Selecting Groups of Audio Features by Statistical Tests and the Group Lasso

Bernd Bischl, Markus Eichhoff, Claus Weihs

Chair of Computational Statistics, TU Dortmund
E-Mail: {bischl,eichhoff,weihs}@statistik.tu-dortmund.de
Web: www.statistik.tu-dortmund.de

Abstract

In this paper we aim at discriminating between two musical instruments by means of different groups of audio features, namely absolute amplitude envelope in the time domain as well as MFCC, pitchless periodogram and simplified spectral envelope in the spectral domain. For this task we utilize common statistical classification algorithms and perform statistical tests to evaluate whether the discriminating power of certain subsets of feature groups dominates other group subsets. We also examine if it is possible to directly select a useful set of groups by applying logistic regression regularized by a group lasso penalty structure. Specifically, we apply our methods to a data set of single piano and guitar tones.

1 Description of Features

Each single tone consists of an audio signal $x[n]$, $n \in \{1,\dots,N\}$, of length 1.2s at sampling rate $sr = 44100$ Hz. From this the following four feature vectors are calculated. In general, the signal is windowed by half-overlapping segments w_s, $s \in \{1,\dots,25\}$ of a size of 4096 samples.

1.1 Absolute Amplitude Envelope

To take the upper and lower part of the envelope into account the absolute values $|x[n]|$ define the so-called *absolute amplitude envelope* $e \in \mathbb{R}^{1\times 132}$ by setting $l = \lfloor \frac{N}{400} \rfloor \cdot 400$ as follows:

$$e = \left(\max_{1\le i\le 400} \{|x[i]|\}, \max_{401\le i\le 800} \{|x[i]|\}, \dots, \max_{l-399\le i\le l} \{|x[i]|\} \right).$$

Note that here non-overlapping segments of size 400 are used.

1.2 Pitchless Periodogram

The periodogram P of each window is calculated at fixed frequencies $\{X_1,\dots,X_{2048}\}$, $\frac{sr/2}{2048} \le x_i \le \frac{sr}{2}$. Additionally for each window the fundamental frequency is estimated (called $\hat{f}_0$) so that overtones can be calculated as $\hat{f}_i = (i+1)\cdot \hat{f}_0$, $i \in \{0,\dots,13\}$. For each fixed $\hat{f}_i$ and each window w_s the periodogram values, i.e. the squared values of the DFT,

$$P^{w_s}_{\hat{f}_i}(x^i), \text{ with } |\hat{f}_i - X^i| = \min_{1\le j\le 2048} |\hat{f}_i - X_j|$$

$\forall s \in \{1,\dots,25\}, \forall i \in \{0,\dots,13\}$ are calculated. Medians of blocks of five subsequent time windows are considered:

$$p^r_i := \text{median}\left(P^{w_r}_{\hat{f}_i}(X^i), P^{w_{r+1}}_{\hat{f}_i}(X^i), \dots, P^{w_{r+4}}_{\hat{f}_i}(X^i) \right)$$

for $i \in \{0,\dots,13\}$ and $r \in \{1,6,11,16,21\}$. The *Pitchless Periodogram* $v \in \mathbb{R}^{1\times 70}$ is then defined as

$$v = \left(p^1_0, p^1_1, \dots, p^1_{13}, p^6_0, \dots, p^6_{13}, \dots, p^{21}_0, \dots, p^{21}_{13} \right).$$

This is called "'pitchless"', because v is independent of the pitch and the distances $X^{i+1} - X^i$.

1.3 Mel Frequency Cepstral Coefficients

The power spectrum is calculated by a DFT using Hamming windows and a subsequent log-transformation. After mapping the powers of the spectrum onto the mel scale by using triangular filters the discrete cosine transformation is applied yielding the MFCC coefficients.

