Hierarchical Multilabel Classification and Voting for Genre Classification

Benjamin Murauer, Maximilian Mayerl, Michael Tschuggnall, Eva Zangerle, Martin Pichl, Günther Specht
University of Innsbruck, Austria
firstname.lastname@uibk.ac.at

ABSTRACT

This paper summarizes our contribution (team DBIS) to the AcousticBrainz Genre Task: Content-based music genre recognition from multiple sources as part of MediaEval 2017. We utilize a hierarchical set of multilabel classifiers to predict genres and subgenres and rely on a voting scheme to predict labels across datasets.

1 INTRODUCTION

In the MediaEval AcousticBrainz Genre Task, the goal is to classify tracks into main and subgenres, using content-based features computed with Essentia [2] and collected by AcousticBrainz [11]. Four separate training and test sets of tracks were provided, stemming from four different sources (AllMusic, Discogs, Lastfm, and Tagtraum). The task features two subtasks, which differ in the amount of data that can be used for solving them: In subtask 1, only training data from the same source as the current test data may be used for the classification; in subtask 2, all provided datasets can be utilized for training. However, the evaluation is performed on a per-dataset basis. Further details can be found in [1].

2 CLASSIFICATION AND CHALLENGES

There are multiple factors that make the posed task difficult to solve, particularly the large amount of data to handle and the multilabel nature of the classification problem make the tasks highly challenging. Subtask 2 is further complicated by the fact that genre and subgenre labels are hardly consistent across the four provided training sets, hence providing a heterogeneous set of labels.

In the following, we firstly sketch our approach to mitigate these difficulties. Next, we detail the classification approaches we used throughout subtasks 1 and 2 and lastly, present the obtained results. We make our implementation available for reproducibility and for promoting research in this direction[1].

Reducing the amount of data. To reduce the amount of data and make the task computationally feasible within the limited time frame, we at first skipped detailed features describing low level energy bands of the energy spectrum and verified on a preliminary basis that the respective central moments are sufficient in terms of classification accuracy. This allowed us to reduce the number of features used for training the genre classifiers to 395 (from over 3,000 features originally provided). The full list of features can be found in our GitHub repository[1].

[1]https://github.com/dbis-uibk/MusicGenreClassification

MediaEval 2017 Workshop, Sept. 13-15, 2017, Dublin, Ireland.

Multilabel classification. The fact that any track may feature multiple genres and subgenres complicates the classification problem, since not all classification algorithm inherently support multilabel classification. We solved this problem by applying the one-vs.-the-rest strategy, effectively training a separate binary classifier for every label.

Different genre labels across data sets. As subtask 2 allows to combine all datasets for training, the (vastly) differing genre labels used in the four available training sets posed a challenge. We tackled this problem by computing a direct mapping between the main class labels of all training sets aiming to find equivalent genre labels across all datasets. Therefore, we applied the Levenshtein string distance measure [9] (as previously used for e.g., entity matching [6]) to find all labels with a distance of at most 1. This slightly fuzzy matching approach allows us to neglect minor syntactic differences in the labels (e.g., hip hop vs. hiphop). Preliminary experiments and manual inspection showed that this allows to increase the number of matching labels while still avoiding false positives. We did not match sub-genres, as our experiments showed that those diverged to a far greater extent.

Classification Algorithms. We implemented our solution with two different classification methods[2]: (1) a linear C-support vector machine [12] and (2) an extra-trees classifier [7]. In addition, multilayer neural networks, that are known to work well for this task (c.f. [4, 5, 8]), and extreme gradient boosting [3] showed promising preliminary results, but were deemed infeasible due to the computational resources required to train full-scale models.

2.1 Subtask 1

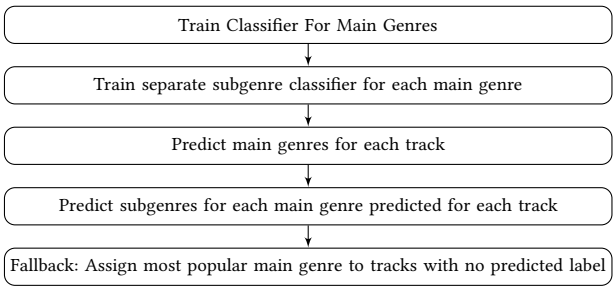

Figure 1: Classification workflow for subtask 1.

