LeaderSTeM-A LSTM model for dynamic leader identification within musical streams

Sutirtha Chakraborty[1], Shyam Kishor[2], Subham Patil[3], and Joseph Timoney[1]

[1] Maynooth University,Ireland
[2] Genpact, Bangalore
[3] SVNIT, India
sutirtha.chakraborty@mu.ie

Abstract. Musical Ensembles have fully connected multiple leader-follower topologies where the leadership role can change dynamically from one musician to another. This makes it a complex task for tracking the correct leader based on audio features such as pitch, rhythm, and amplitude. In this study, we propose 'LeaderSTeM', a deep learning model to predict and follow the leader by tempo. We built different models and evaluated the results for this problem and found the LSTM models to be most effective. Furthermore, we examined the hyper-tuning of 3-, 4- and 5-layered LSTMs to determine the best possible models.

Keywords: Leader-follower, LSTM, Musical ensemble, Deep Learning

1 Introduction

A musical ensemble is a collaborative performance of vocal and/or instrumental music by a group of musicians and singers. In a jazz or popular music, there are generally melodic instruments, accompanying instruments providing a harmonic underpinning, a bass instrument, and a percussionist. Rock ensembles typically have vocals, keyboards and guitars and a rhythm section composed of a drum kit and bass guitar (Chang, Livingstone, Bosnyak, & Trainor, 2017). Classical ensembles exist in different forms. The presence of percussionists depends on the size of the group, and the harmonic accompaniment could be provided by multiple monophonic instruments, such as violas and cellos playing separate lines, or polyphonic instruments such as a piano or guitar. Small ensembles are groups of musicians numbering from two to eight, while larger group can vary from 10-100 players, and sometimes more.

Musicians continuously communicate with each other during a group performance through sound, body movements, and facial gestures to bring life to their art and to connect with audience emotion. There is a time-varying leader-follower relationship where the leader plays a key role in the appropriate functioning of the system (Harrell, 2008) . The leader-follower relationship is typical of an interconnected network of musicians (Niemeyer & Cavazotte, 2016) (Kawase, 2014) and Fig.1 illustrates how such a topology may appear, where the conductor is the leader.

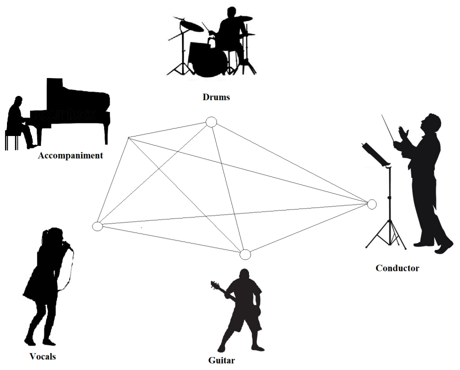

Fig. 1. Interaction Topology among musicians during a performance

The motivation for this paper has come from attempting to understand live human-robot musical ensemble interactions, such as a single robot with three or four musicians. The robot should respond and synchronize to the human musicians. To do this effectively they need to observe the tempo, or BPM (beats per minute), of the musicians in real-time, because most music is played with some tempo variation. A well-known real-time BPM trackers is AUBIO (Brossier, 2006). Ideally, the robot should be driven with a signal that consistently captures a combined, representative BPM of the ensemble. However, if the robot can observe the real-time BPM of the different instruments, it will be seen that the most representative BPM values can shift between these instruments depending on the notes and rhythm they are playing. Thus, while itself is playing, or following, the robot must dynamically identify and track this representative BPM to stay in time. In the research into musical ensembles there has been little effort in the literature on identifying the tempo leader in an ensemble who is establishing the BPM of the combined musical output. This is only now an emerging area as the field of inter-entity synchronization. This is a difficult problem and the intention in this paper is to use a machine learning approach to develop this dynamic 'Leader' tracker. This algorithm would extract the individual real-time BPM signals from each instrument along with a set of associated audio features, and then discern from these at each time interval which instrument track is the 'leader' that currently dominates the overall representative BPM. This BPM signal could be used to drive a synchronization algorithm if more than one musical robot is present. To test the algorithm we used an audio stem dataset to represent the musical ensemble because within each music piece the leadership control of the BPM was observed to shift between the stems. This learning model was designed to identify and follow the correct stem leader based on the dynamic audio features of the different instruments and voice sources, and output an overall, representative BPM signal. Our investigation included the machine learning techniques of Random forest, Support Vector Machine(SVM), and a Long Short-

Term Memory (LSTM) based model. Once it was established that the LSTM deep learning model performed best, it was termed 'LeaderSTeM'. Six different hyper tuning algorithms were then applied to identify an optimal LSTM model. The next section introduces some background to our work.

