Content-Based Music Classification Using Ensemble of Classifiers

Manikanta Durga Srinivas Anisetty[1(✉)], Gagan K Shetty[1(✉)], Srinidhi Hiriyannaiah[1], Siddesh Gaddadevara Matt[1], K. G. Srinivasa[2], and Anita Kanavalli[1]

[1] Ramaiah Institute of Technology, Bengaluru, India
amd.srinivas@gmail.com, gagankshetty@gmail.com, srinidhi.hiriyannaiah@gmail.com, siddeshgm@gmail.com, anita.kanavalli@gmail.com
[2] Ch. Brahm Prakash Government Engineering College, Delhi, India
kgsrinivasa@gmail.com

Abstract. This paper presents an application of Ensemble learning in the field of audio data analytics. We propose a system using Hierarchical ensemble model to classify the genre of a music track based on the contents of the track. The hierarchical ensemble comprised of 7 classifiers trained on different sections of the dataset that can co-relate the output of each other for classifying the data. Using this hierarchical ensemble model, we achieved an accuracy boost of 15% over machine learning models. This hierarchical ensemble has been proven better than an ensemble model with hard voting logic in term of accuracy. This work describes the comparison of basic models with hierarchical model and its characteristics.

Keywords: Music classification · Machine learning
Ensemble learning · Free music archive

1 Introduction

In the rising world of technology and intelligent systems, audio sensing and voice commands can be a vital interaction means that can help in improving the user interaction with a system. Building accurate systems to work with audio data helps in achieving this.

Classification corresponds to the task of training a system to identify different types of entities in the data. This involves probabilistic inference that a data point belongs to certain class based on the training data provided to the system. These trained systems are called classifiers or estimators. There are various applications for this process in the fields of medical imaging, image recognition, sentiment analysis and many more. One such application is classifying music into different genres.

Ensemble learning corresponds to training multiple classifiers and co-relating each other to achieve a collaborative output. In this technique, we use multiple

U. S. Tiwary (Ed.): IHCI 2018, LNCS 11278, pp. 285–292, 2018.
https://doi.org/10.1007/978-3-030-04021-5_26

classifiers that are trained on different sections of the dataset and combined using an ensemble logic, that defines the behaviour of the ensemble model. This method helps in increasing the performance of the system by narrowing down the problem solved by each classifier, eventually solving the entire problem through ensemble.

In this work, we propose an ensemble model that is trained to classify the genre of a music track. As discussed in further sections, the ensemble approach helped in increasing the accuracy over standalone systems.

2 Related Works

Transfer learning is a very promising technique in machine learning. It involves using a pre-trained model's parameters for a task different than the one it has been trained for. This method is very helpful for problems with limited data available. The authors of this paper [1,2] suggest using deep learning techniques for feature extraction. The concept of a deep neural network is that as the data goes through the forward phase, where the lower level features are learnt in the first few layers, for example the edges of objects in case of image classification tasks, then the later layers build upon the lower features and learn the higher features, for example the objects themselves. A similar logic exists when this is applied to a music classification task, hence the idea for the transfer learning task for the classification of music. The final list of features generated are not just the output of the network, but also the activations of the intermediate layers, which are concatenated to the features list. Kapre provides on-GPU preprocessing layers for music data which can aid in feature extraction tasks [3]. Kapre is built specifically for audio data using keras. It provides easy-to-use classes for Spectogram and Melspectrogram that are used for audio pre-processing tasks.

In computer vision tasks, there are several well established datasets for bench marking, such as the MNIST [4], CIFAR [5], ImageNet [6]. The Free music archive (FMA) [7] intends to provide a dataset for bench marking for music classification tasks, as the lack of standardised datasets hinders the research in music information retrieval systems.

This work [9] proposes a new technique called 'DWCHs', which captures the information of the music files based on the wavelet histogram. Wavelets has many applications in the fields of data mining [10], which makes it a promising technique in this field.

A huge set of features is a burden on classifiers and they generally end up overfitting. A novel method was proposed for dimensionality reduction called "Locality Preserving Non-Negative Tensor Factorization (LPNTF)" [11]. Using this technique, they were able to beat the state of the art models on the GTZAN and ISMIR2004 datasets. Dimensionality reduction also reduces the effects of the outliers, noise and missing data.

Support vector machine is a very effective algorithm. The authors of this paper [12] have described the advantages of a support vector machine for the task of music classification. Since we work on classifying multiple genres, typically a Multi-class Support Vector Machine is suitable. One way is to use the

One against the rest method [13] which is widely used in Multi-class classification problems. The features extracted for the music files are not linearly separable [8] and hence a linear classification algorithm will fail to classify the music accurately. SVMs can learn non-linear functions using a kernel trick [14].

