Music Genre Classification Using Sparsity-Eager Support Vector Machines

Kamelia Aryafar  
*Drexel University*  
Philadelphia, PA, USA  
kca26@cs.drexel.edu

Sina Jafarpour  
*Multimedia Research Group*  
Yahoo! Research  
sina2jp@yahoo-inc.com

Ali Shokoufandeh  
*Drexel University*  
Philadelphia, PA, USA  
ashokouf@cs.drexel.edu

Abstract

Constructing robust categorical and typological classifiers, i.e., finding auditory constructs utilized for describing music categories, is an important problem in music genre classification. Supervised methods such as support vector machine (SVM) achieve state of the art performance for genre classification but suffer from over-fitting on training examples. In this paper, we introduce a supervised classifier, $\ell_1$-SVM, that utilizes sparse methods to deal with over-fitting for genre classification. We compare our proposed algorithm to competing learning methods such as SVM, logistic regression, and $\ell_1$-regression for genre classification. Experimental results suggest that the proposed method using short-time audio features (MFCCs) outperforms the baseline algorithms in terms of average classification accuracy rate of musical genres.

1 Introduction

Over the past two decades, advances in the digital music industry have resulted in an exponential growth in music data sets. This exponential growth has in turn spurred great interest in music information retrieval (MIR) problems, organizing large music collections, and content-based search methods for digital music libraries. Equally important are the related problems in music classification such as genre classification, music mood analysis, and artist identification. Musical genres are categorical and typological constructs utilized for describing music, often characterized by statistical properties of its musical instruments and rhythmic structure. Music genre classification (MGC) is a well-studied problem in the music information retrieval community and has a wide range of applications, e.g., music playlist generation [3, 12]. Due to its
subjectivity, MGC has been traditionally performed manually [20]; However, over the past decade automatic genre classification has received increasing interest [27, 2].

The two major challenges in automatic music genre classification are robust representation of audio signals in terms of audio features and construction of a learning schema to classify these features into music genres. Various features have been proposed in the literature of the MIR community to represent short-time or long-time audio characteristics [8]. The short-time audio features are mainly derived from succinct segments of signal spectrum and include spectral centroids, Mel-frequency cepstral coefficients (MFCC) [26], and octave based spectral contrast (OSC) [16]. The long-time audio features are mainly based on variation of spectral or beat information over a long segment of the audio signal. Typical examples of long-time audio features include Daubechies wavelet coefficients histogram (DWCH) [18], octave-based modulation spectral contrast (OMSC), low-energy and beat histogram [26].

Selecting the superior classifier for a given data set is an important part of developing an automatic music genre classifier. Typically, the accuracy and the convergence time are the most important factors for evaluating a classifier. Anglade et al. [1] combine a low-level classifier with the harmony-based method to obtain some of the most promising classification results. Support vector machines (SVM) [11, 7], ℓ1-regression [8], logistic regression [23] and k-nearest neighbors are among the widely used baseline methods for audio classification.

In this paper, we introduce the ℓ1-SVM classifier for music genre classification which integrates structural risk minimization benefits of the SVM and the over-fitting resilience of the ℓ1-regression method. The proposed ℓ1-SVM classifier (not to be confused with the sparse-SVM of Yuan et al. [29]), finds a sparse linear combination of the training examples which maximally separates the examples belonging to distinct classes. The proposed classifier can be efficiently trained by solving a linear optimization problem. In contrast to the original ℓ2-SVM, only a small subset of the training examples participate in formation of the final classifier. We will show that this in turn leads to simpler classification complexity and higher generalization (multitasking) accuracy. Experimental results confirm the superiority of the proposed method over the existing ones using only the MFCC features.

The remainder of this paper is organized as follows. In Section 2, we review the structural risk minimization using support vector machines, and the statistical model selection using the ℓ1-regression classifier. In Section 3, we introduce the ℓ1-SVM with the goal of providing structural risk minimization and statistical model selection simultaneously. Section 4 describes the experimental setup, the data set and the details of our experimental results. We conclude the paper in Section 5 and propose future improvements.