2 Literature Review

Glowinski et al. (Glowinski, Badino, Ausilio, Camurri, & Fadiga, 2012) studied an ensemble of a string quartet to understand how the leadership behavior among musicians related to their non-verbal cues. The used Granger Causality to investigate how each musician's behavior was influenced by their distance from other musicians when choosing their potential leader. Measuring the inter-musician communication showed, as expected, that the leader had a significant impact on the interaction of the whole network. Similarly,Timmers et al. (Timmers, Endo, Bradbury, & Wing, 2014) did a case study on auditory and visual cues within a string quartet. They repeated the experiment many times on the same musical piece. They observed that the first Violin player was acting as the primary leader and the speed of their bow movement during the first onset tone was responsible for setting the tempo of the whole performance. They also found that visual gaze is an important feature for establishing the connection in the beginning. Kawase (Kawase, 2014) investigated the leadership phenomenon by experimenting with six piano players. He created constraints under which musicians could be compared and evaluated. His study found that the leader's gaze towards other musicians was shorter than that of the followers. He also found that before any tempo changes occurred that it was visual cues that helped them to maintain the synchrony. Although not entirely relevant, Martin et al.(Martin, Ellefsen, & Torresen, 2017) used an artificial neural network (ANN) model to create an ensemble experience using a gesturally-controlled touch-screen interface to make music. They demonstrated the value of long short-term memory (LSTM) neural networks to force synchronization among the participants.

While it is clear that gaze and visual cues, or gesture, are a the primary mode of communication, this work, in contrast, will take a different approach by relying only on the dynamic relationship between the audio features of the individual waveforms produced by the ensemble of musicians, and identify the leader among them. This result would be especially useful in an analysis scenario where no video or eye-tracking or gestural data is available, but only the audio data, which is often the case.

3 Dataset

In this experiment we used the Musdb18 dataset which consists of a total of 150 complete music tracks (Rafii, Liutkus, Stöter, Mimilakis, & Bittner, 2017)(10 hours in total). These are of distinct genres that contain a mix of vocals, drums, bass, and accompaniment tracks. All tracks are stereophonic as well as being sampled at 44.1kHz. To replicate an ensemble scenario, we separated the main, full-mix, track into four stems or individual instrument sub-tracks (drums, bass, accompaniment, voice). We used PYAudio (Pham, 2006) to capture the tempo, pitch, and amplitude/volume for each sub-track and the full-mix track in real-time with a window size of 1024 and hop-size 512 using AUBIO's (Brossier, 2006) library. For training the various machine learning techniques, the input included the audio features derived from the four sub-tracks and the BPM of the full-mix track as shown in Fig.2. The sample code and dataset are available in the GitHub repository (GitHubRepo, 2020).

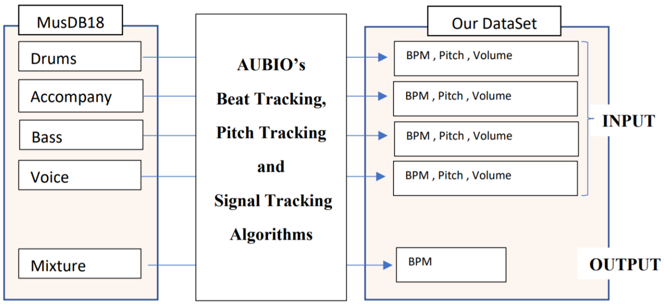

Fig. 2. Dataset Generation from MUSDB18 using AUBIO for Machine learning algorithms

4 Learning models and hyper tuning

We split our dataset into training and testing data (8:2 ratio) . We explored three different machine learning models. These were random forest, SVM, and LSTM. Random forest is a learning algorithm that constructs groups of decision trees during the training phase. The higher the number of trees, the more precise the outcome (Farner, Solvang, Sæbo, & Svensson, 2009), and it is known for its capability to avoid overfitting. SVMs are based on the principle of structural risk minimization. The SVM uses a non-linear representation of the data in a high-dimensional character space for function approximation and then consecutively performs a linear regression (Cortes & Vapnik, 1995). It is more efficient than Random Forest and works well in high dimension space problems. LSTM is a neural network where each network layer has a set of recurrently connected memory blocks (Borovkova & Tsiamas, 2019) . The LSTM is known to be capable of learning long-term dependencies.

