Ensemble-based stacking model on music genre classification

Sirong Huang, Aalto University

September 10, 2018

Abstract

Ensemble methods combine several models into one aggregated model in attempt to achieve better performance. In this study, we apply ensemble-based stacking method with Logistic regression, Support Vector Machine(SVM), Naive Bayes classifier, Random Forest and K-Nearest Neighbor on over 4000 songs to build an aggregated meta-classifier model to classify music into 10 different genres. The result shows promising performance increase. Further improvements can be made in terms of dimensionality reduction methods, finer hyperparameter tuning on individual model training, increased diversity in base model and new ensemble techniques.

I. Introduction

Music information analysis is an increasingly important task as music industry digitizes. Automatic music genre recognition is of great value for digital music consumers and businesses alike.

Music genre classification is a multi-class classification problem. Each song is represented by a set of features extracted with certain mathematically formulation to describe musical characteristics such as tonality, pitch and rhythm. Researchers have proposed a variety of features such as Mel-frequency Cepstral Coeffi-cients (MFCCs), Wavelet Transform (WT) and Daubechies Wavelet Coefficient Histograms(DWCH) [1], and have successfully implemented various machine learning algorithms such as K-Nearest Neighbor(KNN), Support Vector Machine(SVM), and neural networks to classify music genre. A similar study done by Tzanetakis and Cook [2] achieved 61% accuracy on 10 genre music classification.

Ensemble methods are meta-algorithms that combine several machine learning algorithms into one aggregated model in order to decrease variance and/or bias. It has been shown by many researches to achieve better performance than single classifiers. [3,4].

In this study, ensemble-based learning are implemented to combine a set of diverse classifiers to achieve better prediction accuracy than any single classifier. First, features are preprocessed to reduce the dimensionality and standardized. Then 5 different classifiers are trained and their prediction output is aggregated as new input for the meta-classifier.

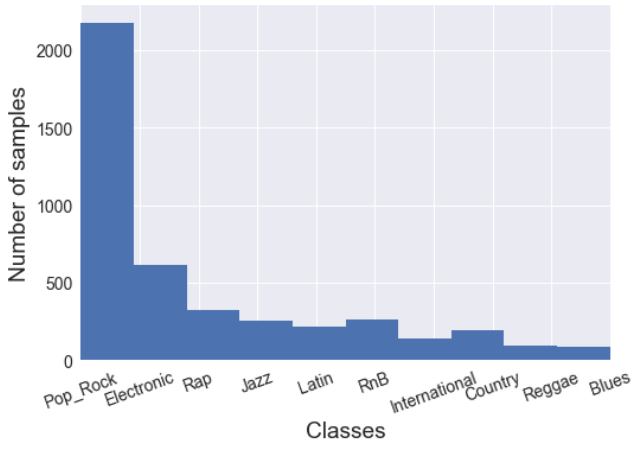
The paper is organized in the following manner: first, general observations of dataset characteristics are made; followed by detailed experiment methods and process; then the experiment results are presented; and finally, the paper ends with a discussion of whole experiment with limitations and future improvements.

II. Data analysis

The data used in this study is from the Million Song Dataset which contains 4363 and 6544 labeled songs for training and testing respectively. Each song is labeled as one of the 10 music genres.

There are considerable class imbalance as shown in Figure 1. Poprock is the most represented class that takes up almost 50 of training data. The large population of Poprock genre songs will cause the classifiers to biased towards it, resulting large misclassification rate for minority classes. Total accuracy doesn't not imply class-specific accuracy [5]. A simple guess of

Figure 1: Class distribution of music genre training data



every single song belonging to Poprock will get about 50% accuracy score while having 100% error rate for all 9 other classes.

The features are summary statistics of MFCCs, chroma and rythmic patterns representing the timbre, pitch and rhythm of the music. There are a total of 264 features, which implies a need for dimensionality reduction. The features take on a range of values from -494 to 18698. This could cause problems to algorithms that use gradient descent to optimize such as SVM and Logistic Regression, and algorithms which are sensitive to feature scale such as Principle Component Analysis(PCA) and KNN. Therefore all features are standardized to zero mean and unit variance.

