On music genre classification via compressive sampling

Sturm, Bob L.

Published in:
International Conference on Multimedia and Expo

Publication date:
2013

Document Version
Early version, also known as pre-print

ON MUSIC GENRE CLASSIFICATION VIA COMPRESSIVE SAMPLING

Bob L. Sturm

Audio Analysis Lab, Dept. Architecture, Design and Media Technology
Aalborg University Copenhagen, A.C. Meyers Vænge 15, DK-2450
bst@create.aau.dk

ABSTRACT

Recent work [1] combines low-level acoustic features and random projection (referred to as "compressed sensing" in [1]) to create a music genre classification system showing an accuracy among the highest reported for a benchmark dataset. This not only contradicts previous findings that suggest low-level features are inadequate for addressing high-level musical problems, but also that a random projection of features can improve classification. We reproduce this work and resolve these contradictions.

Index Terms— Music genre classification, sparse approximation, random projection, compressive sampling

1. INTRODUCTION

We address the confusion arising from findings in music genre classification (MGR) by Chang et al. [1] that contradict established findings. First, their results (Fig. 4 in [1]) suggest that the random projection of features can make them more discriminative than the original features. Several works in machine learning research, however, observe random projection can, *at best*, lower computational load while not significantly hurting discriminability [2–6]. Second, the results of Chang et al. suggest that low-level features of sampled musical audio are imbued with musical meaning by a transformation devoid of musical principles. Several works in music information research, however, observe that low-level features do not effectively address a problem such as MGR, e.g., [7–11].

The MGR system of Chang et al. [1] is quite similar to that of Panagakis et al. [12–14], and we see a close agreement between their classification accuracies on the benchmark dataset GTZAN [15, 16]. Both use sparse representation classification (SRC) [6], but while Panagakis et al. use as features frequency-modulation rates from half-minute spectrograms, Chang et al. use random projections of the statistics of low-level features over 3 s. However, the classification accuracies reported in [12–14] arise from a systematic mistake in evaluation,[1] inflating classification accuracies in GTZAN from 60-70% [17] to over 90%. While we have shown [10, 11] that the system of Panagakis et al. can be altered to produce classification accuracies above 80% in GTZAN, and that evaluations of MGR systems using features similar in nature to those of Panagakis et al. find similar performance in GTZAN [11, 18], this is still 10 points below those reported by Chang et al.

MGR systems using features similar to those used by Chang et al. [1], but without random projection, and taking other approaches to machine learning, appear to perform inferior to the 92.7% accuracy in [1]. The works from which Chang et al. take their features [15, 19–21] all report accuracies far below 92.7%, but for different datasets. The system appearing to perform closest to that of Chang et al. in GTZAN is that of Bergstra et al. [22]. Using some of the same features as Chang et al., but a different classifier and no random projection, its accuracies in GTZAN are reportedly 78–83% [10].

At least four works [23–26] make comparisons to the results of Chang et al. [1]; but only Aryafar et al. [23] reproduces a part of the system in [1]. Using SRC and a small subset of features used by Chang et al. without random projection, they report a classification accuracy of 30.47% for the HOMBURG dataset [27]. For the same dataset, Homburg et al. [27] report a mean classification accuracy of 53.23% using a k-nearest neighbor classifier and low-level features [28].

In light of all this work then, three practical questions arise: 1) is our work in [10, 11, 17] incorrect?; 2) are the results of Chang et al. [1] incorrect?; and 3) if these results are correct, how does the random projection of low-level features, which demonstrably lack high-level musical information, significantly boost classification accuracy in GTZAN? We answer these questions by first reproducing the system of Chang et al., which we detail in the next section. In the third section, we present our experimental results, and compare them with those of Chang et al. We make available all code to reproduce the figures and results contained herein: http://imi.aau.dk/~bst.

2. MGR VIA COMPRESSIVE SAMPLING

Two significant hurdles to reproducing the results of Chang et al. [1], are implementing the feature extraction and the classifier. We detail our attempts at these here, and make clear with justification the numerous assumptions we must make.

BLS is supported in part by Independent Postdoc Grant 11-105218 from Det Frie Forskningsråd; and the Danish Council for Strategic Research of the Danish Agency for Science Technology and Innovation in the CoSound project, case number 11-115328.

[1] Personal communication with Y. Panagakis.