4.1 Ray hyper tuner

To find the best model parameters for the LSTM Neural Networks, we investigated six different hyper tuning algorithms using Ray tuner (Liaw et al., 2018).

This provided a large search space. The following tuning algorithms applied were as follows:

AxSearch It uses the Ax platform for hyperparameter optimization. The Ax program is used to recognize, maintain, expand, and to automate adaptive tests. It presents a simple interface with BoTorch, a manageable and functional library for Bayesian optimization.

SkoptSearch Skopt or scikit-optimize is a black-box optimization library.A skopt-optimizer is required for the search algorithm (Markov, 2017).

HyperOptSearch This is a Python library for serial and parallel optimization. It includes optimization over difficult search spaces consisting of valued, discrete, and conditional dimensions (Bergstra, Komer, Eliasmith, Yamins, & Cox, 2015). It uses the Tree-structured Parzen Estimators algorithm.

BayesOptSearch It uses Bayesian Optimization for hyperparameters optimization which is a library for Bayesian optimization (Martinez-Cantin, 2013).

TuneBOHB BOBH or Bayesian Optimization HyperBand has dual functions. It terminates bad tests and improves the hyperparameter search through Bayesian Optimization, supported by 'HpBandSter' library.

ZOOptSearch It is a derivative-free optimization that is supported by ZOOpt package. Presently, it uses Asynchronous Sequential RAndomized COordinate Shrinking (ASRacos) for adjusting the hyper tuning.

5 Experimental Trails

To check if there was any redundant information in the set of features, we used Principal Component Analysis (PCA). The relationship between the number of components and the variance is shown in Fig.3. From the figure the relative contributions of the features can be seen. Observing the cumulative value of the variance, it reaches one when all the features were included. It was decided that there was insufficient redundancy so that all the features should be used to build the model.

On evaluating all three machine learning models first of all we found that a two-layered LSTM model performed much better than both the SVM and Random Forest when using the mean square error value, as shown in Table 1.

We then built three, four, and five-layer LSTM models to do a comparative study across these models. We used six different hyper-tuning algorithms to find the best model. The search space explored for the learning rate was 0.001

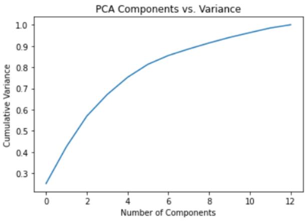

Fig. 3. PCA on dataset

Table 1. Performance of Time-Series Models on Dataset

	Random Forest	**SVM**	**2-Layered LSTM**
Mean Square Error	481	402	345

to 0.1 and the number of LSTM units for each layer ranged from 2 to 512. To perform the intensive computations, we used four CPU cores and a 12GB NVIDIA Tesla K80 GPU to run the code. We generated a total of 25 samples for each tuning algorithm, each with 10 iterations. The batch size was 32 for each case. We chose the best model for LeaderSTeM by comparing the performance of 18 models generated by each tuning algorithm, as shown in Table 2, using the mean squared error (MSE) value. It was found that the best overall was the three-layered LSTM model with hidden nodes of 445, 481, 37 in the first, second, and third layers respectively, having a learning rate of 0.082, and an output layer with one unit. The AxSearch tuning for this model gave an MSE of 245.

6 Results

To illustrate the operation of the LeaderSTeM model we used the mixer-track of the file 'A Classic Education – NightOwl' from MUSDB18. This track was specifically chosen as it had a significant tempo/BPM changes in the middle of the song which would clearly demonstrate how the model could adapt to tempo variations. The BPM, pitch, and volume of each of the four source-separated sub-tracks of the mixer-track were extracted and fed to our model as input.