The authors of the paper [15] proposed a novel ensemble-based approach based on a segmentation strategy presented. Several representations of the digital audio signal were used since each segment generates a different feature vector. When using this segmentation strategy, it is possible to train a specific classifier for each one of the segments, and to compute the final decision about the class from the ensemble of the results provided by each classifier. The output of each classifier is taken and a combination of the results is achieved through the majority voting rule, max rule, sum rule and product rule.

The papers [16–18] discuss the applications of hierarchical ensemble in the fields of biology and computer vision. The authors have reported significant improvement in tasks such as solving protein function prediction and image classification problems using this technique.

3 Dataset

We used Free Music Archive dataset for the training purpose. This dataset comprised of 106,574 tracks arranged in 161 hierarchical genres. We used a small subset of the dataset which comprised of 8000 tracks from 8 genres. Each music track in the dataset is of 30 s duration.

4 System Developed

The system is built as 2 blocks:

- Feature Extraction
- Classification

The feature extraction module is used to extract the features from the given music files. The extracted features are fed to the classifier for prediction of the genre. The system design is shown in Fig. 1.

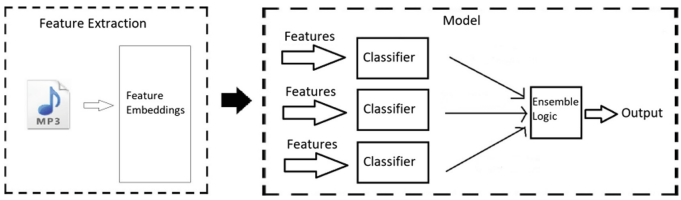

Fig. 1. System design

4.1 Feature Extraction

We used the kapre audio processing library for the feature extraction. Kapre is an audio processing module built over Keras to accelerate the deep learning tasks involved in audio processing tasks. We used a 5 layer convolutional neural network trained using Theano and Kapre by the authors of this paper [1]. The network outputs a 160 dimensional vector on the input file which serves as the extracted feature vector.

4.2 Classification

The vital step in a classification task is to choose an estimator that is suitable for the dataset used. The dataset used is inseparable with linear models due to the overlap of the data points across multiple genres. This resulted in poor accuracy with linear models. In order to achieve better accuracy, Support Vector Machines are used to learn the non-linearity in the data. Since we are working with a hierarchical ensemble model, we train a series of binary SVM estimators.

SVM classifies the data into classes by using a maximum margin separator. Kernel trick is used for non-linear fitting by allowing to fit the separator in a higher dimensional transformed space. The non linear function used for the purpose of this experiment is the Gaussian radial basis function.

4.3 Ensemble of Classifiers

Ensemble methods use multiple learning algorithms to obtain better predictive performance than could be obtained from any of the individual learning algorithms alone. This enables the system to intelligently decide the appropriate classifier suitable for the data point to be classified.

Hierarchical Ensemble. Hierarchical ensemble is an ensemble method which involves multiple classifiers with each classifier trying to predict a subset of the classes. Subsets that are highly separable are at higher levels in the hierarchy and as we go down the hierarchy, the genres get difficult to separate. Initially, we have a root classifier that segregates the classes into subsets of classes. The lower level classifiers are then trained on these subsets to either predict a final class or further divide the class into subsets of classes. This segregation of the further possible classes at each level results in reaching the final result with a better confidence even if the final result is inaccurate. This allows for backtracking of the classification result along with the performance boost.

Algorithm. The decision-making path will be based on the output of the classifiers, i.e., the sub-tree of the predicted class will be considered. For example, if root class predicts Class-A, then the Class-A classifier's result will be considered, and Class-B classifier's result will be ignored. The algorithm developed is as described in Algorithm 1.

```
Data: Features
Result: Genre of the music file
PrimaryPrediction ← RootClassifier(Features);
if PrimaryPrediction = ClassA then
    SecondaryPrediction ← ClassAClassifier(Features);
    if SecondaryPrediction = ClassAA then
        FinalPrediction ← ClassAAClassifier(Features);
    else
        FinalPrediction ← ClassABClassifier(Features);
    end
else
    SecondaryPrediction ← ClassBClassifier(Features);
    if SecondaryPrediction = ClassBA then
        FinalPrediction ← ClassBAClassifier(Features);
    else
        FinalPrediction ← ClassBBClassifier(Features);
    end
end
return FinalPrediction;
```

Algorithm 1. Hierarchical ensemble algorithm

5 Experiment

The dataset was randomly split with 70–30 basis into training and testing sets, resulting in 5,600 tracks for training and 2,400 tracks for testing.