2.1. Feature Extraction

The system of Chang et al. [1] (SRCRP) uses six short-term features: octave-based spectral contrast (OSC) [19]; Mel-frequency cepstral coefficients (MFCCs); spectral centroid, rolloff, and flux; zero-crossings [15]; and four long-term features: octave-based modulation spectral contrast (OMSC) [20]: "low-energy" [15]; modulation spectral flatness measure (MSFM); and modulation spectral crest measure (MSCM) [21]. Chang et al. imply they compute features from analysis frames of 93 ms (they do not specify the shape of analysis window, so we assume a Hann window), translated half their duration along a 30 s excerpt. Then, they compute "the statistics" of the short-term features in 3 s texture windows. We assume this means SRCRP disjointly partitions the set of analysis frames into subsets of 63 consecutive frames, i.e., the texture window of 3 s. Since Chang et al. describe computing the mean and variance of "the 13-dimensional MFCCs," we assume "the statistics" for the other short-term features are mean and variance as well. From [15, 20, 21], we assume SRCRP computes long-term features by processing the analysis frames, or by comparing the statistics of the analysis frames to the texture window of sound. We now explicitly describe how we compute each feature.

Although MFCCs are widely used in audio signal processing, there are often differences between implementations [29], and which coefficients are used. Chang et al. [1] do not specify how they compute their MFCCs, and which 13 coefficients they keep. Thus, we compute for each frame 40 MFCCs using [30], and retain the first thirteen including the zeroth coefficient. For each coefficient, we compute its mean and variance in a texture window starting at frame index t, thus producing the feature $\mathrm{MFCC}(t)$ having 26 dimensions.

The zero-crossings of a frame is the number of times the time-domain signal amplitude changes sign. For the spectral centroid of the frame at index t, we compute its magnitude Fourier spectrum by a length-F DFT, and keep the magnitudes of the positive frequencies to create $X(\omega;t)$, where $\omega \in \mathcal{F} := \{n/(F/2) : n = 0, 1, \ldots F/2\}$. We normalize $X(\omega;t)$ such that it sums to one, to produce the probability mass function $p_X(\omega;t)$ and cumulative distribution function $P_X(\omega;t)$. In terms of these, we compute the spectral centroid by expectation $E[\omega]$, and the spectral rolloff is $\arg\min_\omega P_X(\omega;t) \geq 0.85$. We compute the spectral flux of consecutive frames by $\|X(\omega;t) - X(\omega;t-1)\|_2/\sqrt{F/2}$, where $\|\cdot\|_2$ is the Euclidean length. Over a texture window, we compute the mean and variance of each feature, producing features of 2 dimensions each, e.g., starting at index t: $\mathrm{zcr}(t), \mathrm{sc}(t), \mathrm{sr}(t), \mathrm{sf}(t)$. For the spectral flux of the first texture window, we ignore the spectral flux of the first frame.

The remaining short-term feature Chang et al. [1] compute is OSC, defined by Jiang et al. [19]. We compute this feature by partitioning the magnitude spectrum of a frame into low- and high-pass bands, and six octave-width frequency bands in-between. First, define $\delta_{\mathcal{W}}(\omega)$ as 1 if $\omega \in \mathcal{W}$, and zero otherwise; and define the following sets:

$$\mathcal{W}_0 := \mathcal{F} \cap [0, 100/22050) \tag{1}$$

$$\mathcal{W}_6 := \mathcal{F} \cap [3200/22050, 8000/22050) \tag{2}$$

$$\mathcal{W}_7 := \mathcal{F} \cap [8000/22050, 1) \tag{3}$$

$$\mathcal{W}_k := \mathcal{F} \cap 100/22050 \cdot [2^{k-1}, 2^k), k \in \{1, \ldots, 5\}. \tag{4}$$

