Detecting Musical Key with Supervised Learning

Robert Mahieu
Department of Electrical Engineering, Stanford University

Overview

In 1982, Krumhansl and Kessler [1] proposed the concept of the tonal profile which assigned relative weights to each pitch class in each key indicating how musically important it was to the key. The study generated two *KK-profiles*, for major and minor modes, which could be transposed to work for any key.

Many modern key-detection algorithms pre-process an input song then use a correlation function, such as cosine similarity, to find which key's tonal profile is most similar. However, this method is very sensitive to the profile used. Slight variations of the KK-profile have been found to produce significantly better or worse classification results [2].

This project aims to use a supervised approach to instead learn tonal *regions* for each key, instead of a single profile. Although the training data was not entirely reliable, results seem to indicate that this system is capable of correct classification over 80% of the time.

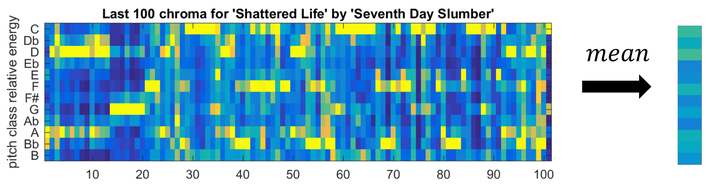
Data

Mode and key metadata, as well as chromogram feature data, was taken from the *Million Song Dataset* [3], which includes audio features and metadata for one million songs.

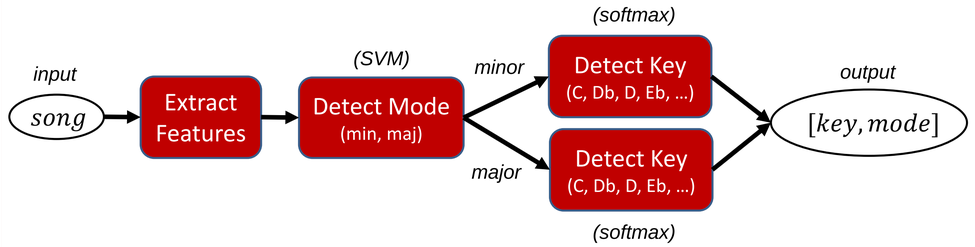
Note that due to the large size of the entire dataset (>300GB), this project was only able to use a 10,000 song subset.

Features

To represent the strength of pitches throughout a song, chromogram features are used. A chromogram represents the amount of relative energy in each pitch class across time windows in a song. To account for variable song lengths, an average chromogram vector was used as the data feature.



System Architecture



SVM Mode Classification

Experimental results comparing logistic regression to an SVM for classifying mode determined SVM to be more successful. The SVM model was trained for the two classes by minimizing the cost function:

$$J(\alpha) = \frac{1}{m}\sum_{i=1}^{m} L_{hinge}\left(\sum_{j=1}^{m} \alpha_j K\left(x^{(i)}, x^{(j)}\right), y^{(i)}\right)$$

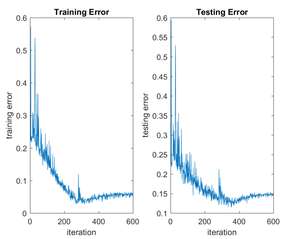
Where the loss function is chosen as hinge loss:

$$L_{hinge}(z, y) = [1 - yz]_+$$

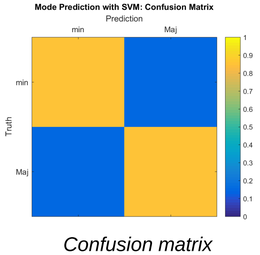
And the kernel $K(x, z)$ is chosen as the radial basis function:

$$K(x, z) = \exp\left(-\frac{1}{2}||x - z||^2\right)$$

Data standardization and box constraint regularization were employed to mitigate overfitting and better fit for underrepresented data.



Error per training iteration



Confusion matrix (normalized over columns)

Softmax Key Classification

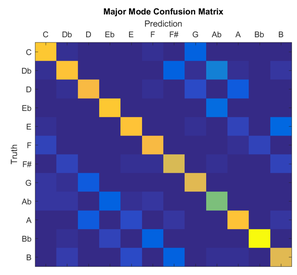
Experimental results comparing multinomial logistic (softmax) regression to an SVM for classifying key determined softmax to be more successful. The model was trained for all the $K = 12$ classes using maximum likelihood estimation on the likelihood function:

$$L(\theta) = \prod_{i=1}^{m}\left(\prod_{k=1}^{K} P\left(y^{(i)} = k \middle| x^{(i)}; \theta\right)^{1\{y^{(i)}=k\}}\right)$$

Where the probability term:

$$P\left(y^{(i)} = k \middle| x^{(i)}; \theta\right) = \frac{\exp\left(\theta_k^T x^{(i)}\right)}{1 + \sum_{j=1}^{K} \exp\left(\theta_j^T x^{(i)}\right)}$$

Confusion matrix for classifying major keys (normalized over columns)

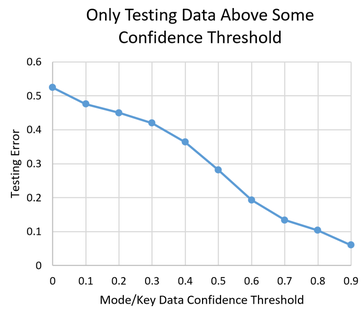


Results are quite good in this example for classifying major keys (testing error of 17.083% on cross-validation set), though **misclassification is common for keys one fifth up and one fifth down from the truth** due to the similarity of the underlying scales.

Results

It is important to note that the "truth" training/testing data from the dataset is not actually ground truth, and is actually given some confidence metric, $c \in [0,1]$. As such, the system was trained only on data with $c \geq 0.5$.

After training the system, all data in the dataset over various confidence thresholds was tested and the error rates are summarized in the plot below:



These results appear to indicate that the system is quite capable of correctly estimating the correct key, since it does very well when data confidence is high. When confidence is ≥ 0.6, we observe error rates less than 20%. This is about equivalent to success rate of the best free key-detection software available [4].

References

[1] Krumhansl, Carol L., and Edward J. Kessler. "Tracing the dynamic changes in perceived tonal organization in a spatial representation of musical keys." *Psychological review* 89.4 (1982): 334.

[2] Sha'ath, Ibrahim. *Estimation of key in digital music recordings*. Diss. Master's Thesis, Birkbeck College, University of London, London, UK, 2011.

[3] Bertin-Mahieux, Thierry, et al. "The million song dataset." *ISMIR*. Vol. 2. No. 9. 2011.

[4] "Key Detection Software Comparison: 2014 Edition." *DJ TechTools*. N.p., 14 Jan. 2014. Web. 20 Nov. 2016.