Electric Guitar Pickups Recognition

Warren Jonhow Lee and Yi-Chun Chen

Objective

Recognize electric pickups from audio records through two machine learning methods:

- Support Vector Machine (SVM)
- Bayesian Networks

Data Extraction

Pickup devices are electric transducers that captures vibrations of guitar strings and converts them to electric signals. There are two commonly used pickups: single coil and humbuckers, and they are shown in Figure 1. Ideally, the classification of pickups can be achieved by selecting features from audio records and learning, since pickups directly affect the sound of guitars. On the other hand, guitar pedals, such as overdrive effect, would distort the sound and thus decrease the classification accuracy. Therefore, the guitar sound used in this project should be clean and recorded directly from amplifier or line in.

Figure: Two guitar pickups: single coil (left) and humbuckers (right)

In this project, the data extraction consists of two stages: preprocessing and feature extraction. In first stage, silence and noise are removed from original audio records, since they have no contribution to later machine learning process. This removing process is achieved by audio segmentation algorithm [1], which is demonstrated in Figure 2. The top plot shows the original audio record. The bottom plot demonstrates the audio segmentation algorithm adapts SVM to distinguish high-energy and low-energy short term frames.

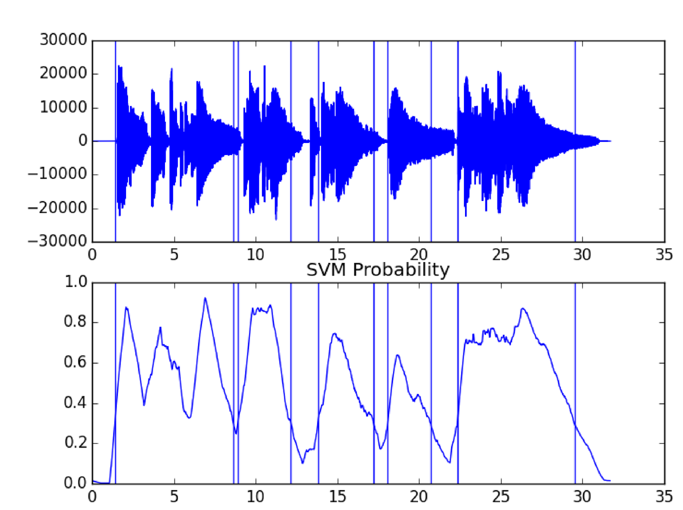

Figure: Demonstration of audio segmentation algorithm

After preprocessing, sixteen features are extracted from audio signals: thirteen Mel-frequency cepstral coefficients (MFCCs), spectral spread, spectral centroid and spectral flatness. MFCCs are commonly used in speech recognition systems as short-term power spectrum of sounds. Spectral spread is associated with the "brightness" of sound. Spectral spread measures the bandwidth of the spectrum. Spectal flatness represents noisiness of the power spectrum. MFCCs and the other three spectral features in a sound are shown in Figure 3 and Figure 4.