2 Previous Work

In this section, we will describe the classification of audio signals using SVM and ℓ1-regression methods. Later in Section 3, we will show how the ideas behind these techniques can be combined to achieve the more robust ℓ1-SVM algorithm. We will begin by describing the audio sampling and construction of MFCC feature vectors.
Throughout our discussions, we assume that a large and diverse repository of music audio sequences is provided as training data. Each classifier then builds a model that assigns samples (points in the feature space) to their corresponding genre categories. The set of hyperplanes that define the gaps between genres, i.e., decision planes, are the outcome of classifier training on the selected data set.

2.1 Feature Extraction

The Mel-frequency cepstral coefficients (MFCCs) are selected as the short-time representation of a given audio sequence. The MFCC descriptors are known to be very effective for music genre classification systems [19, 2]. These features represent short-duration musical textures by encoding a sound’s timbral information. The process of constructing MFCCs begins by applying a Fourier transform to fixed-size, overlapping windows. A series of transformations, combining an auditory filter bank with a cosine transform, will result in a discrete representation of each window in terms of MFCC descriptors. In practice, the filter bank is often constructed using 13 linearly-spaced filters followed by 27 log-spaced filters [25, 14]. Each short-time audio window is then represented using an MFCC vector composed of 13 cepstral coefficients. Finally, a random finite subset of MFCC descriptors will be used to describe an audio sample.

It should be noted that the main contribution of this paper is providing evidences on the advantages of ℓ1-SVM classifier over the baseline classification methods. The MFCC features are well-studied and easily calculable features, and therefore we study the performance of the classification algorithms using this type of features. This by no means implies that the MFCC features induce the best space for solving the MGC problem. In fact, one can argue, a richer space with more informative features may increase the generalization accuracy of all classifiers.

2.2 Support Vector Machines

Support vector machines (SVM) [5] is a linear threshold classifier in a prescribed feature space, with maximum margin and consistent with a set of training examples. Throughout this section, we only consider the problem of using SVM for binary classification. A multiclass SVM is obtainable from a binary SVM using the output coding techniques [9]. In the binary classification setting, every instance $x$ has a corresponding label $y \in \{-1, 1\}$ indicating whether it belongs to a specific genre or not.

In the sequel, we denote vectors and matrices with bold lowercase and capital letters, respectively. A linear threshold classifier $w(x)$ corresponds to a vector $w \in \mathbb{R}^n$ and its prediction for the instance $x$ is the outcome of the inner product $w(x) = \text{sign} (w^T x)$. As a result, we identify the linear threshold classifiers with their corresponding threshold vectors. For simplicity, we will only focus on classifiers passing through the origin. The results can be simply extended to the general case. Since the prediction $w(x)$ is invariant under rescaling, without loss of generality we may assume that every instance $x$ is normalized to have unit ℓ2 norm, i.e., $\|x\|_2 = 1$.

\^1The feature space representation of the music frame in our case.
Observe that if the training examples are not linearly separable, then soft margin SVM can be used. The idea is to simultaneously maximize the margin and minimize the empirical hinge loss. More precisely, let

\[ H(x) = (1 + x)_+ = \max\{0, 1 + x\} \]

denote the Hinge function, and let \( S \) \( \doteq \{(x_1, y_1), \cdots, (x_M, y_M)\} \) be a set of \( M \) labeled training data. For any linear classifier \( w \in \mathbb{R}^n \) we define its empirical hinge loss, an upper bound for the classification error, as

\[ \hat{H}_S(w) \doteq \mathbb{E}_{(x, y) \sim S} \left[ (1 - y_i w^\top x_i)_+ \right]. \]

The empirical(\( \ell_2 \)) regularization loss is similarly defined as

\[ \hat{L}(w) = \hat{H}_S(w) + \frac{1}{2C} \|w\|^2, \tag{1} \]

where \( C \) is the regularization constant.

Soft margin SVM then minimizes the empirical regularization loss which is a convex optimization program. The following theorem is a direct consequence of the convex duality. It is immediate from this theorem that the classification problem can be restated as a convex optimization program.