1.4 LPC Simplified Spectral Envelope

For each time window the coefficients of a pth-order linear predictor (FIR filter) are calculated with $p = \lfloor 2 + sr/1000 \rfloor = 46$ (rule of thumb of formant estimation). So the current value of the signal $x[n]$ in segment k can be estimated by the past samples:

$$\hat{x}^k(n) = -a^k_2 x^k(n-1) - a^k_3 x^k(n-2) - \dots - a^k_{p+1} x^k(n-p).$$

The 512-points complex frequency response vector H of the filter can be interpreted as the transfer function evaluated at $z = e^{i\omega}$:

$$H^k(e^{i\omega}) = \left(\sum_{l=1}^{p+1} a^k_l e^{-i\omega l} \right)^{-1}, \; k \in \{1,\dots,25\}, a^k_1 = 1$$

where a^k_l are the linear predictor coefficients. This frequency response is calculated for each time window k and so yields a matrix $K \in \mathbb{R}^{512\times 25}$, with $K_{\cdot,j} = 20\log_{10}|H^j|$, $j \in \{1,\dots,25\}$. With $r \in \{1,6,11,16,21\}$ define

$$v^r := \text{median}\left(K_{\cdot,r}, K_{\cdot,r+1}, K_{\cdot,r+2}, K_{\cdot,r+3}, K_{\cdot,r+4}\right).$$

This yields $V = \left(v^1, v^6, v^{11}, v^{16}, v^{21}\right) \in \mathbb{R}^{512\times 5}$. The *Simplified LPC Spectral Envelope* $s \in \mathbb{R}^{1\times 125}$ is then the maximum of each subsequent 20 rows of V:

$$s = \Bigg(\max_{1\le j\le 20} \{V_{j,1}\}, \max_{21\le j\le 40} \{V_{j,1}\}, \dots, \max_{501\le j\le 512} \{V_{j,1}\},$$
$$\vdots$$
$$\max_{1\le j\le 20} \{V_{j,21}\}, \max_{21\le j\le 40} \{V_{j,21}\}, \dots, \max_{481\le j\le 501} \{V_{j,21}\} \Bigg).$$

2 Statistical Modeling and Evaluation

In order to identify which of the above groups are most useful to discriminate between tones of different musical instruments, we do not employ a usual feature selection algorithm. We are not primarily interested in an optimal set of features chosen arbitrarily across all groups, but rather want to statistically evaluate which of complete groups are most useful for our classification task at hand. To put it differently, we would like to identify a minimal set of

groups classifying optimally. This does not only reduce runtime and storage requirements in applications, but also stabilizes the fitting process of classification models, as the number of features compared to the number of observations might be quite large. We follow a two-fold approach to achieve these objectives.

2.1 Testing Generalization Performance

First, we employ the framework for benchmark experiments by Hothorn et al. [6] to compare the discriminating power of different sets of feature groups. By applying a resampling strategy like bootstrapping or subsampling one independently generates training sets from a given data set, uses a classification algorithm to fit models on these, predicts the out-of-bag test samples and measures their performance according to an appropriate loss function. This generates a population of performance values for every classifier, which now can be compared by using standard statistical inference methodology. But instead of the usual approach of fixing a certain set of features and then comparing the generalization performance of different kinds of classifiers, we fix the classifier and then vary the sets of features. We are generalizing a similar approach for a comparable setting in [14].

2.2 Group Lasso for Logistic Regression

The lasso penalty is a well-known way to directly encode the aim of variable selection into the problem of minimizing the empirical error of a generalized linear predictor:

$$\min_{\beta,\beta_0} \left(\sum_{i=1}^{n} L(y_i, \beta^T x_i + \beta_0) + \lambda \sum_{j=1}^{p} |\beta_i| \right) .$$

Here, the first term is the empirical error on the training data $\{(x_i, y_i)\}_{i=1}^{n}$ from $\mathbb{R}^p \times \{0,1\}$, measured by some loss function $L(\cdot,\cdot)$. If we set

$$L(y,x) = -y(\beta^T x + \beta_0) + \log(1 + \exp(\beta^T x + \beta_0)) ,$$

i.e. to the negative log-likelihood of a logistic model and ignore the second summand we arrive at simple logistic regression. The second term penalizes large entries of the coefficient vector β through the l_1 norm. It is well-known that for this specific norm the optimal estimator for β will usually contain at least some zero entries, which can be interpreted as an implicit form of feature selection [5]. The constant λ controls the influence of the penalty term.
The group lasso for logistic regression [11] generalizes this by penalizing disjoint groups of features individually by the l_2 norm, but combines all group terms with a penalty of the l_1 type

$$\min_{\beta,\beta_0} \left(\sum_{i=1}^{n} L(y_i, \beta^T x_i + \beta_0) + \lambda \sum_{g=1}^{G} ||\beta_{I_g}||_2 \right) ,$$

where I_g is an indicator function referring to the index set of all features in group g. This achieves feature selection at the group level. Both optimization problems for the simple lasso and the group lasso are non-trivial to solve because of the non-differentiability of the l_1 norm. Therefore special purpose algorithms have been developed, for the latter problem often a group-coordinate version of gradient descent is used [11].

