

# Music Genre Classification: A Semi-supervised Approach

Soujanya Poria,<sup>1</sup> Alexandar Gelbukh,<sup>2</sup> Amir Hussain,<sup>3</sup>  
Sivaji Bandyopadhyay,<sup>1</sup> Newton Howard<sup>4</sup>

<sup>1</sup> Computer Science and Engineering Department, Jadavpur University, India,  
soujanya.poria@ieee.org, sbandyopadhyay@cse.jdvu.ac.in

<sup>2</sup> CIC, Instituto Politécnico Nacional, 07738 DF, Mexico, www.Gelbukh.com

<sup>3</sup> Dept. of Computing Science and Mathematics, University of Stirling,  
United Kingdom, ahu@cs.stir.ac.uk

<sup>4</sup> Brain Science Foundation, nhmit@mit.edu

**Abstract.** Music genres can be seen as categorical descriptions used to classify music basing on various characteristics such as instrumentation, pitch, rhythmic structure, and harmonic contents. Automatic music genre classification is important for music retrieval in large music collections on the web. We build a classifier that learns from very few labeled examples plus a large quantity of unlabeled data, and show that our methodology outperforms existing supervised and unsupervised approaches. We also identify salient features useful for music genre classification. We achieve 97.1% accuracy of 10-way classification on real-world audio collections.

## 1 Introduction

Downloading and purchasing music from online music collections has become part of the daily life of probably the majority of people in the world, and quality of music recommendation affects quality of life of billions of people. The users often formulate their preferences in terms of genre, such as jazz or disco. However, many tracks in existing collections are not classified by genre, or a genre is specified for an artist or an album but not for a particular track. Given huge size of existing collections, automatic genre classification is crucial for organization, search, retrieval, and recommendation of music.

While huge amount of unlabeled data is readily available, labeled data—tracks with the genre reliably assigned by human annotators—are scarce. In this paper we propose to use for genre classification a methodology that was proven to work well in a similar situation: affective labeling of words in natural language texts, where, similarly, unlabeled texts abound but few words have a manually assigned affective label [1]. For brevity we refer to this methodology as semi-supervised learning, to emphasize that it uses two kinds of data: few labeled examples and a large quantity of unlabeled data; however, internally our two-step procedure works differently from a typical semi-supervised learner. We show that this methodology outperforms a number of standard supervised learning techniques, such as Support Vector Machine (SVM) and k-Nearest Neighbor (kNN).

In addition, we present several features salient for the genre classification task. According to Lee *et al.* [3], musical features can be divided into three categories: short-time features, long-time features, and beat features. We show that long-term features are more salient for music genre classification than short-time ones, though using features of both types together gives best results. We explore various feature combinations and identify those that perform best on our task.

The rest of the paper is organized as follows: Section 2 describes related work. Section 3 gives an overview of our method. Section 4 presents the dataset and the features used in our experiments. Sections 5, 6, and 7 describe the three main steps of our algorithm: fuzzy clustering, mapping of the obtained clusters to labels, and the final hard categorization. Section 8 gives the experimental results and evaluation. Finally, Section 9 presents conclusions and future work.

## 2 Related Work

The state-of-the-art music genre classification systems can be classified into those based on supervised or unsupervised approach [4].

**Unsupervised Approaches** These approaches mainly concern with determination of a genre taxonomy. Music files are clustered basing on an objective function to dynamically build a taxonomy depending on the clustering outcome. Shao *et al.* [5] used Agglomerative Hierarchical Clustering.

The main drawback of unsupervised methods is that the clusters are not labeled and the boundaries between clusters are not reliably defined. In this paper, we rely on a genre taxonomy well-defined by music experts and well-known to the users, which suggest using a supervised approach.

**Supervised Approaches** Pampalk *et al.* [7] used a kNN classifier; Mandel *et al.* [10], Lidy *et al.* [11], and Scaringella *et al.* [12] obtained good results with SVM.

**Semi-Supervised Approaches** Xu *et al.* [23] used co-training based semi-supervised classifier based on some novel “multi-view” [23] features. Yaslan *et al.* [24] stated how Random Subspace Method for Co-training [24] can help the genre classification process.

Our approach is a bit different from existing semi-supervised approaches in a manner that we used fuzzy for training supervised classifier in order to map a 10-way classification problem to 2-way or 3-way classification problem.