**Theorem 1** Let \( \{(x_1, y_1), \cdots, (x_M, y_M)\} \) be a set of \( M \) labeled training examples, and let \( w \) be the SVM classifier that minimizes Equation (1). Then the SVM classifier can be represented as a linear combination of the training examples, i.e., \( w = \sum_{i=1}^{M} \alpha_i y_i x_i \). Moreover, \( \alpha_i \in [0, \frac{C}{M}] \), for all \( i \in \{1, \ldots, M\} \).

**Proof 1** The optimization problem of Equation (1) can be restated as

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \|w\|^2 + C \frac{1}{M} \sum_{i=1}^{M} \xi_i \\
\text{subject to} & \quad 1 - y_i w^\top x_i \leq \xi_i, \quad i = 1, \ldots, M \\
& \quad \xi_i \geq 0, \quad i = 1, \ldots, M.
\end{align*}
\tag{2}
\]

We can state the Lagrangian of the above optimization problem as

\[
\mathcal{L}(w, \xi, s, \eta) = \frac{1}{2} \|w\|^2 + C \frac{1}{M} \sum_{i=1}^{M} \xi_i

+ \sum_{i=1}^{M} \alpha_i (1 - y_i w^\top x_i - \xi_i)

- \sum_{i=1}^{M} \eta_i \xi_i.
\tag{3}
\]

The followings are the consequences of KKT conditions for the saddle point of the Lagrangian function:
• The optimal classifier is a linear combination of the training examples, i.e.,

\[ w - \sum_{i=1}^{M} \alpha_i y_i x_i = 0. \]

• The optimal dual variables \( \alpha_i \) and \( \eta_i \) are all non-negative.

• We have \( \frac{C}{M} - \alpha_i - \eta_i = 0 \), which implies \( \alpha_i \leq \frac{C}{M} \) for all \( i = 1, \ldots, M \)

Therefore, the optimization problem in equation (2) can be written as

\[
\begin{align*}
\text{minimize} & \quad \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j x_i^\top x_j + \frac{C}{M} \sum_{i=1}^{M} \xi_i \\
\text{subject to} & \quad 1 - y_i \sum_{j=1}^{M} \alpha_j y_j x_j^\top x_i \leq \xi_i, \quad \forall i \\
& \quad 0 \leq \alpha_i \leq \frac{C}{M}, \quad \forall i \\
& \quad \xi_i \geq 0, \quad \forall i.
\end{align*}
\]

(4)

Furthermore, its dual program is

\[
\begin{align*}
\text{maximize} & \quad \sum_{i=1}^{M} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j x_i^\top x_j \\
\text{subject to} & \quad \forall i \in \{1, \ldots, M\} : \quad 0 \leq \alpha_i \leq \frac{C}{M}.
\end{align*}
\]

(5)

The optimization problem in (5) is a convex quadratic program and can be solved efficiently using a nonlinear solver [4].

While SVM provides an efficient minimum margin classifier, it suffers from overfitting. Specifically, a majority of the training examples are involved in making the final decisions for the unseen (test) examples, since the final decision for any new instance \( x \) is \( \text{Sign}(\sum_{i=1}^{M} \alpha_i y_i x_i^\top x) \). In contrast, principles of learning theory suggest that simpler models with a small well-chosen subset of training examples should suffice in defining the classifiers [28]. Next, we will review the sparse approximation framework which bases the classification decision on such a small number of well-chosen training examples.

### 2.3 Sparse Reconstruction and \( \ell_1 \)-Regression

The sparse approximation approach for music genre classification has been recently proposed by Chang et al. [8] based on recent advances in using compressed sensing for speaker identification [17] and speech recognition [24]. This approach relies on the
assumption that for a sufficiently large number of training instances per class, each new (unseen) instance lies on the subspace spanned by all training examples belonging to that class, and is therefore representable by a sparse linear combination of all training examples.

