DATA DRIVEN AND DISCRIMINATIVE PROJECTIONS FOR LARGE-SCALE COVER SONG IDENTIFICATION

Eric J. Humphrey, Oriol Nieto, Juan P. Bello
Music and Audio Research Laboratory
New York University
{ejhumphrey, oriol, jpbello}@nyu.edu

ABSTRACT

The predominant approach to computing document similarity in web scale applications proceeds by encoding task-specific invariance in a vectorized representation, such that the relationship between items can be computed efficiently by a simple scoring function, e.g. Euclidean distance. Here, we improve upon previous work in large-scale cover song identification by using data-driven projections at different time-scales to capture local features and embed summary vectors into a semantically organized space. We achieve this by projecting 2D-Fourier Magnitude Coefficients (2D-FMCs) of beat-chroma patches into a sparse, high dimensional representation which, due to the shift invariance properties of the Fourier Transform, is similar in principle to convolutional sparse coding. After aggregating these local beat-chroma projections, we apply supervised dimensionality reduction to recover an embedding where distance is useful for cover song retrieval. Evaluating on the Million Song Dataset, we find our method outperforms the current state of the art overall, but significantly so for top-k metrics, which indicate improved usability.

1. INTRODUCTION

Cover song identification is a well-established task in the MIR community, motivated by both theoretical and practical interest. On one hand, a “cover” is an abstract form of musical variation and presents a challenging computer audition problem. Alternatively, music collections continue to expand to unprecedented volumes, particularly in terms of amateur and user-generated content. As evidenced by even a brief review of websites like YouTube [1], Vimeo [2], or Soundcloud [3], a considerable portion of online musical content now consists of covers.

In light of this, previous research in cover song identification explores a variety of approaches, including the cross-correlation of beat-synchronous chroma features [4], dynamic time warping on binary chroma similarities [10], cross-recurrence quantification [11], etc. For a comprehensive review, the reader is referred to [9]. Over time, it has been shown that these methods can achieve robustness to specific kinds of musical variation, e.g., tempo changes, differences in structure, or key transpositions. In practice however, making use of these non-trivial operations yields complex systems that are computationally prohibitive to evaluate, let alone deploy, on large music databases.

Recognizing this limitation, recent work in cover song retrieval explores a slightly different approach to the task [1]. Rather than attempting to resolve irrelevant musical variation in the process of comparing two tracks, this particular system tries to encode this invariance directly with a multi-stage, feed-forward architecture. Local beat-chroma patterns are efficiently transformed into shift-invariant features via the 2D-Fourier Transform, median-pooled over time into a summary representation, and projected into a PCA subspace. Having transformed a collection of tracks into a much lower dimensional space, pairwise comparisons can be efficiently computed by Euclidean distance. As a result, this approach scales well to large collections like the Million Song Dataset [4] (MSD), and offers a promising research direction for pursuing general web-scale music similarity.

Here, we seek to advance this initial work by improving the feed-forward architecture to yield better representations for cover song retrieval. After fine-tuning the previously developed system, we propose two major modifications: sparse, high-dimensional data driven component estimation to improve separability, and supervised dimensionality reduction to recover a cover-similarity space. Our initial analysis on a training subset shows how the combination of sparse projections and supervised embeddings can lead to better organized spaces and improve cover song retrieval. Interestingly, evaluating on the MSD results in two notable findings: one, that our approach significantly improves performance at top-k metrics; and two, though our supervised embedding can be prone to over-fitting, PCA subspaces help alleviate this issue.

The remainder of this paper is organized as follows. Section 2 formally motivates and introduces the approach in [2] upon which our work is based, while Section 3 de-

[1] http://youtube.com
[2] http://vimeo.com
[3] http://soundcloud.com
[4] http://labrosa.ee.columbia.edu/millionsong/

tails our proposed modifications. In Section 4 we present results on the development set and give detailed analyses on the impact of each proposed modification. Section 5 discusses results on the MSD and tests strategies for minimizing the generalization error. Finally, Section 6 draws conclusions and advances a number of ideas for future work.