For the implementation of the classifiers, we used the Sklearn [19] library of Python to generate the individual machine learning models. For the ensemble classifier, we used 7 of such machine learning models, each trained individually on its own special handcrafted dataset. To achieve this, we modified the labels of the original dataset as described below:

1. For the root classifier, the genres Folk, International, Pop, Rock were treated as Class A and the genres Electronic, Experimental, Hip-Hop, Instrumental as Class B. The root classifier was hence trained on 8000 tracks, with two labels: Class A and Class B.
2. For the next level, the subset of the genres were considered. For example, Class A classifier only dealt with the genres Folk, International, Pop, Rock and the Class B classifier dealt with the remaining 4. The genres Folk and International were treated as ClassAA and the genres Pop and Rock were treated as ClassAB. Hence, the Class A classifier was trained on 4000 tracks, with the two labels: ClassAA and ClassAB. Class B follows the same logic.
3. For the third level, again a subset of the genres were considered. For example, Class AA classifier only dealt with the genres Folk and International and Class AB classifier the remaining 2, and similarly the Class BA and Class BB classifier. The split of the genres is illustrated in the Fig. 2.

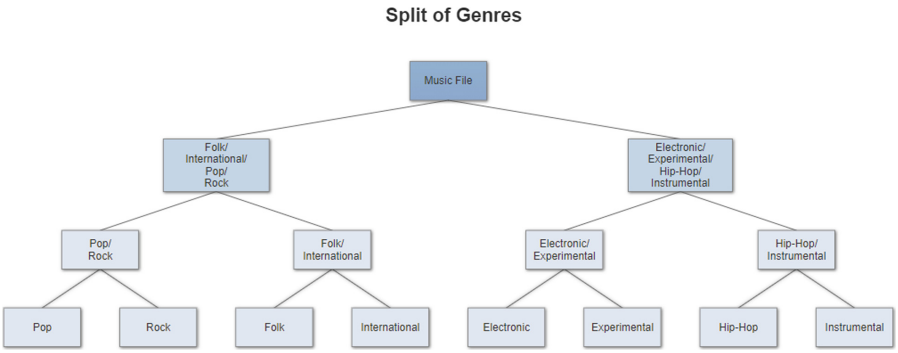

Fig. 2. Genre split

In this way 7 classifiers were trained and the decision making path is decided by the ensemble algorithm. The masking of the genres, i.e, choosing the correct tracks at each step was done using the Pandas [20] library in python.

6 Results

6.1 Ensemble Evaluation

The system achieved an accuracy of 75% over the testing set. The confusion matrix for the testing dataset is as shown in Table 1. The time taken for training the models on the dataset is 43 s (after feature extraction). The average time taken by the system to classify a music file of 30 s duration is 15 s (feature extraction + prediction).

Table 1. Confusion matrix

Predicted	Electronic	Experimental	Folk	Hip-Hop	Instrumental	International	Pop	Rock
Actual								
Electronic	243	22	4	8	13	2	4	4
Experimental	38	176	17	7	17	12	9	24
Folk	0	4	250	1	18	12	4	11
Hip-Hop	23	6	2	251	8	1	7	2
Instrumental	19	14	9	6	233	2	1	16
International	16	2	16	5	1	245	4	11
Pop	16	9	33	11	7	21	146	57
Rock	4	8	11	4	5	5	14	249

6.2 Comparison with Different Models

Along with evaluating the model developed, a comparative study was conducted to understand the improvement in performance of the ensemble model in terms

