Guitar Playing Techniques Classification with Different Machine Learning Approaches

Yudi Wang, He Peng, Yudi Zhu, Jishi Lyu, Weixin Li, Jialun Lyu
University of California, San Diego

Abstract

The use of various instrumental techniques is essential in guitar performances. Guitar playing techniques classification is beneficial to novice guitar players in helping them learn guitar techniques correctly on their own. Although many traditional signal processing methods have been adopted for this problem, the application of machine learning methods on this problem are limited comparing with the new algorithm constantly emerging in recent years. In this paper, we consider 7 different playing techniques of electric guitar and compare the classification performance of different machine learning approaches, including kNN, AdaBoost, SVM and Neural Network. We first adapt MFCCs and CSSFR these two feature extraction methods to pre-process data set; then use different machine learning approaches for classification; finally analyze the PoE and confusion matrix of each methods and do comparison. In our evaluation, SVM achieves best classification performance with 75% accuracy, while other methods achieve an accuracy around 70%.

Index Terms - *kNN, AdaBoost, SVM, Neural Network, guitar playing techniques, feature extraction, machine learning, classification*

1. Introduction

1.1. Motivation

Guitar is one of the most widely used instruments in the world. For guitar playing, there are various fingering styles and playing techniques such as pull-off, hammer-on or bending. [1, 14, 15] Different techniques could achieve entirely different performances. For novice guitar players, complicated playing techniques should be the first obstacle that they need to face with. For this case, there already exists some online auto-grader like Chordify can give the learner some useful instruction. Similarly, a tool recording guitar playing note by note and recognizing the matching degree of different playing techniques can enhance the interactivity of guitar learning experience. [2] This tool should also has important educational, recreational and cultural values.

On the other hand, extracting and analyzing the onset, chord, pitch, and instrument information from a clip of music has got so much attention for researchers in the related area.[2] However, limited effort has been spent in the playing techniques of the instrument. Especially for the classification of guitar playing techniques, there is still a lot of research value and space. Researchers tend to view this problem as a typical audio signal process problem. However, it has been proved that machine learning algorithm can be another efficient tools to solve this, according to many other similar category classification problems.

Therefore, motivated by the above situation, in this paper we present a comparative study of adopting different machine learning algorithm to analyze the classification performance, and try to do some optimization on the basic algorithm, which is one of the most crucial technique cores we should focus on.[3]

1.2. Related Work

Music feature extraction and classification has been a long-studied topic and has achieved high performance in some fundamental problems such as instrument classification of monophonic signals. [8] There are many features in an audio signal such as Mel-frequency cepstral coefficients (MFCCs) which are often employed in audio signal processing. [11, 6] Based on audio features, much work done nowadays. For instance, Zhu *et al*. used audio features to analyze scene segmentation and classification. [10] Moreover, Abeer *et al.* and Reboursi'ere *et al.* [13] pioneered the problem of automatic guitar playing technique classification, and used timber descriptors such as spectral flux, weighted phase. However, these studies were not evaluated using a dataset comprising of various playing techniques and guitar tones.

In addition to some researches on music feature extraction, novel machine learning algorithms might help play technique classification. Recently, Namunu *et al.*, pioneered using stacked generalization SVM outputs to improve automatic music tag annotation.[12] Besides using

Technique	Description
Normal	Normal sound
Muting	Sounds muted (by right hand) to create great attenuation
Vibrato	Trilled sound produced by twisting left hand finger on the string
Pull-off	Sound similar to normal but with the smoother attack created by pulling off the string by left hand finger
Hammer-on	Sound similar to normal but with the smoother attack created by hammering on the string by left hand finger
Sliding	Discrete change to the target note with a smooth attack by left hand finger sliding through the string
Bending	Continuous change to the target note without an apparent attack by bending the string by left hand fingers

Table 1. Results.Description of the playing techniques.