The workflow underlying our approach for subtask 1 is outlined in Figure 1. First, we train one classifier for main genres and a separate classifier for each main genre's subgenres. After that, we

[2]We relied on the python library *scikit-learn* [10] for implementing the machine learning parts of the tasks.

utilize the main genre classifier to predict the main genres of every track in the test set. Following that, for every track in the test set and every main genre predicted for that track, the corresponding subgenre classifier is used to predict the subgenre labels for the track. Lastly, as it is possible in multilabel classification that no label is assigned to a track (i.e., if every binary classifier predicts a *'no'* for its respective label), we apply a 'most popular genre' fallback approach and assign the most common main genre label for the respective dataset to ensure that each track is assigned a main genre.

To exploit the possibility to submit five submission runs, this basic approach was implemented with the following configurations of classification algorithms for main and subgenre, which are also listed in Table 1:

- Run #1 uses a SVM with C = 1.0 and no class weight balancing for the main genre classifier; an extra-trees classifier with 50 trees, $\sqrt{|features|}$ features considered when searching for the best split and balanced class weights for the subgenre classifiers.
- Run #2 uses a SVM with C = 1.0 and balanced class weights for the main genre classifier; an extra-trees classifier with 50 trees, $\sqrt{|features|}$ features considered when searching for the best split and balanced class weights for the subgenre classifiers.
- Run #3 includes a SVM with C = 10.0 and balanced class weights for the main- and subgenre classifiers.

The C value for the SVMs was selected after a grid search on a smaller test set of 10,000 randomly sampled tracks. The chosen amount of features and trees for the extra trees classifier was a trade off between classification runtime and accuracy, as more features would possibly have provided more accuracy. For runs #4 and #5, the results of run #3 were used.

2.2 Subtask 2

For subtask 2, the set of all provided datasets could be utilized to classify each of the four test sets. We chose to implement this using a voting mechanism. First, SVMs as main genre classifiers were trained as in subtask 1, independently for every training set. These classifiers were then used to predict the main genres of a given track as follows:

(1) Predict the main genres of the track with all four classifiers.

(2) Utilize the genre mapping as described above to map the predicted genres to the genre labels of the current test set (otherwise, the predicted labels would not be compatible and hence, create false positives). Thereby, classification results where no class label was contained in (or could be mapped to) the test set were discarded.

(3) For every genre predicted by any of the four classifiers, count the number of classifiers that predicted this genre (using two different weighing schemes) and weigh this by the number of classifiers that produced a usable result.

To arrive at the final set of main genres for every track, we applied two different variants, which can be seen in Table 1 for runs #4 and #5: (1) weigh every prediction equally and retain genres predicted by at least 50% of the usable classifiers—for example, if three of the four classifiers predict the label `rock/pop`, that label was predicted by 75% of the classifiers and is retained (run #4); (2) double the weight of the prediction of the classifier trained specifically on the training set corresponding to the current test set and retain genres predicted by at least 60% of the usable classifiers (e.g., if we did predictions for the Lastfm test set and the Lastfm and Discogs classifiers predicted `rock/pop`, then that label was assigned three votes out of five (i.e., 60%) and retained (run #5)). This puts more emphasis on the predictions of the training set and hence, classifier that is trained on the naturally best training data (stemming from the same data source as the current test set).

Prediction of subgenres and handling of tracks with no predicted labels was handled the same way as in subtask 1. For this subtask, support vector machines were used as classifiers, with C = 10.0 and balanced class weights as determined in preliminary experiments.

3 RESULTS AND OUTLOOK

The results of the evaluation of our approach for subtasks 1 and 2 can be found in Tables 2 and 3, respectively. Table 2 shows the results of run #3, which provided the best overall performance in both subtasks. Table 3 contains the results of run #5, which performed better in some measures (in bold font) compared to run #3 in subtask 2. Due to space limitations, the results of the other runs are omitted.

Possible improvements of the presented approaches include different classifying methods such as deep neural networks and a more detailed feature selection process. These steps were rendered impossible due to time constraints and technical limitations of the available hardware.

Table 1: Submitted Runs

Run #	Subtask 1	Subtask 2
1	unbalanced SVM + ET	unbalanced SVM + ET
2	balanced SVM + ET	balanced SVM + ET
3	bal. SVM + bal. SVM	bal. SVM + bal. SVM
4	bal. SVM + bal. SVM	bal. SVM + bal. SVM + voting 50
5	bal. SVM + bal. SVM	bal. SVM + bal. SVM + voting 60

Table 2: F-scores for subtask 1 with run # 3.

