Analyzing the Dictionary Properties and Sparsity Constraints for a Dictionary-based Music Genre Classification System

Ping-Keng Jao, Li Su and Yi-Hsuan Yang
Research Center for Information Technology Innovation, Academia Sinica, Taipei, Taiwan.
E-mail: {nafraw, lisu, yang}@citi.sinica.edu.tw Tel: +886-2-2787-2388

***Abstract*—Learning dictionaries from a large-scale music database is a burgeoning research topic in the music information retrieval (MIR) community. It has been shown that classification systems based on such learned features exhibit state-of-the-art accuracy in many music classification benchmarks. Although the general approach of dictionary-based MIR has been shown effective, little work has been done to investigate the relationship between system performance and dictionary properties, such as the dictionary sparsity, coherence, and conditional number of the dictionary. This paper aims at addressing this issue by systematically evaluating the performance of three types of dictionary learning algorithms for the task of genre classification, including the least-square based RLS (recursive least square) algorithm, and two variants of the stochastic gradient descent-based algorithm ODL (online dictionary learning) with different regularization functions. Specifically, we learn the dictionary with the USPOP2002 dataset and perform genre classification with the GTZAN dataset. Our result shows that setting strict sparsity constraints in the RLS-based dictionary learning (i.e., <1% of the signal dimension) leads to better accuracy in genre classification (around 80% when linear kernel support vector classifier is adopted). Moreover, we find that different sparsity constraints are needed for the dictionary learning phase and the encoding phase. Important links between dictionary properties and classification accuracy are also identified, such as a strong correlation between reconstruction error and classification accuracy in all algorithms. These findings help the design of future dictionary-based MIR systems and the selection of important dictionary learning parameters.**

I. INTRODUCTION

Online music service is one of the most essential entertainments in modern people's daily life. However, the vast amounts of digital music contents have made it difficult for people to find a preferred song in million-scale online music libraries. Music recommendation systems [1][2][3] aim to solve the above problem, either by utilizing human-tagged metadata, or by learning the musical contents such genres, emotions, instruments, and any other information. As a high-level descriptor, genre information suggests possible characteristics that help people to understand, retrieve or categorize music.

Music genre classification is one of the most widely-investigated topics in MIR field. Various approaches have been studies previously, for example, different classifiers such as K-nearest neighbor [4], Gaussian mixture models (GMM) [5], hidden Markov model (HMM) [6], linear discriminant analysis (LDA) [7], and support vector machines (SVM) [8], have been applied in the literature. In addition to classification method, feature representation is also important. For example, Lidy and Rauber [9] evaluated multiple feature extraction algorithms on genre classification and reported 7% to 15% accuracy difference.

In recent years, learning a dictionary (codebook) from a large database as a means to improving the musical feature representations has attracted increasing attentions. Such a dictionary-based approach converts low-level features (e.g., spectrum) of an input signal into a finite set of dictionary atoms using algorithms such as vector quantization (VQ) or L_1-regularized sparse coding (SC). This approach has been shown useful in various music information retreival problems [10][11].

Dictionary learning algorithms [12]-[15] can be generally categorized into non-incremental type and incremental (online) type. The former needs to read the whole training data at one time before learning, while the latter allows for updating the dictionary adaptively as new training data are received. Obviously, incremental type algorithms, such as recursive least square dictionary learning algorithm (RLS-DLA) [12] and online dictionary learning (ODL) [13], provide memory efficient solutions for a music database which is likely to be changed or extended rapidly. In consequence, we consider incremental type algorithms in this work.

In addition to merely evaluating the system performance by a testing dataset, we are interested in how the quality of a learned dictionary influences the classification accuracy. In particular, we are interested in the following issues: What are the optimal sparsity constraints in dictionary learning phases for these dictionary learning algorithms? How does the size of a dictionary influence performance? Is it possible to identify some dictionary properties that are indicative of the resulting system performance? Inspired by O'Hanlon and Plumbley's work [16], we investigate the relationships among various property indicators of a learnt dictionary, the quality of reconstruction, the codeword sparsity, and the accuracy in classifying music genre using the dictionary-based feature. To this end, we present an empirical performance study using the well-known GTZAN dataset.

The main findings of this paper include:

- Strict sparsity constraints in dictionary learning phase (i.e., <1% of the signal dimension) generally leads to better accuracy in genre classification.
- In contrast, we do not need strict sparsity constraints in the encoding phase.
- The value of some dictionary property indicators is correlated with the classification accuracy of a dictionary-based system.

The rest of this paper is organized as follows. Section II introduces the dictionary learning algorithms and the dictionary indicators. Section III gives an overview of used system. Section IV evaluates the parameters in the whole system, after which Section V explores the relationships among the dictionary properties and classification accuracy. Finally, Section VI concludes the paper.

