Exploration of Cognition Impact: An Experiment with Cover Song Retrieval Through Indexing

D. Khasim Vali[1](✉) and Nagappa U. Bhajantri[2]

[1] Department of Computer Science and Engineering, Vidyavardhaka College of Engineering,
Mysuru 570017, India
dkv@vvce.ac.in

[2] Department of Computer Science and Engineering, Government College of Engineering,
Chamarajanagara 571313, India
bhajan3nu@gmail.com

Abstract. Large-scale cover song retrieval systems frameworks can figure tune-to-tune likeness and suit contrasts in timing, key, and beat. Straightforward vector separation measure is not enough incredible to perform spread melody acknowledgment and high arrangements, for example, dynamic time-traveling does not scale to a large number of cases, making spread tune recovery erroneous for business-scale applications. In this work, the substance-based music highlights of tunes are utilized as information and changed them into vectors by utilizing the 2D Fourier change approach. By anticipating the tunes into a combination vector of PCA and LDA, the effective KD-tree and R-tree indexing calculation is utilized to look at the comparability of melodies and recover the most comparable tunes from the enormous scale database. The proposed framework is not just effective enough to perform adaptable substance-based music recovery, yet can likewise build up the capability of making comparative music acknowledgment applications quicker and increasingly exact.

Keywords: CNN · MFCC · Deep learning

1 Introduction

Cover song retrieval (CSR) by and large alludes to the issue of recognizing various translations of similar melodic work. Since 2006, this test has been enrolled in the brought together yearly Music Information Retrieval (MIR) assessment sessions known as the MIR EXchange (MIREX). From that point onward, a few frameworks for this nature of errand have been submitted and assessed on a fixed, however undisclosed dataset. As the outcomes obtained by these frameworks are communicated as general execution numbers, no data is given that could uncover the impact of explicit CSR framework part plan decisions and the organization of the assessment dataset on the obtained recovery results.

C. R. Panigrahi et al. (eds.), *Progress in Advanced Computing and Intelligent Engineering*, Advances in Intelligent Systems and Computing 1198, https://doi.org/10.1007/978-981-15-6584-7_33

CSR has all the earmarks of being more explicit than exemplary music classification recovery, spread tunes still length a wide scope of sorts, each with their variations and invariants, presenting explicit difficulties on the structure of the CSR framework. To approve plan inspirations and distinguish which framework perspectives are most basic for execution results, it is important to consider CSR frameworks as mixes of general framework parts and audit execution as for these segments. Also, the plan of the assessment dataset is basic for getting genuine knowledge into the presentation of CSR frameworks.

2 Related Works

Interestingly and recently, in the area of cover song identification, there has been a considerable amount of new approaches [1, 2] that tries to handle different issues. The typical goal is trying to effect new algorithms or combinations of them to improve the results in comparison, but the recent focus by most researchers has been diverted toward scalable strategies [2, 3]. The most common way to calculate the similarity between two different songs is with alignment-based methods, and they have shown to be able to produce good results 1 (75% MAP in MIREX'2009). However, these methods are computationally expensive and, when applied to large databases, they can become impractical: The best performing algorithm [4] in MIREX'2008 implemented a modified version of the Smith–Waterman algorithm and taken approximately 104 h to compute the results for 1000 songs. In other words, the algorithm exercised to the Million Song Dataset (MSD), and the estimated time would be of 6 years [5]. Martin et al. [2] suggest the use of Basic Local Alignment Search Tool (BLAST), a bioinformatics sequence searching algorithm, as an alternative to dynamic programming solutions. The data is indexed based on the similarity between songs, and to compute the similarity value, the best subsequences are chosen and then compared. Khadkevich et al. [3] extract information about chords and store them using locality-sensitive hashing (LSH). Bertin-Mahieux et al. [2] adopt the 2D Fourier transform magnitude for large-scale cover detection, further improved by Humphrey et al. [6], who modified the original work to use a sparse, high-dimensional data-driven component, and a supervised reduction of dimensions. Balen et al. [1] extract high-level musical features that describe the harmony, melody, and rhythm of a musical piece. In other words, those descriptors are stored with LSH, which allows retrieving the most similar songs. Lu and Cabrera [5] employed hierarchical K-means clustering on chroma features to find audio words or centroids. A song will then be represented by its audio words. Moreover, similarity with other songs will be determined by audio words share with the same location.

