MUSIC GENRE CLASSIFICATION USING SELF-TAUGHT LEARNING VIA SPARSE CODING

Konstantin Markov

Human Interface Lab.,
University of Aizu,
Fukushima, Japan

Tomoko Matsui

Dep. of Statistical Modeling,
Institute of Statistical Mathematics,
Tokyo, Japan

ABSTRACT

Availability of large amounts of raw unlabeled data has sparked the recent surge in semi-supervised learning research. In most works, however, it is assumed that labeled and unlabeled data come from the same distribution. This restriction is removed in the self-taught learning approach where unlabeled data can be different, but nevertheless have similar structure. First, a representation is learned from the unlabeled data via sparse coding and then it is applied to the labeled data used for classification. In this work, we implemented this method for the music genre classification task using two different databases: one as unlabeled data pool and the other for supervised classifier training. Music pieces come from 10 and 6 genres for each database respectively, while only one genre is common for both of them. Results from wide variety of experimental settings show that the self-taught learning method improves the classification rate when the amount of labeled data is small and, more interestingly, that consistent improvement can be achieved for a wide range of unlabeled data sizes.

Index Terms— Music genre classification, Self-taught learning, Sparse coding, Transfer learning.

1. INTRODUCTION

A tremendous amount of music-related data has recently become available either locally or remotely over networks, and technology for searching this content and retrieving music-related information efficiently is demanded. This consists of several elemental tasks such as genre classification, artist identification, music mood classification, cover song identification, fundamental frequency estimation, and melody extraction. Essential for each task is the feature extraction as well as the model or classifier selection. Audio signals are conventionally analyzed frame-by-frame using Fourier or Wavelet transform, and coded as spectral feature vectors or chroma features extracted for several tens or hundreds of milliseconds [1, 2, 3]. However, it is an open question how precisely music audio should be coded depending on the task kind and the succeeding classifier. Recently proposed compressive sampling / sparse coding approaches to feature generation have also been applied in music processing with promising results [4, 5, 6].

For the classification, classical supervised pattern recognition approaches require large amount of labeled data which is difficult and expensive to obtain. On the other hand, in the real world, a massive amount of musical data is created day by day and various musical databases are newly composed. There may be no labels for some databases and musical genres may be very specific. Thus, recent music information retrieval research has been increasingly adopting semi-supervised learning methods where unlabeled data are utilized to help the classification task. Common assumption in this case is that both labeled and unlabeled data come from the same distribution [7] which, however, may not be easily achieved during the data collection. This restriction is alleviated in the transfer learning framework [8] which allows the domains, tasks, and distributions used in training and testing to be different. Utilizing this framework and the semi-supervised learning ideas, the recently proposed self-taught learning approach [9] seems be a good fit for the music genre classification task. First, an unsupervised learning algorithm finds a sparse and high-level representation of the inputs given only unlabeled data. Then, using this representation the labeled data which are assumed to be few are transformed into new feature vectors. Finally, classical supervised classifier is trained using the new features. It can be considered that by utilizing the sparse coding, sparse and key features for all classes are possibly captured.

In order to manage various types of new musical databases, we utilize the self-taught learning algorithm based on sparse coding and investigate a genre classification method for new musical database using an existing one which has a well-balanced design for all genres.

2. THE SELF-TAUGHT LEARNING ALGORITHM

A classification task is considered with small labeled training data set $\mathcal{X}^l = \{x_i^l\}, i = 1, \ldots, M$ drawn i.i.d. from an un-

known distribution $\mathcal{D}$. Each $x_i^l \in R^n$ is an input feature vector which is assigned a class label $y_i \in \mathcal{Y} = \{1, \ldots, C\}$. In addition, a larger unlabeled training data set $\mathcal{X}^u = \{x_i^u\}, i = 1, \ldots, K$ is available, which is assumed only to be of the "same type" as $\mathcal{X}^l$ and may not be associated with the class labels $\mathcal{Y}$ and distribution $\mathcal{D}$. Obviously, in order $\mathcal{X}^u$ to help the classification of the labeled data, it should not be totally different or unrelated.

The main idea of the self-taught learning approach is to use the unlabeled data to learn in an unsupervised way slightly higher level representation of the data [9]. In other words, to discover some hidden structures in the data which can be considered as basic building blocks. For example, if the data represent images, the algorithm would find simple elements such as edges, curves, etc., so that the image can be represented in terms of these more abstract, higher level features. Once learned, this representation is applied to the labeled data $\mathcal{X}^l$ resulting in a new set of features which lighten the supervised learning task.

