Sample Paper in the Format for MediaEval 2017

Kijung Kim[1], Jaeyoung Choi[2]
[1]University of California, Berkeley, United States
[2]Dublin City University, Ireland
kijung@berkeley.edu

ABSTRACT

This paper will present our results for the MediaEval 2017 AcousticBrainz Content-based music genre recognition task. We proposed an approach based two machine learning classifiers: logistic regression and random forest. Experimental results show that the best results come from random forest classifiers with partial feature selection.

1 INTRODUCTION

The goal of this task is to understand how genre classification can explore and address the subjective and culturally-dependent nature of genre categories. Traditionally genre classification is performed using a single source of ground truth with broad genre categories as class labels. In contrast, this task is aimed at exploring how to explore and combine multiple sources of annotations, each of which are more detailed. Each source has a different genre class space, providing an opportunity to analyze the problem of music genre recognition from new perspectives and with the potential of reducing evaluation bias.

2 RELATED WORK

Language modeling is used in placing photos on a map. In particular, Pavel et. al [7] place a grid of fixed degree over the world map and map train instances to cells based on their coordinates.They learn a model which allows them to predict the location of the test instances on the grid. Though this work provides several smoothing techniques to predict the location of a test instance whose tags are not seen, it does not differentiate between general and location specific tags. Giorgos et. al in [4] use a similar model but capture information regarding how many users use a particular tag in a particular region. Additionally, they use Shannon's Entropy to give small weights to tags which are user specific or general. Our base model is the same, as it provides a weighting of each tag based on its popular-ity among users in describing a place, but we experiment with additional components to boost the performance. An- other related weighting scheme is that of Aibek et. al in [6] which uses the Kullback-Leibler divergence to differentiate between class-specific and general terms. Even though that work is done in a different context, we experiment with this model in identifying location specific tags.

3 APPROACH

The proposed framework can be divided into two phases: (1) Feature Selection and (2) Model selection and predictions.

MediaEval'17, 13-15 September 2017, Dublin, Ireland

(1) **Feature Selection** Each song has three groups of features: tonal, rhythm, and low-level. A feature vector for each song was made through a concatenation of all the individual features. For features with specifics labels such as mean, max, and min, these were ignored and concatenated together. For simplicity, categorical features were not considered. These features were: "key_key", "key_scale", "chords_key", and "chords_scale". In addition, "beats_position" feature was also not included because the feature length varied for each song.

(2) **Model Selection and Predictions** The models used were from the python library sci-kit learn. The two classifiers used were the sklearn.linear_model.SGDClassifier with hinge loss and sklearn.ensemble.RandomForestClassifier with 16 n_estimators. The method used for genre classification was to train a binary classifier for each genre/subgenre and then conglomerate the results together.

The first two runs consisted of concatenating every feature (minus ones mentioned above) and using the SGDClassifier.

3.1 Run 1

This run consisted of each song having a concatenated feature vector of all features minus the ones mentioned above with the SGDClassifier. To accommodate for large data, the function "partial_fit" was used.

3.2 Run 2

Run 2's feature selection and model was the same as Run 1. The only difference came from the prediction process. The procedure was to look at the results for each song and mainly go with the genre prediction. For example, given main genre A has subgenres B,C and main genre D has subgenres E, F, if the classifiers classified a song as genre A with subgenres C,D, and F, because main genre D was not predicted, the predictions will ignore subgenre F, and the final prediction will be genre A with subgenres C,D.

3.3 Run 3,4,5

For the next three runs, the procedure differed. Unlike the SGDClassifier, sklearn's RandomForestClassifier did not have the method "partial_fit". Instead, RFC had a method "feature_importances", of which we took advantage. We first took a subset of the train data (around 100k songs) and fit the concatenated feature vector mentioned above to the RFC for each genre and subgenre. Then, we used the "feature_importances" method to select the x% best features. From there, we trained all-for-one RFC's using the top x% features using a subset of the train data (around 150k songs) and then predicted the genres based on a conglomeration of all the RFC's.