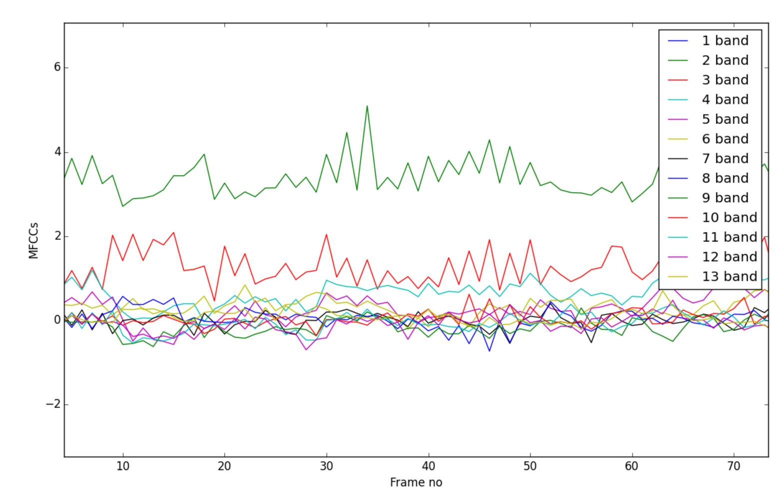

Figure: Variation of thirteen Mel-frequency cepstral coefficients

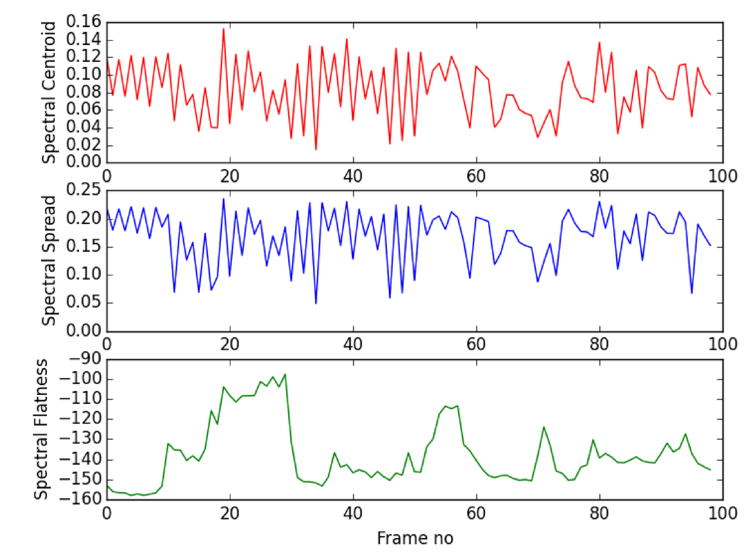

Figure: Variation of three spectral features

SVM

After obtaining features, SVM is applied to classify two pickups. Note that the training data is arranged chronologically, since the temporal property of music can not be ignored. In our tests, such arrangement can improve the learning curves.

SVM is applied with several kernels and various amount of penalty. The following four plots show the learning curves of SVM with linear kernel. In each plot, the green curve is training score (accuracy) versus size of training data. The blue curve is cross-validation score versus size of training data, which can be considered as test accuracy. The desired result is that the green curve and the blue curve converge to the same value. As shown in figures, low penalty $C = 0.001$ SVM with linear kernel achieves such convergence.