With these, we define the partitioned spectrum of a frame as $f_k(\omega;t) := X(\omega;t)\delta_{\mathcal{W}_k}(\omega), k \in \mathcal{K} = \{1, \ldots, 8\}$. Define the set of frequencies Ω_k ordering from largest to smallest the magnitudes of the kth band, i.e., $f_k(\Omega_k(l);t) \geq f_k(\Omega_k(l+1);t)$ for $l \in \{1, \ldots, |\Omega_k| - 1\}$; and the set $\mho_k$ that orders from smallest to largest the non-zero values in the same band. Jiang et al. define the OSC of the kth band

$$\mathrm{OSC}(k;t) := \mathrm{peak}(k;t) - \mathrm{valley}(k;t) \tag{5}$$

where

$$\mathrm{peak}(k;t) := \log\left[\frac{1}{\lceil\alpha|\Omega_k|\rceil}\sum_{l=1}^{\lceil\alpha|\Omega_k|\rceil} f_k(\Omega_k(l);t)\right] \tag{6}$$

$$\mathrm{valley}(k;t) := \log\left[\frac{1}{\lceil\alpha|\Omega_k|\rceil}\sum_{l=1}^{\lceil\alpha|\Omega_k|\rceil} f_k(\mho_k(l);t)\right] \tag{7}$$

and the parameter $\alpha \in [|\Omega_k|^{-1}, 1)$ determines how many of the largest and smallest values of each band are considered.

Chang et al. [1] do not state the value of α they use; however, Jiang et al. [19] claim, with respect to music classification performance, they find little difference for $\alpha \in [0.02, 0.2]$. Chang et al. use a sampling rate 44.1 kHz, and so a 93 ms analysis frame consists of 4101 samples. Since Chang et al. do not state the size of the DFT they use to compute $X(\omega;t)$, we zeropad each frame to length $F = 8192$, which means for us $|\mathcal{W}_k| \geq 19$, and $\alpha > 0.053$. We thus set $\alpha = 0.2$. Finally, Chang et al. only mention that the dimension of the OSC feature is 32. We assume that, like Jiang et al., Chang et al. create a feature for each frame using the 8 values of $\mathrm{OSC}(k;t)$ and the 8 values of $\mathrm{valley}(k;t)$, and then compute the mean and variance of the dimensions for 63 consecutive frames, creating a 32-dimensional feature.

The "low-energy" feature appears verbatim in [1,21]: "the percentage of analysis windows that have energy less than the average energy across the texture window;" however, "energy" and "average energy" are not clearly defined. Tzanetakis and Cook [15] define this feature "as the percentage of analysis windows that have less RMS energy than the average RMS energy across the texture window." We assume "average RMS energy" is the same as "RMS energy." Since the RMS energy of a discrete set of N time-domain samples y is defined $RMS(y) := \|y\|_2/\sqrt{N}$, we compute the percentage of 63 consecutive RMS frames energies that are below the RMS energy of the associated 3 s portion of the signal. This creates the scalar $\mathrm{le}(t)$.

The OMSC feature is described by Lee et al. [20], but in a way inconsistent with that of Chang et al. [1]. While Lee et al. define this feature over all texture windows, Chang et al. appear to not compute any feature beyond the length of a texture window. Hence, we proceed as follows. We first build the octave-resolution magnitude spectrogram starting at t

$$\mathbf{X}_t^{(p)} := \begin{bmatrix} \|f_0(\omega;t)\|_p^p & \cdots & \|f_8(\omega;t)\|_p^p \\ \|f_0(\omega;t+1)\|_p^p & \cdots & \|f_8(\omega;t+1)\|_p^p \\ \vdots & \ddots & \vdots \\ \|f_0(\omega;t+62)\|_p^p & \cdots & \|f_8(\omega;t+62)\|_p^p \end{bmatrix} \quad (8)$$

where time increases with row, and band increases with column. We the apply from the left the appropriately sized DFT matrix $\mathbf{W}$, which creates the amplitude modulation spectrum: $\mathbf{M}_t^{(1)} := |[\mathbf{I}|\mathbf{0}]\mathbf{W}^H\mathbf{X}_t^{(1)}|$, where $[\mathbf{I}|\mathbf{0}]$ retains only the positive frequencies. The first column of $\mathbf{M}_t^{(1)}$ is the amplitude spectrum of the variations in the sum magnitudes of the first band across the 63 frames of the texture window. We then disjointly partition $\mathbf{M}_t^{(1)}$ into J equal-sized sets of rows, where we define $\mathbf{P}_j\mathbf{M}_t^{(1)}$ to retain the jth set of rows of $\mathbf{M}_t^{(1)}$, $j \in \mathcal{J} = \{1,\ldots,J\}$. Finally, we compute $\mathrm{OMSC}(j;t) := \mathrm{peak}(\mathbf{P}_j\mathbf{M}_t^{(1)}) - \mathrm{valley}(\mathbf{P}_j\mathbf{M}_t^{(1)})$, where $\mathrm{peak}(\mathbf{B})$ is the log of the maximum element in $\mathbf{B}$, and $\mathrm{valley}(\mathbf{B})$ is the log of the minimum element in $\mathbf{B}$. While Chang et al. do not mention this, we append zeros to each column of $\mathbf{X}_t^{(1)}$ such that it has 512 rows. Thus, $\mathbf{W}$ is 512-square, and $\mathbf{M}_t^{(1)}$ is 257×8.