SVM in music, there is a stable k-NN regression algorithm which can estimate the music tempo[7] and this method also can apply on music emotion classification.[16] Moreover, Adaboost also could be a good ML method for music classification: Douglas *et al.* proposed such method to classify different music tags and make a recommendation for people,[5] which is one of most vital application for the music industry. Nowadays, some music researching expert like Keunwoo *et al.* fully utilized convolutional neural networks to make music classification.[4] In conclusion, for now, there are so many machine learning applications towards music classification or distinction. Consequently, our work will do more on playing techniques classification by using different kinds of machine learning methods and find the most suitable one.

1.3. Playing Techniques

Table [1] lists the seven playing techniques we consider in this work. Most guitar solos are constructed with these techniques. Among these techniques, muting is widely used alternatively in place of normal in guitar riffs for rhythmic and punched phrases in rock and metal music; bending is commonly considered to be the most crucial technique for expressing emotion; Vibrato can highlight important notes in musical form; pull-off and hammer-on can often be used in the solo part of the song. And the quality of the sliding string can directly affect the correctness of the chord. [1, 13]

2. Methodology

2.1. Feature extraction

Two feature extraction methods have been implemented in our research before the classification, MFCCs and centralized spectrogram with selected frequency ranges.

By comparing the spectrogram of audios of different guitar playing techniques, it is apparent that most differences exist around the onsite point, which is the exact time certain playing techniques. The audio files we used have already been adjusted so that the onsite points for each audio clips have been moved to the same time, so that the classification performance won't be influenced by inconsistent onsite points of different audio files.

Method One: MFCCs (Mel-frequency cepstral coefficients). MFCCs are coefficients that collectively make up an MFC, which is the most popular method to extract the spectral based parameter from the audio.[9] During the experiment, we adopt MFCCs to achieve the dimension reductions. The larger MFCCs numbers is, the more details of the original audio will be kept. Since the number of MFCCs may effect the accuracy of different machine learning methods in the later process. We try different numbers n = 20, 40, 60 and after comparison, choose n = 60 for the later use. The visual result for the seven techniques are shown as Figure [1].

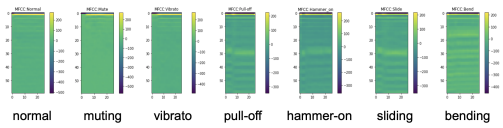

Figure 1. MFCCs (Mel-frequency cepstral coefficients) for 7 different guitar playing techiniques.

Method Two: CSSFR(Centralized spectrogram with selected frequency ranges). It can be noticed in Figure [2] that the spectrograms are much stronger from fundamental frequency to third harmonic frequency. Frequencies above the third harmonic frequency are not obvious, and frequencies below the fundamental frequency can be some noise. Since what we only care about is how the relative frequencies change by time, but not the exact value of the frequencies, so the fundamental frequency of spectrograms of each audio are centralized to roughly the same position and are cut off so that the features used for training can be reduced from 18441 (spectrogram of full range) to 495 (spectrogram of frequencies ranging from $0.5*$fundamental frequency to $3.5*$fundamental frequency).

2.2. Classification

The method we used for classification are four popular machine learning approaches: k Nearest Neighbor, Ad-

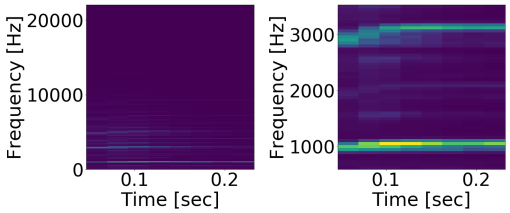

Figure 2. Full range spectrogram (left) and centralized spectrogram ranging from fundamental frequency to third harmonic frequency.

aboost, linear SVM and Neural Network.

2.3. k-Nearest Neighbor Algorithm (kNN)

For each test sample, kNN finds the closest k training samples in the training set, and assigns the predict label which is most frequent among the k closest training samples. The k-nearest neighbour classifier can be viewed as assigning the k nearest neighbors a weight $1/k$ and all others 0 weight.