Let \( k, r, \) and \( n \) denote the number of music genre classes, the number of training music examples per genre, and the dimension of the extracted feature space, respectively. We will denote the extracted feature vector of the \( j \)th training example in the \( i \)th class as \( x_{ij} \in \mathbb{R}^n \). Again, without loss of generality, we will assume that each feature vector \( x_{ij} \) is normalized, i.e., \( \|x_{ij}\|_2 = 1 \). We will also assume the \( i \)th class has a corresponding music repository \( \mathbf{X}_i \) which is an \( n \times r \) matrix, obtained from all \( r \) training examples belonging to class \( i \). The training music genre repository (TMGR) is then the \( n \times M \) matrix \( \mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_k] \), where \( M = k \times r \) denotes the total number of training examples. A vector \( x \) is said to be \( s \)-sparse if it has at most \( s \) non-zero entries. The support of a \( s \)-sparse vector is the set of indices of its non-zero entries. The \( \ell_0 \) pseudo-norm of a vector counts the number of its non-zero entries. In other words, a vector \( x \) is \( s \)-sparse if and only if \( \|x\|_0 \leq s \).

It follows from the subspace model assumption \cite{10} that if \( \mathbf{X} \) contains a sufficiently rich set of training examples for each music genre, then every new test example can be represented by a sparse linear combination of all training examples in the training music genre repository. Specifically, let \( f \in \mathbb{R}^n \) be a test example. The sparse-approximation classifier first finds the solution \( \hat{\alpha} \) of

\[
\begin{align*}
\text{minimize} & \quad \|\alpha'\|_0 \\
\text{subject to} & \quad \|X\alpha' - f\|_2 \leq \epsilon
\end{align*}
\]

for sufficiently small parameter \( \epsilon \). For each class \( i \), let \( \delta_i \) be the indicator function that selects the coefficients associated with the \( i \)th class. The sparse-approximation classifier then outputs its prediction \( \hat{y} \) as

\[
\hat{y} = \arg\min_{i \in \{1, \ldots, k\}} \|f - \mathbf{X}_i \delta_i(\hat{\alpha})\|_2.
\]

It is known that solving the optimization problem of Equation (6) is non-convex, and in general, NP-hard \cite{21}. However, the emerging theory of compressive sensing \cite{6, 10}, asserts that under certain conditions for many practical applications, the solution of the convex optimization problem

\[
\begin{align*}
\text{minimize} & \quad \|\alpha'\|_1 \\
\text{subject to} & \quad \|X\alpha' - f\|_2 \leq \epsilon
\end{align*}
\]

known as \( \ell_1 \)-minimization, \( \ell_1 \)-regression, or Basis Pursuit optimization coincides with the solution of Equation (6). The optimization problem of Equation (7) is a second order cone optimization and can be solved in \( O(M^3) \) time. Moreover, the \( \ell_1 \)-regression classifier is a lazy classifier, that is, the optimization of Equation (7) for each test example \( f \) must be solved independently. As a result, scalability is a fundamental issue for this approach. For a survey of alternative formulation and techniques for this problem see \cite[Ch. 2]{15}. 

6
3 The $\ell_1$-SVM classifier

In this section we introduce the $\ell_1$-SVM for music genre classification which combines the ideas of the classic SVM with the sparse approximation techniques. The main objective of the proposed classifier includes obtaining higher generalization accuracy on new (test) examples, while increasing the robustness against over-fitting to the training examples, and providing scalability in terms of the classification complexity. Given a set $\langle (x_1, y_1), \cdots, (x_M, y_M) \rangle$ of $M$ training examples, we aim to find a vector $\alpha \in \mathbb{R}^M$ such that (i) $\alpha$ is sufficiently sparse, and (ii) the classifier $w = \sum_{i=1}^{M} \alpha_i y_i x_i$ has a sufficiently low empirical loss and therefore sufficiently large separating margin.

Recall that the objective function of a regular ($\ell_2$)-SVM can be rewritten as the optimization problem in (4). To avoid the curse of dimensionality and over-fitting for training examples, we wish to find a solution $\alpha$ which is as sparse as possible. Therefore, by replacing the maximal margin achieved from minimizing $\sum_{i,j=1}^{M} \alpha_i \alpha_j y_i y_j x_i^\top x_j$, with the new objective of minimizing $\|\alpha\|_0$, we obtain the following objective function for training a sparse linear threshold classifier

$$
\text{minimize} \quad \|\alpha\|_0 + C M \sum_{i=1}^{M} \xi_i \\
\text{subject to} \quad 1 - y_i \sum_{j=1}^{M} \alpha_j y_j x_j^\top x_i \leq \xi_i, \; i \in \{1, \ldots, M\} \\
\quad \quad \quad 0 \leq \alpha_i \leq C M, \; i \in \{1, \ldots, M\} \\
\quad \quad \quad \xi_i \geq 0, \; i \in \{1, \ldots, M\}.
$$