As can be seen in Figure 2, some classes are distinctly different whereas others have very similar features, which poses great challenge in distinguishing between those similar classes. Also note that the music genre are labeled manually by human listeners, who are prone to subjective judgment since the line between different music genres are often not well defined.

III. Methods and experiments

i. Dimensionality reduction

The first step of data analysis project is usually feature dimensionality reduction. High dimensional features leads to sparsity of data in high dimensional space,

Figure 2: Linear Discriminant Analysis 2-components

which can greatly impair the performance for certain algorithms such as K-Nearest Means. [6] By reducing the number of variables, the noise in data is also reduced, leading to improved prediction performance for most algorithms.

There are many practical dimensionality reduction techniques, from the simplest Variance Threshold, to Decision Tree feature importance score, PCA, Linear Discriminant Analysis(LDA) and autoencoder etc. Most of them are unsupervised method, except for LDA which is a supervised method that maximizes the separability between classes given labels [7].

Equation (1) shows the mathematical formula for LDA, which assumes each class a Gaussian distribution with same covariance matrix Σ. It then projects the data into a lower dimensional linear subspace that maximizes the separability between classes [8]:

$$X^* = D^{-1/2}U^t X with \Sigma = UDU^t \tag{1}$$

In this study, Linear Discriminant Analysis is used to reduce the feature dimensionality due to its suitability for supervised multiclass classification task. The final feature set contains **9** features reduced from 264 features. Figure 2 is an example of LDA reducing all features to 2 compressed features.

ii. Ensemble model

A. **Introduction**

There are three common subcategories of ensemble

methods: bagging, boosting and stacking.

Bagging stands for bootstrap aggregation, which is a method that aggregates multiple versions of an algorithm with bootstraps of randomly drawn with replacement samples from the data and use majority vote (for classification) to get an aggregated predictor [9]. Bagging has proven to work well for high variation algorithms such as decision tree.

Boosting is very similar to bagging, but differs in its iterative model building process. The training sample selected is based on previously incorrectly labeled data. This way, the model builds new classifiers that improve the weakness of previous classifier [10].

Unlike bagging and boosting, stacking is a meta-learner that combines different algorithms [11]. The motivation behind stacking models is that algorithms that perform well in classifying a subset of the data might have lower overall accuracy, and therefore not being used as final model. Combining those classifiers together can exploit the prediction capability of various algorithms to improve the final prediction accuracy.

The choice of the base classifiers is crucial for the success of ensemble-based learning. For stacking model, algorithms with fundamentally different mechanism and are highly uncorrelated are preferred as base classifiers to include as much diverse information as possible in the meta-learner [12].

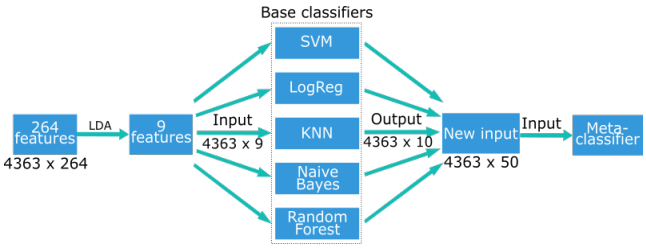
In this study, we focus on using stacking ensemble method to build a meta-classifier that combines 5 distinct classifiers on music genre classification task.

B. **Choice of classifiers**

As mentioned in previous section, the choice of base classifiers is of crucial importance in stacking model building. 5 highly uncorrelated and dissimilar classification algorithms are chosen as base classifiers for the experiment: SVM, Logistic regression, Naive Bayes classifier, KNN and Random Forest.