2.4. AdaBoost

Boosting is a general machine learning frame which tries to maximize the margin between positive and negative examples. By combining simple weak learners during hundreds of iterations, it can finally achieve a complex non-linear classification. We adopt a set of decision stump as the weak leaner and use the exponential loss to encourage large positive margin during the iteration. The weak learner set is U

$$U = \{u(x; j, t)|j \in 1, \cdots, d\}, t \in T$$

$$t = \tfrac{i}{N}, i \in \{0, \cdots, N\} \text{ and } N = 50$$

After the feature extraction, each sample is a 60-dimensional vectors. In order to apply the AdaBoost, first we do the normalization to map the value of each element into the [0,1] range. The iteration number is 100 in this case.

2.5. SVM

The definition of w and b is like perceptron. For support vectors, the inequality can be forced to be an equality

$$| < x^i, w^i > +b| = 1$$

Then, the distance between any support vector and the hyper plane is:

$$|r| = \frac{| < x^i, w^i > +b|}{\|w\|} = \frac{1}{\|w\|}$$

The margin is (by denition)

$$\rho = |r| = \frac{1}{\|w\|}$$

Maximizing the margin is then equivalent to minimizing $\|w\|^2$, providing the hyper plane (w,b) separates the data. Therefore, the problem can be recast as

$$min \frac{1}{2}\|w\|^2 \quad subject \quad to \quad d^i(< x^i, w > +b) \geq 1 \quad for \quad all \quad i.$$

For a fixed α, the minimum is achieved by simultaneously canceling the gradients with respect to w and b, then the problem becomes a dual problem:

$$\max_{\alpha} L_D(\alpha) = \sum_i \alpha^i - \frac{1}{2}\sum_i \sum_j \alpha^i \alpha^j d^i d^j < x^i, x^j >$$

subject to $\alpha^i \geq 0$ and $\sum_i \alpha^i d^i = 0$, for all i.
Each non-zero α^i indicates that corresponding x^i is a support vector.

The decision function could be reformulated as :

$$y = \sum_{i(s.v.)} \alpha^i d^i < x, x^i > +b$$

which only depends on the dot products $< x, x^i >$ between x and the support vectors.

If the case is not separable,just relax the constraints by permitting errors to some extent:

$$\min_{w,b,\epsilon} \frac{1}{2}\|w\|^2 + C\sum_i \epsilon_i$$

subject to $d^i(< x^i, w > +b) \geq 1\epsilon_i$ and $\epsilon_i \geq 0$, for all i.
Quantities ϵ_i are called slack variables.

2.6. Neural Network

We used 3 kinds of activation functions (sigmoid, ReLU and tanh) into our network to compare their result. In order to do the change and connection easily, we set up them into a class first.

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$\sigma(z) = \tanh(z)$$

$$\sigma(z) = \max(0, z)$$

Similar to the activation functions, we set up the linear layers of the neural network. In the forward pass, we just need use matrix multiplication of the weights with inputs and addition of biases. In the backward pass, with the gradient of

the loss that calculated in the output layer, we can calculate all the gradient in each layer. The gradient will help us to update weights and bias in each layer.

$$a_j = w_{ij} z_i + b$$

$$z_j = g(a_j)$$

$$w_{ij} = w_{ij} - \frac{\partial J}{\partial w_{ij}}$$

$$\frac{\partial J}{\partial w_{ij}} = \frac{\partial J}{\partial a_j} \frac{\partial a_j}{\partial w_{ij}} = -\delta_j z_i$$

With the lower level abstractions of layers and activation functions, next step is just use neural network class to connect them. Later, construct the structure we want and do the forward and backward pass in this class.

3. Experiment

After feature extraction, the whole dataset is randomly shuffled and splits into two parts: 80% for training and 20% for testing.