2. SCALABLE COVER SONG RETRIEVAL

2.1 Problem Formulation

Expressed symbolically, cover song retrieval proceeds by determining the relationship $S_{i,j}$ between a query A_i and reference track B_j via the composite of feature extraction f and a pairwise comparison[5] function g:

$$S_{i,j} = g(f(A_i), f(B_j)) \tag{1}$$

Note then that computing the full comparison matrix S between a set of Q queries against a collection of R reference tracks requires a double-for loop, and the total computational cost $\mathcal{C}_S$ is expressed as $QR\bar{\mathcal{C}}_{S_{i,j}}$, where $\bar{\mathcal{C}}_{S_{i,j}}$ is the expected cost of computing a single pairwise relationship. However, when f and g are independent, feature extraction can be performed separately, and the total computational load can be re-written as follows:

$$\mathcal{C}_S = QR\bar{\mathcal{C}}_g + (Q+R)\bar{\mathcal{C}}_f \tag{2}$$

Importantly, though the average comparison cost $\bar{\mathcal{C}}_g$ scales quadratically, the start-up cost of feature extraction $\bar{\mathcal{C}}_f$ is linear. The intuition for this trick is a common optimization in software engineering —minimize the amount of computation inside for-loops— and pinpoints the fundamental deficiency of many cover song retrieval systems: comparison functions often rely on expensive operations like cross-correlation or dynamic time warping, which *must* remain inside a nested for-loop. Thus, scalable cover song retrieval necessitates choosing an efficient comparison function g; the challenge then becomes one of designing the feature extraction stage f so as to maximize the accuracy of the rankings according to S.

2.2 Relating to Previous Work

To these ends, the authors of [1] propose an astute solution to this challenge. Intuitively, cover song retrieval algorithms are designed to be invariant to time and key transpositions. One cleverly efficient way of achieving this behavior is by computing the 2-dimensional Discrete Fourier Transform (2D-DFT) of local patches of beat-synchronous chroma features and keeping only the magnitude coefficients. Whereas the phase component of the 2D-DFT encodes circular rotations in time and pitch class, 2D-Fourier Magnitude Coefficients (2D-FMC) capture these patterns regardless of absolute position. As shown by [8] in the context of rhythm analysis, the DFT is sensitive to the order of events in a sequence, where the addition of different sinusoids results in patterns of cancellations that affect the magnitude coefficients.

Describing holistically, the system presented in [1] defines f as a feed-forward embedding function that operates at multiple time scales. First, 2D-FMC are computed on a moving window of 75 beat-synchronous chroma vectors, with a 1 beat hop size. A track is then pooled over time by taking the coefficient-wise median across all 2D-FMC vectors and L_2-normalized. Having sampled a collection of summary 2D-FMC vectors, PCA is performed and used to embed tracks in a low dimensional subspace; the authors experimentally found that preserving anywhere between 50 and 200 principal components returns better results. Importantly, once tracks are embedded in this feature space via f, they define g as Euclidean distance to efficiently compute pairwise comparisons.

3. IMPROVING FEATURE EXTRACTION

Starting from the work presented in [1], we now propose a series of modifications to make the feature extraction process more robust and improve cover song retrieval, outlined in its entirety in Figure 1. First, we discuss various data pre-processing strategies, including non-linear scaling and normalization. Next, we describe our approach to data driven component estimation, addressing its motivation and conceptual parallels to recent developments in information processing strategies. Lastly, a supervised learning stage is introduced to realize an embedding where summary representations of covers are significantly closer.

3.1 Data Pre-processing

As an initial step, here we apply three operations to the 2D-FMC representation preparing it for further processing: logarithmic compression, vector normalization, and dimensionality reduction via PCA. Expressed formally, the first two are achieved by

$$\hat{X} = \log\left(\frac{CX}{\|X\|_2} + 1\right) \tag{3}$$

where C is a constant hyperparameter, X is a 2D-FMC vector, and $\|\cdot\|_2$ is the L_2-norm. We empirically observed that L_2-normalization followed by log-scaling with $C = 5$ yields slightly better results than the inverse order with $C = 100$. Intuitively speaking, while log-compression scales all coefficients independently, unit normalization adjusts the dynamic range of each vector *relative* to the other dimensions, and can be viewed as a form of adaptive gain control. This contrast adjustment turns out to be quite necessary, as certain coefficients, e.g., the DC component, are prone to dominating the overall representation and unfavorably biasing distance calculations downstream.

