SCALABLE GENRE AND TAG PREDICTION WITH SPECTRAL COVARIANCE

James Bergstra
University of Montreal
bergstrj@iro.umontreal.ca

Michael Mandel
University of Montreal
mandelm@iro.umontreal.ca

Douglas Eck
University of Montreal
eckdoug@iro.umontreal.ca

ABSTRACT

Cepstral analysis is effective in separating source from filter in vocal and monophonic [pitched] recordings, but is it a good general-purpose framework for working with music audio? We evaluate covariance in spectral features as an alternative to means and variances in cepstral features (particularly MFCCs) as summaries of frame-level features. We find that spectral covariance is more effective than mean, variance, and covariance statistics of MFCCs for genre and social tag prediction. Support for our model comes from strong and state-of-the-art performance on the GTZAN genre dataset, MajorMiner, and MagnaTagatune. Our classification strategy based on linear classifiers is easy to implement, exhibits very little sensitivity to hyper-parameters, trains quickly (even for web-scale datasets), is fast to apply, and offers competitive performance in genre and tag prediction.

1. INTRODUCTION

Many features for music classification have a longer history in speech recognition. One of the first steps taken by most speech recognizers is to transform audio containing speech into a sequence of phonemes. A phoneme is the smallest segmental unit of sound, for example the /b/ in "boy". For a speech recognizer to work with multiple speakers, it needs to generalize over a range of voice types (adult verus child, male verus female). To achieve this generalization it can be useful to separate the audio signal into two parts: the source excitation at the vocal cords and the transfer function (filtering) of the vocal tract. Cepstral analysis is commonly used to achieve this separation. The cepstrum C is defined as

$$C = |G\log(|Fx|^2)|^2 \quad (1)$$

where F is the discrete Fourier transform and G is the inverse discrete Fourier transform or the discrete cosine transform. One important property of the cepstrum is that convolution of two signals can be expressed as the addition of their cepstra.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page.
© 2010 International Society for Music Information Retrieval.

In general cepstral analysis for speech recognition or music analysis is done on a power spectrum $|Fx|^2$ that has been downsampled (compressed) non-linearly to better model human perception of equidistant pitches (the Mel scale). The resulting cepstral-domain values are called Mel-Frequency Cepstral Coefficients (MFCCs). See Section 2 for details.

In the domain of music, cepstral analysis can be used to model musical sounds as a convolution of a pitched source (perhaps a vibrating string) and an instrument body that acts as a filter. This deconvolution of musical source (pitch) from musical filter (instrument) can be seen for several instruments in Figure 1. If our goal is to generalize across different pitches for a single instrument timbre, it is clear that the cepstral domain has advantages over the spectral domain.

The main use of source / filter deconvolution in speech is allow for the *elimination* of the source. This is achieved by only retaining the first few cepstral coefficients (usually 12 MFCCs for speech recognition). However, music is not speech. The assumption that recorded music consists of a filter with it's own sound quality (instrument timbre) acting on an irrelevant source is certainly false. For many instruments, it is difficult to distinguish between pitch and timbre, and certainly the assumption breaks down in polyphonic recordings.

This paper presents segment covariance features as an alternative to means and variances in MFCCs for condensing spectral features for linear classification. The covariance, together with mean and variance in spectral features provides a better basis for genre and tag prediction than those same statistics of MFCCs. These features are quick and easy to compute (pseudo-code given below) work well with linear classification. Together they offer a viable approach to web-scale music classification, and a competitive null model for research on new datasets.

2. FEATURES

We follow [1] in distinguishing two levels, or stages, of feature extraction. Firstly, at the *frame level* (typically 20-50 milliseconds) we extract features such as Mel Frequency scale Cepstral Coefficients (MFCCs). Secondly, at the *segment level* (typically 3-10 seconds) we summarize frame-level features via statistics such as the mean, variance and covariance. For our frame-level features we partitioned the audio into contiguous *frames* of either 512,