Goal	AllMusic	Discogs	Lastfm	Tagtraum
Per Track (all)	0.249	0.374	0.340	0.363
Per Track (genre)	0.587	0.680	0.512	0.478
Per Track (subgenre)	0.193	0.219	0.251	0.303
Per Label (all)	0.070	0.144	0.155	0.153
Per Label (genre)	0.266	0.441	0.313	0.345
Per Label (subgenre)	0.065	0.129	0.139	0.131

Table 3: F-scores for subtask 2 with run # 5. Bold numbers mark better results than in subtask 1.

Goal	AllMusic	Discogs	Lastfm	Tagtraum
Per Track (all)	0.183	**0.426**	**0.366**	**0.401**
Per Track (genre)	0.516	0.668	**0.523**	**0.629**
Per Track (subgenre)	0.065	0.014	0.056	0.166
Per Label (all)	0.019	0.026	0.055	0.059
Per Label (genre)	0.230	0.395	0.309	0.272
Per Label (subgenre)	0.013	0.008	0.030	0.034

REFERENCES

[1] Dmitry Bogdanov, Alastair Porter, Juliàn Urbano, and Hendrik Schreiber. 2017. The MediaEval 2017 AcousticBrainz Genre Task: Content-based Music Genre Recognition from Multiple Sources. In *Proc. of the MediaEval 2017 Workshop, Dublin, Ireland, Sept. 13-15, 2017.*

[2] Dmitry Bogdanov, Nicolas Wack, Emilia Gómez, Sankalp Gulati, Perfecto Herrera, O. Mayor, Gerard Roma, Justin Salamon, J. R. Zapata, and Xavier Serra. 2013. ESSENTIA: an Audio Analysis Library for Music Information Retrieval. In *International Society for Music Information Retrieval Conference (ISMIR'13)*. Curitiba, Brazil, 493–498. http://hdl.handle.net/10230/32252

[3] Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*. ACM, 785–794.

[4] Sander Dieleman, Philemon Brakel, and Benjamin Schrauwen. 2011. Audio-based Music Classification with a Pretrained Convolutional Network.. In *In Proceedings of the 12th International Society for Music Information Retrieval Conference (ISMIR 2011)*. 669–674.

[5] Shyamala Doraisamy, Shahram Golzari, Noris Mohd. Norowi, Md Nasir Sulaiman, and Nur Izura Udzir. 2008. A Study on Feature Selection and Classification Techniques for Automatic Genre Classification of Traditional Malay Music. In *In Proceedings of the 9th International Society for Music Information Retrieval Conference (ISMIR 2008)*. 331–336.

[6] A. K. Elmagarmid, P. G. Ipeirotis, and V. S. Verykios. 2007. Duplicate Record Detection: A Survey. *IEEE Transactions on Knowledge and Data Engineering* 19, 1 (Jan 2007), 1–16. https://doi.org/10.1109/TKDE.2007.250581

[7] Pierre Geurts, Damien Ernst, and Louis Wehenkel. 2006. Extremely randomized trees. *Machine learning* 63, 1 (2006), 3–42.

[8] Arijit Ghosal, Rudrasis Chakraborty, Bibhas Chandra Dhara, and Sanjoy Kumar Saha. 2015. Perceptual feature-based song genre classification using RANSAC. *International Journal of Computational Intelligence Studies* 4, 1 (2015), 31–49.

[9] Vladimir I Levenshtein. 1966. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, Vol. 10. 707–710.

[10] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, and others. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* 12, Oct (2011), 2825–2830.

[11] Alastair Porter, Dmitry Bogdanov, Robert Kaye, Roman Tsukanov, and Xavier Serra. 2015. AcousticBrainz: a community platform for gathering music information obtained from audio. In *16th International Society for Music Information Retrieval Conference (ISMIR 2015)*. Malaga, Spain, 786–792. http://dblp.org/rec/html/conf/ismir/PorterBKTS15

[12] Ting-Fan Wu, Chih-Jen Lin, and Ruby C Weng. 2004. Probability estimates for multi-class classification by pairwise coupling. *Journal of Machine Learning Research* 5, Aug (2004), 975–1005.