Finally, PCA is applied for two reasons. First, as we will see, it is important to center the representation such that each coefficient has zero mean. Additionally, we discard the redundant components of the Fourier transform, reducing the dimensionality from 900 to 450 coefficients.

[5] The choice of similarity or distance is only a matter of preference.

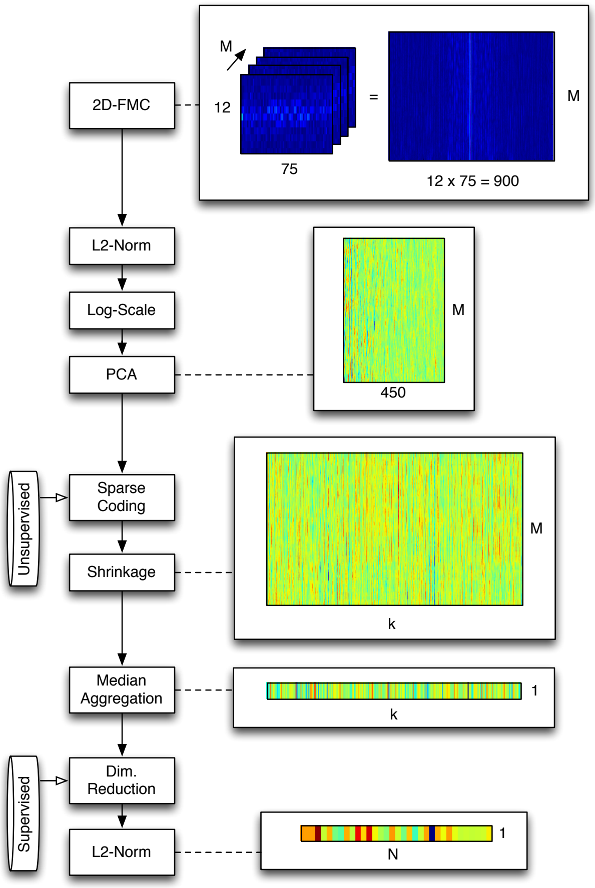

Figure 1. Diagram of the proposed method.

3.2 Sparse Component Estimation

Previous work used 2D-FMCs as a clever way to represent meaningful, rotation-invariant beat-chroma patterns. While this is, strictly speaking, an accurate insight, the Fourier bases themselves do not necessarily make for good feature extraction. Generally speaking, when the bases of a projection are unlike the data to which it is applied, it is unable to compactly represent this information. As a result, the noise floor of the resulting representation is higher and most coefficients tend to be active; furthermore, this behavior becomes especially problematic when pooling features. This observation is quite relevant to this particular instance, as the data being projected —non-negative chroma— is not sinusoidal.

Alternatively, data driven transformations learn a set of bases [6] , or a *dictionary*, from a sampling of data, and are often able to encode meaningful behavior with a small number of active components. This is typically realized by the dot-product of an input X with a dictionary W, followed by an activation function $h(\cdot)$, expressed as follows:

$$Z = h(W \cdot X) \tag{4}$$

Interestingly, this formulation draws strong parallels to previous work both neural networks and sparse coding. Recent research in these areas has emphasized the importance of $h(\cdot)$ being defined as the shrinkage operator, $h_\theta(x) = sgn(x) * max(\|x\| - \theta, 0)$, where θ controls the knee or threshold of the function [5]. This non-linearity exhibits the desirable behavior of suppressing low-level activations while passing sufficiently large ones. Such a process inherently leads to sparser outputs, with the rationale being that only the most representative attributes are encoded.