Table 2. Experimental Values of different LSTM Models

	3 Layered LSTM					**4 Layered LSTM**						**5 Layered LSTM**						
Tune Algorithm	*Learning Rate*	*Unit 1*	*Unit 2*	*Unit 3*	*MSE*	*Learning Rate*	*Unit 1*	*Unit 2*	*Unit 3*	*Unit 4*	*MSE*	*Learning Rate*	*Unit 1*	*Unit 2*	*Unit 3*	*Unit 4*	*Unit 5*	*MSE*
Ax Search	0.082	445	481	37	245	0.075	390	391	197	114	271	0.076	238	388	69	141	477	347
Skopt Search	0.0835	457	491	288	273	0.083	433	330	70	9	289	0.083	457	491	288	48	510	290
HyperOpt Search	0.0979	308	500	358	471	0.095	309	352	155	413	292	0.098	450	173	434	424	426	275
BayesOpt Search	0.038	485	375	307	308	0.073	342	509	209	276	279	0.075	439	302	188	226	402	294
TuneBOHB Search	0.0096	294	256	313	504	0.0082	394	407	349	486	514	0.008	177	452	422	470	377	539
ZOOpt Seach	0.06	413	269	160	285	0.08	160	210	305	206	315	0.1	230	185	488	413	114	277

In Fig.4 four subplots can be seen that show the BPM signals extracted. Each plot superimposes the AUBIO-identified real-time BPM signal of the different sub-tracks (in orange) onto the AUBIO-identified mixer-track BPM signal (in blue). We can observe that at different parts of the song the mixer-track BPM signal matches more closely with that of the different sub-tracks. Initially, the mixer-track BPM follows the accompanying track. In the next phase, the mixer-track BPM significantly slows down. The bass and drum BPM signals produce similar behaviour to each other, and these then dominate the BPM mixer-track trajectory. In the last phase, the tempo increases. The vocal track BPM signal is now primary alongside the drum and bass BPM signal, thus having the greatest impact on the overall mixer-track BPM.

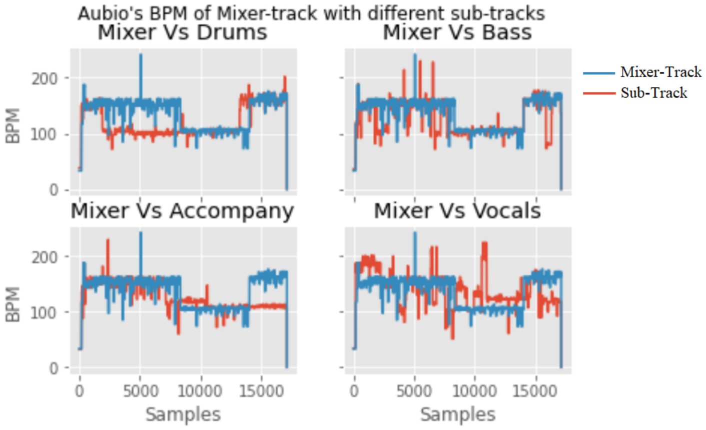

Fig. 4. Mixer Track vs Sub-tracks BPM signal comparison for 'A Classic Education – NightOwl'

The output of the LeaderSTeM model BPM output is now shown (blue line) in the panels Fig.5 against the BPM signal of the four sub-tracks (orange line). The LeaderStem output initially follows the vocals sub-track as the song starts with a strong vocal amplitude. This contrasts with the identification of the accompaniment as the leader sub-track. Following this, the model then correctly identifies the BPM signal similarity between the bass and voice signals. It outputs BPM values that are close to those of the BPM signal for the bass sub-track. When the tempo reduces mid-song, LeaderSTeM was able to identify the tempo slowdown and followed the BPM signals of the bass and drums. In the last phase, the model was able to correctly judge the dominant three instruments (Vocals, Bass, and Drums) and follow the increase in tempo. In summary, while it was initially unable to find the right sub-track this improved, and afterwards it maintained tracking correctly.

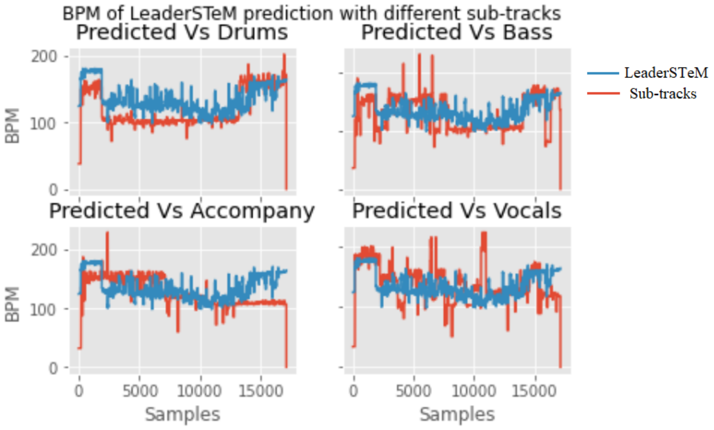

Fig. 5. LeaderSTeM Prediction for 'A Classic Education – NightOwl'