3.1. k-Nearest Neighbor (kNN)

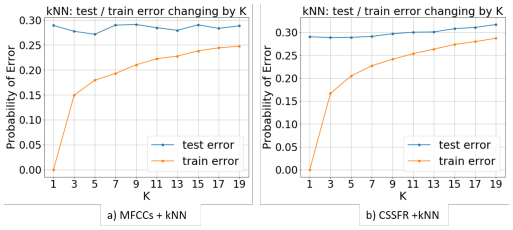

Figure 3. Test and train error changing by k using kNN with MFCCs and CSSFR respectively, k = 1, 3, 5, .. 19.

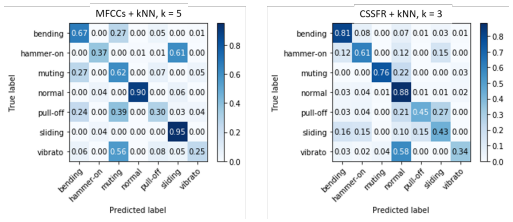

Figure 4. Confusion matrices of kNN with MFCCs and CSSFR with k=3 and k=5 respectively (both are of their lowest overall test error).

First use two feature extraction methods mentioned above, then use k-Nearest Neighbor for classification. The curve of test errors changing by k and confusion matrices at most suitable k are shown in Figure [3], [4].

According to figure [3], when k = 5, MFCCs + kNN got the lowest overall test error of 28%, while when k = 3, CSSFR + kNN got the lowest overall test error of 29%. So the overall test error of kNN with both feature extraction methods are very close, and MFCCs performs a little better than CSSFR.

According to the confusion matrices (Figure [4]) using kNN for classification with MFCCs and CSSFR, it can be noticed that MFCCs is better at classifying normal and sliding, while CSSFR is better at classifying bending, hammer-on and muting. So if we're only trying to classify between normal and sliding, MFCCs feature extraction method could be a good choice, while if we're focusing on classifying bending, hammer-on and muting, CSSFR actually performs better. The classification performance could be improved if these two feature extraction methods are combined to select better representative features for guitar playing techniques classification.

3.2. AdaBoost

First, we use the MFCCs feature extraction method, then apply the AdaBoost. The error rate of training set keeps going down as the iteration times increase as in shown in the Figure[5].

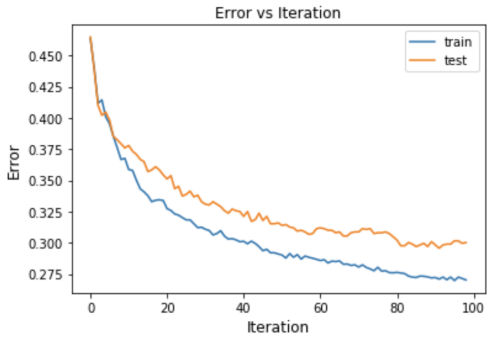

Figure 5. Train/Test PoE v.s.Iteration times for the AdaBoost method

However, the test error keeps quite stable after the 100 iterations and the final test PoE is 30.03%. When we apply the CSSFR (centralized spectrogram with selected frequency range) for the feature extraction, the final result and the tendency has no big difference. Thus here only show the result with the MFCCs feature extraction method.

From the confusion matrix in Figure[6], the easiest classified technique is sliding, which can reach at 90% accuracy; while the classification result for hammer-on, pull-off and vibrato are not so good, especially for the hammer-on and sliding pairs. The error rate for misclassifying the hammer-on as sliding are as high as 78%, which should be the main reason why the overall test error rate cannot goes down to 30%.