Additionally, convolutional variants [7] apply this dictionary, also referred to as *kernels*, at all translations over an input to achieve shift-invariant feature extraction, given by the following:

$$Z = h(W \circledast X) \tag{5}$$

Reusing these kernels at all positions, known as *weight sharing*, results in fewer parameters to learn, reduced overfitting, and thus better generalization. Despite these advantages, convolution is a computationally expensive operation. Therefore, to reap the benefits of data driven transformations while still learning shift-invariant features, we leverage the convolution-multiplication duality of the DFT and operate on the Fourier magnitude representation:

$$Z = h(W \circledast X) = h(W \cdot \|\mathcal{F}(X)\|) \tag{6}$$

Note that it is unnecessary to also take the Fourier transform of W, as Eq (6) is now equivalent to Eq (4) and the dictionary can be learned directly on the pre-processed 2D-FMC representation. Additionally, by first centering the data, a bias term is unnecessary and both W and Z will also be approximately zero-mean.

3.3 Semantically Organizing the Space

Having designed a transform to project 2D-FMCs into a sparse representation, we subsequently pool features over a track by taking the median over each coefficient. However, while discriminative power can be achieved by projecting into higher dimensional spaces, it is often necessary to recover a lower dimensional embedding where distance encodes the desired semantic relationship between vectors, i.e. covers are near-neighbors. There are at least two conceptual justifications motivating an embedding transform. First, high dimensional representations are known to suffer from the curse of dimensionality, i.e. distance is not well behaved. Second, and more specific to this approach, the dictionary used in the previous stage is learned as an unsupervised process. As a result, there are no guarantees that the representation it produces provides the latent organization necessary for this task.

Therefore, using known relationships between songs in a training set, we can treat covers as distinct classes in a large, multi-class problem, and apply supervised learning to recover an embedding that tries to preserve these relationships. The resulting projection can then be used to transform unseen data into a cover-similarity space for computing distances between tracks.

[6] These are not strictly bases in the orthogonal, linear algebra sense, but it is a commonly used term in the literature.

4. EXPERIMENTAL DESIGN

4.1 Methodology

Having introduced our main contributions, we now turn our attention to a discussion of implementation details and explore various hyperparameters. To quantitatively navigate this space, we use the training split of the Second Hand Song (SHS) dataset[7] for development and save the test split for our final evaluation. The SHS is a collection of 18,196 tracks from 5,854 "cliques", or distinct classes, with 12,960 from 4,128, respectively, set aside for training; the remainder constitutes the test set. Importantly, the SHS is also a subset of the MSD, which allows for large scale evaluation by using the entire MSD as background noise in a cover song retrieval task.

In line with previous work, the primary metrics of interest here are mean average precision (MAP) and average rank (AR). MAP is computed as the mean of the average precision over a set of queries, and reflects not only accuracy but also the order of correct documents in a ranked list. As an additional statistic, AR is computed as the average position of relevant documents, and measures where relevant documents fall in a ranked list. For evaluating performance in the training condition, each track in the training set is treated as a query and ranked relative to the remaining items in the training set, i.e. 1-vs-12,959; alternatively, in the test condition, each track in the test set is treated as a query and ranked relative to all other tracks in the MSD, i.e. 1-vs-999,999.

4.2 Impact of Sparse Projections

Here, we propose using the k-means algorithm to learn various dictionaries, inspired by recent work in [3]. While we acknowledge that there are alternative methods that could be applied to learn the bases of this transform, k-means is particularly attractive being unsupervised and relatively simple, having a single hyperparameter k. Noting that k-means is a batch, as opposed to on-line, learning algorithm, we first draw 50,000 2D-FMC vectors randomly from the SHS training set. This subset is used for both fitting PCA in the pre-processing set as well as learning dictionaries for various values of k; at this stage, we consider $k \in [128, 512, 1024, 2048]$. It is worth mentioning that due to the nuances of the algorithm —we use the Scipy implementation[8] — only 2045 elements were returned for $k = 2048$, as three of the centroids did not change. Additionally, after inspecting the data to determine a reasonable knee for the shrinkage function, we set $\theta = 0.2$ for our experiments.