To learn the higher level representation, a sparse coding method is used. Given the unlabeled data set $\mathcal{X}^u$, the following optimization procedure is defined:

$$\min_{a,b} \quad \sum_i \|x_i^u - \sum_j a_j^i b_j\|_2^2 + \beta \|a^i\|_1 \qquad (1)$$
$$\text{subject to} \quad \|b_j\|_2 \leq 1, \quad j = 1, \ldots, s$$

where basis vectors $b_j \in R^n, j = 1, \ldots, s$ and activations $a^i \in R^s, i = 1, \ldots, M$ are subject to optimization. The first term of the above objective tries to represent each data vector as a linear combination of the bases b_j with weights given by the corresponding activations. The second term, on the other hand, tries to reduce the L_1 norm of the activation vectors, thus making them sparse. The optimization problem is convex only in terms of basis vectors or activations alone and these sub-problems are solved iteratively by alternatingly holing a^i or b_j fixed. An efficient algorithm for solving this sparse coding problem is reported in [10].

3. FEATURE GENERATION USING UNLABELED DATA

After learning the basis vectors b_j from the unlabeled training data $\mathcal{X}^u$ as described in the previous section, we can use them to obtain activations for the labeled data $\mathcal{X}^l$. This procedure can be viewed as a non-linear mapping or transformation of vectors $x_i^l \in R^n$ into vectors $a_i^\square \in R^s$, which will be further used as features in the supervised classification task. The new features are computed by solving the following optimization problem:

$$a_i^\square = \arg\min_a \|x_i^l - \sum_j a_j b_j\|_2^2 + \beta \|a\|_1 \qquad (2)$$

This is a convex L_1-norm regularized least squares task which is the same as the optimization problem (1) with fixed bases b_j and can be solved in the same way. Vectors $a_i^\square$ are sparse and approximate labeled data x_i^l as a linear combination of the bases which, however, are learned using the unlabeled data x_i^u. Considering that basis vectors represent information discovered automatically from different data, and that activation vectors $a_i^\square$ in fact use this information to encode the labeled data, the self-taught learning approach can be viewed as an instance of *knowledge transfer* between tasks [8].

Finally, using labels y_i and the new feature vectors $a_i^\square$ we can train standard supervised classification models, such as SVMs. During testing, for each test vector an activation vector is computed using Eq.(2) which is then used as input to the trained classifier. Following three steps summarize the self-taught learning algorithm:

1. Compute basis vectors b_j using unlabeled data x_i^u by solving the optimization problem (1).

2. Fix the basis vectors and obtain activation vectors $a_i^\square$ for each labeled data vector x_i^l using Eq.(2).

3. Use activation vectors $a_i^\square$ as new labeled features to train standard supervised classifier. Transform each test vector into an activation vector (using Eq.(2) again) and apply the classifier to obtain its label.

4. EXPERIMENTS

4.1. Databases

As unlabeled data we used the GTZAN collection of musical pieces [1]. It consists of 1000 30-second audio clips, each belonging to one of the following ten genres: Classical, Country, Disco, Hip-Hop, Jazz, Rock, Blues, Reggae, Pop and Metal. There are 100 clips per genre and all of them have been downsampled to 22050 Hz. The other database which we used as labeled data is the corpus used in the ISMIR 2004 audio contest [11]. It contains of 729 whole tracks for training, but since the number of tracks per genre is non-uniform, the original nine genres are usually mapped in the following six classes: Classical, Electronic, Jazz-Blues, Metal-Punk, Rock-Pop and World. Another 729 tracks are used for testing. Note that the only common class between the two databases is the "Classical" genre.

Audio data from both databases are divided into 5 sec. pieces which were further randomly selected in order to make several training sets with different amount of data, keeping the same number of such pieces per genre. Table 1 summarizes the contents of the data sets we used in our experiments. For example, set GT-50 has 50 randomly selected 5 sec. pieces per genre, 500 pieces in total or 0.69 hours of music. All sets are constructed such that each larger set contains all the pieces from the smaller set. The test set consists of 250 pieces per genre randomly selected from the ISMIR 2004 test tracks.

Table 1. Data sets used in the experiments.