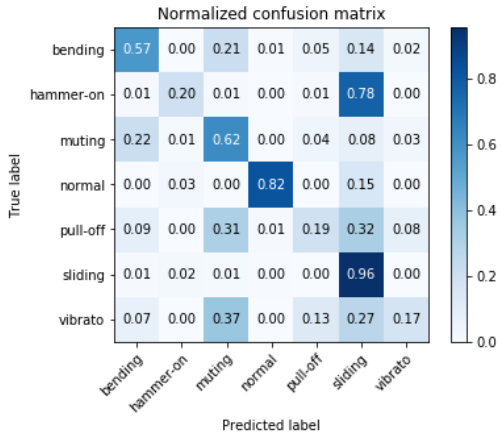

Figure 6. Confusion Matrix for the AdaBoost Method

3.3. SVM

We set the parameter C to 2 and use the linear SVM to do the classification, our experiment result is following:

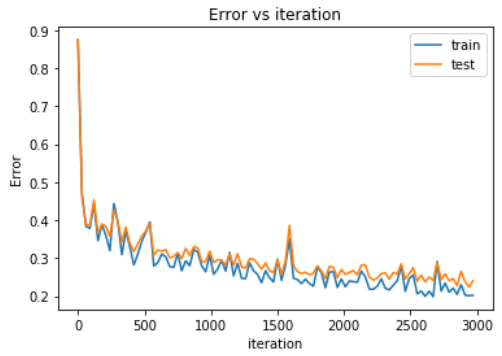

Figure 7. Train error vs. iteration for SVM (C=2,linear, feature extraction = 'MFCCs').

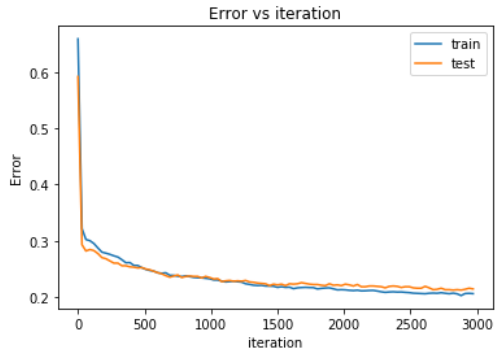

Figure 8. Train error vs. iteration for SVM (C=2,linear, feature extraction = 'CSSFR').

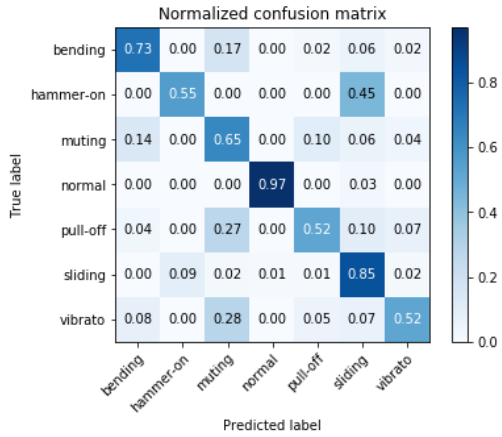

Figure 9. Confusion Matrix for SVM (C=2,linear, feature extraction = 'MFCCs').

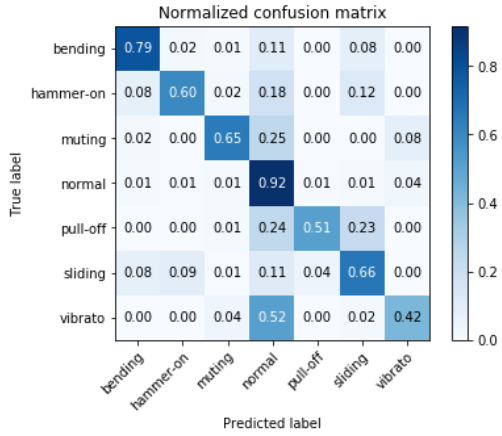

Figure 10. Confusion Matrix for SVM (C=2,linear, feature extraction = 'CSSFR').

The PoE with linear SVM is about 25.71% on the test set. From Figure [7],we can see that the error is noisy at both beginning and end of the iteration. This phenomenon leads to the fact that even between two neighbor iteration, the model which undergoes a small update could lead to a huge difference in final result. So, the error finally would hardly converge to a constant value which means that the classification is very sensitive to the updates.