**Feature Selection** Proper feature selection is crucial for classification. For this, a segment of audio is represented by numerical values of several audio features. Peeters [14] proposed a variety of features to characterize the timbre of instruments. These features are called low level features [15] because they usually describe sound on small scale, such as slices of 10 to 60 seconds. Spectral features have been used to distinguish between speech and music [4] and to identify isolated sounds [16] and instruments [17]. Rauber *et al.* [6] used psychoacoustic features of music to determine similarities between music files. The importance of the size of texture window for extracting timbral features has been explored by Meng *et al.* [18], who stated that

texture window of 1 sec. works best for music retrieval task and there is no significant gain in increasing the size of the texture window, while the accuracy decreases with smaller window size.

Tzanetakis [19] designed a basic music genre classification system based upon timbral, temporal, and beat features with 61% accuracy, outperformed by Lee *et al.* [3] with a spectral modulation-based approach.

While we work with a conventional genre classification, we explore the performance of different features and feature combinations, achieving 97.1% accuracy.

### 3 Overview of the Procedure

We followed a procedure suggested in [1] for a quite different task: effective classification of words [2]. The procedure consists in the following steps:

- Feature extraction: the real-world data, both labelled and unlabelled, are represented by numerical vectors, which then are used for classification.
- Fuzzy clustering: the whole available dataset, including both labelled and unlabelled data (the labels are ignored even when available), is clustered in unsupervised manner into the number  $c$  of clusters corresponding to the number of target categories (in our case,  $c = 10$  music genres), in hope that the found clusters would roughly correspond to the target categories. Fuzziness accounts for uncertainty: a data point can be assigned more than one label, with a different degree. This ambiguity is resolved at the last step.
- Mapping: the obtained fuzzy clusters are one-to-one identified with the  $c$  target categories. The classes are identified through a majority voting technique, performed within each of the clusters. In our case we have all annotated music samples but for that situation when we have maximum number of unlabeled data and much lower number of labelled data we can still carry out our method by taking all of those unlabeled and labelled data for clustering and determine the fuzzy classes of the clusters through the majority voting with the help of available labelled data taking part in the clustering step.
- Hard clustering: the ambiguity of the fuzzy assignment of category labels to data items is resolved, leaving each data item assigned to exactly one category using a supervised technique.

### 4 Dataset and Features

As a dataset for the music genre classification task, we used the one presented by Tzanetakis [19]. The dataset is publicly available for research purposes.<sup>1</sup> It consists of 1000 audio tracks, each being 30 sec. long, classified into 10 genres: BLUES, CLASSICAL, HIPHOP, COUNTRY, DISCO, POP, ROCK, JAZZ, METAL and REGGAE. Each genre is represented by 100 tracks. We followed this taxonomy for our classification.

---

<sup>1</sup> <http://opihi.cs.uvic.ca/sound/genres.tar.gz>

All tracks are 22,050 Hz mono 16-bit audio files in .wav format, collected in 2000–2001 from a variety of sources including personal CDs, radio, and microphone recordings, in order for a variety of recording conditions to be represented.

For feature extraction, we used the Jaudio toolkit [20], a music feature extraction toolkit written in Java, freely available for research purposes.<sup>2</sup> As we have mentioned, we used the following three kinds of features: short-time features, long-time features, and beat features.

Short-time features are mainly used to distinguish the timbral characteristics of music and are usually extracted independently from each short time window (frame) during which the audio signal is assumed to be stationary. We used the following features [3, 20]: mel-frequency cepstral coefficients (MFCC; Jaudio gives first five coefficients of 13, and [3] states that these five coefficients give best classification result), spectral centroid, spectral roll-off, spectral flux, root mean square, compactness, and time domain zero crossing;

Long-time features can be obtained by aggregating the short-term features extracted from several consecutive frames within a time window. We have used derivative, standard deviation, running mean, derivative of running mean, and standard derivative of running mean as the aggregation methods of short-time features.

Beat features give meanings to audio signals in human-recognizable terms which generally reveal the human interpretation or perception of certain audio properties such as mood, emotion, tempo, genre, etc. We used the following four main beat features: beat histogram, beats per minute, beat sum, and strongest beat in the audio signal.

## 5 Fuzzy Clustering

The first step in our process is unsupervised: we cluster the music files into 10 categories, given that we consider 10 genres. On output, we define for each music file and each of the ten classes the membership value between 0 and 1 with which the given music file belong to the given class.

**Fuzzy C-means Clustering Algorithm** For fuzzy clustering, we used the fuzzy c-means clustering algorithm [21] with a modified objective function as described in Section 5.2 below.