(8)

Similar to the optimization problem in (6), the optimization problem in (8) is intractable. To overcome this, we will relax the non-convex pseudo-norm $\|\alpha\|_0$ with the convex $\ell_1$ norm $\|\alpha\|_1$ which is the closest convex norm to $\ell_0$. As a result, the optimization objective for the $\ell_1$-SVM classifier can be stated as

$$
\text{minimize} \quad \sum_{i=1}^{M} \alpha_i + C M \sum_{i=1}^{M} \xi_i \\
\text{subject to} \quad 1 - y_i \sum_{j=1}^{M} \alpha_j y_j x_j^\top x_i \leq \xi_i, \; \forall i \\
\quad \quad \quad 0 \leq \alpha_i \leq C M, \; \forall i \\
\quad \quad \quad \xi_i \geq 0, \; \forall i.
$$

(9)
The optimization program in (8) is a linear program with the dual objective

\[
\text{minimize } \sum_{i=1}^{M} \lambda_i + \frac{C}{M} \sum_{i=1}^{M} \theta_i \\
\text{subject to } 1 - y_i \sum_{j=1}^{M} \lambda_j y_j x_j^\top x_i \geq \theta_i, \; \forall i\\
0 \leq \lambda_i \leq \frac{C}{M}, \; \forall i\\
\theta_i \leq 0, \; \forall i.
\]

(10)

This linear program can be efficiently solved using fast gradient descent techniques [22]. In contrast to the \(\ell_1\)-regression framework, this optimization needs to be solved only once to learn the optimal values of \(\alpha\). Finally, the classification decision for a new sample \(x\) will be based on

\[
\hat{y} = \text{Sign} \left( \sum_{i: \alpha_i \neq 0} \alpha_i y_i x_i^\top x \right).
\]

4 Experimental Results

In this section, we compare the performance of the \(\ell_1\)-SVM classifier with existing baseline classifiers for music genre classification. In our experiments, we use the publicly available benchmark data set for audio classification proposed by Homburg et al. [13]. The data set contains samples of 1886 songs and is comprised of nine music genres: pop, rock, folk-country, alternative, jazz, electronic, blues, rap/hip-hop, and funk soul/R&B. As illustrated in Table 1, the number of available samples varies by genre. The funk soul/R&B genre is excluded from all experiments due to small numbers of available samples. We use the parameter \(minG = 113\) as a minimum number of available songs per genres. For each music genre, \(minG\) songs were chosen uniformly at random to represent the corpus of our data set. Each song is associated with a ten-second audio sample drawn from a random position in the corresponding song. All audio samples were encoded using mp3 format with a sampling rate of 44100Hz and bit-rate of 128mbit/s. The short-time MFCC features are extracted using the Auditory toolbox [25]. Each sample is then represented by \(n = 500\) random MFCC feature vectors.

We compared our \(\ell_1\)-SVM method to three different machine learning algorithms for music genre classification: \(\ell_1\)-regression [8], logistic regression [23], and SVM optimization [11]. We performed a 10-fold cross validation to evaluate the accuracy of the MGC. In this approach \(0.9 \times minG\) of the data set is chosen uniformly at random to serve as the training set, while the remaining \(0.1 \times minG\) forms the testing set. We will use the classification accuracy as our performance evaluation criteria. This measure corresponds to the number of audio samples correctly classified divided by the total number of audio samples in the corpus of our data set. Table 2 shows the average classification accuracy rate for the four learning methods.
<table>
<thead>
<tr>
<th>Genre</th>
<th>Number of Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>alternative</td>
<td>145</td>
</tr>
<tr>
<td>blues</td>
<td>120</td>
</tr>
<tr>
<td>electronic</td>
<td>113</td>
</tr>
<tr>
<td>folk-country</td>
<td>222</td>
</tr>
<tr>
<td>funk soul/R&amp;B</td>
<td>47</td>
</tr>
<tr>
<td>jazz</td>
<td>319</td>
</tr>
<tr>
<td>pop</td>
<td>116</td>
</tr>
<tr>
<td>rap/hip-hop</td>
<td>300</td>
</tr>
<tr>
<td>rock</td>
<td>504</td>
</tr>
</tbody>
</table>

Table 1: Number of songs per genre.