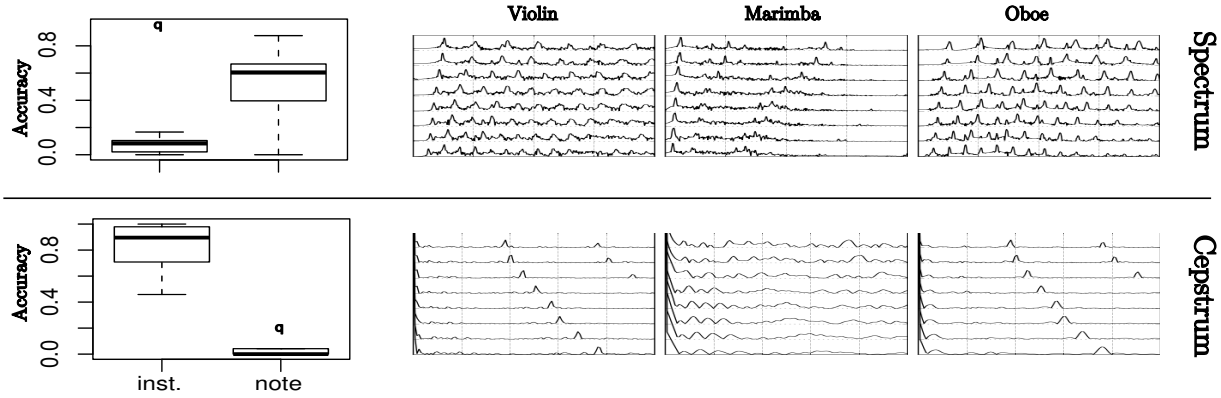

Figure 1. Comparison of spectral and cepstral analysis for eight notes from twelve instruments. **Right:** The six panels show spectral and cepstral anslysis for three of the twelve instruments. The horizontal axis is frequency for the three spectral panels, and time for the three cepstrum panels. The vertical axis for each of the eight subplots is coefficient magnitude. Spectral analysis (above) highlights the overtone series. Cepstral analysis separates pitch and timbre for pitched instruments, separation less clear for marimba. **Left:** Nearest neighbor classifiers using spectral and cepstral features to predict note and instrument labels–cepstrum predicts instruments, spectrum predicts notes.

1024, or 2048 samples, depending on the samplerate – whichever corresponded to a duration between 25 and 50 ms.

We compare two kinds of frame-level features in this work: MFCCs and Log-scaled Mel-Frequency Spectrograms (denoted LM features). MFCCs have been used extensively for music classification [1, 3, 6–8, 12, 14]. As our null model, we implemented MFCCs as in Dan Ellis' Rastamat toolbox's implementation of HTK's MFCC. [1] Our MFCC frame-level feature was 16 coefficients computed from 36 critical bands spanning the frequency range 0-4kHz using the HTK Mel-frequency mapping

$$\mathrm{Mel}(\mathrm{freq}) = 2595 \log_{10}(1 + \mathrm{freq}/700), \qquad (2)$$

using a type-II DCT, and with Hamming windowing. Here the variable freq denotes the frequency of an FFT band, and it is in units of Hz.

The definition of our LM features differs subtly from the MFCCs in ways that make the feature computation faster and the implementation simpler. The audio is scaled to the range (-1.0,1.0). No windowing is applied to the raw audio frames prior to the Fourier transform. A small constant value (10^{-4}) is added in the Fourier domain to represent a small amount of white noise and prevent division by zero. The Fourier magnitudes (not the power of each band) were projected according to a Mel-scale filterbank (the `M` in Table 2). The loudness in each critical band was approximated by the logarithm of its energy. Our experiments used from 16 to 128 critical bands (LM coefficients) to cover the frequencies from $0Hz$ to $16kHz$.

$$\mathrm{LM} = \log_{10}(|FA|M + 10^{-4}) + 4 \qquad (3)$$

$$\mathrm{MFCC} = G(\mathrm{LM}) \qquad (4)$$

Equations 3 and 4 describe the computation of the MFCC and LM features in terms of an audio matrix A that has a row for each frame in a segment, and a column for each sample in a frame. The FFT is taken over columns. The constant 10^{-4} quantifies our uncertainty in $|FA|$ and prevents taking a logarithm of zero. The subtraction of 4 from the logarithm ensures that LM features are non-negative, but take value zero when the audio is silent. The MFCCs are the discrete cosine transform G of LM.