GTZAN database			ISMIR 2004 database		
Set	pieces	hours	Set	pieces	hours
GT-50	500	0.69	IS-20	120	0.17
GT-100	1000	1.39	IS-50	300	0.42
GT-250	2500	3.47	IS-100	600	0.83
GT-500	5000	6.95	IS-250	1500	2.08

4.2. Audio signal preprocessing

When it comes to feature extraction for music information processing, in contrast to the case of speech where the MFCC is dominant, there exists wide variety of approaches - from carefully crafted multiple music specific tonal, chroma, etc. features to single and simple "don't care about the content" spectrum. In our experiments, we used spectral representation tailored for music signals, such as Constant-Q transformed (CQT) FFT spectrum. The main property of this transform is the log-like frequency scale where the consecutive musical notes are linearly spaced [12].

The CQT transform is applied to FFT spectrum vectors computed from 23.2ms (512 samples) frames with 50% overlap in a way that there are 12 Constant-Q filters per octave resulting in a filter-bank of 89 filters which covers the whole bandwidth of 11025 Hz. The filter-bank outputs of 20 consecutive frames are further stacked into a 1780 (89x20) dimensional super-vector which is used in the experiments. This is the same as to have a 20 frame time-frequency spectrum image. There is a overlap of 10 frames between such two consecutive spectrum images. This way, each 5 sec. music piece is represented by 41 spectrum images or super-vectors.

4.3. Bases learning

For each dataset given in Table 1 we learned several basis vector sets or dictionaries as they are often called in the literature. The sets sizes are: 100, 200, 300 and 500. Contrary to the conventional sparse coding scheme, where the dictionary size is much bigger than the vectors dimension (for overcomplete representation), in our case we in fact do dimension reduction. This is motivated by the fact that our super-vectors are highly redundant and that basis vectors actually represent higher level spectral image features, not just arbitrary projection directions.

Before bases learning, all data are pooled and randomly shuffled. Then, 100 iterations of the learning algorithm (see Section 2) are performed. Figure 1 shows the first 68 of 200 basis vectors learned from GT-100 as 89x20 pixel spectrum images. Specific spectrum shapes and transitions are clearly captured by these bases.

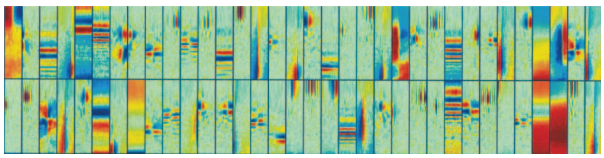

Fig. 1. Example of learned basis vectors shown as spectrum images.

4.4. Supervised classification

After all labeled training data, sets IS-20, IS-50, IS-100 and IS-250, have been transformed into activation vectors for each dictionary learned from each unlabeled data set, we obtained in total 64 (4 labeled datasets x 4 dictionary sizes x 4 unlabeled datasets) labeled training data sets. Then, using the LIBSVM tool, we learned 64 SVM classifiers each consisting of 6 SVMs trained in one-versus-all mode. The SVM input vectors were linearly scaled to fit the [0,1] range. This significantly reduces their sparsity, but it is tolerable since our goal is not the sparse representation itself. Linear kernel was used as distance measure and the SVMs were trained to produce probabilistic outputs.

During testing, each 5 sec. musical piece represented by 41 feature (activation) vectors is considered as a sample for classification. Outputs of all genre specific SVMs are aggregated (summed in the log domain) and the label of the maximum output is taken as the classification result.

4.5. Results

In order to assess the effect of the self-taught learning, we need a performance comparison with a system build under the same conditions but without unlabeled data. We will refer to this system as a baseline. In this case, the basis vectors are learned using labeled training data $\mathcal{X}^l$ instead of the unlabeled $\mathcal{X}^u$. Then, the activations are obtained in the same way as if the bases were learned from the unlabeled data. Table 2 shows the baseline system performance for several dictionary sizes and different amounts of training data. As can be expected, the more training data are available the better is the classification accuracy. It also increases with the size of the

Table 2. Baseline classification accuracy. Bases are learned from the labeled training data.

Training Set	Dictionary size			
	100	200	300	500
IS-20	52.5%	52.4%	51.5%	54.1%
IS-50	56.3%	58.6%	58.9%	60.9%
IS-100	57.0%	58.9%	60.1%	62.3%
IS-250	57.4%	59.9%	61.7%	64.0%

Table 3. Absolute improvement wrt baseline when bases are learned from the unlabeled GT-50 dataset.