The well-known fuzzy c-means clustering algorithm takes as input a set of  $N$  data points  $x_1, x_1, \dots, x_N$  described via their coordinates in a  $P$ -dimensional feature space:  $x_k = (x_{k1}, x_{k2}, \dots, x_{kP})$ . As output, it constructs two sets: a set of  $c$  centroids  $v_1, v_2, \dots, v_c$ , points in the same feature space, that represent the constructed  $c$  clusters, and a set of  $cN$  membership values  $\mu_{ik}$ ,  $i = 1, \dots, c$ ;  $k = 1, \dots, N$ , which represent the degree of membership of a point  $x_k$  in a class  $c_i$ , such that  $0 \leq \mu_{ik} \leq 1$  and the values sum up to a unity for each point:

$$\sum_{i=1}^c \mu_{ik} = 1, \quad k = 1, \dots, N. \quad (1)$$

---

<sup>2</sup> <http://sourceforge.net/projects/jmir/files/>

To find the optimal distribution of points by clusters and optimal placement of the centroids, it uses an given objective function  $J$ , which is minimized when the distribution is optimal:  $(\mu_0, v_0) = \arg \min J(\mu, v)$ , where  $\mu = \{\mu_{ik}\}$  and  $v = \{v_i\}$  represent the sets of the variables to be found and  $\mu_0, v_0$  are the optimal solutions. An expression often used for  $J$  is

$$J_p(\mu, v) = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^p \|x_k - v_i\|^2, \quad \|x - y\|^2 = \sum_{l=1}^P (x_l - y_l)^2 \quad (2)$$

where the power  $p > 1$  is a given parameter that controls the degree of fuzziness of the obtained clusters (we used  $p = 2$ ). The optimal solution of a constraint optimization problem defined by (1) and (2) is given [1] by

$$v_i = \frac{\sum_{k=1}^N \mu_{ik}^p x_k}{\sum_{k=1}^N \mu_{ik}^p}, \quad \mu_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{\|x_k - v_i\|}{\|x_k - v_j\|} \right)^{\frac{2}{p-1}}} \quad (3)$$

A stationary point  $(\mu_0, v_0)$  of the system (3), which is the desired result of the algorithm, was found iteratively:

- Assigning random values to all  $\mu_{ik}$ , normalized to satisfy the constraints (1);
- Iteratively re-calculate the values for all  $v_i$  and then all  $\mu_{ik}$  according to (3);
- Stop when the objective function  $J$  changes from the previous iteration less than by a small number  $\varepsilon$ , a given parameter (we used  $\varepsilon = 0.01$ ).

**Modified Objective Function** To achieve more compact clusters in which the most similar elements are clustered together, we incorporated an additional term in the original objective function (2):

$$J_p(\mu, v) = \sum_{i=1}^c \sum_{k=1}^N \mu_{ik}^p \left( \|x_k - v_i\|^2 + \rho \sum_{x \in N_k} \|x - v_i\|^2 \right), \quad (4)$$

where the parameter  $\rho$  is intended to control the effect of the new term (we used  $\rho = 1$ ) and  $N_k$  is the set constructed in the following way:

- For each data point  $x$ , we identified the nearest centroid

$$v(x) = \arg \min_i \|v_i - x\| \quad (5)$$

(in case of a tie an arbitrary one was chosen);

- Now,  $N_k = \{x \mid v(x) = v(x_k)\}$  is the set of all data points with the same nearest centroid as  $x_k$ .

This additional term forces the algorithm to increase the membership of a data point in the cluster with the nearest centroid, grouping similar points together.

In our implementation we constructed these sets on the fly while re-calculating the positions of the centroids according to (6) below, which is a modification of (3). I.e., when re-calculating  $v_2$ , we considered in (5) already re-calculated value for  $v_1$ .

The change of the objective function required modification of the formulas (3):

$$v_i = \frac{\sum_{k=1}^N \mu_{ik}^p \left( x_k + \rho \sum_{x \in N_k} x \right)}{\sum_{k=1}^N \mu_{ik}^p (1 + \rho |N_k|)}, \quad \mu_{ik} = \frac{1}{\sum_{j=1}^c \left( \frac{\|x_k - v_i\|^2 + \rho \sum_{x \in N_k} \|x - v_i\|^2}{\|x_k - v_j\|^2 + \rho \sum_{x \in N_k} \|x - v_j\|^2} \right)^{\frac{1}{p-1}}}; \quad (6)$$

the derivation can be found in [1].