Chang et al. [1] state their OMSC feature has 32 dimensions, but do not make clear what dimensions those are. Lee et al. [20] compute $\{\mathrm{OMSC}(j;t), \mathrm{valley}(\mathbf{P}_j\mathbf{M}_t^{(1)})\}$ for all texture windows over a music excerpt, and then build a feature vector for the entire excerpt by computing the mean and variance of each modulation band across all texture windows. If Chang et al. do this, then some features come from only 3 s, while others come from 30 s. Since Chang et al. do not describe computing the statistics for an excerpt from the statistics of the texture windows, we assume that all features come from windows no longer than 3 s. Hence, we assume SRCRP retains both $\{\mathrm{OMSC}(j;t)\}$ and $\{\mathrm{valley}(\mathbf{P}_j\mathbf{M}_t^{(1)})\}$, and uses the entire modulation bandwidth. We thus set $J = 16$ so that these features together have 32 dimensions.

The MSFM and MSCM features are both defined in [21]. Though Chang et al. [1] mention using both of these features, only MSFM appears in their table listing the features they use. We assume that they use both. Here, we evaluate the DFT of $\mathbf{X}_t^{(2)}$, and keep the positive modulation frequencies: $\mathbf{M}_t^{(2)} := |[\mathbf{I}|\mathbf{0}]\mathbf{W}^H\mathbf{X}_t^{(2)}|$. Then, for each band $k \in \mathcal{K}$

$$\mathrm{MSFM}(k;t) := \frac{\mathrm{geom}(\mathbf{M}_t^{(2)}\mathbf{e}_k)}{\mathrm{mean}(\mathbf{M}_t^{(2)}\mathbf{e}_k)} \quad (9)$$

$$\mathrm{MSCM}(k;t) := \frac{\|\mathbf{M}_t^{(2)}\mathbf{e}_k\|_\infty}{\mathrm{mean}(\mathbf{M}_t^{(2)}\mathbf{e}_k)} \quad (10)$$

where $\mathrm{geom}(\mathbf{y})$ is the geometric mean of the elements of $\mathbf{y}$, $\mathrm{mean}(\mathbf{y})$ is the arithmetic mean, and $\mathbf{e}_k$ is the k standard vector. Though Chang et al. do not mention this, we append zeros to the columns of $\mathbf{X}_t^{(2)}$ such that it has 512 rows.

To conclude, from a texture window that begins at index t, we create by concatenation the 115-dimensional feature:

$$\begin{aligned}\mathbf{v}_t := \Big\{&\mathrm{MFCC}(t), \mathrm{zcr}(t), \mathrm{sc}(t), \mathrm{sr}(t), \mathrm{sf}(t), \mathrm{le}(t), \\ &\{\mathrm{OMSC}(j;t), \mathrm{valley}(\mathbf{P}_j\mathbf{M}_t^{(1)})\}_{j\in\mathcal{J}}, \{\mathrm{OSC}(k;t), \\ &\mathrm{valley}(k;t), \mathrm{MSFM}(k;t), \mathrm{MSCM}(k;t)\}_{k\in\mathcal{K}}\Big\} \quad (11)\end{aligned}$$

For a music excerpt longer than 3 s, we create a feature vector for each disjoint 3 second segment. For the 30 s excerpts of GTZAN, this produces 10 feature vectors for each excerpt.

2.2. Classification

SRCRP classifies an observation $\mathbf{v}$ by SRC [6], which is a non-parametric method for machine learning, and was first applied to MGR by Pangakis et al. [12]. SRCRP first creates a "dictionary" matrix of N features computed from training signals in $|\mathcal{C}|$ classes, $\mathbf{D} := [\mathbf{V}_1|\mathbf{V}_2|\cdots|\mathbf{V}_{|\mathcal{C}|}]$, where the columns of the matrix $\mathbf{V}_c$ are from the set of class-c unit-norm feature vectors $\{\mathbf{v}_t/\|\mathbf{v}_t\|_2 : t \in \mathcal{T}_c\}$. Chang et al. [1] do not say how they construct $\mathbf{D}$, but we assume they use all 10 feature vectors from every excerpt in the training set, i.e., $|\mathcal{T}_c| = 10$ for all classes.