Table 1: Subtask 1 Average of Per label results, only genres.

Run #	Precision	Recall	f-score
1	0.0963	0.7399	0.1471
2	0.0963	0.7399	0.1471
3	0.1375	0.3149	0.1638
4	0.1095	0.351	0.1409
5	0.0889	0.3908	0.117

Table 2: Subtask 1 Average of Per track results, only with genres

Run #	Precision	Recall	f-score
1	0.0989	0.791	0.172
2	0.0989	0.791	0.172
3	0.2059	0.4023	0.2475
4	0.1725	0.4595	0.2326
5	0.0999	0.3943	0.1513

3.4 Run 3

This run used the top 25% of the features obtained from feature _importances function.

3.5 Run 4

This run used the top 50% of the features obtained from feature _importances function.

3.6 Run 5

This run used the top 75% of the features obtained from feature _importances function.

4 RESULTS AND ANALYSIS

In this section, we report accumulated results on the sub- task based on our two different approaches. The run approaches are applied on the MediaEval 2017 test set and results are reported in Figure 1. The test set is composed of three different databases (Discogs, Lastfm, Tagtraum), and we took the average of precision, recall, and f-score to get one number.

We observe that the approaches based on the RandomForestClassifiers (run 3,4,5) outperforms the SGDClassifier approaches (runs 1, 2). In particular, we note that the recall in runs 1,2 is especially high while the precision is especially low, which meant the classifiers predicted each song with almost every label. For runs 3,4,5 we note a significantly lower recall with a better precision.

Out of the last three runs, it first seems to be that by adding additional features, the recall improves at the cost of precision, but run5 disproves the trend, as it shows that run5 gives only better recall for per-label results while does worse in all metrics in per-track results.

5 CONCLUSION

The results were disappointing. Runs 1 and 2 clearly suffered from oversampling, which lead the classifiers in most genres to predict positive, which resulted in high recall and low precision. Runs 3,4,5 did not suffer like Runs 1 and 2, but upon observing precision, recall, and f-scores for each genre, the classifiers did far worse on non-popular genres and subgenres, which lead to overall lower precision and recall.

The shortcomings came, for Runs 1 and 2, from errors in sampling. For runs 3,4,5, the shortcomings came from a lack of a system to combine results from different classifiers. For one, we could have exploited the function predict_proba to ascertain a threshold for each genre and subgenre. This would have helped especially for sparse subgenres.

For future works, for runs 1, 2, it would it would be interesting to see if taking less top % of the features obtained from feature_importances function will improve precision. Also, it may be worth trying majority voting by training several different random forest classifiers using the features obtained from the features_importances function.

ACKNOWLEDGMENTS

The authors would like to thank the reviewer for the helpful comments. This work was supported in part by the International Computer Science Institute.

REFERENCES

[1] American Mathematical Society 2015. *Using the amsthm Package*. American Mathematical Society. http://www.ctan.org/pkg/amsthm.
[2] Mic Bowman, Saumya K. Debray, and Larry L. Peterson. 1993. Reasoning About Naming Systems. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 795–825. https://doi.org/10.1145/161468.161471
[3] Johannes Braams. 1991. Babel, a Multilingual Style-Option System for Use with LaTeX's Standard Document Styles. *TUGboat* 12, 2 (June 1991), 291–301.
[4] Malcolm Clark. 1991. Post Congress Tristesse. In *TeX90 Conference Proceedings*. TeX Users Group, 84–89.
[5] Simon Fear. 2005. *Publication quality tables in LaTeX*. http://www.ctan.org/pkg/booktabs.
[6] Maurice Herlihy. 1993. A Methodology for Implementing Highly Concurrent Data Objects. *ACM Trans. Program. Lang. Syst.* 15, 5 (November 1993), 745–770. https://doi.org/10.1145/161468.161469
[7] Leslie Lamport. 1986. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley Publishing Company, Reading, Massachusetts.
[8] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[9] S.L. Salas and Einar Hille. 1978. *Calculus: One and Several Variable.* John Wiley and Sons, New York.