## 6 Mapping Fuzzy Classes to Music Genre Labels

After the fuzzy clustering process had been completed, we identified which one of the  $c = 10$  classes corresponded to which one of the ten music genre labels. For this, first we converted the fuzzy clustering into hard clustering; in our implementation we chose for each data point  $x_k$  a cluster  $C(x_k) = \arg \max_i \mu_{ik}$  (in case of a tie, arbitrary class was chosen). Now, the music genre label for each hard cluster was chosen by majority voting.

Such procedure does not guarantee for a hard cluster to be non-empty, for the majority voting not to result in a tie, or for two clusters not to share the same genre label, in which case some labels would not be assigned at all. However, this is low probable and did not happen in our experiments. Moreover, correctness of the obtained mapping of the classes to genre labels is confirmed by the fact that we obtained over 90% accuracy of the final results, which is not possible with incorrectly mapped labels.

## 7 Hard Clustering

In our evaluation, we consider a label to be assigned correctly if the evaluation dataset assigns this label to the music file. To choose only one class for a token under classification, we used a two-step process.

**Reducing the Confusion Set** For each data point, we chose  $K$  classes for which the fuzzy clustering gave the highest value of the membership function. The hard clustering technique used afterwards was only allowed to choose between those  $K$  labels pre-selected for a given music file.

In case of  $K = 1$  no further processing is needed and the final result is determined by the greatest membership value of the fuzzy clustering. The case of  $K = 10$  means no reducing of the confusion set. In case of  $K = 2$  or  $K = 3$ —the values we experimented with—the confusion set is reduced to 2 or 3 options, correspondingly. We show in Section 8 that reducing the confusion set to 2 candidates increased the accuracy. However, selection of the proper size of confusion set depends to the problem: on another task a confusion set of, say, 3 might result in better accuracy.

**Final Hard Categorization** Given the  $K$  options left after reducing the confusion set, we trained different classifiers for each of the  $\binom{10}{K} = \frac{c!}{K!(c-K)!}$ ,  $c = 10$ , possible combinations of  $K$  genre labels: for example, with  $K = 3$ , a separate classifier was trained for choosing between BLUE, METAL, and COUNTRY, another one to choose between JAZZ, METAL, and ROCK, etc. For  $K = 3$ , therefore, 120 different classifiers were trained; for  $K = 2$ , 45 different classifiers were trained. To assign a label to data point, the  $K$  genre labels for the point are selected as explained above, and then the corresponding classifier is used.

For training, we have taken care of all training music files in ten genre lists of our dataset, i.e., we used 60 music files of each 10 genres lists for training. For example, to train a classifier for the confusion set {BLUE, METAL}, all music files extracted from the lists and have their either the label BLUE or the label METAL were used.

As features, we used the same feature vectors as for fuzzy clustering, extended by 10 extra dimensions: the membership values generated by the fuzzy classifier for the 10 genre labels, except the experiments where the fuzzy clustering was not used. As classes, the  $K$  selected labels were used for each classifier; in case of  $K = 2$  the classification was binary, for  $K = 3$  the classification is ternary.

As a hard clustering algorithm, we used the SVM framework. Specifically, we used the `libsvm` library of the WEKA toolset [22], which, for the case of  $K > 2$ , provides an implementation of a multiclass SVM. As a result, we obtained one music genre label for each music file in the test dataset.

## 8 Evaluation

**Impact of different feature combinations** Table I shows that we obtained better accuracy when we used long-time features than using short-time features. However, a much lower accuracy was obtained when we used only beat features. The highest accuracy was obtained when we used all three types of features: long-time, short-time, and beat features.

TABLE I. Accuracy with different feature combinations and different classifier combinations

Feature Combination	Fuzzy	SVM	Fuzzy + SVM
Long-time features	59.12%	61.20%	63.25%
Short-time features	42.54%	44.15%	48.92%
Long-time + short-time features	68.21%	71.24%	75.34%
Beat features	39.15%	39.46%	41.27%
Long-time + short-time + semantic	76.33%	87.45%	96.23%
Long-time + beat features	68.67%	72.35%	76.25%
Long + short + semantic + fuzzy vector	79.21%	—	<b>97.10%</b>

We have done the evaluation in two ways. In one experiment we performed tenfold cross-validation on each of the 120 and 45 classifiers mentioned in Section 7.2 using all 1000 music files. The result and the corresponding confusion sets are given in

TABLE II. Impact of the selection of most likely fuzzy cluster

$K$	Accuracy
1	76.33%
2	<b>97.10%</b>
3	79.38%
4	77.51%
10	67.45%

TABLE III. Accuracy obtained using different classifiers

Classifier	Accuracy
KNN	54.21%
Naïve Bayes	65.88%
MLP	74.23%
Our procedure	<b>97.10%</b>

Section 8. Using all three feature sets along with fuzzy membership vector as a feature, we obtained 97.10% accuracy.