With the dictionary built, SRCRP then solves for a $\mathbf{v}$

$$\arg\min_{\mathbf{s}} \|\mathbf{s}\|_1 \text{ subject to } \mathbf{\Phi v} = \mathbf{\Phi D \Sigma s} \quad (12)$$

where every element of $\mathbf{\Phi} \in \mathbb{R}^{m\times 115}$ is iid Normal, and $\mathbf{\Sigma}$ is diagonal and defined such that each column of $\mathbf{\Phi D \Sigma}$ has unit ℓ_2-norm. The role of $\mathbf{\Phi}$ here is exactly that of random projection, which has been applied in machine learning before [2–5]. Chang et al. [1] refer to this as "compressive sampling" [31], which is not entirely appropriate. According to the theory of compressive sampling [31], the solution to (12) is guaranteed with high probability to be the "true solution," *as long as* $\mathbf{\Phi D \Sigma}$ satisfies some special conditions, and the sparsity s of the "true solution" (its number of non-zero elements) is less than $Qm[\log(115/s)]^{-1}$ for some scalar Q depending on $\mathbf{\Phi D \Sigma}$. For our problem, however, we make no assumption of there being a "true solution," and furthermore we are concerned with efficient signal description, not signal acquisition — the application domain of compressed sensing.

From the solution to (12), SRCRP builds a set of class-restricted weights $\{\mathbf{s}_c\}_{c\in\mathcal{C}}$ defined by

$$[\mathbf{s}_c]_i = \begin{cases} [\mathbf{s}]_i, & i \in \mathcal{I}(\mathcal{T}_c) \\ 0, & \text{else} \end{cases} \quad (13)$$

where $[\mathbf{s}]_i$ is the ith row of $\mathbf{s}$, and $\mathcal{I}(\mathcal{T}_c)$ indexes the columns of $\mathbf{D}$ from class c. Hence, $\mathbf{s}_2$ are the weights in $\mathbf{s}$ related to the

training features from class 2. Finally, SRCRP selects a class for the texture window by a minimum distortion criterion:

$$\hat{c} = \arg\min_{c \in \mathcal{C}} \|\mathbf{\Phi v} - \mathbf{\Phi D \Sigma s}_c\|_2. \quad (14)$$

Chang et al. do not say how SRCRP classifies music *excerpts* instead of texture windows, e.g., voting methods [10, 11, 21]. We make further assumptions about this below.

Finally, Chang et al. [1] do not explicitly specify m. They show in one table that their feature dimension is 64, and write in another table, "The sampling rate takes 67%", which would make $m = 77$ (assuming that is to what they refer). Hence, we define $m = 64$.

3. EXPERIMENTS

We now present our experimental results, and compare them to those of Chang et al. [1]. Since we wish to validate their results, we use the same dataset — GTZAN [15] — though it is now known to have serious faults [16]. Chang et al. mention setting up their "experimental parameters to be as close as possible to those used in [32]. In particular, the recognition rate is obtained from 10-fold cross validation." We find no overlap between the work of Chang et al. and of Sainath et al. [32] — who do not apply SRC, address musical signals, or use 10-fold cross validation (10fCV). Nonetheless, we use 10fCV for training, and sample $\mathbf{\Phi}$ anew in each fold.

We might assume the experimental results of Chang et al. [1] come from training and testing using only one 3 s texture window from each 30 s music excerpt; however, since more data for training typically leads to better performance, we create $\mathbf{D}$ from all 10 observations of each excerpt of the training set, and test on only one randomly-selected observation from each excerpt of the testing set. The final problem to address is how to solve (12). Chang et al. do not state the methods of convex optimization [33] they use. We use the SPGL1 solver [34] since in our work [10, 11, 17] we find it works well with reasonable efficiency. We set the maximum iterations to find a solution to 200.