[1] HTK: `http://htk.eng.cam.ac.uk/`

```
sr <-samplerate
nf <-number of Fourier bins
nm <-number of mel-scale coefficients

nyq = sr/2
nyq_mel = 2595 * log10(1 + nyq/700.)

# Build a Mel-Frequency Scale
# Filterbank matrix M
M = zero_matrix(rows=nf, cols=nf)
for i in 0..(nf-1):
  f     = i * nyq / nf
  f_mel = 2595 * log10(1 + f/700.)
  m_idx = f_mel/nyq_mel*nm
  j     = floor(m_idx)
  M(j+1,i) = m_idx-j
  M(j+0,i) = 1.0-m_idx+j
```

Table 1. Pseudo-code for building a Mel-Frequency Scaled filterbank in the frequency domain. This code assumes 0-based array indexing.

3. PERFORMANCE

We used three datasets to explore the value of these features in different descriptor prediction settings: tag frequency, tag presence, and genre. Segments were summarized by the mean, variance, and/or covariances of frame-level features. The summaries were classified by either a

logistic regression model, or where noted, by a Support-Vector Machine (SVM) with an RBF kernel.

In linear models, we trained binary and multi-class logistic regression to maximize the expected log-likelihood

$$-\sum_{x\in X}\sum_{l\in L} P_{true}(l|x)\log(P_{predicted}(l|x)) \qquad (5)$$

where X is the set of examples, and L the set of labels. The linear (technically affine) prediction was normalized across the classes using a Gibbs distribution.

$$P_{predicted}(l|x) = \frac{e^{Z_l f(x)+b_l}}{\sum_{k\in L} e^{Z_k f(x)+b_k}} \qquad (6)$$

where $f(x)$ is the features we extract, Z_l is the l^{th} row of the linear model, and b_l is a scalar bias for the l^{th} predictor. We fit this model by gradient descent with a small learning rate and regularize it by early stopping on held-out data. Our SVM experiments were performed using LIBSVM with the RBF kernel to implement all-pairs multiclass prediction. Held-out data was used to choose the best kernel and cost parameters (typically called γ and C in SVM literature). Features [by dimension] were normalized to have zero mean and unit variance prior to training a classifier.

3.1 Genre Prediction

Genre classification performance was estimated using the GTZAN dataset of 1000 monophonic audio clips [12], each of which is 30 seconds long. The dataset features 100 clips each of the 10 genres: blues, classical, country, disco, hiphop, pop, jazz, metal, reggae, and rock. Although this collection is relatively small, it has been used in several studies of genre classification [1, 5, 10, 13]. Classification was performed by partitioning the audio into 3-second segments and voting over the segments, as in [1]. Following standard practice, performance was measured by standard 10-fold cross-validation. For each fold, the training set was divided into a hold-out set (of 100 randomly chosen songs) and fitting set (of the remaining 800). The results of our models and some results from the literature are shown in Table 2

Our best results with both the linear and SVM classifiers were with the mean and covariance (or correlation) in LM features rather than MFCC features. Our linear model using covariance in LM features was approximately 77% accurate, while the more expensive SVM was 81% accurate. The small size of the GTZAN dataset leaves considerable variance in these performance estimates (scores are accurate to within about 4 percentage points with 95% probability). To our knowledge, the only systems to surpass our baseline linear classifier are the AdaBoost-based model of [1] and the model of [10] based on L1-regularized inference and non-negative tensor factorization. Both of these superior models are significantly more complicated and CPU intensive than our baseline. In larger datasets, the capacity of the linear model to use more data in a tractable amount of time should make its performance improve in comparison to the SVM.