Outside the field of large-scale cover identification, several solutions regarding distance fusion [3] have been suggested. Salamon et al. [7] extract the melodic line, the bass-line, and HPCP 12-bins for each song. They explore the fusion of those features to discover which results in the best performance. Distance fusion is also the main focus in the work of Degani et al. [8], where they propose a heuristic for distance fusion. The work consists of normalizing all values to [0, 1], computing a refined distance value, and produce a single matrix of results.

The regular methodologies portrayed above figure the separation between an inquiry and the tunes to be analyzed and verify that the melody with the closest separation is almost certain to be a spread. Since the procedure is discrete from each inquiry, the outcome from “another variant of a similar spread” cannot be considered. On the off chance that it is potential, tunes with various lengths can be spoken to in a similar space. Besides, if comparable/different melody sets are known, the measurement to quantify the tune separation can be streamlined, instead of utilizing the Euclidean separation. Nonetheless, rather than taking the separation network legitimately to rank the comparability, here first play out a change utilizing PCA and LDA to modify every melody in the high-dimensional space. In this manner, the separation metric found out from the melody combines in the new portrayal and their names. Likewise, select center melodies with assorted melodic properties and apply them for both installing and preparing. In outline, the methodology accepts that the separation between the center sets, and every tune can be a segregating highlight to effortlessly gather similar spreads. The calculated delineation of this new portrayal has appeared in Fig. 1. Here, exertion is to look at whether the K-D-tree and R-tree can be powerful to recover the likeness between tunes. Moreover, the best execution is uncovered overspread tune recovery by applying the PCA and LDA to the grid produced.

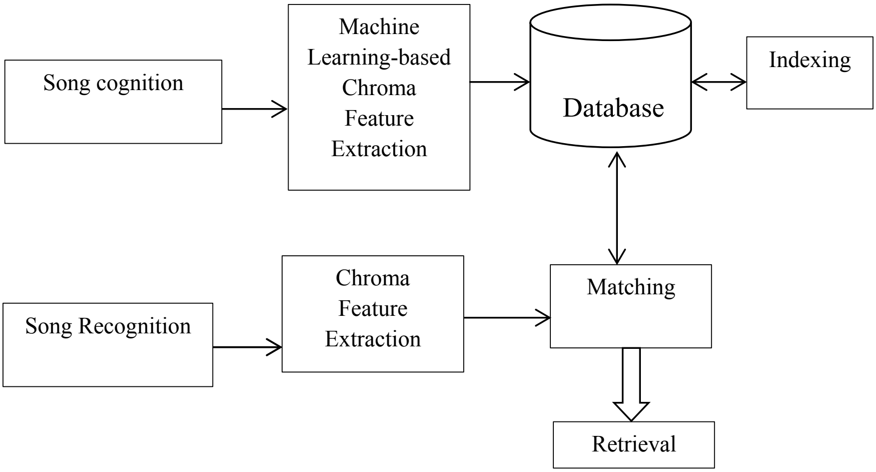
Song cognition
Machine Learning-based Chroma Feature Extraction
Database
Indexing
Song Recognition
Chroma Feature Extraction
Matching
Retrieval

Fig. 1. Proposed work

3 Proposed Work

Here, an effort to develop a method to build an efficient song retrieval strategy can retrieve a similar pattern for a given query. Firstly, the preprocessing pillar attempts to convert audio signals into chroma features for song and reduce those through PCA and LDA. Further, reduced features are fusion to a vector. Eventually, tree indexing structures such as KD-tree and R-tree are exercised to effectively retrieve the more relevant songs.