It could be explained by that the difference between some classes is very small. Then the separate hyper plane is very close to the data point (which means small margin) and a small perturbation to the hyper plane would let many points cross the hyper plane. On the other hand, using this signal processing could not present all the features of them. In this case, some selected features, to some extent, could not seperate them well.

For example, in Figure [9] the Hammer-on technique has

a probability of 45% to be misclassified as a Sliding one, however for a Sliding playing technique, we achieve huge confidence in correctly classifying it. It means in the feature dimension, the Hammer-on and Sliding data points are mixed with each other. In conclusion, we may try some other feature extraction methods to lead to a better performance.

When we try the second feature extraction method (Figure [8]), it is shown that CSSFR achieves a better performance and the noise during the training is smaller. According to Figure [10], though normal playing technique achieves a high accuracy, it has a strong negative impact on classifying other playing techniques, showing that the influence of feature extraction method is huge. And a better feature extraction method needs to be explored.

3.4. Neural Network

Set the learning rate as 0.0001, batch size as 100, and train with three kinds of activation functions for 250 epoch. The structure of the network is 520, 100, 7. The results are shown in Figure [11], [12],[13].

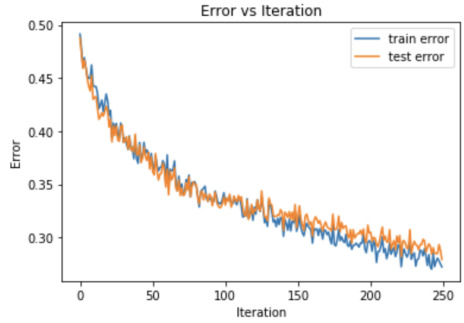

Figure 11. Loss plot with ReLU.

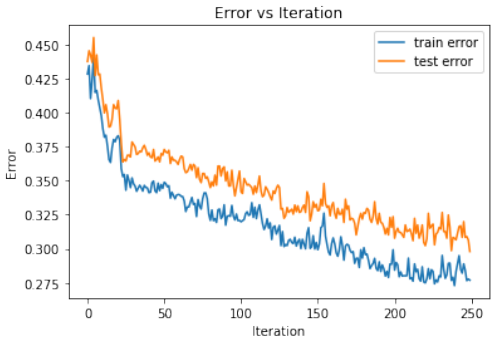

Figure 12. Loss plot with Sigmoid.

By contrast, the ReLU has the best performance among these three. The noise during the training is smaller, and its final test train loss reached about 27%. The confusion matrix for ReLu is shown in Figure [14].

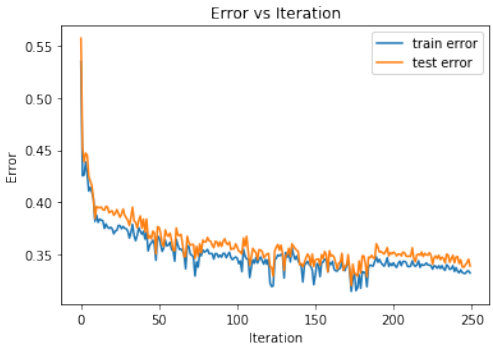

Figure 13. Loss plot with tanh.

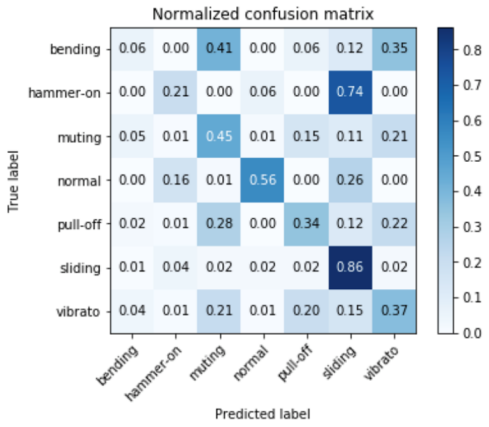

Figure 14. Confusion matrix with Neural Network.

Again the easiest classified playing technique is sliding. However, it is also hard to classify between sliding and hammer-on. The reason may be the unbalanced data set, which contain too many sliding samples. Add more training data of hammer-on playing technique could be a way to improve the accuracy.