Algorithm	Acc.(%)
Sparse rep. + tensor factor. [10]	92
AdaBoost + many features [1]	83
*RBF-SVM with LM (m32,r32)	81
*RBF-SVM with LM (m32,c32)	79
*Log. Reg. with LM (m32,r32)	77
*RBF-SVM with MFCC (m32,c32)	76
*Log. Reg. with LM (m32,c32)	76
*RBF-SVM with MFCC (m32,r32)	74
*Log. Reg. with MFCC (m32,r32)	72
*Log. Reg. with MFCC (m32,c32)	70
RBF+MFCC [11]	72
LDA+MFCC(m5,v5) + other [5]	71
GMM+MFCC(m5,v5) + other [13]	61

Table 2. Classification accuracies on the GTZAN dataset of our algorithms (*) and others in the literature. 'm32' is the means of 32 frame-level features, 'c32' is the upper triangle in their covariance matrix, 'r32' is the upper triangle in their correlation matrix. These scores are accurate to within about 4 percentage points with 95% probability.

3.2 Tag Frequency

We estimated our models' performance at tag frequency prediction using the MajorMiner dataset. The MajorMiner dataset consists of short audio clips labeled by players of the MajorMiner web game [7]. In this game, players listen to 10-second clips of songs and describe them with free-form tags. They score points for using tags that agree with other players' tags, but score more points for using original tags that subsequent players agree with. There are 1000 unique tags that have been verified by two players and there are a total of 13,000 such verified usages on 2600 clips. The average clip has been seen by 7 users and described with 30 tags, 5 of which have been verified. The tags describe genres, instruments, the singer (if present), and general musical or sonic qualities.

The MajorMiner dataset includes the number of times each tag was applied to each clip, which gives a graded relevance for each (tag, clip) pair. After removing clips that were tagged 'silent','end','nothing' or had a length less than eight seconds, 2578 remained. Clips are typically nine seconds in length but we used the first 8 seconds. As second-level features, we summarized each clip as a single 8-second segment. We followed [7] in using the top 25 tags (drum, guitar, male, synth, rock, electronic, pop, vocal, bass, female, dance, techno, piano, jazz, hip hop, rap, slow, beat, voice, 80s, electronica, instrumental, fast, saxophone, keyboard) which accounted for about half of the tag usages in the dataset. To ensure that each clip had a valid distribution over these tags, we added to every clip an extra virtual tag with a usage of 0.1. For most clips this usage accounted for about 1.5% of the total tag usage, but for a small number of clips with none of the top 25 tags it accounted for 100%. The clips in the dataset were sorted by their order of occurrence in the "three_columns.txt" file. The dataset was partitioned into train, validation, and test

sets by taking the 6^{th} of every 10 clips for validation, and by taking the 7^{th} and 8^{th} for testing.

We interpreted the tag usages in the MajorMiner dataset as being samples from clip-conditional multinomial distributions. We would like a function to infer that *whole distribution* by conditioning on the audio of the clip. This learning setup differs from multiple independent label prediction, for that case see the Tagatune results below.

Our results are summarized in Figure 2. Our best results were obtained by using LM features and keeping many coefficients: 128 means, 128 variances, covariance in a 32-dimensional downsampling of all 128 features. The best model (m128,v128,c32) had an average (across tag) AUC-ROC of 0.78, which is competitive with the best models from the MIREX 2009 MajorMiner evaluation. The average AUC-ROC values across the 25 tags used here for the MIREX 2009 submissions range from 0.72 (submission "HBC") to 0.79 (submission "LWW2"). [2]

There are two interesting things to note about the effect of covariance features in the MajorMiner experiments. Firstly, the covariance in MFCC features was strictly harmful in this prediction task, when used alongside the mean and variance. It has been argued that MFCC features are already relatively decorrelated compared with spectral magnitudes [9]. Perhaps when we normalize the MFCC covariance features we add features with a very low signal to noise ratio. Secondly, the LM features without covariance are poorer than a like number of MFCC features. This is similar to the simple experiment in the introduction that demonstrated the superior ability of cepstral features to generalize across pitch—that simple experiment corresponds to using the mean frame-level feature. Spectral features do not generalize as well across pitch without covariance information, but with covariance information they are better at it.

All the models were trained in just a few minutes on a desktop PC. Extracting features and training the model were fast enough that the decoding of the MP3 files was a significant part of the overall experiment time.