3 Experimental Setup

For the experiment 4309 guitar and 1345 piano tones are used. They are recorded as wav-files and arise in three different databases: McGill University Master Samples [10], RWC Database [4] and the Musical Instrument Samples of the University of Iowa [12]. The calculation of the total 407 features is carried out by using the *MIRtoolbox* [8] for Matlab and the *tuneR* package [9] for the statistical programming language *R*.
Using subsampling with 100 repetitions, we generate 100 out-of-bag performance values for each of the $2^4 - 1 = 15$ possible sets of feature groups. As learning algorithms we consider linear discriminant analysis (LDA), logistic regression (LReg), decision trees (CART) and support vector machines (SVM) with a radial basis kernel [5]. Hyperparameters for the SVM are tuned by nested resampling to ensure unbiased estimates of the prediction performance. To reduce runtime, hyperparameter optimization is performed by Nelder-Mead with a heuristically selected, data-dependent starting point from the training data. After descriptively checking that the normality assumption for the error terms holds at least approximately, we use a linear mixed effects model to analyze significant differences in performance between the feature groups [3]. Lastly, we compare and contrast the results from these tests to the groups selected by the group lasso. For all experiments we use the *mlr* [1] package for machine learning in *R*, for the all-pairs tests and order relations (see below) the *benchmark* package [3].

4 Results and Discussion

Table 1 shows the classification results for linear discriminant analysis, logistic regression, decision trees and support vector machines and all different combinations of feature groups. (In the table and following figures these abbreviations are used: C = Pitchless Chromagram, E = Absolute Amplitude Envelope, L = LPC Simplified Spectral Envelope, M = MFCC.) We calculate the mean classification error and the standard deviation as a measure of spread for the respective predictions of the 100 test sets from subsampling.
In figure 1 we report orderings of feature groups according to significant differences in misclassification error. Per learning algorithm, we generate a graph of partially ordered feature groups by including an edge between A and B if feature group A significantly outperforms feature group B, locating A below B in the figure.
One can see quite clearly that the MFCCs would be chosen, if one would have to select a single feature group. This confirms fact that MFCCs have already proven to be very useful for classification tasks in speech processing [13] as well as in musical instrument recognition [7], [15], [2] in many other studies. Combining MFCCs with LPCs probably achieves the best trade-off between a low number of groups and best predictive performance. By the chroma and time envelope features a small gain can be achieved only if they are added to the MFCC / LPC groups. The best classification result of 0.9% error, significanlty outperforming all other ones, is produced by SVM with the MFCCs, LPCs and the time envelope features. CART results are most often worst.
These observations are essentially reproduced by the group lasso results in figure 2. Along the solution path from a large penalization parameter λ to a lower one, most useful