II. Dictionary Properties

Dictionary learning algorithms aim at constructing a finite set of representative elements called *atoms* from a training database. In this section, we introduce three types of incremental dictionary learning algorithms, including the on-line dictionary learning (ODL) algorithm[1] [13] with two different objective functions, and recursive-least squares dictionary learning algorithm[2] (RLS-DLA) [12]. Moreover, we describe several indicators measuring the quality of a dictionary. We are interested in whether these indicators can be used to predict the performance of the corresponding dictionary for discriminating musical genre. If this is possible, we can automatically select the most promising dictionary for constructing the classification system.

A. Dictionary Learning

The objective functions of RLS-DLA and ODL are both:

$$D^* = \arg\min_D \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left\|x_i - D\alpha_i\right\|_2^2 \quad s.t.\ \left\|\alpha_i\right\|_0 \le \lambda_d\,, \tag{1}$$

where λ_d is the regularization parameter to be determined. *Smaller λ_d represent more strict constraint*, and there is no universal value for λ_d. One has to empirically determine the best value through evaluation. The dictionary $D \in R^{m\times k}$ is to be learnt, and $k > m$ in general, and each column d_j is normalized to unit Euclidean norm. The $\alpha_i \in R^k$ is a column vector approximates an input datum x_i by (a few) columns of D, and is referred to as codeword in this paper. Although the hard constraint of L_0-norrm leads the problem to be NP-hard [17], this can be solved in polynomial time by a greedy algorithm based on orthogonal matching pursuits (OMP) [18]. As the objective function is not convex, both RLS-DLA and ODL adopt a sub-optimal alternating minimization strategy, fixing D while optimizing α and vice versa, to make the objective convex when fixing one variable.

[1] http://spams-devel.gforge.inria.fr/

[2] http://www.ux.uis.no/~karlsk/dle/index.html

Both RLS-DLA and ODL algorithms build a dictionary by an incremental approach, updating a dictionary by newly-collected data instead of the whole dataset. This is a memory-efficient solution when we are given a large amount of data for learning the dictionary.

The two algorithms differ in the updating strategies. RLS-DLA updates the D iteratively by

$$D^{(i)} = B^{(i)} A^{(i)^{-1}}. \tag{2}$$

This product of matrix B and the inverse of matrix A is the solution for the objective value in (1) when all α_i are given [15]. The two matrices B and A are defined as:

$$A^{(i)} = \gamma A^{(i-1)} + \alpha_i \alpha_i^T\,, \tag{3}$$

$$B^{(i)} = \gamma B^{(i-1)} + x_i \alpha_i^T\,. \tag{4}$$

Matrix A is the product between each element of a codeword α_i, and matrix B is the product between each element of an input datum, x_i and the corresponding codeword, α_i. The superscript i denotes the iteration step, i.e., the corresponding state when given i^{th} datum. The parameter $\gamma \in (0, 1]$ is the forgetting factor, whose purpose is to de-emphasize the effects of previous solution, since the choice of initial solution should not affect the final solution.

On the other hand, ODL updates each atom by stochastic approximation, a family of steepest descent optimization. The update rule is as follows,

$$d_j^{(i)} = d_j^{(i-1)} + \frac{1}{A^{(i)}_{[j,j]}}\left(b_j^{(i)} - D a_j^{(i)}\right). \tag{5}$$

In (5), d_j is the j^{th} atom (column) of D, and $A^{(i)}_{[j,j]}$ is the element in j^{th} row and j^{th} column of matrix A defined in (3). The lower case symbol b is the column vector of B defined in (4). In ODL, the forgetting factor is not considered.

Besides above algorithmic difference, ODL also supports other objective functions, including the following one,

$$D^* = \arg\min_D \frac{1}{n}\sum_{i=1}^{n}\frac{1}{2}\left\|x_i - D\alpha_i\right\|_2^2 + \lambda_d\left\|\alpha_i\right\|_1. \tag{6}$$

Unlike (1), λ_d in (6) uses a soft regularization parameter over α_i, instead of a hard constraint. Larger λ_d in (6) forces α_i sparser, whereas smaller λ_d forces α_i denser. In the following discussion we refer to the ODL algorithm based on formulation (1) as ODL1, and the one based on formulation (6) as ODL2, respectively. Please note that because (6) takes a soft regularization form, in ODL2 the LARS-Lasso algorithm [19] is used to solve α_i instead of using OMP.

B. Indicators

The goal of dictionary learning is to find suitable atoms to approximate the data. To define the quality of dictionary, it is straightforward to measure the quality of reconstruction error for applications aiming at perfect reconstruction, e.g. image

and audio compression. However, for other applications such as classification or similarity estimation, low reconstruction error does not guarantee good performance, because of the difference in objective function. For example, for genre classification, the dictionary of interest is able to help extract the common characteristic in the same genre.

To examine the property of learnt dictionaries, in this study we resort to the measurements employed in a recent work [16]. The first indicator is the coherence μ defined as

$$\mu = \max_{i \neq j} \left| d_i^T d_j \right|. \tag{7}$$

Coherence is the maximum absolute value of inner product between different normalized atoms. This is a similarity indicator of a dictionary D. Larger μ indicates higher similarity between the dictionary atoms. However, this is only an estimation, as it considers only the maximum value.