7 Conclusion

To emulate an ensemble scenario this paper used the MUSDB18 dataset, and then stems were extracted from it. Using only features derived from the audio stem sources an investigation into machine learning tools was made. It tested how well they could identify the dynamic leading instrument among the individual tracks that contributed most to the representative BPM signal at various times in the musical piece. The LeaderSTeM machine learning algorithm was observed as being the best at "following the leader" among the ensemble. This was a three-layer LSTM with an optimized set of parameters. The result is useful to robotic-human musical interaction scenarios that requiring real-time BPM analysis of an ensemble musical outputs.

For future work, we will experiment with a more extensive range of audio features along with exploring an expanded search space for the model parameter values. Incorporating data from tracking the visual gaze of the musicians to identify the leader over time would also be valuable (Bishop, 2019). The integrating of these visual features representing gestural intent would complement and corroborate the sound features in the dataset and would most likely strengthen the predictive performance.

References

Bergstra, J., Komer, B., Eliasmith, C., Yamins, D., & Cox, D. D. (2015). Hyperopt: a python library for model selection and hyperparameter optimization. *Computational Science & Discovery*, *8*(1), 014008.

Borovkova, S., & Tsiamas, I. (2019). An ensemble of lstm neural networks for high-frequency stock market classification. *Journal of Forecasting*, *38*(6), 600–619.

Brossier, P. M. (2006). *Automatic annotation of musical audio for interactive applications* (Unpublished doctoral dissertation).

Chang, A., Livingstone, S. R., Bosnyak, D. J., & Trainor, L. J. (2017). Body sway reflects leadership in joint music performance. *Proceedings of the National Academy of Sciences*, *114*(21), E4134–E4141.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine learning*, *20*(3), 273–297.

Farner, S., Solvang, A., Sæbo, A., & Svensson, U. P. (2009). Ensemble hand-clapping experiments under the influence of delay and various acoustic environments. *Journal of the Audio Engineering Society*, *57*(12), 1028–1041.

GitHubRepo. (2020). Sutirthachakraborty/leaderstem. *GitHub*. Retrieved from `https://github.com/SutirthaChakraborty/LeaderSTeM`

Glowinski, D., Badino, L., Ausilio, A., Camurri, A., & Fadiga, L. (2012). Analysis of leadership in a string quartet. In *Third international workshop on social behaviour in music at acm icmi 2012* (pp. 763–774).

Harrell, M. (2008). The relationships between leader behavior, follower motivation, and performance.

Kawase, S. (2014). Assignment of leadership role changes performers' gaze during piano duo performances. *Ecological Psychology*, *26*(3), 198–215.

Liaw, R., Liang, E., Nishihara, R., Moritz, P., Gonzalez, J. E., & Stoica, I. (2018). Tune: A research platform for distributed model selection and training. *arXiv preprint arXiv:1807.05118*.

Markov, S. (2017). Skopt documentation.

Martin, C. P., Ellefsen, K. O., & Torresen, J. (2017). Deep models for ensemble touch-screen improvisation. In *Proceedings of the 12th international audio mostly conference on augmented and participatory sound and music experiences* (pp. 1–4).

Martinez-Cantin, R. (2013). Bayesopt: A library for bayesian optimization with robotics applications. *arXiv preprint arXiv:1309.0671*.

Niemeyer, J. R. L., & Cavazotte, F. D. S. C. N. (2016). Ethical leadership, leader-follower relationship and performance: a study in a telecommunications company. *RAM. Revista de Administração Mackenzie*, *17*(2), 67–92.

Pham, H. (2006). Pyaudio: Portaudio v19 python bindings. *URL: https: // people. csail. mit. edu / hubert / pyaudio*.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., & Bittner, R. (2017). Musdb18-a corpus for music separation.

Timmers, R., Endo, S., Bradbury, A., & Wing, A. M. (2014). Synchronization and leadership in string quartet performance: a case study of auditory and visual cues. *Frontiers in Psychology*, *5*, 645.