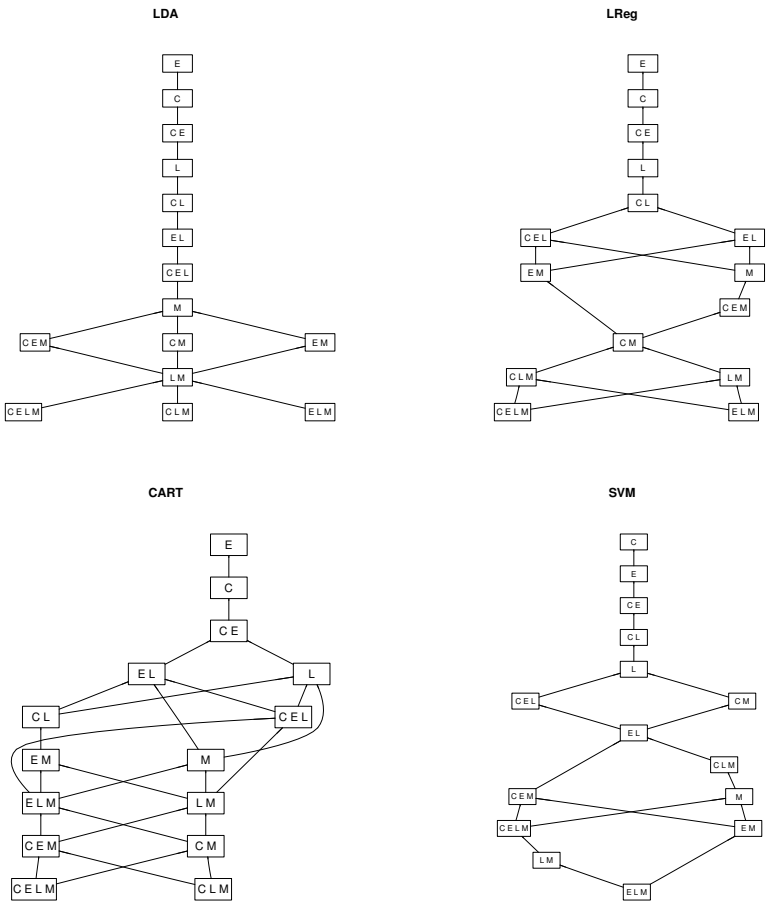

Figure 1: Ordering of feature groups for considered classifiers.

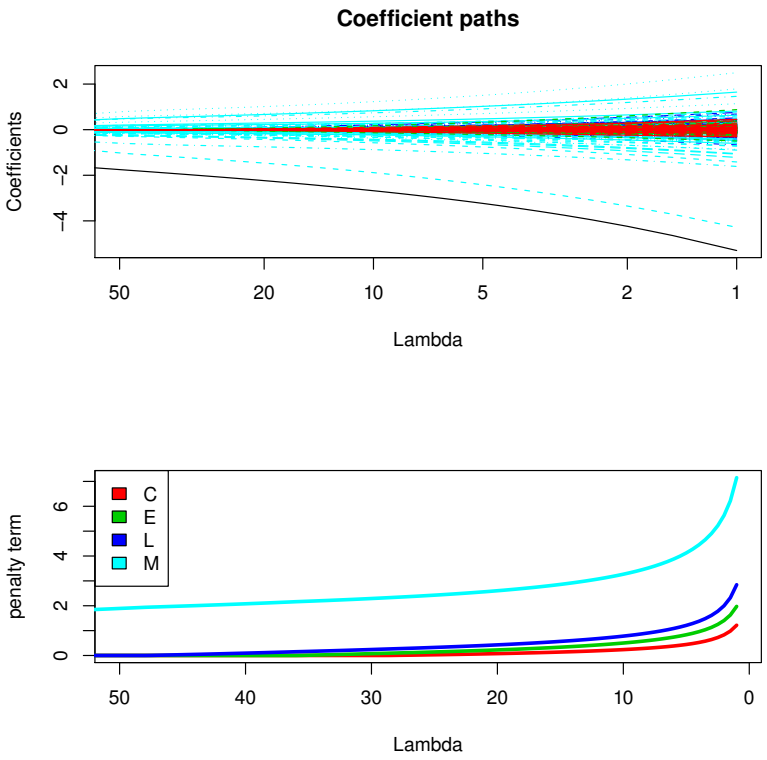

Figure 2: Coefficient paths for group lasso and penalty term per group. Path for intercept is shown in black in top plot.

	LDA	LReg	CART	SVM
	mean sd	mean sd	mean sd	mean sd
C	0.205 0.010	0.197 0.012	0.122 0.010	0.182 0.011
E	0.215 0.011	0.206 0.011	0.173 0.011	0.171 0.010
C E	0.191 0.009	0.162 0.009	0.116 0.010	0.114 0.009
L	0.101 0.008	0.072 0.007	0.101 0.009	0.039 0.006
C L	0.096 0.008	0.068 0.007	0.080 0.009	0.051 0.006
E L	0.089 0.007	0.063 0.006	0.101 0.010	0.026 0.004
C E L	0.084 0.007	0.060 0.007	0.078 0.009	0.031 0.005
M	0.039 0.005	0.039 0.005	0.077 0.009	0.019 0.004
C M	0.035 0.005	0.033 0.005	0.061 0.008	0.030 0.005
E M	0.035 0.005	0.037 0.005	0.076 0.009	0.013 0.004
C E M	0.033 0.005	0.036 0.005	0.059 0.008	0.021 0.004
L M	0.028 0.004	0.028 0.004	0.068 0.008	0.012 0.003
C L M	0.025 0.004	0.026 0.005	0.049 0.006	0.022 0.004
E L M	0.024 0.004	0.021 0.004	0.068 0.008	0.009 0.003
C E L M	0.024 0.004	0.021 0.004	0.047 0.007	0.015 0.004