The second indicator is the condition number κ:

$$\kappa(D) = \frac{\sigma_{\max}(D)}{\sigma_{\min}(D)}, \tag{8}$$

where σ_{max} and σ_{min} stand for the maximum and minimum singular value of D, respectively. It can be found that κ is also an indicator of extreme value, instead of the whole picture. However, the condition number has been considered important in numeric analysis, as it indicates whether the matrix is stable, or how sensitive the matrix is. A matrix D with condition number $\gg 1$ is often called ill-conditioned, and well-conditioned otherwise. In other words, larger condition number may lead to larger variations even when two different data are very close. As similar input data should normally be associated to the same genre, one might expect that a small value would lead to better accuracy.

The third indicator, *redundancy*, is defined as

$$\|D^T D - I\|_F, \tag{9}$$

where D^TD is the Gram matrix of normalized D, and I is the identity matrix. The Frobenius norm is used to summarize the similarity between all different atoms, with larger $\|D^TD-I\|_F$ indicating the atoms in D are more similar. The subtraction of I is to remove the self-correlation of each d in D^TD, since we only consider the relationship between different atoms. This indicator can measure the redundancy of the dictionary, for it equals to zero when the codewords in the dictionary D are perfectly orthogonal.

In addition to the aforementioned indicators, we also study the relationship between the accuracy of genre classification and combinations of the reconstruction error and sparsity of codewords.

III. System Overview

Fig. 1 shows the block diagram of the implemented dictionary-based genre classification system. At the outset, we extract spectrograms from two music datasets, USPOP2002 and GTZAN, which are used for dictionary learning and classifier training/test, respectively. To reduce the computational power, the codewords are pooled along with short-time frames, giving rise to the so-called bag-of-frames (BoF) features. Moreover, as proper normalization might help reduce the effects of outliers, we considered two normalization techniques before applying support vector machine for classification.

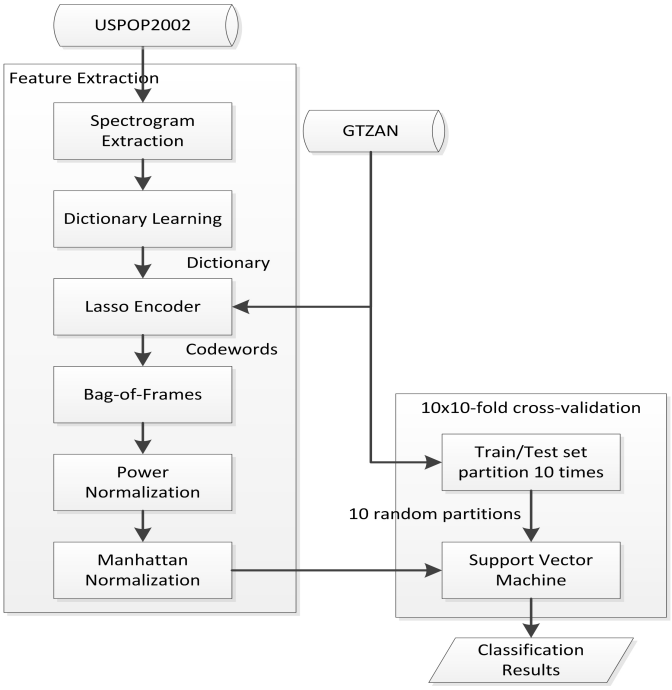

Fig. 1. Framework of Algorithm and Experimental Settings.

A. Spectrogram

As Fig. 1 shows, the first phase of the feature extraction process computes the spectrogram of each audio file in USPOP2002 and GTZAN dataset. The sampling rate of each file is 22.05 kHz. A spectrogram is obtained by computing decibel of short-time Fourier transform (STFT) by using a window function with 1025 samples (46.5 ms) in window size and 512 samples in hop size, resulting in a 513-dimensional spectral feature (plus the DC term).

B. Sparse Coding

Given a dictionary $D \in R^{m \times k}$, sparse coding seeks the vector $\alpha \in R^k$ with minimal number of non-zero weighting coefficients that approximates input vector $x \in R^m$ by atoms in the dictionary D. The sparse coding problem can be formulated as

$$\alpha^* = \arg\min_{\alpha} \left\| x - D\alpha \right\|_2^2 + \lambda_c \left\| \alpha \right\|_1, \tag{10}$$

where λ_c controls the trade-off between approximation error and sparsity. Please note that the value of λ_c can be different from the value of λ_d in (1) and (6). The sparse property makes it possible to capture the most "important" characteristic of x

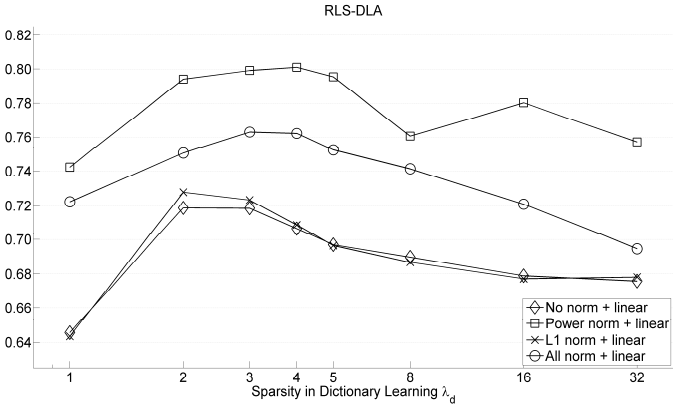

Fig. 2 Averaged accuracies of four configurations in normalizations with linear kernel in SVM for RLS-DLA bag-of-frames features. Diamond marks: no normalization. Square marks: power normalization only. X-marks: L_1 normalization. Circle markers: power normalization and L_1 normalization.