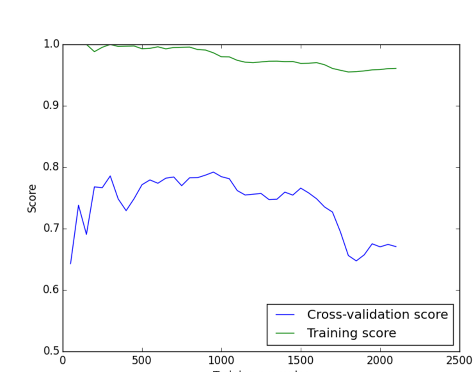

Figure: SVM with linear kernel and penalty $C = 1$

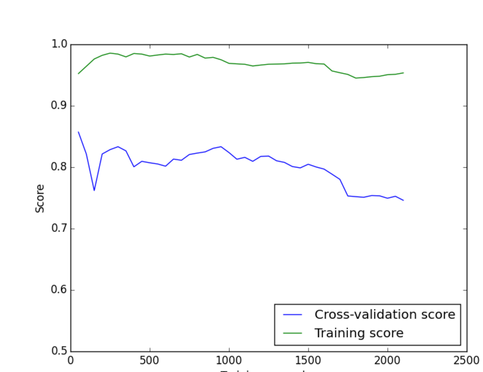

Figure: SVM with linear kernel and penalty $C = 0.1$

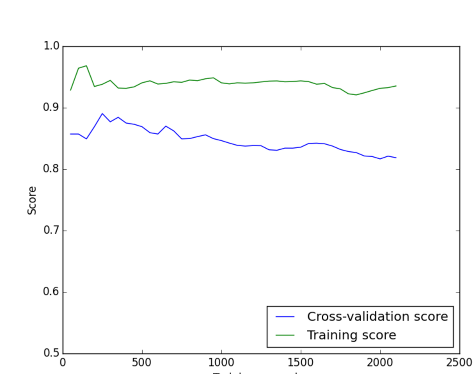

Figure: SVM with linear kernel and penalty $C = 0.01$

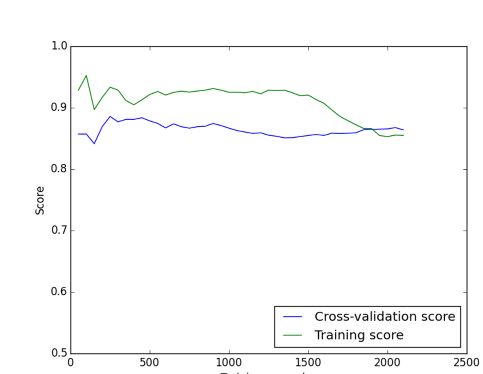

Figure: SVM with linear kernel and penalty $C = 0.001$

SVM with polynomial kernel, which is $\{1, x, x^2, x^3\}$, is also tested. The result is shown in following two plots. It illustrates that penalty does not affect the learning curves under polynomial kernel. In addition, the learning curves indicate SVM with polynomial kernel is over-fitting.

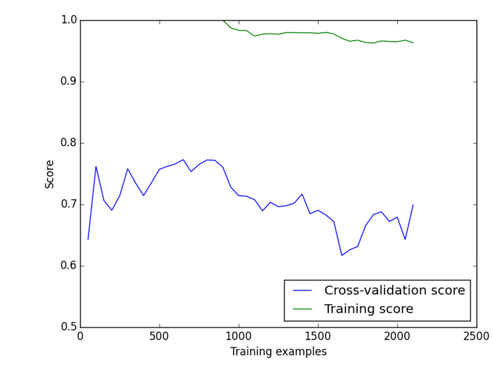

Figure: SVM with polynomial kernel and penality $C = 1$

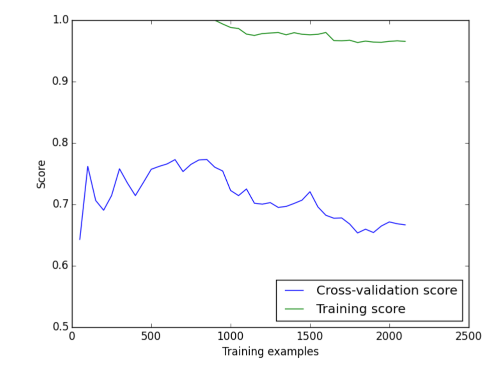

Figure: SVM with polynomial kernel and penality $C = 0.001$

After learning the desired SVM (linear kernel and 0.001 penalty), the next step is to test on new audio files [2], which consist of different pitches, different playing techniques, and different tones. Table 1 shows the accuracy of the learned SVM on five test audio files. SingleCoil1 is composed of only one note and the other four are mixtures of chords and notes. Table 1 indicates that the SVM performed bad on the single note audio file. This matches our expectation, since the SVM is learned from audio files with several chords and notes. In addition, the learned SVM has high accuracy on the other four audio files. It demonstrates that **SVM is a good classifier for guitar pickups, even if the recording data come from different players on different guitars**.

Table: Applying the learned SVM to new audio files

Testing File Name	Accuracy	Sample Size
SingleCoil1 (single note)	41.57%	777
SingleCoil2 (mixture)	96.09%	179
SingleCoil3 (mixture)	87.26%	377
Humbucker1 (mixture)	72.77%	459
Humbucker2 (mixture)	91.5%	459

Bayesian Network

Bayesian network is a probabilistic graphical model that represents random variables and their conditional dependencies via a directed acyclic graph. It has been widely applied to artificial intelligence, medical diagnosis, etc. However, in this pickup classification problem, there are two challenging points. First, the network structure of Bayesian network is not known in advance. Second, feature data are continuous-valued.

To solve the problems, the recent research of one team member at the Stanford Intelligent System Lab has been used. The research applies Bayesian statistics and the proposed priors to find the most probable discretization policy on each continuous variable according to the data of variables in its Markov blanket. In addition, the discretization procedure is incorporated with K2 structure learning algorithm to learn a discrete Bayesian network. For more detail, please refer to [3].

Once the discrete Bayesian network is learned from the continuous data, the prediction on testing data is done as follows: assume X_n is the categorical variable and $(x_1, x_2, \ldots, x_n)$ is the testing data, then the prediction is made by calculating

$$P(X_n \mid x_1, x_2, \ldots, x_{n-1}) \propto P(X_n, x_1, x_2, \ldots, x_{n-1}),$$

and choosing the value of X_n with higher probability.