In another experiment we split our dataset into 60% training and 40% test data. Using 60% training data we trained our 60 or 45 classifiers correspondingly, depending on the value of  $K$ , and tested them on the unseen test data. With this, we obtained 91.50% accuracy, which is probably explained by smaller size of the training data.

In particular, we observed that spectral centroid and MFCC are the most important features, because removing these two features significantly decreases accuracy.

**Impact of the fuzzy clustering and hard categorization** In addition to the data presented in Table I for fuzzy-only and hard-only classifiers, we experimented with different values of  $K$ : the size of the confusion set after reduction based on the result of fuzzy clustering; see Table II:

- $K = 1$  means that the final classification is made basing on the results of the fuzzy clustering and no further hard clustering is necessary;
- $K = 2$  means that the hard classification has to do only binary choices;
- $K = 3$  reduces the confusion set for the hard classification to three options;
- $K = 10$  means no reduction of the confusion set. It is not the same as not to use the fuzzy clustering phase at all, because the fuzzy clustering results are still used as additional features for final categorization.

We can see that SVM performed better on choosing between the category with the highest membership value and that of the second highest one. Here, we used all features, which corresponds to the last row of Table I.

**Comparing with other classifiers** We tried several classifiers, such as Multi-Layer Perception (MLP), Naïve Bayes, and kNN. While MLP performed better than Naïve Bayes and kNN, none of them outperformed our two-stage procedure; see Table III.

**Confusion matrices** The confusion matrices obtained with our procedure are shown in Table IV. We can observe that misclassification problems are very rare, and quite similar in both cases.

## 9 Conclusions and Future Work

We have proposed a method of music genre classification in a large music dataset using a two-stage classification methodology. The methodology consists in fuzzy



Table IV. Confusion Matrices

Classified as	Tenfold cross-validation										60% training / 40% test split									
	a	b	c	d	e	f	g	h	i	j	a	b	c	d	e	f	g	h	i	j
a. blues	100	0	0	0	0	0	0	0	0	0	38	0	0	0	1	0	0	0	0	1
b. country	0	98	0	0	0	0	0	0	0	2	0	37	0	0	0	0	0	0	0	3
c. classical	1	0	99	0	0	0	0	0	0	0	0	0	40	0	0	0	0	0	0	0
d. disco	1	0	1	94	4	0	0	0	0	0	1	0	1	34	4	0	0	0	0	0
e. hiphop	1	0	0	5	93	0	0	0	1	0	1	0	0	3	35	0	0	1	0	0
f. jazz	0	0	0	0	0	100	0	0	0	0	0	0	0	5	0	34	0	1	0	0
g. metal	0	0	0	0	0	0	100	0	0	0	0	0	1	0	0	0	39	0	0	0
h. pop	0	0	0	0	0	0	3	95	0	2	0	0	1	0	0	0	2	36	0	1
i. reggae	0	3	0	0	0	0	0	0	97	0	0	3	0	0	0	0	0	0	37	0
j. rock	0	4	0	0	0	0	0	1	0	95	0	2	0	0	0	1	0	1	0	36

clustering followed by disambiguation using a hard classifier. As features of musical data, we used the values obtained with the Jaudio toolkit.

Soon, we plan to expand the classification tool on a different level by taking into account also lyrics associated with music tracks. In particular, we will extract conceptual and affective information associated with songs by means of semantic multi-dimensional scaling [25] and, hence, add these as additional semantic and sentic [26] features for classification.

**Acknowledgements** The work was partly supported by the Governments of Mexico and India via the project “Answer Validation through Textual Entailment”, CONACYT 122030 / Mexico-India 2011–2014; the Government of Mexico via the Instituto Politécnico Nacional, SIP 20131702 and CONACYT 50206-H; and the Government of UK via UK EPSRC grant ref: EP/G501750/1 “Common sense Computing to enable the development of next generation Sematic Web Applications”.