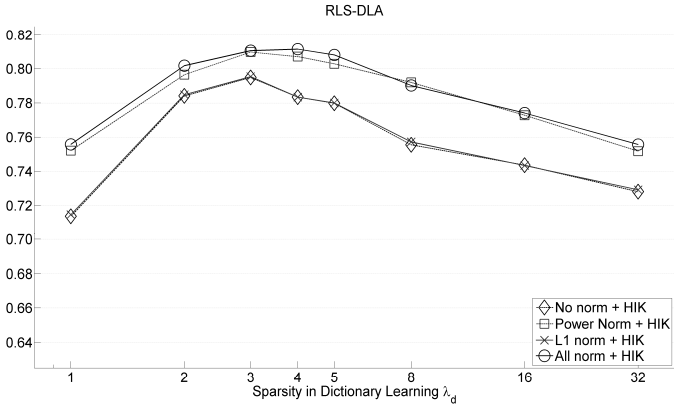

Fig. 3. Averaged accuracies of four configurations in normalizations with HIK kernel in SVM for RLS-DLA bag-of-frames features. Diamond marks: no normalization. Square marks: power normalization only. X-marks: L_1 normalization. Circle markers: power normalization and L_1 normalization.

with prominent atoms in D. Different from λ_d, we set the value of λ_c to λ_0, the inverse of square root of m, the value recommended in [13] for it leads to good performance both empirically and theoretically.

C. Bag-of-Frames

Temporal pooling has been shown useful in music and audio processing [20][21][22]. Obviously, the concept of genre can be identified only when the temporal information is accumulated for at least several seconds. Therefore, just concerning short-time frames independently may not be enough. Furthermore, temporal pooling would reduce the computational complexity for later process, where the size of feature representing a song reduces from a matrix to a column vector as we sum up the codewords α over the whole sequences of each song.

D. Normalization and Support Vector Machine

We adopted LIBSVM for the implementation of support vector machine [23], and the used kernels of SVM are histogram intersection kernel (HIK) and linear kernel, and we adopt power normalization and Manhattan normalization serially to enhance the distribution of features before classification. Power normalization is a transformation that makes data distributes more like Gaussian distribution by a power function. Given a datum $x_i \in R^k$, a typical power normalization technique outputs $\text{sign}(x_i)|x_i|^p$, where $\text{sign}(\cdot)$ is a sign function and $p \in [0, 1]$. In this paper, we set p to 0.5, which reduces to square root of absolute value of x with original sign. This technique has been shown useful in [20]. The Manhattan normalization here is L_1-normalization of data x_i.

For readability, in all the figures we use "No norm" to denote the case of no normalization, and use "All norm" for power normalization plus L_1 normalization.

IV. Evaluations

In this section, we first introduce the evaluation datasets. The first part of the experiments focuses on various post-processing methods and SVM kernels of the bag-of-frames features learned from RLS-DLA dictionary. Then, we compare different dictionary learning algorithms and objective functions. Extensive experiments on sparsity parameter λ_d in both dictionary learning algorithms and dictionary size k are also performed. Moreover, we also studied the influence of the encoding sparsity constraint λ_c.

All experiments are performed under a ten-fold cross-validation scheme, repeated for 10 times for each experiment. Each fold has the same number of training (testing) songs for each genre. All training and testing sets are fixed over all experiments for no bias. HIK and linear kernel for SVM are both investigated with the tuning parameter C swept from 2^{-10} to 2^{10}, totaling four configurations over two selected normalization techniques are tested in most cases.

A. Datasets

Since transductive dictionary learning is potentially biased [24], we used two different datasets for dictionary learning and genre classification. Specifically, we used USPOP2002 to train the dictionary and used the resultant dictionary to generate the features for predicting the genres in GTZAN dataset. Both datasets have different formats, and we converted all songs to standard mono-channel with 22,050 Hz sampling rate in WAV.

USPOP2002 were collected with 6700 preview audio from 7digital[3] according to the list of [25]. The length of the audio files ranges from 30 to 60 seconds. On the other hand, The GTZAN dataset consists of 1,000 30-second clips with 10 genres included, and there are 100 clips for each genre. These genres are blues, classical, country, disco, hip-hop, jazz, metal, pop, reggae, and rock. Although this dataset is the most widely-used dataset in evaluating genre classification, 10.6% of the files have been found mislabeled recently [26].