We first validate that our implementation of SRCRP is working by testing it on the USPS handwritten digits dataset [35]. We use 10fCV, and test random projections to subspaces of various dimensions. The features are length-256 vectors of pixel values. Figure 1(a) shows the results for randomly projected features, which agree closely with those of Dasgupta [2] for the same dataset and random projection. We find by a McNemar's test on the contingency table of every pair of algorithms that only for $m = 141$ and $m = 256$ there is no significant difference (statistical significance $\alpha = 0.05$). We are thus satisfied that our implementation of SRCRP is working as expected. We also see that the classification accuracy is highest without random projection, i.e., for $m = 256$.

We next test whether our music features are discriminative for GTZAN. Using the same experimental design above, we test a quadratic discriminate classifier (QDC) [36] using

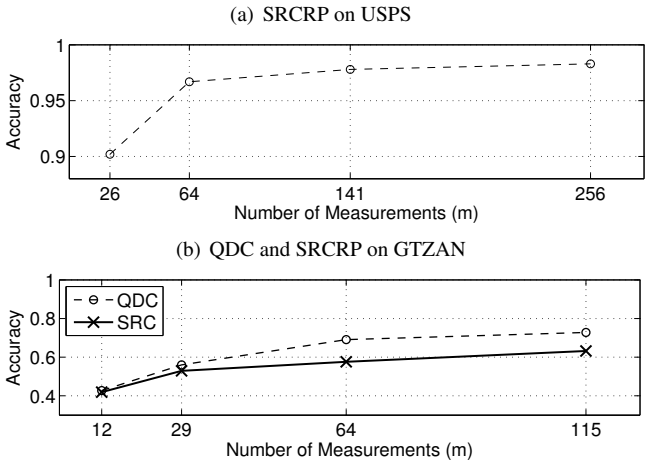

Fig. 1. Accuracies of classifiers for two datasets using various amounts of dimension reduction by random projection.

as training data the columns of $\mathbf{\Phi D \Sigma}$, and making unit norm each test feature $\mathbf{\Phi v}/\|\mathbf{\Phi v}\|_2$. Figure 1(b) shows an accuracy 0.69 when $m = 64$. This increases to 0.73 without random projection. With a McNemar's test, we find significant differences in performance between every pair of m. We are thus satisfied that our features are discriminative.

Figure 1(b) shows the classification accuracies of SRCRP in GTZAN using several m. For $m = 64$, which we assume was used by Chang et al. [1], we find SRCRP obtains a classification accuracy 0.576, which is far below the 0.927 reported. Furthermore, we find with a McNemar's test that QDC performs significantly better than SRCRP at this m, as well as at $m = 115$.

Figure 2 shows several figures of merit from our tests of SRCRP at these two m. These confusion behaviors are relatively comparable — though "worse" — with those of the SRC system we test in [10, 11]. Unlike in [1], our confusion table is not perfectly symmetric. While we find that SRCRP misclassifies Rock observations most often as Metal and Country, Chang et al. find SRCRP misclassifies Rock most often as Reggae and Pop. While we see SRCRP misclassifies Country observations rarely as Disco, Chang et al. find SRCRP misclassifies Country most often as Disco. Furthermore, we see SRCRP misclassifies Disco observations most often as Pop and Hip hop, but Chang et al. find SRCRP never misclassifies Disco as those. Compared to Fig. 2(b), we see random projection does not help many of the figures of merit.

4. CONCLUSION

We have reproduced, to the furthest extent possible, the MGR system SRCRP described by Chang et al. [1], as well as the attendant references [15, 19–21]. We make clear the considerable number of assumptions we have had to make, and have provided justifications for our decisions. Our experiments show that our implementation of SRC is working cor-

(a) $m = 64$

	bl	cl	co	di	hi	ja	me	po	re	ro	Pr
bl	55	0	9	6	2	9	0	2	5	9	58.76
cl	1	88	0	2	1	8	0	0	2	2	85.71
co	5	3	48	4	0	6	2	4	7	10	56.52
di	3	0	5	29	4	1	1	3	1	3	54.45
hi	9	0	0	12	63	0	3	12	18	3	55.16
ja	2	6	6	1	0	70	3	4	5	4	70.89
me	13	1	8	14	7	3	83	2	1	33	52.29
po	4	0	4	13	14	1	0	65	11	3	56.16
re	4	0	2	12	5	1	1	3	45	3	58.84
ro	4	2	18	7	4	1	7	5	5	30	36.89
F	55.87	86.07	51.09	36.91	58.05	69.82	63.40	59.96	50.35	32.47	57.60