Support Vector Machine is a geometry inspired classification algorithm that finds optimal decision boundary by maximizing the margin between classes [13]. Logistic regression is a binary linear regressor that uses

Figure 3: Experiment work-flow



logic function to model real number input into probability output [14]. For binary classifiers, one-versus-all method is used to classify multiclass music genre labels. Naive Bayes is a generative model that applies Bayes' rule with the 'naive' assumption of independence between every pair of features [15]. Random Forest is based on many decision trees, which is a rule-based classifier [16]. K-Nearest Neighbors is distance based algorithms that produce the prediction based on the K nearest data points [17].

Each base classifier has unique mechanism, and therefore the resulting combinations should contain diverse information and prediction capabilities. Meta-classifier is chosen from one of the five base classifiers.

C. **Experiment design**

The experiment is designed around building a meta-classifier with probability outputs of 5 base classifiers with the best performing meta-classifier configuration and then compare the ensemble-based meta-classifier with the best performing single classifier.

The stacking process is illustrated in Figure 3. To elaborate the procedures in more detail:

- Firstly, 5 base classifiers are trained using 5-fold cross validation and grid search on the whole training data to select the best hyperparameters.
- Secondly, each base model with optimal configuration is trained again on each iteration's train set of 5-fold cross validation and its probability output on validation set is stacked row-wise into a new feature vector of dimension 4363x10. Repeat this for each base model and then stack the feature vectors column-wise to get a feature matrix of dimension 4363x50.
- Finally, use the new feature matrix that represents the probability prediction output of each base clas-

Figure 4: Single classifier performance: accuracy

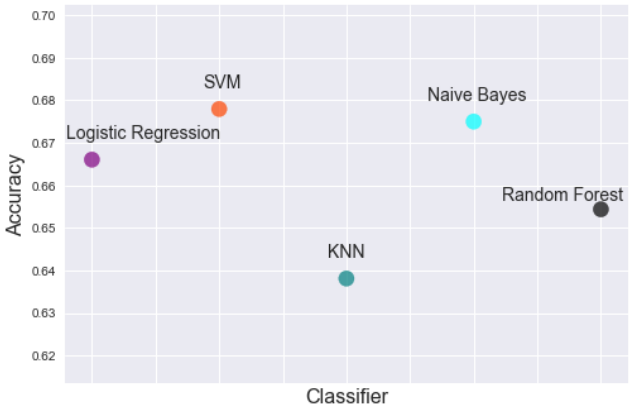
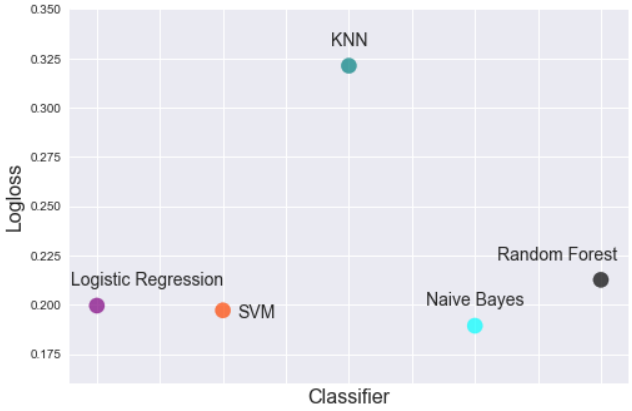


Figure 5: Single classifier performance: logloss



sifier as input and the corresponding true labels as output to train a meta-classifier. Optimize its hyperparameters using 5-fold cross validation and grid search on the new feature data.

D. **Performance evaluation method**

Both base classifiers and meta-classifier use 5-fold cross validation and grid search method to tune hyperparameters. This method avoids overfitting caused by selecting models on train error and also does not waste any data as holdout validation method does.Both accuracy and logloss is implemented as performance measurement metrics in both model training and evaluation.