Training Set	Dictionary size			
	100	200	300	500
IS-20	-1.67%	0.66%	4.60%	2.54%
IS-50	-1.06%	-1.00%	1.83%	-0.86%
IS-100	-2.00%	-1.67%	0.34%	-1.40%
IS-250	-1.40%	-0.60%	-0.20%	-0.33%

Table 4. Absolute improvement wrt baseline when bases are learned from the unlabeled GT-100 dataset.

Training Set	Dictionary size			
	100	200	300	500
IS-20	-0.67%	0.40%	3.40%	1.94%
IS-50	-2.06%	-0.06%	1.76%	2.74%
IS-100	-2.80%	-0.14%	-0.60%	1.14%
IS-250	-0.33%	0.00%	-0.27%	0.60%

Table 5. Absolute improvement wrt baseline when bases are learned from the unlabeled GT-250 dataset.

Training Set	Dictionary size			
	100	200	300	500
IS-20	-0.80%	0.20%	3.80%	3.67%
IS-50	-3.06%	-0.73%	1.56%	0.80%
IS-100	-2.73%	0.00%	-0.40%	-0.60%
IS-250	-2.07%	-0.60%	0.06%	-0.80%

Table 6. Absolute improvement wrt baseline when bases are learned from the unlabeled GT-500 dataset.

Training Set	Dictionary size			
	100	200	300	500
IS-20	-1.27%	0.46%	2.73%	3.20%
IS-50	-1.53%	-1.20%	0.16%	1.60%
IS-100	-2.13%	-1.07%	-0.60%	-0.40%
IS-250	-0.87%	-0.73%	-1.34%	0.33%

dictionary and this seems to be independent of the amount of training data except for the case with very few data: IS-20.

Tables 3-6 show the absolute difference of the classification accuracy between the self-taught learning based system and the baseline when basis vectors are learned with different amounts of unlabeled data: GT-50 to GT-250 respectively.

The results clearly show that for the smallest labeled training set, IS-20, consistent improvement is achieved with dictionary sizes 300 and 500 no matter how much unlabeled data are used for bases learning. As the amount of labeled training data increases, the effect of the self-taught learning gradually decreases, which is an expected result.

5. CONCLUSION

In this study, we investigated the effect of the self-taught learning algorithm when applied to the music genre classification task. Results of the experiments conducted under various conditions confirmed that when small size of labeled training is available, higher level features extracted from a bigger unlabeled dataset in an unsupervised manner via sparse coding can indeed help the supervised classification task.

6. REFERENCES

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

[2] R. Typke, F. Wiering, and R. C. Veltkamp, "A survey of music information retrieval systems," in *Proc. International Conference on Music Information Retrieval*, 2005, pp. 153–160.

[3] M. Casey, R. Veltkamp, M. Goto, M. Leman, C. Rhodes, and M. Slaney, "Content-based music information retrieval: Current directions and future challenges," *Proceedings of the IEEE*, vol. 96, no. 4, pp. 668–696, 2008.

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

[5] Plumbley M., Blumensath T., Daudet L., Gribonval R., and Davies M., "Sparse representations in audio and music: From coding to source separation," *Proc. IEEE*, vol. 98, no. 6, pp. 995–1005, 2010.

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

[7] K. Nigam, A. McCallum, S. Thrun, and T. Mitchell, "Text classification from labeled and unlabeled documents using EM," *Machine Learning*, vol. 39, no. 2-3, 2000.

[8] S. Pan and Q. Yang, "A survey on transfer learning," *IEEE Trans. Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1349–1359, 2010.

[9] R. Raina, A. Battle, H. Lee, B. Packer, and A. Ng, "Self-taught learning: Transfer learning from unlabeled data," in *Proc. ICML*, 2007.

[10] H. Lee, A. Battle, R. Raina, and A. Ng, "Efficient sparse coding algorithms," in *Advances in Neural Information Processing Systems (NIPS)*, 2007, vol. 19.

[11] P. Cano, E. Gomes, F. Gouyon, P. Herrera, M. Koppenberger, B. Ong, X. Serra, S. Streich, and N. Wack, "ISMIR 2004 audio description contest," Tech. Rep. MTG-TR-2006-02, Universitat Pompeu Fabra, 2006.

[12] C. Schoerkhuber and A. Klapuri, "Constant-Q transform toolbox for music processing," in *Proc. 7th. Sound and Music Computing Conference*, 2010.