(b) $m = 115$ (no random projection)

	bl	cl	co	di	hi	ja	me	po	re	ro	Pr
bl	62	0	9	3	5	4	2	0	10	7	60.74
cl	1	91	2	2	2	10	0	0	2	1	82.83
co	9	2	53	2	1	8	2	6	1	16	53.19
di	0	0	3	47	2	1	1	5	5	1	74.19
hi	3	0	1	11	65	2	3	8	14	3	60.94
ja	5	3	6	4	1	64	2	4	1	2	70.63
me	9	0	9	5	4	4	84	4	3	22	58.88
po	2	0	3	11	14	3	1	70	9	4	62.23
re	4	0	2	3	4	2	0	0	53	1	76.25
ro	5	4	12	12	2	2	5	3	2	43	51.60
F	60.90	86.29	52.20	55.17	61.94	65.62	68.98	64.72	62.06	45.05	63.20

Fig. 2. Confusions of SRCRP in GTZAN. Columns: "true" genres, with mean precision (Pr $\times 100$) shown in last column. Rows: predicted genres, with mean F-measure (F $\times 100$) in last row. Bottom right corner is classification accuracy. Classes: Blues (bl), Classical (cl), Country (co), Disco (di), Hip hop (hi), Jazz (ja), Metal (me), Pop (po), Reggae (re), Rock (ro).

rectly, and that the features we extract are indeed discriminative. Furthermore, we see that a simple classifier and the same features performs significantly better than SRCRP solving the high-computation problem in (12).

In conclusion, since we are unable to reproduce the high results reported by Chang et al. [1], and since our results comport with findings that are challenged by their results [2–11], the contradictions arising from their work no longer appear real. Our results and attendant code clearly show that low-level features of sampled musical audio are not magically imbued with high-level musical meaning just from projecting them randomly onto even lower-dimensional subspaces.

5. REFERENCES

[1] K. Chang, J.-S. R. Jang, and C. S. Iliopoulos, "Music genre classification via compressive sampling," in *Proc. ISMIR*, Amsterdam, The Netherlands, Aug. 2010, pp. 387–392.

[2] S. Dasgupta, "Experiments with random projection," in *Proc. Conf. Uncertainty in Artificial Intelligence*, Stanford, CA, USA, June 2000, pp. 143–151.

[3] E. Bingham and H. Mannila, "Random projection in dimensionality reduction: Application to image and text data," in *Proc. Int. Conf. Knowledge Discovery Data Mining*, San Francisco, CA, Aug. 2001, pp. 245–250.

[4] D. Fradkin and D. Madigan, "Experiments with random projections for machine learning," in *Proc. SIGKDD*, Washington, DC, USA, Aug. 2003.

[5] C. Hegde, M. Davenport, M. B. Wakin, and R. G. Baraniuk, "Efficient machine learning using random projections," in *Proc. Neural Info. Process. Syst.*, Dec. 2007.

[6] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[7] J.-J. Aucouturier, F. Pachet, P. Roy, and A. Beurivé, "Signal + context = better classification," in *ISMIR*, 2007, pp. 425–430.

[8] G. Marques, M. Lopes, M. Sordo, T. Langlois, and F. Gouyon, "Additional evidence that common low-level features of individual audio frames are not representative of music genres," in *Proc. SMC*, Barcelona, Spain, July 2010.

[9] G. Marques, T. Langlois, F. Gouyon, M. Lopes, and M. Sordo, "Short-term feature space and music genre classification," *J. New Music Research*, vol. 40, no. 2, pp. 127–137, 2011.

[10] B. L. Sturm, "Two systems for automatic music genre recognition: What are they really recognizing?," in *Proc. ACM MIRUM Workshop*, Nara, Japan, Nov. 2012.

[11] B. L. Sturm, "Classification accuracy is not enough: On the evaluation of music genre recognition systems," *J. Intell. Info. Systems (accepted)*, 2013.

[12] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music genre classification via sparse representations of auditory temporal modulations," in *Proc. EUSIPCO*, Aug. 2009.

[13] Y. Panagakis, C. Kotropoulos, and G. R. Arce, "Music genre classification using locality preserving non-negative tensor factorization and sparse representations," in *Proc. ISMIR*, Kobe, Japan, Oct. 2009, pp. 249–254.