Shown in Table 1, we find that applying learned k-means dictionaries as sparse projections, followed by median pooling and L_2-normalization, leads to slightly worse performance than the baseline system. This negative result illustrates that a sparse, higher dimensional feature space does not necessarily exhibit the organization necessary for distance to be meaningful. However, the goal of a sparse projection is only to make the information more separable, and this behavior must be explored further to determine its true impact on system performance.

k	128	512	1024	2045	Baseline
MAP	3.44%	4.54%	4.92%	5.51%	**8.91%**
AR	3,248	3,154	3,112	**3,026**	3,097

Table 1. Exploring values of k on the Training set.

4.3 Semantically Organizing the Space

In light of this, we now seek to better encode semantic relationships with distance measures. Linear Discriminant Analysis (LDA) is a natural choice for learning a supervised embedding that jointly minimizes intra-class variance and inter-class discrimination. This approach also has a single hyperparameter N, the dimensionality of the projection, and we explore $N \in [50, 100, 200]$.

As shown in Table 2, the combination of sparse projections *and* supervised dimensionality reduction leads to considerably better performance on the training set. While this result says nothing about generalization, it more than demonstrates that the representation produced by projecting onto a learned dictionary is indeed significantly more separable. It is interesting to note how performance degrades sharply as a function of decreasing k, and less so with decreasing N. The interpretation of this is two-fold: one, because the dictionary learning is unsupervised, it requires an over-complete set of bases to adequately capture the "right" information for LDA to recover; and two, model complexity can be constrained by limiting N, and therefore serve as a type of regularization.

Before proceeding, it is necessary to ensure that this increase in performance is in fact due to the sparse projection and not just the supervised embedding. To test this hypothesis, we apply LDA to the baseline system, with the 2D-FMC pre-processing pipeline discussed in Section 3.1. Table 3 clearly shows that, though there is some improvement to be had via LDA alone, projecting into a higher dimensional space first is indeed significant, almost doubling MAP as a linear function of N.

Mean Average Precision			
$k \setminus N$	200	100	50
128	5.34%	4.82%	4.19%
512	9.30%	7.38%	4.95%
1024	13.99%	9.63%	5.63%
2045	**28.51%**	17.35%	9.05%
Average Rank			
$k \setminus N$	200	100	50
128	2,915	3,116	3,345
512	2,719	3,153	3,688
1024	2,420	2,980	3,665
2045	**1,844**	2,539	3,249

Table 2. Exploring impact of both k-means *and* LDA on the Training set.

[7] http://labrosa.ee.columbia.edu/millionsong/secondhand

[8] http://docs.scipy.org/doc/scipy/reference/cluster.vq.html

Method	MAP	AR
Baseline + LDA(50)	5.35%	3,666
Baseline + LDA(100)	9.85%	3,034
Baseline + LDA(200)	14.31%	2,434
k-means(2045) + LDA(50)	9.05%	3,249
k-means(2045) + LDA(100)	17.35%	2,539
k-means(2045) + LDA(200)	**28.51%**	**1,844**

Table 3. Results for the SHS Training set applying LDA to the baseline, versus the best performing sparse projection.

Method	MAP	AR
Random	$\sim$ 0.001%	500,000
2DFTM + PCA(50) [1]	1.99%	**173,117**
2DFTM + PCA(200) [1]	2.95%	180,304
k-LDA(50)	**13.41%**	343,522
k-LDA(200)	0.83%	398,005
k-PCA(200) + LDA(200)	12.76%	338,882

Table 4. Results for the SHS Test set over the full MSD. Note that we contract k-means(2045) here simply as "k-".

5. LARGE-SCALE EVALUATION

So far, we have focused exclusively on the SHS training set, both as a computational simplification and an approach to system development. We now turn our evaluation to the test split of the SHS dataset to investigate how our approach generalizes to unseen data. Based on the results of the previous section, we reduce the parameter space by fixing $k = 2045$ but continue to observe performance as a function of N.