Notice that the joint probability on the RHS can be factorized as

$$P(x_1, x_2, \ldots, x_n) = \Pi_{i=1}^{n} P(x_i \mid \text{parent}_{x_i}).$$

Figure 11 is the learned discrete Bayesian network. In order to reduce the runtime, only seven important features (MFCC2 to MFCC6, Spectral Spread, Spectral Flatness) are used in the learning process and the upper bound of parents for each node is limited by two. This network has 93% accuarcy on training data and 70% accuracy over all testing data in Table 1. The performace is just slightly weeker than SVM.

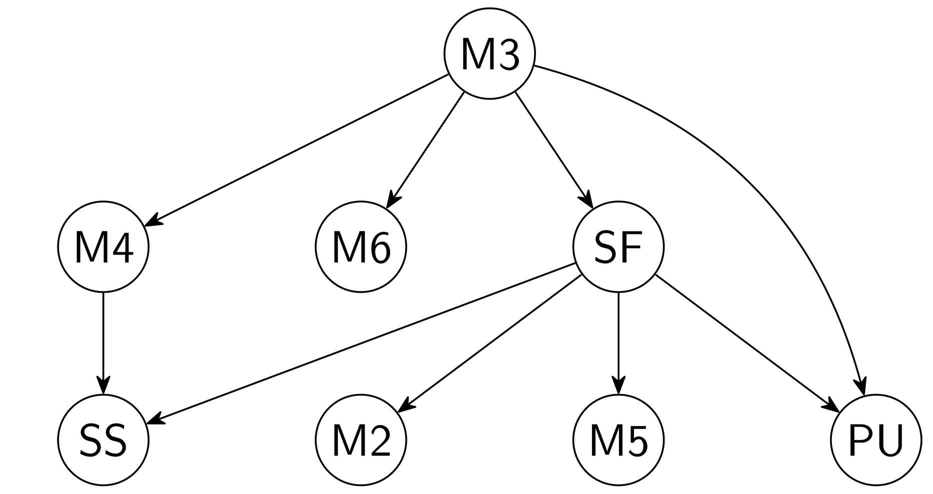

Figure: The learned Bayesian network from seven selected features and the pickup.

On the other hand, the Bayesian network method has an advantage that new non-audio features can be easily incorporated into the network. For example, in the future, guitar brands and wood materials obtained from image processing might be introduced to determine price of guitars along with pickup information. Then Figure 12 shows a possible network to predict price.

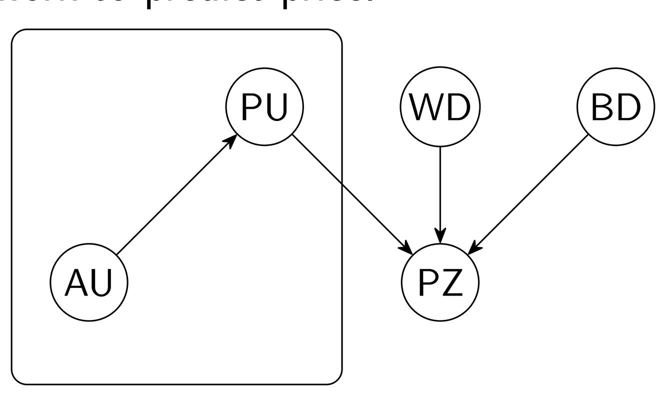

Figure: A possible Bayesian network to predict price of guitars. The structure in the box corresponds to the network in Fig 11.

Conclusion

In this project, SVM with linear kernel is shown to be a good classifier for recognizing pickups from audio files. It has 85% predict accuracy on training data and 84% predict accuracy on audio files played by different players on various guitars. Bayesian network is also a good classifier for pickup recognition. In addition, it can easily incorporate with other non-audio features. In the future, a price prediction model might be proposed and determine whether a guitar is overpriced.

References

[1] Theodoros Giannakopoulos and Aggelos Pikrakis, Introduction to Audio Analysis: A MATLAB Approach. Academic Press, 2014.

[2] Ted Drozdowski, http://www.gibson.com/News-Lifestyle/Features/en-us/tone-hunting-0309-2011.aspx.

[3] Yi-Chun Chen, Tim Wheeler, and Mykel Kochenderfer, Learn Discrete Bayesian Network from Continuous Data, http://arxiv.org/abs/1512.02406, submitted to *Machine Learning*.