Table 1: Ensemble meta-classifier vs single classifier performance

Model	Accuracy	Logloss
LogReg	0.6660	0.1996
SVM	0.6779	0.1972
KNN	0.6381	0.3213
NB	0.6749	0.1894
RF	0.6555	0.2127
Meta-classifier (LogReg)	0.6827	0.1696
Meta-classifier (SVM)	**0.1654**	**0.1672**
Meta-classifier (KNN)	0.6623	0.2213
Meta-classifier (NB)	0.6861	0.1684
Meta-classifier (RF)	0.6571	0.1727

IV. Results

For single classifiers, SVM performs the best on classification accuracy with 0.6779 cross-validation accuracy. Naive Bayes classifier performs the best on logloss althought most classifiers are rather close except KNN. Therefore SVM and Naive Bayes classifier will be used as the benchmarks for ensemble meta-classifier vs single classifier performance comparison.

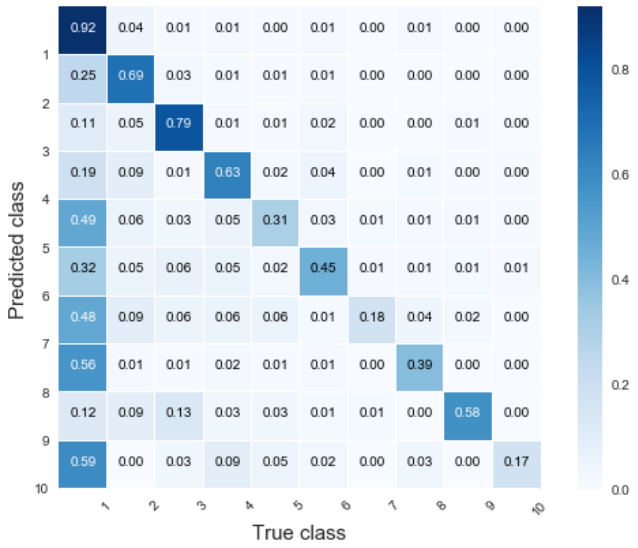
Table 1 shows the accuracy and logloss performance of single classifiers and ensemble meta-classifiers with with different choices of meta-classifier with 5-fold cross validation on the stacked features.

As experiment result shows, ensemble meta-classifier with SVM as aggregate classifier yielded the best cross-validation accuracy of 0.6871 by a small margin. Meta-classifier with SVM also performed the best on logloss with a score of 0.1672.

In Kaggle competition, the score of best models for each performance metrics are: **0.6515** accuracy and **0.1806** logloss. The scores are slightly worse than the cross validation accuracy estimation. The possible explanations will be explored in the Discussion section.

Figure 6 shows the confusion matrix of the best performing meta-classifier with SVM. As can be seen, most music genres can be predicted with above 62% accuracy. Most of the genres have relatively high possibility to be misclassified as Poprock genre, as Poprock is the most dominant class.

Figure 6: Confusion matrix of meta-classifier



V. Discussion

The purpose of this study is to ensemble different classification algorithms to build a meta-classifier with improved generalization capability. The best ensemble model configuration uses SVM as meta-classifier to combine the probability output of 5 distinct classification algorithms. Both performance metrics on Kaggle competition have shown that ensemble-based stacking model do outperform individual classifiers on multiclass classification task. This result corresponds to the literature research results which have shown performance increase using ensemble-based model as opposed to single classifier models.

As shown in Figure 6, the imbalance in class distribution resulted in larger misclassification rate for minority classes. For this reason, accuracy score can be incomplete measurement and misleading since it does not take into consideration false positive and false negative error rates on other classes. Logloss, on the other hand, punishes wrong predictions based on the model's confidence in its predictions. Wrong prediction with higher confidence gets punishes more. This way, we can more objectively measure the model performance on imbalanced data.

There are a number of limitations in this study. First of all, LDA as dimensionality reduction tends to cause inflated cross-validation accuracy score due to information leak. Since LDA is a supervised algorithm that uses the true label, which is used again in the model building process. However, Kaggle score indicates no significant different between the performance of LDA and PCA. More dimensionality reduction method can be investigated to improve single classifier's performance, and therefore improve the whole model. Moreover, more base classifiers can be included to improve the ensemble model capabilities, as well as experimenting other ensemble methods such as bagging and boosting.