B. SVM Kernels and Normalization Methods

[3] http://us.7digital.com/

By comparing Fig. 2 and Fig. 3, we see that combining HIK kernel with both normalization methods leads to the best accuracy in most cases for RLS-DLA. Although the case with linear kernel + power normalization is the most accurate when $\lambda_d = 16$, the accuracy is still behind HIK + both normalization methods at $\lambda_d = 4$. The setting "All Norm + HIK" is found significant better than all other settings ($p < 0.01$, d.f. = 198, under a two-tail t-test), except for Power Norm + HIK.

Linear kernel is comparable to histogram intersection kernel only when power normalization is performed, suggesting that histogram-like kernel better suites our data. Power normalization improves the accuracy a lot as it remedies the problem of large outliers and the majority of small-valued data. The classification process then benefit from this as long as the distribution of different genres overlap less.

C. Different Algorithms and Objective Functions

Fig. 4 (a) illustrates the results of RLS-DLA and ODL1 and ODL2 for linear kernel with power normalizations. We can see from both Figs. 2 and 3 (a) that a loose constraint (larger λ_d) generally results in worse accuracy for RLS-DLA, regardless of the learning kernel. Better accuracy is achieved with stricter sparisty, except for the case when λ_d equals 1. It seems that setting a strict constraint forces RLS-DLA to learn more "prominent" features in different genres. Best result is obtained with $\lambda_d = 4$, which is less than 1% dimension of signals ($4/513 \approx 0.8\%$).

We cannot obtain fair classification accuracy with $\lambda_d = 1$, possibly because RLS-DLA would degenerate into K-means algorithm [11] in such as case.

As for ODL1, we see that larger λ_d is preferred. Even though RLS-DLA and ODL1 use the same objective function, the performance trend of ODL1 is different from that of RLS-DLA. Fig. 4 (a) suggests that for ODL1, the accuracy is positively correlated with λ_d. However, we do not observe strong performance difference between different sparsity levels for this algorithm. Considering that the major differences between these two algorithms are the forgetting factor in RLS-DLA and the dictionary update strategy, and that we have set the forgetting factor close to one (which reduces the effect of the forgetting factor), we infer that ODL1 is less sensitive to the degree of sparsity constraint because of dictionary update strategy.

Fig. 4 (b) shows the average accuracies with different λ_d. Because larger λ_d stands for strict constraint in ODL2, λ_d goes from large values to small ones in this figure. In addition, λ_d is in terms of the multiple of λ_0, the suggested optimal value in [13]. We can see that the performance trend of ODL2 is similar to that of RLS-DLA; imposing a strong emphasis on sparsity degrades the classification accuracy. The optimal λ_d turns out to be λ_0, confirming the suggestion in [13].

D. Influences of Dictionary Size

In addition to the sparsity, the dictionary size also plays an important role in dictionary-based framework. Fig. 5 depicts the results by varying k, graphed in four lines standing for two dictionary learning algorithms (RLS-DLA and ODL1) and two post-processing settings (Power Norm + Linear and L_1 Norm + linear). The regularization parameter λ_d is set to 4 for both ODL1 and RLS-DLA. We can see that using larger k improves the classification accuracy, which is expected since larger dictionary typically contains more information. Nevertheless, the rate of growth is not significant for some cases, such as RLS-DLA with L_1 normalization and linear kernel.

The rate converges quickly possibly due to limited testing data, where only 1,000 songs are included for training and testing, and each song contains only one vector, due to bag-of-frames, as the feature.

E. Encoding Sparsity

Fig. 6 depicts the result of RLS-DLA, ODL1 and ODL2 with power normalization and linear kernel with various λ_c, the regularization parameter of LASSO encoder. In addition to the classical $\lambda_c = \lambda_0$ in this experiment, we also examined the value from $0.1\lambda_0$ to $10\lambda_0$. Results show that loose constraint (smaller λ_c) generally works better. Serious degradation may be found when too strict L_1-norm constraint is applied, e.g., ODL1. This trend is different for RLS-DLA in dictionary learning, showing that RLS-DLA prefers strict constraint but loose constraint is desired in sparse coding. It

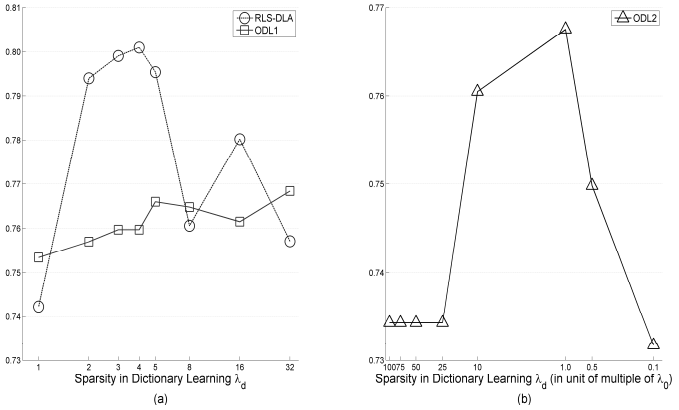

Fig. 4. The averaged accuracies of (a) RLS-DLA, ODL1 and (b) ODL2 with power normalization and linear kernel are employed. Where λ_0 in the right figure is $1/\sqrt{513}$