4. Conclusion

In this paper we classified seven guitar playing techniques based on two feature extraction methods and four machine learning approaches, then compared their classification performances. The results show that among these four machine learning approaches, SVM can achieve the best classification result of 75% accuracy. The results also shown that different feature extraction methods can have significant influences on the classification results. In future research, our research can further focus on the more comprehensive raw signal pre-processing to extract better representative features. We should also work to build a

better classifier model specifically for the guitar playing techniques classification problem. Besides, deep learning method is another tools worth to apply in the later research.

References

[1] A. M. Barbancho, A. Klapuri, L. J. Tardón, and I. Barbancho. Automatic transcription of guitar chords and fingering from audio. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(3):915–921, 2012. 1, 2

[2] E. Benetos, S. Dixon, D. Giannoulis, H. Kirchhoff, and A. Klapuri. Automatic music transcription: challenges and future directions. *Journal of Intelligent Information Systems*, 41(3):407–434, 2013. 1

[3] Y. Chathuranga and K. Jayaratne. Automatic music genre classification of audio signals with machine learning approaches. *GSTF Journal on Computing (JoC)*, 3(2), 2018. 1

[4] K. Choi, G. Fazekas, M. Sandler, and K. Cho. Convolutional recurrent neural networks for music classification. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2392–2396. IEEE, 2017. 2

[5] D. Eck, P. Lamere, T. Bertin-Mahieux, and S. Green. Automatic generation of social tags for music recommendation. In *Advances in neural information processing systems*, pages 385–392, 2008. 2

[6] A. Eronen and A. Klapuri. Musical instrument recognition using cepstral coefficients and temporal features. In *2000 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (Cat. No. 00CH37100)*, volume 2, pages II753–II756. IEEE, 2000. 1

[7] A. J. Eronen and A. P. Klapuri. Music tempo estimation with k-nn regression. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(1):50–57, 2010. 2

[8] A. Klapuri and M. Davy. *Signal processing methods for music transcription*. Springer Science & Business Media, 2007. 1

[9] S. Z. Li, G. Guo, and H. Zhang. Boosting for content-based audio classification and retrieval: An evaluation. In *2001 IEEE International Conference on Multimedia and Expo*, page 253, Los Alamitos, CA, USA, aug 2001. IEEE Computer Society. 2

[10] Z. Liu, Y. Wang, and T. Chen. Audio feature extraction and analysis for scene segmentation and classification. *Journal of VLSI signal processing systems for signal, image and video technology*, 20(1-2):61–79, 1998. 1

[11] M. Muller, D. P. Ellis, A. Klapuri, and G. Richard. Signal processing for music analysis. *IEEE Journal of Selected Topics in Signal Processing*, 5(6):1088–1110, 2011. 1

[12] S. R. Ness, A. Theocharis, G. Tzanetakis, and L. G. Martins. Improving automatic music tag annotation using stacked generalization of probabilistic svm outputs. In *Proceedings of the 17th ACM international conference on Multimedia*, pages 705–708. ACM, 2009. 1

[13] L. Reboursière, O. Lähdeoja, T. Drugman, S. Dupont, C. Picard-Limpens, and N. Riche. Left and right-hand guitar playing techniques detection. In *NIME*, 2012. 1, 2

[14] N. Scaringella, G. Zoia, and D. Mlynek. Automatic genre classification of music content: a survey. *IEEE Signal Processing Magazine*, 23(2):133–141, 2006. 1

[15] G. Tzanetakis and P. Cook. Musical genre classification of audio signals. *IEEE Transactions on speech and audio processing*, 10(5):293–302, 2002. 1

[16] Y.-H. Yang, C.-C. Liu, and H. H. Chen. Music emotion classification: A fuzzy approach. In *Proceedings of the 14th ACM international conference on Multimedia*, pages 81–84. ACM, 2006. 2