References

[1] Tao Li, Mitsunori Ogihara, and Qi Li. A comparative study on content-based music genre classification. pages 282–289, 01 2003.

[2] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. ieee trans speech audio process. 10:293 – 302, 08 2002.

[3] Ludmila I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[4] Zhi-Hua Zhou. *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[5] Aida Ali, Siti Mariyam Hj. Shamsuddin, and Anca L. Ralescu. Classification with class imbalance problem: A review. 2015.

[6] Laurens Van Der Maaten, Eric Postma, and Jaap Van den Herik. Dimensionality reduction: a comparative. *J Mach Learn Res*, 10:66–71, 2009.

[7] Masashi Sugiyama. Local fisher discriminant analysis for supervised dimensionality reduction. In *Proceedings of the 23rd international conference on Machine learning*, pages 905–912. ACM, 2006.

[8] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. *The elements of statistical learning*, volume 1. Springer series in statistics New York, 2001.

[9] Leo Breiman. Bagging predictors. *Machine learning*, 24(2):123–140, 1996.

[10] Yoav Freund, Robert Schapire, and Naoki Abe. A short introduction to boosting. *Journal-Japanese*

Society For Artificial Intelligence, 14(771-780):1612, 1999.

[11] David W Fan, Philip K Chan, and Salvatore J Stolfo. A comparative evaluation of combiner and stacked generalization. In *Proceedings of AAAI-96 workshop on Integrating Multiple Learned Models*, pages 40–46. AAAI Press, 1996.

[12] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[13] Johan AK Suykens and Joos Vandewalle. Least squares support vector machine classifiers. *Neural processing letters*, 9(3):293–300, 1999.

[14] David W Hosmer Jr, Stanley Lemeshow, and Rodney X Sturdivant. *Applied logistic regression*, volume 398. John Wiley & Sons, 2013.

[15] David D Lewis. Naive (bayes) at forty: The independence assumption in information retrieval. In *European conference on machine learning*, pages 4–15. Springer, 1998.

[16] Andy Liaw, Matthew Wiener, et al. Classification and regression by randomforest. *R news*, 2(3):18–22, 2002.

[17] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2006.

Appendices

Figure 7: Source code for key functions

```
# base classifiers
clfs = [LogisticRegression(C=0.06),
        SVC(kernel='rbf',C=4,decision_function_shape='ovr',
            probability=True),
        KNeighborsClassifier(n_neighbors=6,weights='distance'),
        GaussianNB(),
        RandomForestClassifier(n_estimators=100, criterion='entropy')]

# stack outputs of base classifiers with 5-fold cross validation
single_model_prediction = np.array([]).reshape(0, len(set(y)))
all_model_predictions = np.array([]).reshape(len(y), 0)

for j, clf in enumerate(clfs):
    for i,(train, validation) in enumerate(skf):
        X_train,X_val = X[train],X[validation]
        y_train,y_val = y[train],y[validation]
        clf.fit(X_train, y_train)
        prediction = clf.predict_proba(X_val)
        single_model_prediction = np.vstack([single_model_prediction,
                                             prediction])
        print(i,single_model_prediction.shape)

    all_model_predictions = np.hstack([all_model_predictions,
                                       single_model_prediction])
    single_model_prediction = np.array([]).reshape(0, len(set(y)))

reordered_labels = np.array([]).astype(y.dtype)
reordered_features = np.array([]).reshape((0, X.shape[1]))\
     .astype(X.dtype)
for train_index, test_index in skf:
     reordered_labels = np.concatenate((reordered_labels,
                                        y[test_index]))
     reordered_features = np.concatenate((reordered_features,
                                          X[test_index]))
# train meta-classifier with new features
meta_clf = LogisticRegression()
meta_clf.fit(np.hstack((reordered_features,all_model_predictions)),
                        reordered_labels)
```