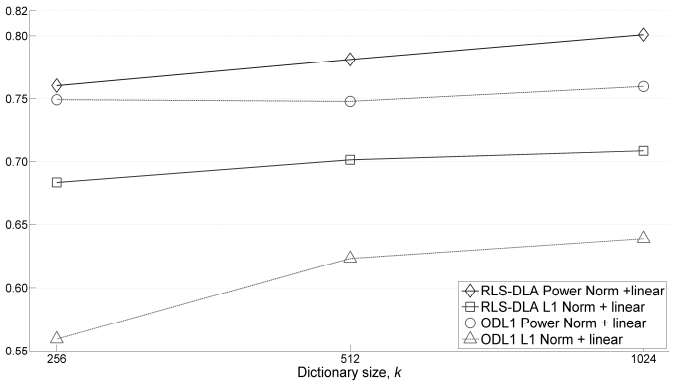

Fig. 5. Influence of dictionary size (horizontal axis) of RLS-DLA and ODL1 algorithms with respect to averaged accuracies (vertical axis).

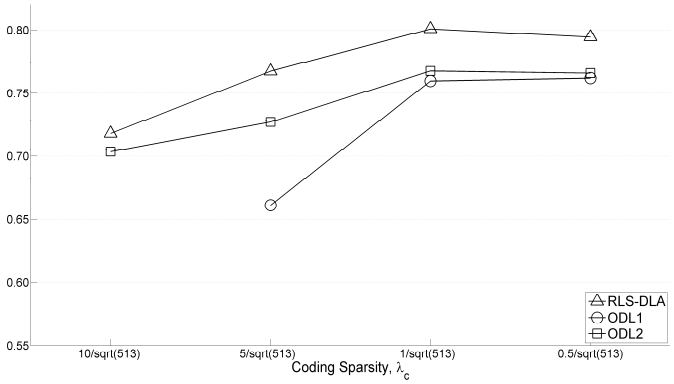

Fig. 6. Influence of coding sparsity, λ_c (horizontal axis) with respect to averaged accuracies (vertical axis). Linear kernel and power normalization are used.

perhaps implies that SVM favors small λ_d because that makes the dictionary learning algorithm RLS-DLA focuses on only a subset of (representative) atoms. Also, the trend of λ_c provides two possible implications. First, SVM can easily identify a musical genre using the proposed codewords, if the genre is associated with a specific set of atoms. Second, the number of atoms associated with a genre might not be sparse. If an overly strict constraint is applied in the encoding process, SVM may not obtain sufficient information to discriminate different genres. In consequence, a loose λ_c enriches the feature representation, making it possible to include atoms that are specifically associated with each genre.

V. Probing the Trend

After extensive evaluations of different dictionary learning algorithm with many parameters, we probed with indicators of dictionary and qualities of signal reconstruction to disclose the correlation with accuracy.

A. Analysis of Dictionary Properties

We first analyzed the correlation between the indicators introduced in Section II and the accuracies obtained from different λ_d. The experimental results used here are all obtained by power normalization with linear kernel.

Left part of TABLE I lists the correlations between each selected indicator and each accuracy using RLS-DLA, ODL1 and ODL2. The entries marked with asterisk represent significant correlation between the indicator and accuracy. The detailed trend of the two variables with the strongest correlation (ODL1 and condition number) is shown in Fig. 7. Obviously, low condition number in ODL1 stands for good accuracy in most cases, as expected by the discussion in Section II. However, such phenomenon is not found in the other two algorithms. It is possible that condition number works fine as an indicator only when it is in a proper range. Because, as given in Fig. 7, the highest value is less than 90, the dictionaries should be well-conditioned. We find in our data that the range of κ varies from 786 to 21860 for RLS-DLA, and from 345 to 551 for ODL2. Such high values imply that the dictionaries are relatively ill-conditioned. Why high κ becomes an unreliable predictor of accuracy for given matrices needs further investigation. We also notice that even RLS-DLA has the largest range of condition number, RLS-DLA has the best accuracy among all. Therefore, we can only infer that κ can be used as a proper indicator only when the value is appropriate and for matrices trained in the same way.

As can be read from TABLE I, the coherence seems to be too brief to be an informative measurement. In contrast, the redundancy of the dictionary ($\|D^TD - I\|_F$) is negatively correlations with the accuracy for all algorithms (same trend); The correlation is significant for the case of RLS-DLA. This indicates that a dictionary achieves higher accuracy when most of atoms are not similar and thus contains more information. However, ODL2 has a low correlation in this indicator, probably due to the soft constraint formulation.