## References

1. S. Poria, A. Gelbukh, E. Cambria, D. Das, S. Bandyopadhyay, “Enriching SenticNet polarity scores through semi-supervised fuzzy clustering”, IEEE ICDM, Brussels, pp. 709–716, 2012.
2. S. Poria, A. Gelbukh, A. Hussain, D. Das, S. Bandyopadhyay. “Enhanced SenticNet with affective labels for concept-based opinion mining”, IEEE Intelligent Systems, DOI: 10.1109/MIS.2013.4, 2013.
3. C. Lee, J. Shih, K. Yu, H. Lin, “Automatic music genre classification based on modulation spectral analysis of spectral and cepstral features”, IEEE Transactions on Multimedia 11(4), 2009.
4. N. Scaringella, G. Zoia, “Automatic genre classification of music content: a survey” Signal Processing Magazine 23(2):133–141, IEEE, 2006.
5. X. Shao, C. Xu, M. Kankanhalli, “Unsupervised classification of musical genre using hidden Markov model”, IEEE Int. Conf. of Multimedia Explore (ICME), Taiwan, 2004.

6. A. Rauber, E. Pampalk, D. Merkl, "Using psycho-acoustic models and self-organizing maps to create a hierarchical structuring of music by sound similarity", 3rd Int. Conf. on Music Information Retrieval, France, 2002.
7. E. Pampalk, A. Flexer, G. Widmer, "Improvements of audio based music similarity and genre classification?", 6th Int. Symposium on Music Information Retrieval, UK, 2005.
8. N. Scaringella, G. Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals", 6th Int. Symposium on Music Information Retrieval, UK, 2005.
9. H. Soltau, T. Schultz, M. Westphal, A. Waibel, "Recognition of music types", IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP), USA, 1998.
10. M. Mandel, D. Ellis, "Song-level features and support vector machines for music classification", 6th Int. Symposium on Music Information Retrieval, UK, 2005.
11. T. Lidy, A. Rauber, "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification", 6th Int. Symposium on Music Information Retrieval, UK, 2005.
12. N. Scaringella, D. Mlynek, "A mixture of support vector machines for audio classification", Music Information Retrieval Evaluation exchange (MIREX) website, [www.music-ir.org/evaluation/mirexresults/articles/audio\\_genre/scaringella.pdf](http://www.music-ir.org/evaluation/mirexresults/articles/audio_genre/scaringella.pdf), 2005.
13. A. Berenzweig, D. Ellis, S. Lawrence, "Using voice segments to improve artist classification of music", AES 22nd International Conference on Virtual, Synthetic and Entertainment Audio, 2002.
14. G. Peeters, "A large set of audio features for sound description (similarity and classification) in the CUIDADO project", CUIDADO I.S.T. Project Report, 2004.
15. J. Saunders, "Real time discrimination of broadcast speech/music", Int. Conf. Acoustics, Speech, Signal Processing (ICASSP), 1996, pp. 993–996.
16. E. Wold, T. Blum, D. Keislar, and J. Wheaton, "Content-based classification, search, and retrieval of audio", IEEE Multimedia 3(2), 1996.
17. MPEG-7, "Information Technology – Multimedia Content Description Interface – Part 4: Audio", ISO/IEC JTC 1/SC29, ISO/IEC FDIS 15938-4:2002, 2002.
18. A. Meng, P. Ahrendt, J. Larsen, "Improving music genre classification by short-time feature integration", 6th Int. Symposium on Music Information Retrieval, UK, 2005.
19. G. Tzanetakis, "Music Genre Classification of Audio Signal," IEEE Transactions on Speech and Audio Processing 10(5), 2002.
20. D. McEnnis, C. McKay, I. Fujinaga, P. Depalle, "Jaudio: A Feature Extraction Library", ISMIR 2005.
21. J. C. Bezdek, 1981. Pattern Recognition with Fuzzy Objective Function Algorithms, Plenum Press, New York.
22. M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, I.H. Witten. The WEKA data mining software: An update; SIGKDD Explorations 11(1), 2009.
23. Y. Xu, C. Zhang, C. Yang. "Semi-supervised classification of musical genre using multi-view features", International Computer Music Conference ICMC, 2005.
24. Y. Yaslan, and C. Zehra. "Audio genre classification with semi-supervised feature ensemble learning", 2nd International Workshop on Machine Learning and Music, 2009.
25. E. Cambria, Y. Song, H. Wang, and N. Howard, "Semantic multi-dimensional scaling for open-domain sentiment analysis", IEEE Intelligent Systems, doi:10.1109/MIS.2012.118, 2013.
26. E. Cambria and A. Hussain, Sentic Computing: Techniques, Tools, and Applications, Dordrecht, Netherlands: Springer, ISBN: 978-94-007-5069-2, 2012.