[14] Y. Panagakis and C. Kotropoulos, "Music genre classification via topology preserving non-negative tensor factorization and sparse representations," in *Proc. ICASSP*, Mar. 2010, pp. 249–252.

[15] G. Tzanetakis and P. Cook, "Musical genre classification of audio signals," *IEEE Trans. Speech Audio Process.*, vol. 10, no. 5, pp. 293–302, July 2002.

[16] B. L. Sturm, "An analysis of the GTZAN music genre dataset," in *Proc. ACM MIRUM Workshop*, Nara, Japan, Nov. 2012.

[17] B. L. Sturm and P. Noorzad, "On automatic music genre recognition by sparse representation classification using auditory temporal modulations," in *Proc. CMMR*, London, UK, June 2012.

[18] Joakim Andén and Stéphane Mallat, "Multiscale scattering for audio classification," in *Proc. ISMIR*, 2011, pp. 657–662.

[19] D.-N. Jiang, L.-Lu, H.-J. Zhang, J.-H. Tao, and L.-H. Cai, "Music type classification by spectral contrast features," in *Proc. ICME*, 2002.

[20] C.-H. Lee, J.-L. Shih, K.-M. Yu, and J.-M. Su, "Automatic music genre classification using modulation spectral contrast feature," in *Proc. ICME*, 2007.

[21] D. Jang, M. Jin, and C. D. Yoo, "Music genre classification using novel features and a weighted voting method," in *Proc. ICME*, 2008, pp. 1377–1380.

[22] J. Bergstra, N. Casagrande, D. Erhan, D. Eck, and B. Kégl, "Aggregate features and AdaBoost for music classification," *Machine Learning*, vol. 65, no. 2-3, pp. 473–484, June 2006.

[23] K. Aryafar, S. Jafarpour, and A. Shokoufandeh, "Music genre classification using sparsity-eager support vector machines," Tech. Rep., Drexel University, 2012.

[24] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun, "Unsupervised learning of sparse features for scalable audio classification," in *Proc. ISMIR*, Miami, FL, Oct. 2011.

[25] Jan Wülfing and Martin Riedmiller, "Unsupervised learning of local features for music classification," in *Proc. ISMIR*, Porto, Portugal, Oct. 2012.

[26] C.-C. M. Yeh and Y.-H. Yang, "Supervised dictionary learning for music genre classification," in *Proc. ACM Int. Conf. Multimedia Retrieval*, Hong Kong, China, Jun. 2012.

[27] H. Homburg, I. Mierswa, B. Möller, K. Morik, and M. Wurst, "A benchmark dataset for audio classification and clustering," in *Proc. ISMIR*, London, U.K., 2005.

[28] I. Mierswa and K. Morik, "Automatic feature extraction for classifying audio data," *Machine Learning*, vol. 58, no. 2-3, pp. 127–149, Feb. 2005.

[29] T. Ganchev, N. Fakotakis, and G. Kokkinakis, "Comparative evaluation of various mfcc implementations on the speaker verification task," in *Proc. Int. Conf. Speech Comp.*, Patras, Greece, Oct. 2005, vol. 1, pp. 191–194.

[30] M. Slaney, "Auditory toolbox," Tech. Rep., Interval Research Corporation, 1998.

[31] E. Candès and M. B. Wakin, "An introduction to compressive sampling," *IEEE Signal Process. Mag.*, vol. 25, no. 2, pp. 21–30, Mar. 2008.

[32] T. N. Sainath, A. Carmi, D. Kanevsky, and B. Ramabhadran, "Bayesian compressive sensing for phonetic classification," in *Proc. ICASSP*, 2010.

[33] S. Boyd and L. Vandenberghe, *Convex Optimization*, Cambridge University Press, Cambridge, UK, 2004.

[34] E. van den Berg and M. P. Friedlander, "Probing the Pareto frontier for basis pursuit solutions," *SIAM J. on Scientific Computing*, vol. 31, no. 2, pp. 890–912, Nov. 2008.

[35] J. J. Hull, "A database for handwritten text recognition research," *IEEE Pattern Anal. Machine Intell.*, vol. 16, no. 5, pp. 550–554, May 1994.

[36] Robert P. W. Duin, Piotr Juszczak, D. de Ridder, Pavel Paclik, Elzbieta Pekalska, and D.M.J. Tax, "PR-Tools4.1, a matlab toolbox for pattern recognition," Delft University of Technology, 2007, `http://prtools.org`.