First, evidenced by the results given in Table 4, the combined k-means and LDA projections —which we contract here on as k-LDA for brevity— observe radically different behavior based on the dimensionality of the embedding. In fact, k-LDA(200), the best performing system on the training set, seemingly fails to generalize at all; MAP and AR are over two-times worse than the baseline system, and these results clearly indicate extreme over-fitting. Setting this observation aside for a moment though, something even more curious occurs with k-LDA(50). While the AR is also much worse than baseline, the MAP improves by a factor of 6. This behavior begs an obvious question: what is occurring under the surface such that these metrics move in drastically different directions?

On closer inspection, a rather surprising observation precipitates: despite a significantly worse AR, the k-LDA(50) projection actually produces a remarkable number of correct *nearest* neighbors, i.e. the top-ranked item in the list is an accurate match. This intuitively explains the discrepancy between these metrics, as MAP weights precision as a function of rank position, e.g. being correct at the top matters more than being correct lower in the list. Furthermore, despite pulling relevant tracks to the top of the list, the k-LDA(50) system also pushes some to the very bottom. As a result, the distribution of relevant items in the ranked list is bimodal, and AR is at a loss to characterize this behavior.

To get a better sense of this behavior, we investigate precision-@-k, defined simply as the precision over the top-k items in a ranked list. Figure 2 clearly illustrates how our proposed method not only yields better performance overall, but offers improved usability as well. For this particular test set, the system gets the top result correct nearly 25% of the time, out of a space of one million possible items. Considering the top 10 results, or approximately the first page of a web search, about 5% of the documents are correct; in other words, there is a 50% chance that a true cover will appear on the first page of a search.

Turning back to the k-LDA(200) projection, the question now becomes how to reduce such substantial over-fitting. Fortunately, projecting into a PCA subspace before fitting LDA has been shown to reduce over-fitting in the image processing and pattern recognition communities, notably for face recognition [6]. This is because PCA dimensionality reduction avoids singularities or near singularities in any of the scatter matrices used in LDA; this problem is exacerbated for small datasets or high dimensional feature spaces, of which this application is both. Furthermore, the cascade of PCA and LDA has been shown to be a general case of other LDA variations like uncorrelated LDA (ULDA), which are also used to avoid the singularity issue. Most importantly, how much PCA alleviates LDA over-fitting depends on the dimensionality of the intermediate PCA subspace. Therefore, selecting the right number of principal components is both crucial for good results, and non-trivial.

In lieu of a more extensive exploration, we perform an initial inquiry into the potential of PCA to address this particular problem. Here, we fit a 200 dimensional PCA subspace by transforming the SHS training set into its mid-level, 2045-dimensional representation, just before the application of LDA. In an effort to help minimize potential singularities and other such problems, we take two additional steps when fitting LDA to encourage better generalization; however, the true impact of such decisions are admittedly uncertain. First, we subsample the training set, only using cliques with 9 or more tracks each. Then, we include an arbitrarily large number of tracks that do not belong to any clique. This resulting embedding, dubbed k-PCA(200)+LDA(200), is then evaluated on the MSD, and we again recover performance roughly on par with k-LDA(50), e.g. relatively low AR, but a significantly higher MAP than baseline.

Finally, in terms of computation time, our method takes three more times to compute than baseline. In a machine with plenty of RAM, 16 cores, and splitting the process into 10 different threads, the baseline takes 8.7 hours to compute the features of 50, 100 and 200 PCA components. However, as our method produces features with the same output dimensionality as the baseline, our distance calculations —the prohibitive computation— requires the same amount of time. More specifically, it takes 0.4, 0.9, and 1.5 hours using 50, 100, and 200 components respectively.

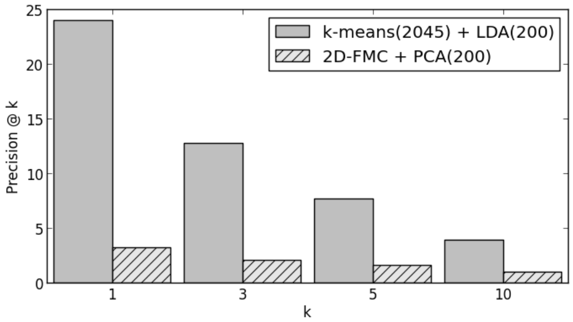

Figure 2. Comparison of Precision@k on the Test set for k-LDA(50), versus the best baseline result.