3.3 Tag Presence

We also tested spectral covariance features in a larger and sparser descriptor-based retrieval setting using the Magna Tagatune (Tagatune) dataset. Tagatune contains 25863 30-second audio clips [4]. Each clip is labeled with one or more binary attributes from a collection of 188 potential descriptors such as 'guitar', 'classical', 'no voices', 'world', 'mellow', 'blues', 'harpsicord', 'sitar'. These descriptors were collected from an online game in pairs of players use these words/phrases to determine whether they are listening to the same song or not. The descriptors generally refer to instrumentation, genre, and mood (see Table 4). Attributes in the dataset are binary and do not reflect the degree to which any attribute applies to a song. Furthermore, the nature of the data-collection game is such that the non-occurrence of an attribute is weak evidence that the attribute does not apply. There are many descriptors which might apply to a song, but no player thought to use them. Still, the Tagatune dataset provides a great deal of human-verified annotations and audio–despite these minor problems it is a great resource.

[2] MIREX 2009 Results:
http://www.music-ir.org/mirex/2009/index.php

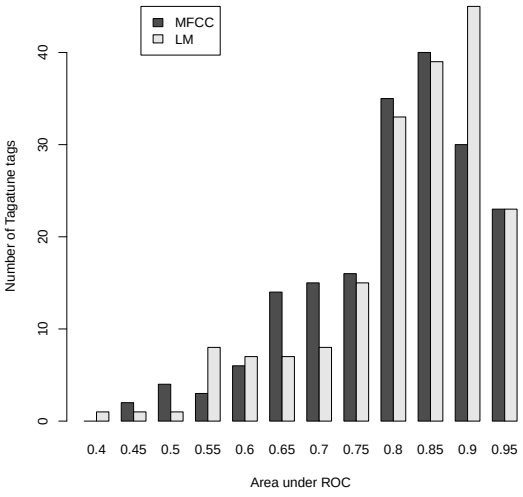

Figure 3. Histogram of the success of a linear classifier at predicting Tagatune attributes from LM (light) and MFCC (dark) means and covariance. AUC-ROC reflects both precision and recall; 0.5 is expected from a random predictor, and 1.0 from a perfect one. The MFCC features get 3 more tags 80% right, but the LM features get 16 more tags 90% right. Overall, LM features give better performance.

For the Tagatune dataset an independent logistic regression model was used to predict each potential attribute. Only logistic regression was used because the size of the dataset was prohibitive of the SVM's quadratic training algorithm. Song classification was performed by partitioning each clip into 3-second segments and voting over the segments, as in [1]. The dataset was partitioned into train, validation, and test sets by taking the 6^{th} of every 10 clips for validation, and by taking the 7^{th} and 8^{th} for testing. A plateau in validation-set accuracy (averaged across all tasks) was reached after about 20 minutes of CPU (wall) time, after about 10 passes through the training data.

The results of a linear classifier applied to the LM and MFCC mean and covariance feature are shown in Figure 3 and Table 4. Again, the LM features with covariance outperformed statistics of MFCC features. For the LM-based model, 140 of 188 descriptors were over .8 ROC and 68 of those are over .9. The MIREX 2009 Tagatune evaluation used a test protocol almost identical to ours, and found that the participants' average ROC scores ranged from 0.67 to 0.83. Our simple model appears actually to score slightly better than the MIREX 2009 participants did.