4 Chroma Features

If the beat following can recognize a similar primary heartbeat in various versions of a similar piece, at that point speaking to the sound against a period base characterized by the distinguished thumps standardizes away varieties in rhythm. Further, record a solitary component vector for every beat, and utilize twelve component chroma highlights to catch both the overwhelming note commonly song just as the wide symphonious backup [8]. Thus, the chroma highlights record the force related to every one of the 12 semitones (e.g., piano keys) inside one octave; however, all octaves are collapsed together. Calculating symphonious highlights over beat-length fragments shows up in [9]. Rather than utilizing a coarse mapping of FFT receptacles to the chroma classes they cover, apply the stage subordinate inside each FFT container both to distinguish solid tonal segments in the range and to get a higher-goals gauge of the hidden recurrence [10]. The methods proposed in [11] help to evacuate non-tonal parts and improve recurrence goals past FFT receptacle level has comparable inspiration and effect to the sinusoid-demonstrating.

5 Dimensionality Reductions

Dimensionality decrease means to interpret high measurement to a low measurement portrayal to such an extent that comparative info items are mapped to every single close-by point. Be that as it may, a large portion of the current dimensionality decrease methods has a couple of impediments. Initially, they do not deliver a capacity from the contribution to complex that can be connected to new focuses, whose relationship to preparing focuses are unsure.

5.1 Principal Component Analysis (PCA)

For the most part, neighborhood highlights can contribute and express to a certain degree conceivable. At the end of the day, the object includes through more extensive highlights, for example, worldwide highlights are fundamental. In this unique circumstance, to raise those highlights through PCA [12] is the appropriate way. Then again, it is an amazing strategy for removing the worldwide structure from the high-dimensional dataset to diminish the dimensionality and to separate the dynamic highlights of the animation pictures. In this manner, to recognize the examples and features the similarities. PCA is a powerful exertion for the dimensionality decrease and points in lessening the elements of an $\times p$ framework X. It alludes to the vital segments and the consequent utilization of these segments in understanding the information. Then again, PCA experiences various weaknesses [12], for example, its understood supposition of Gaussian appropriations and its limitation to direct mixes.

5.2 Linear Discriminant Analysis (LDA)

Linear discriminant analysis (LDA) is a traditional procedure in example acknowledgment [12]; it is utilized to locate a straight blend of highlights, which recognizes or isolates at least two classes of the articles, and the subsequent mix can be utilized as a direct classifier. LDA is a directed subspace learning strategy dependent on the Fisher criterion [13]. Fisher paradigm assumes a vitalizing job in dimensionality decrease, which points in finding a component portrayal by which the inside class separation is made least and the between-class separation is greatest.

Further, it aims to find a linear transformation $W \in R^{d \times m}$ that maps X_i in the d-dimensional space to m-dimensional space, in which the between-class scatter is maximized while the within-class scatter is minimized, i.e.,

$$\arg\max_{w} tr\left(\left(W^T S_w W\right)^{-1}\left(W^T S_b W\right)\right) \tag{1}$$

where S_b and S_w are the between-class scatter matrix and within-class scatter matrix, respectively, which are defined as:

$$S_b = \sum_{k=1}^{c} n_k(\mu_k - \mu)(\mu_k - \mu)^T \tag{2}$$

$$S_w = \sum_{k=1}^{c} \sum_{i \in c_k}^{1} (x_i - \mu_k)(x_i - \mu_k)^T \tag{3}$$

where C_k is the index set of the k-th class, μ_k and n_k are mean vector and size of k-th class, respectively, in the input data space. It is easy to show that Eq. (1) is equivalent to:

$$\arg\max_{w} tr\left(\left(W^T S_t W\right)^{-1}\left(W^T S_b W\right)\right) \tag{4}$$

where S_t is the total scatter matrix, defined as follows,

$$S_t = \sum_{i=1}^{n} (x_i - \mu)(x_i - \mu)^T \tag{5}$$

when the total scatter matrix S_t is non-singular, then the solution of Eq. (4) consists of the top eigenvectors of the matrix $S_t^{-1} S_b$ corresponding to nonzero eigenvalues.

6 KD-Tree

There is a dearth of general approach to effectively retrieve the songs from the large databases. Most of the current approaches are designed to handle a large dataset. Here the size of the database is taken into consideration. The KD-tree (K-dimensional) indexing schema supports the range search with good pruning and lays its efficiency from the point of search time.