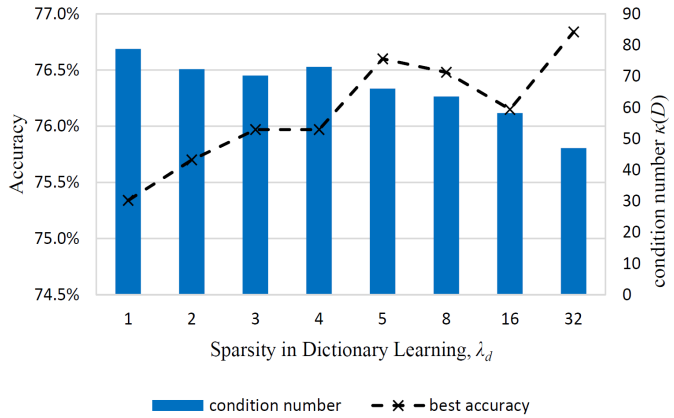

Fig. 7. Dictionary sparsity λ_d, versus the condition number $\kappa(D)$ (bars) and the classification accuracy (dashed line), using ODL1 for dictionary learning.

B. Analysis of Reconstruction Error

Besides the dictionary atoms, we sought for the relation between the codewords and the accuracy. In this subsection, regularization parameter of encoding, λ_c, is set to λ_0 in all experiments and analysis. We computed several indicators related to objective functions and constraints used in dictionary learning for all codewords of GTZAN, including $\|x\text{-}D\alpha\|_F$, which is the Frobenius norm of reconstruction error, $\|\alpha\|_1$, and $\|x\text{-}D\alpha\|_F+\lambda_c\|\alpha\|_1$.

Right-hand side of TABLE I gives the correlations between

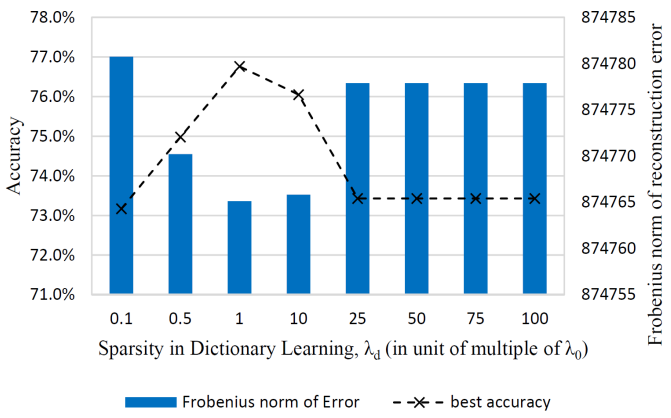

Fig. 8. Dictionary sparsity λ_d, versus reconstruction error $\|x\text{-}D\alpha\|_F$ (bars) and classification accuracy (dashed line), using ODL2 for dictionary learning.

TABLE I
Correlation coefficients between 1) High level dictionary properties and classification accuracy, 2) Objective values and classification accuracy.

	Dictionary Properties			Objective Values		
	μ	$\kappa(D)$	$\|\|D^TD - I\|\|_F$	$\|\|x-D\alpha\|\|_F$	$\|\|\alpha\|\|_1$	$\|\|x-D\alpha\|\|_F+\lambda_c\|\|\alpha\|\|_1$
RLS-DLA	-0.0884	0.1291	-0.7986*	-0.8721**	0.8932**	0.8924**
ODL1	-0.6105	-0.8422 **	-0.7008	0.8115*	0.7764*	0.7769*
ODL2	-0.3287	-0.1911	-0.2717	-0.9870**	0.5013	0.4999

Note: the entries marked with * stand for $p \leq 0.05$ and ** for $p \leq 0.01$

accuracies and the indicators. The reconstruction error shows the strongest correlation among all indicators. Detail behavior of ODL2 is shown in Fig. 8. Both RLS-DLA and ODL2 shows a negative correlation while ODL1 shows a positive correlation. For RLS-DLA and ODL2, higher accuracy benefits from lower error, because acceptable low reconstruction error is required in a classification problem. Surprisingly, ODL1 exhibits an opposite trend, implying that perfect reconstruction is not necessary in all cases. The reason of this might stem from the properties of musical genre. For example, different songs would share similar contents if they are associated with the same genre. Such similar characteristics can be recognized by human listeners and hopefully also by an automatic classification system. This application then tends to extract the characteristic for each genre rather than using original songs for classification.

Because larger L_1-norm generally leads to less error, we expect that if smaller reconstruction error leads to higher accuracy, so does higher L_1-norm. This can be seen from the result of RLS-DLA and ODL2. Moreover, the last column shows that L_1-norm dominates the reconstruction error in correlation, since the sign and the values are almost the same as L_1-norm for three algorithms.

