°, and 0.49° respectively. Fig. 5 depicts graphically every case. The yellow vector is the reference (front), the blue is the actual DOA, and the red vector the estimated one.

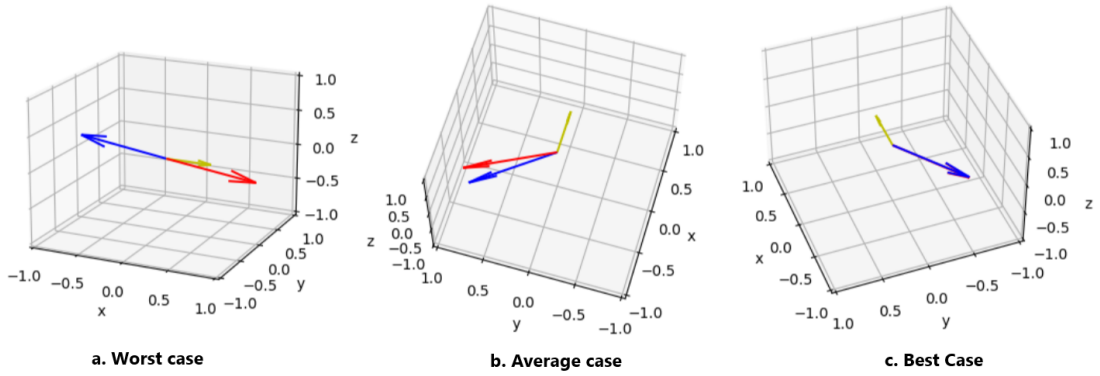

Figure 5: Three cases of interest for DOA estimation.

6 Conclusions and Reflection

The estimation of DOA for FOA files described as in Section 2 is a complex task that has been solved with high accuracy through *Deep Learning Techniques* by combining and modifying different types of ANN such as *Convolutional Neural Networks (CNN)* and *Recurrent Neural Networks* (RNN) as Kapka and Lewandowski (2019), whose solution scored as the best one in the *Sound Event Localization and Detection Challenge (2019)* (DCASE 2019) with an angle error of 4.75° for the training set. In this project the DOA error presented for the same set is 30.235° as shown in section 5, which is an acceptable metric considering a $R^2 = 0.614$ and the lack of precision of humans for an exact estimation.

The comparison between cases in Fig. 5 denotes that the average example is sufficient for the goal of this solution, however the worst case shows an almost completely opposite prediction. Therefore, this solution can be used in applications that do not demand a high precision for DOA, e.g. a system that process FOA files to show feedback as artistic visual objects to reinforce the spatial perception. Other applications, like rescue systems driven by sound, could need a better solution since the task is critical for the situation.

The prediction process needs to be tested in a real-time environment to corroborate whether it is suitable for such conditions or not. The size of the audio buffer and the inference process are

important to determine its usefulness in live setups. Further research is required on small audio segments and with different data than the provided from DCASE (2019) to verify the usefulness to this extent.

Finally, important improvements can be achieved when the ANN architecture is modified in detail from its elemental components (loss function, connectivity, combination of ANN types, etc) as described in previous works. In that sense, it is possible that the failed approaches for feature extraction or the one presented in this project have higher accuracy.

The current solution was challenging to find given the limitations, thus this work intends to give lights to build up an understanding of the process for DOA from a ML perspective.

References

Adavanne, Sharath et al. (2019). "Sound Event Localization and Detection of Overlapping Sources Using Convolutional Recurrent Neural Networks". In: *IEEE Journal on Selected Topics in Signal Processing* 13.1, pp. 34–48. ISSN: 19324553. DOI: 10.1109/JSTSP.2018.2885636. arXiv: 1807.00129.

Bui, Nguyen Khanh, Daisuke Morikawa, and Masashi Unoki (2018). "Method of Estimating Direction of Arrival of Sound Source for Monaural Hearing Based on Temporal Modulation Perception". In: *ICASSP, IEEE International Conference on Acoustics, Speech and Signal Processing - Proceedings* 2018-April, pp. 5014–5018. ISSN: 15206149. DOI: 10.1109/ICASSP.2018.8461359.

Cao, Yin et al. (June 2019). *Two-Stage Sound Event Localization and Detection using Intensity Vector and Generalized Cross-Correlation*. Tech. rep. DCASE2019 Challenge.

Choi, Jeonghwan and Joon Hyuk Chang (2020). "Convolutional neural network-based direction-of-arrival estimation using stereo microphones for drone". In: *2020 International Conference on Electronics, Information, and Communication, ICEIC 2020*. DOI: 10.1109/ICEIC49074.2020.9051364.

DCASE (2019). *Sound Event Localization and Detection - DCASE*. URL: http://dcase.community/challenge2019/task-sound-event-localization-and-detection (visited on 09/14/2021).

Hirvonen, Toni (2015). "Classification of spatial audio location and content using Convolutional neural networks". In: *138th Audio Engineering Society Convention 2015* 2.May, pp. 622–631.

Kapka, Slawomir and Mateusz Lewandowski (June 2019). *Sound Source Detection, Localization and Classification Using Consecutive Ensemble of CRNN Models*. Tech. rep. DCASE2019 Challenge.

MathWorks (2021). *Cross-Correlation of Delayed Signal in Noise - MATLAB and Simulink*. URL: https://www.mathworks.com/help/signal/ug/cross-correlation-of-delayed-signal-in-noise.html (visited on 09/14/2021).

Nguyen, Thi Ngoc Tho et al. (July 2021). "What Makes Sound Event Localization and Detection Difficult? Insights from Error Analysis". In: November. arXiv: 2107.10469. URL: http://arxiv.org/abs/2107.10469.

Schmidt, R. (Mar. 1986). "Multiple emitter location and signal parameter estimation". In: *IEEE Transactions on Antennas and Propagation* 34.3, pp. 276–280. ISSN: 0096-1973. DOI: 10.1109/TAP.1986.1143830. URL: http://ieeexplore.ieee.org/search/srchabstract.jsp?arnumber=5265379%20http://ieeexplore.ieee.org/document/1143830/.

Wierzbicki, J., P. Małecki, and J. Wiciak (2013). "Localization of the sound source with the use of the first-order ambisconic microphone". In: *Acta Physica Polonica A* 123.6, pp. 1114–1117. ISSN: 05874246. DOI: 10.12693/APhysPolA.123.1114.

Yalta, Nelson, Kazuhiro Nakadai, and Tetsuya Ogata (2017). "Sound source localization using deep learning models". In: *Journal of Robotics and Mechatronics* 29.1, pp. 37–48. ISSN: 18838049. DOI: 10.20965/jrm.2017.p0037.