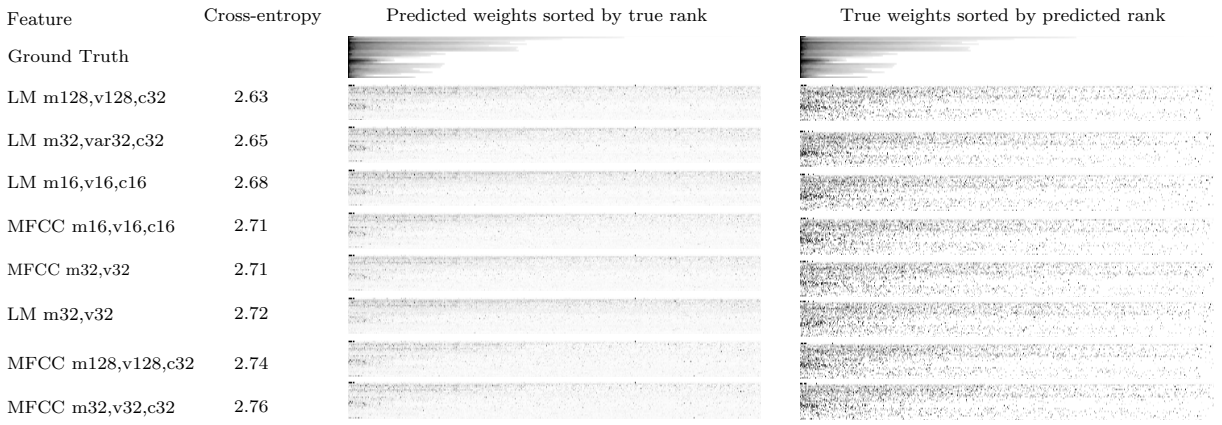

Figure 2. LM features outperform MFCC features for predicting tag distributions in the MajorMiner dataset. 'm32' (similarly 'm16','m128') denotes the mean in 32 coefficients, 'v32' the variance, and 'c32' the covariance (the upper triangle of the 32x32 matrix). When covariance information is used, LM features outperform MFCCs. Images in the middle column illustrate the recall of each model for each of the 25 most popular tags: the greater the density toward the left, the higher the recall of the model. Images in the right column illustrate the corresponding precision. The best model has AUC-ROC of 0.78.

4. DISCUSSION

Our covariance and correlation features help to summarize (in a time-independent way) the different kinds of timbres observed during a segment. The covariance (or correlation) in LM frame-level features summarizes the way that energy/loudness in different frequency bands co-occur in a signal. Recall that the timbre of a sustained instrument note can be crudely approximated by the shape of the spectral envelope. The difficulty in recognizing instruments in a segment after it has been summarized by the mean or variance is that when different instruments are present in one segment, the mean envelope can be indistinguishable from the mean that would come from other instrument combinations. The covariance is an improvement in this respect. If different instruments in a segment do not play simultaneously, then the covariance will encode activity corresponding to the (pairs of) peaks in the envelope of each instrument. To the extent that these instruments have timbres with different pairs of peaks, the instruments will not interfere with one another in the segment-level feature. Still, when instruments are played simultaneously (as often happens!) there is more interference.

4.1 MAP vs. ROC

Table 4 lists the mean average precision (MAP) and classification error rate (ERR) for some of the most popular and least popular attributes. The classification accuracies for most descriptors is quite close to the baseline of random guessing. That is because most descriptors are so rare that when a trained model is forced to take a hard classification decision, it is very difficult for the evidence from the song feature to outweigh the overwhelming prior that a rare descriptor does not apply. A similar finding is described in [2]. Since the attributes are rare, even the best models rank many negative-labeled examples also near the beginning of the clip collection, so precision and classification error are low. But the high rate of false-negative labels (see Table 3) biases the MAP and ERR measures more than the ROC. In the case of MagnaTagatune, where there are many false-negative labels (true instances, labeled as non-instances) MAP and ERR criteria are potentially very biased estimators of model performance. The ROC measure a more appropriate criterion in this context.

Attr	Count
female vocals	386
male vocals	465
female vocal	644
no vocal	995
male vocal	1002
no vocals	1158
vocals	1184
vocal	1729

Table 3. The number of applications of several Tagatune attributes over all 25863 clips. One of the difficulties with Tagatune is the frequency of false negative attributes in the dataset. For example, although vocal was applied 1729 times, the 'no vocal' attribute was applied only 995 times; 90% of the clips are labeled as neither 'vocal' nor 'no vocal'.

5. CONCLUSION

The covariance of log-magnitude Mel frequency scale *spectral* coefficients (LM features) offer a superior alternative to statistics of MFCCs when summarizing frame-level audio features for genre and tag prediction. We have demonstrated the advantage of our LM features on three standard genre and tag prediction datasets: GTZAN, MajorMiner, and Tagatune. Furthermore, these features make state of the art performance available with just a linear classifier (such as L1- or L2-regularized logistic regression, or lin-