6. CONCLUSIONS

In this work we have presented an improved system for large-scale cover song retrieval, demonstrating how sparse, high-dimensional projections can be combined with low-dimensional embeddings to achieve greater performance than either piece alone. This semantically organized space is recovered by efficiently capturing shift-invariant features by effectively performing convolutional sparse coding in the Fourier magnitude domain, and learning a supervised cover-similarity space where distance is meaningful. Our system not only achieves state-of-the-art performance with respect to previously used evaluation metrics (MAP), but greatly improves precision-at-k for k less than 10, indicative of a more useful system. This encourages the additional observation that top-k, as opposed to full-list, metrics may be more informative for characterizing the usability of large scale information retrieval systems.

Looking toward future work, we identify several areas with the potential for improvement. As mentioned, there are a variety of ways the sparse dictionary could be learned; and, depending on the temporal pooling strategy defined, it would be possible to fine-tune the overall architecture like a deep network via backpropagation. Additionally, there are other pooling strategies that could be employed, leveraging structural knowledge to summarize the information over a full track in more musically meaningful ways. Lastly, the challenge of realizing a semantically organized space for computing distances between tracks is hardly a solved problem. Over-fitting seems to be a problem in higher dimensions, but the PCA-subspace trick discussed offers encouraging results, complementing those obtained directly from low-dimensional LDA.

Finally, to facilitate reproduction of results and encourage future work, we provide an open source implementation of our method in a public repository [9] .

7. REFERENCES

[1] Thierry Bertin-Mahieux and Daniel P. W. Ellis. Large-Scale Cover Song Recognition Using The 2D Fourier Transform Magnitude. In *Proc. of the 13th International Society for Music Information Retrieval Conference*, pages 241–246, 2012.

[2] Thierry Bertin-Mahieux, Daniel P. W. Ellis, Brian Whitman, and Paul Lamere. The Million Song Dataset. In *Proc of the 12th International Society of Music Information Retrieval*, Miami, FL, USA, 2011.

[3] Adam Coates and Andrew Y Ng. Learning feature representations with k-means. In *Neural Networks: Tricks of the Trade*, pages 561–580. Springer, 2012.

[4] Daniel PW Ellis and Graham E Poliner. Identifying cover songs' with chroma features and dynamic programming beat tracking. In *Acoustics, Speech and Signal Processing, 2007. ICASSP 2007. IEEE International Conference on*, volume 4, pages IV–1429. IEEE, 2007.

[5] M. Henaff, K. Jarrett, K. Kavukcuoglu, and Y. LeCun. Unsupervised learning of sparse features for scalable audio classification. In *Proc. 12th Int. Conf. on Music Information Retrieval (ISMIR)*, 2011.

[6] Shuiwang Ji and Jieping Ye. Generalized linear discriminant analysis: a unified framework and efficient model selection. *Neural Networks, IEEE Transactions on*, 19(10):1768–1782, 2008.

[7] Koray Kavukcuoglu, Pierre Sermanet, Y-Lan Boureau, Karol Gregor, Michaël Mathieu, and Yann LeCun. Learning convolutional feature hierachies for visual recognition. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[8] Geoffroy Peeters. Spectral and temporal periodicity representations of rhythm for the automatic classification of music audio signal. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(5):1242–1252, 2011.

[9] Joan Serrà, Emilia Gómez, and Perfecto Herrera. Audio cover song identification and similarity: background, approaches, evaluation, and beyond. In *Advances in Music Information Retrieval*, pages 307–332. Springer, 2010.

[10] Joan Serrà, Emilia Gómez, Perfecto Herrera, and Xavier Serra. Chroma binary similarity and local alignment applied to cover song identification. *Audio, Speech, and Language Processing, IEEE Transactions on*, 16(6):1138–1151, 2008.

[11] Joan Serrà, Xavier Serra, and Ralph G Andrzejak. Cross recurrence quantification for cover song identification. *New Journal of Physics*, 11(9):093017, 2009.

[9] https://github.com/urinieto/LargeScaleCoverSongId