Table 1: Misclassification error on subsampling test sets.

groups for the model fit will enter the solution first. This leads to the ordering: MFCCs, LPCs, time envelope and chroma vector.

One should also note, although the general results in figure 1 are very similar for all classifiers, the detailed ordering of the feature groups is somewhat different and especially the best performing feature groups are not always the same. Indicating again the often observed phenomenon, that for optimal results one has to perform feature selection with respect to a particular classification algorithm.

5 Conclusion and Outlook

We have demonstrated two different statistical approaches to select relevant feature groups in the domain of musical instrument classification and very successfully solved the problem at hand. While the analysis by logistic regression with the group lasso is computationally cheap and also already produces the relevant orderings, the statistical testing of the discriminating power of feature sets provides more detailed results with higher computational costs.

If resources are sparse, a reasonable result is achieved by considering only the MFCCs, with an error rate $2-3\%$ worse than the optimum. It remains unclear whether the additional inclusion of other feature groups is worthwhile. We aim to answer these questions in a larger follow-up study, where more groups of musical instruments, more tones, and polyphonic signals are considered.

References

[1] B. Bischl. mlr: Machine learning in R, 2010. `http://mlr.r-forge.r-project.org`.

[2] J.C. Brown. Computer identification of musical instruments using pattern recognition with cepstral coefficients as features. *The Journal of the Acoustical Society of America*, 105(3):1933–1941, 1999.

[3] Manuel J. A. Eugster, Torsten Hothorn, and Friedrich Leisch. Exploratory and inferential analysis of benchmark experiments. Technical Report 30, Institut für Statistik, Ludwig-Maximilians-Universität München, Germany, 2008.

[4] Masataka Goto and Takuichi Nishimura. RWC music database: Music genre database and musical instrument sound database. In *in ISMIR*, pages 229–230, 2003.

[5] T. Hastie, R. Tibshirani, and J.H. Friedman. *The Elements of Statistical Learning*. Springer, 2001.

[6] T. Hothorn, F. Leisch, A. Zeileis, and K. Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14:675–699, 2005.

[7] S. Krey and U. Ligges. SVM based instrument and timbre classification. In H. Locarek-Junge and C. Weihs, editors, *Classification as a Tool for Research*, Berlin-Heidelberg-New York, 2009. Springer.

[8] O. Lartillot, P. Toiviainen, and T. Eerola. A matlab toolbox for music information retrieval. In C. Preisach, H. Burkhardt, L. Schmidt-Thieme, and R. Decker, editors, *GfKl*, Studies in Classification, Data Analysis, and Knowledge Organization, pages 261–268. Springer, 2007.

[9] U. Ligges. tuneR: Analysis of music. `http://r-forge.r-project.org/projects/tuner`.

[10] McGill. Master samples collection on dvd, 2010. `http://www.music.mcgill.ca/resources/mums/html`.

[11] L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(1):53–71, 2008.

[12] University of Iowa: Electronic Music Studios. Musical instrument samples. `http://theremin.music.uiowa.edu`.

[13] L. Rabiner and B.H. Juang. *Fundamentals of Speech Recognition*. Prentice Hall PTR, 1993.

[14] G. Szepannek, T. Harczos, F. Klefenz, and C. Weihs. Extending features for automatic speech recognition by means of auditory modelling. In *Proc. EUSIPCO 2009*, 2009.

[15] E. Wold, T. Blum, D. Keislar, and J. Wheaton. *Handbook of Multimedia Computing*, chapter Classification, search and retrieval of audio, pages 207–226. CRC Press, 1999.