<table>
<thead>
<tr>
<th>Classification method</th>
<th>Average accuracy rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>ℓ₁-SVM</td>
<td>37.43%</td>
</tr>
<tr>
<td>log-regression</td>
<td>34.43%</td>
</tr>
<tr>
<td>ℓ₂-SVM</td>
<td>32.90%</td>
</tr>
<tr>
<td>ℓ₁ regression</td>
<td>30.45%</td>
</tr>
</tbody>
</table>

Table 2: Average classification accuracy rate for music genre classification on the benchmark data set is illustrated. Each experiment is repeated independently 50 times and the average accuracy rate is reported.

Figure 1: True positive genre classification rate is illustrated for each music genre.

In our next experiment we evaluated the proper classification ratios of audio sequences based on each genre. In this experiment, we computed the average classification rate of 50 rounds of independent experiments obtained under different learning methods. Figure 1 illustrates the accuracy rate per genre.
The suitability or performance of classifiers using different learning schemes is often tested using different number of training samples and varying training time. Figure 2 illustrates the classification accuracy rates of 50 rounds of independent experiments using different number of training samples. We note that the $\ell_1$-SVM outperforms $\ell_2$-SVM, logistic regression and $\ell_1$-regression given the same number of training samples.

Figure 2: $\ell_1$-regression [8], logistic regression [23], $\ell_2$-SVM optimization [11] and $\ell_1$-SVM classification accuracy rates are illustrated using different number of training samples. The number of training samples have been limited to 20 samples to reduce the convergence time of the classifier. There is no deviation in the classification accuracy rate for a larger training set.

In addition to classification accuracy rate, the average training time is a second important factor that affects the suitability of a classification algorithm. An algorithm with a slightly lower accuracy might be preferred if its training time is significantly lower. An estimation of the required training time for a genre classification task is very useful if the result has to be available in a certain amount of time. In this experiment we use 20 random training songs to train our classifiers. The $\ell_1$-regression method is a lazy classifier without a training phase and thus is excluded from this experiment. Figure 3 shows the average classification accuracy rate of $\ell_2$-SVM, logistic regression and $\ell_1$-SVM on the corpus of our data set against the average training time for each classifier. We note that the $\ell_1$-SVM outperforms $\ell_2$-SVM and logistic regression once the convergence threshold is set properly.

The experimental results show improvements in genre classification using short-time audio features in combination with $\ell_1$-SVM learning method. An interesting question is whether the choice of different audio features will affect the performance of genre classification using the same machine learning algorithms. This will be part of our future studies of the genre classification problem.
5 Conclusions

In this paper we proposed a novel learning model to perform music genre classification on short-time representations of audio datasets. We introduced the $\ell_1$-SVM classifier which combines the ideas of the classical SVM with the sparse approximation techniques. It achieves higher generalization accuracy on new (test) samples, while increasing the robustness against over-fitting to the training examples, and providing scalability in terms of the classification complexity. Through a set of experiments we have demonstrated the utility of the proposed method for genre classification and compared the results to $\ell_1$-regression [8], logistic regression [23] and $\ell_2$-SVM optimization [11]. The results indicate that $\ell_1$-SVM classifier will improve the classification accuracy of audio samples using MFCCs.

In the future, we intend to study the use of long-time audio features such as Daubechies wavelet coefficients histogram (DWCH) [18], octave-based modulation spectral contrast (OMSC), low-energy and beat histogram [26] to enhance classification accuracy; our current method focuses primarily on short-time MFCC features. We anticipate that a combination of other audio features will enhance genre classification accuracy.

References


information retrieval with user-centered and multimodal strategies, MIRUM ’11, pages 33–38, New York, NY, USA, 2011. ACM.