Attr	Freq	ERR	MAP	ROC
guitar	0.187	0.159	0.62	0.85
classical	0.169	0.157	0.63	0.91
slow	0.135	0.135	0.32	0.77
techno	0.110	0.090	0.58	0.92
strings	0.110	0.110	0.44	0.87
drums	0.102	0.102	0.34	0.84
electronic	0.096	0.097	0.33	0.84
rock	0.093	0.057	0.71	0.96
fast	0.093	0.093	0.31	0.79
piano	0.077	0.074	0.54	0.84
repetitive	0.001	0.001	0.15	0.77
scary	0.001	0.001	0.02	0.97
woodwind	0.001	0.001	0.01	0.82
viola	0.001	0.001	0.08	0.95
quick	0.001	0.001	0.00	0.56
soprano	0.001	0.001	0.17	0.97
horns	0.001	0.001	0.00	0.68
soft rock	0.001	0.001	0.00	0.70
monks	0.001	0.001	0.31	0.99
clasical	0.001	0.001	0.00	0.83
happy	0.001	0.001	0.00	0.61

Table 4. Test set performance of best model on the 10 most popular (above) and least popular (below) descriptors in Tagatune. **Freq** is application frequency, **ERR** is classification error, **MAP** is mean average precision, and **ROC** is the area under the ROC.

ear SVM). Our results on the GTZAN dataset suggest that RBF SVMs may offer slightly better performance when time allows for training.

The LM features are straightforward to implement, computationally cheap, and the use of a linear classifier makes our model viable on any size of genre or tag-prediction dataset. We believe that the model presented here has great potential for working with industrial-scale audio datasets.

Finally, the results presented here demonstrate that the discrete cosine transform (or inverse Fourier transform) responsible for the cepstrum's deconvolution property actually hinders performance in some circumstances. One explanation of this behavior is that our model learns a better deconvolution-like transform of the spectral data than is provided by the cepstrum. We admit that this is only one possible explanation of these results and that further analysis is necessary in order to make conclusions. We believe that this is one fruitful direction for future research.

6. REFERENCES

[1] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kegl. Aggregate features and AdaBoost for music classification. *Machine Learning*, 65:473–484, 2006.

[2] T. Bertin-Mahieux, D. Eck, F. Maillet, and P. Lamere. Autotagger: A model for predicting social tags from acoustic features on large music databases. *Journal of New Music Research*, 37(2):115–135, 2008.

[3] Thibault Langlois and Gonalo Marques. A music classification method based on timbral features. In *ISMIR*, Kobe, Japan, October 2009.

[4] Edith L. M. Law and Luis von Ahn. Input-agreement: A new mechanism for data collection using human computation games. In *CHI*, pages 1197–1206, 2009.

[5] Tao Li and George Tzanetakis. Factors in automatic musical genre classification. In *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics*, New Paltz, New York, 2003.

[6] Beth Logan. Mel frequency cepstral coefficients for music modeling. In *ISMIR*, Plymouth, Mass., October 2000.

[7] M. Mandel and D. Ellis. A web-based game for collecting music metadata. *J. New Music Research*, 37(2):151–165, 2008.

[8] Michael I. Mandel and Daniel P.W. Ellis. Song-level features and support vector machines for music classification. In *ISMIR*, pages 594–599, London, UK, September 2005.

[9] P. Mermelstein. Distance measures for speech recognition, psychological and instrumental. *Pattern Recognition and Artificial Intelligence*, pages 374–388, 1976.

[10] Yannis Panagakis, Constantine Kotropoulos, and Gonzalo R. Arce. Music genre classification using locality preserving non-negative tensor factorization and sparse representations. In *ISMIR*, Kobe, Japan, October 2009.

[11] Douglas Turnbull and Charles Elkan. Fast recognition of musical genres using RBF networks. *IEEE Transactions on Knowledge and Data Engineering*, 17(4):1–5, April 2005.

[12] G. Tzanetakis, G. Essl, and P. Cook. Automatic musical genre classification of audio signals. In *ISMIR*, Bloomington, Indiana, October 2001.

[13] George Tzanetakis and Perry Cook. Musical genre classification of audio signals. *IEEE Transactions on Speech and Audio Processing*, 10(5):293–302, July 2002.

[14] Kris West and Stephen Cox. Finding an optimal segmentation for audio genre classification. In *ISMIR*, London, UK, October 2005.