VI. CONCLUSIONS

In this paper, we have compared three different algorithms, and all of them exhibit different trends, even RLS-DLA and ODL1 share the same objective function. This suggests none of them definitely dominates others in the classification system. In other words, the widely-adopted objective functions in sparse-coding may not have a direct and relevant relation in accuracy. Optimal λ_d is between 3 and 5 for RLS-DLA, and the suggested value in [13] for ODL2. On the other hand, a loose sparsity constraint is preferred for codewords to capture more information about the signal for classification. That is, different sparstiy constraints are needed for dictionary learning, especially different algorithms, and encoding. Our analysis shows that the condition number of the dictionary is a reliable indicator for ODL1, whereas the redundancy of the dictionary is suitable for RLS-DLA. In addition, the reconstruction error is highly correlated with the classification accuracy for all the three algorithms, but the trend for ODL1 is in the opposite direction against the case for the other two. Therefore, we may use reconstruction error as a predictor only when the trend is discovered in advance.

VII. ACKNOWLEDGEMENT

This work was supported by the National Science Council of Taiwan under Grant NSC 102-2221-E-001-004-MY3 and the Academia Sinica Career Development Award.

REFERENCES

[1] B. Logan, "Music recommendation from song sets," in *Proc. ISMIR*, 2004.

[2] P. Cano, M. Koppenberger, and N. Wack, "Content-based music audio recommendation," in *Proc. ACM Int. Conf. Multimedia*, pp. 211−212, 2005.

[3] J.-H. Su, H.-H. Yeh, P.S. Yu, and V.S. Tseng, "Music recommendation using content and context information mining," *IEEE Intelligent Systems,* vol. 25, no.1, pp.16−26, 2010.

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

[5] J.J. Burred and A. Lerch, "A hierarchical approach to automatic musical genre classification," in *Proc. Int. Conf. on Digital Audio Effects*, 2003.

[6] N. Scaringella and G. Zoia, "On the modeling of time information for automatic genre recognition systems in audio signals", in *Proc. ISMIR*, 2005.

[7] K. West and S. Cox, "Finding an optimal segmentation for audio genre classification", in *Proc. ISMIR*, 2005.

[8] R. Tao, Z. Li, and Y. Ji, "Music genre classification using temporal information and support vector machine," In *Proc. ASCI Conf.*, 2010.

[9] T. Lidy and A. Rauber. "Evaluation of feature extractors and psycho-acoustic transformations for music genre classification," in *Proc. ISMIR*. 2005.

[10] J. Nam, J. Herrera, M. Slaney, and J. Smith, "Learning sparse feature representations for music annotation and retrieval," in *Proc. ISMIR*, 2012.

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

[12] K. Skretting and K. Engan, "Recursive least squares dictionary learning algorithm," *IEEE Trans. Signal Processing,* vol. 58, no. 4, pp. 2121−2130, 2010.

[13] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online dictionary learning for sparse coding," in *Proc. ICML*, pp. 689–696, 2009.

[14] M. Aharon, M. Elad, and A. Bruckstein. “K−SVD: Design of dictionaries for sparse representation.” in *Proc. of SPARS*, pp. 9−12, 2005.
[15] K. Engan, K. Skretting and J. H. Husøy, “Family of iterative LS-based dictionary learning algorithms, ILS-DLA, for sparse signal representation,” *Digital Signal Processing*, vol. 17, no. 1, pp. 32−49, 2007.
[16] K. O’Hanlon and M. D. Plumbley, “Automatic music transcription using row weighted decompositions,” in *Proc. ICASSP*, 2013.
[17] G. Davis, “Adaptive nonlinear approximations,” Ph.D. dissertation, New York University, Sep. 1994.
[18] M. Gharavi-Alkhansari and T. S. Huang, “A fast orthogonal matching pursuit algorithm,” in *Proc. ICASSP*, pp. 1389−1392, 1998.
[19] B. Efron, T. Hastie, I. Johnstone and R. Tibshirani, “Least angle regression,” *Annals of Statistics*, vol. 32, pp. 407−499, 2004
[20] C.-C. Yeh, L. Su and Y.-H. Yang, “Dual-layer bag-of-frames model for music genre classification,” in *Proc. ICASSP*, 2013.
[21] J.-J. Aucouturier, B. Defreville, and F. Pachet, “The bag-of-frames approach to audio pattern recognition: A sufficient model for urban soundscapes but not for polyphonic music,” *J. Acoustical Society of America*, vol. 122, no. 2, pp. 881–891, 2007.
[22] L. Su, C.-C. M. Yeh, J.-Y. Liu, J.-C. Wang, and Y.-H. Yang, “A systematic evaluation of the bag-of-frames representation for music information retrieval,” *IEEE Trans. Multimedia,* accepted.
[23] C.-C. Chang and C.-J. Lin, “LIBSVM: a library for support vector machines,” *ACM Trans. Intelligent Systems and Technology*, Article 27, 2011.
[24] K. Markov and T. Matsui, “High level feature extraction for the self-taught learning algorithm,” *EURASIP J. Audio, Speech, and Music Processing*, 2013.
[25] D. Ellis, A. Berenzweig and B. Whitman. “The "uspop2002" pop music data set,” Web resource, 2003. Available at: http://labrosa.ee.columbia.edu/projects/musicsim/uspop2002.html.
[26] B. L. Sturm, “An analysis of the GTZAN music genre dataset,” in *Proc. Int. ACM workshop on MIRUM*, pp. 7–12, 2012.