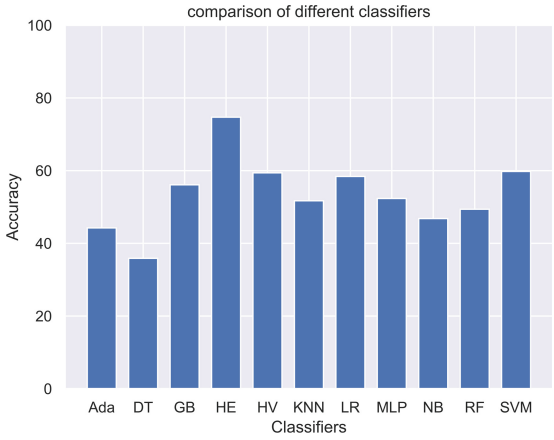

Fig. 3. Comparison of the models

of accuracy is shown in Fig. 3. We considered 10 existing estimators along with the hierarchical ensemble (HE). They are Naive Bayes classifier (NB), Logistic Regression model (LR), SVM with rbf kernel (SVM), Decision tree classifier (DT), Decision tree classifier (DT), K-nearest neighbours estimator (KNN), Multi-Layer Perceptron (MLP), Random forest classifier (RF), AdaBoost Classifier (Ada) [21], Gradient Boost Classifier (GB), Hard-voting ensemble model (HV) [a combined model of previously mentioned models]. Each of these models were trained with a 3 fold cross validation, and the average is taken as the final accuracy in order to reduce the bias effect.

7 Conclusion

Using a hierarchical ensemble has evidently improved the accuracy of the classification. Using more intelligent ensemble algorithms can lead to more accurate ensemble models. This approach can be extended to various other audio analytic tasks to arrive at accurate and improved ensemble algorithms.

References

1. Choi, K., Fazekas, G., Sandler, M., Cho, K.: Transfer learning for music classification and regression tasks. arXiv preprint arXiv:1703.09179 (2017)
2. Zeiler, M.D., Fergus, R.: Visualizing and understanding convolutional networks. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8689, pp. 818–833. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10590-1_53
3. Choi, K., Joo, D., Kim, J.: Kapre: On-GPU audio preprocessing layers for a quick implementation of deep neural network models with keras. arXiv preprint arXiv:1706.05781 (2017)

4. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-based learning applied to document. Proc. IEEE **86**(11), 2278–2324 (1998)
5. Krizhevsky, A., Hinton, G.: Learning multiple layers of features from tiny images (2009)
6. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: a large-scale hierarchical image database. In: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2009, pp. 248–255. IEEE, June 2009
7. Defferrard, M., Benzi, K., Vandergheynst, P., Bresson, X.: FMA: A dataset for music analysis. arXiv preprint arXiv:1612.01840 (2016)
8. Tzanetakis, G., Cook, P.: Musical genre classification of audio signals. IEEE Trans. Speech Audio Process. **10**(5), 293–302 (2002)
9. Li, T., Ogihara, M., Li, Q.: A comparative study on content-based music genre classification. In: Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 282–289. ACM, July 2003
10. Li, T., Li, Q., Zhu, S., Ogihara, M.: A survey on wavelet applications in data mining. ACM SIGKDD Explor. Newsl. **4**(2), 49–68 (2002)
11. Panagakis, Y., Kotropoulos, C., Arce, G.R.: Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In: ISMIR, vol. 14, no. 1, pp. 249–254, October 2009
12. Mandel, M.I., Ellis, D.: Song-level features and support vector machines for music classification. In: ISMIR, vol. 2005, pp. 594–599, September 2005
13. Weston, J., Watkins, C.: Multi-class support vector machines. Technical report CSD-TR-98-04, Department of Computer Science, Royal Holloway, University of London, May 1998
14. Suykens, J.A.: Nonlinear modelling and support vector machines. In: Proceedings of the 18th IEEE Instrumentation and Measurement Technology Conference, IMTC 2001, vol. 1, pp. 287–294. IEEE, May 2001
15. Silla Jr., C.N., Kaestner, C.A., Koerich, A.L.: Automatic music genre classification using ensemble of classifiers. In: IEEE International Conference on Systems, Man and Cybernetics, 2007, ISIC, pp. 1687–1692. IEEE, October 2007
16. Valentini, G.: Hierarchical ensemble methods for protein function prediction. ISRN Bioinform. **2014** (2014). https://doi.org/10.1155/2014/901419
17. Kim, B.S., Park, J.Y., Gilbert, A.C., Savarese, S.: Hierarchical classification of images by sparse approximation. Image Vis. Comput. **31**(12), 982–991 (2013)
18. Huang, J., Kumar, S.R., Zabih, R.: An automatic hierarchical image classification scheme. In: Proceedings of the Sixth ACM International Conference on Multimedia, pp. 219–228. ACM, September 1998
19. Pedregosa, F., et al.: Scikit-learn: machine learning in Python. J. Mach. Learn. Res. **12**, 2825–2830 (2011)
20. McKinney, W.: Pandas: a foundational Python library for data analysis and statistics. Python High Perform. Sci. Comput., 1–9 (2011)
21. Hastie, T., Rosset, S., Zhu, J., Zou, H.: Multi-class adaboost. Stat. Interface **2**(3), 349–360 (2009)