6.1 Construction

An extraordinary instance of the parallel space dividing trees is KD-tree, which is built in a recursive style. At the root, every one of the information focuses is part of equivalent parts by a parcel hyperplane. At that point, every half is doled out to one of the kid hubs and is recursively split to make a decent doubletree. The leaf hub may contain a solitary point or more than one point in various executions. Every hub in the developed KD-tree compares to a cell in Rk, limited by a lot of parcel hyperplanes. A parcel hyperplane is opposite to a segment hub, and it is chosen by a segment esteem. The segment hub in ordinary KD-tree is the organized hub which has the best fluctuation, and segment worth is the middle of the projections of the considerable number of information focuses through the partition.

6.2 Search

A top-down looking is done from the root to leaf hubs to discover the closest neighbor of a question point. At each interior hub, it is required to check which side of the segment hyperplane the question point falsehoods, and after that, the related youngster hub is gotten to likewise. The plummet down procedure requires the examinations of the log2*n* stature of the KD-tree to arrive at a leaf hub. The information focuses related to the principal leaf hub turns into the main component for the closest neighbor, which may not be the genuine closest neighbor. It must be pursuing a procedure of backtracking technique or iterative quest to other leaf hubs for looking through the better components. The generally utilized plan with high opportunity to locate the genuine closest neighbor is a need search dependent on need line, in which every one of the cells is looked arranged by good ways from the question point. The hunt procedure will end when there are no more cells inside the separation which is characterized as the best point.

7 R-Tree

Given past research, there are various spatial information file techniques proposed. One such case of these strategies is the R-tree; it is utilized in a spatial database as a spatial access technique. The point of spatial access strategy is improving question execution by joining an added substance information structure since the high inquiry execution is one of the key highlights of fruitful recovery frameworks [13]. R-tree is made out of the root, moderate hub, and leaf hub. Each leaf hub is made as a minimum bounding rectangle (MBR) [14]. Leaf hub does not store the genuine spatial articles, yet it stores the base bouncing square shape of the real spatial items.

R-Tree Steps: Query calculation of R-tree is like B-tree. It uses top-down inquiry from the root hub, yet MBR of R-tree will cover to result in the non-uniqueness of question way. Since the B-tree utilizes just the location, the B-Tree results will rely upon the presence of the required question. R-tree looks through the database utilizing the miles space from the required question. The R-tree has been utilized in the hunt procedure in this way when the client scans for a particular melody, the framework utilizes R-tree spatial record which searches the database and returns the outcomes to the client on the interface. The client scans for a particular melody in the framework database. On the off chance that the framework discovers this information, it legitimately contrasts it and the present database, at that point if this information is found; at last it will recover the information that is in the database on the guide. Be that as it may, if the hunt information was not found in the present database or web database, there will not be any information to be given to the client.

Utilization of R-tree: The R-tree calculation is one of the ordering calculations. It is adaptable since it is composed of the information list structure [14]. The record can be built up without the forecast of the whole spatial degree since it is a characteristic expansion of B-tree, maybe has comparable structure and attributes. One of the primary utilization of R-tree is to incorporate with conventional social database. Thus, numerous R-tree spatial databases are picked for the spatial list. Be that as it may, when the R-tree hub passages surpass M, the hub must be part.

8 Matching

The melody has some arrangement of highlights when the list of capabilities is acquired, and it is put away in the database. Notwithstanding, putting away in the database in a proficient way is required with the end goal that the conceivable up-and-comer rundown is chosen the coordinating procedure ought to be compelling. Consequently, a backend instrument of ordering system is utilized, which stores the information in some pre-characterized way so that during the coordinating stage just a couple of likely applicants are chosen. Consequently, the exertion implements KD-tree- and R-tree-based methodology for ordering the got highlights. The accompanying subsection gives the outline of the KD-tree ordering guideline.

9 Datasets

The created dataset covers 30 contains 30 sets of original and cover songs—spanning across genres, styles, live and recorded music, and the dataset is biased toward regional languages. Most songs contain a cover version; however, some songs contain up to three. Similarly, the extended covers 80 [15] proposed at MIREX 2007 to benchmark cover song recognition systems. The dataset contains 80 sets of original and cover songs, 166 in total—which comprises genres, styles, live and recorded music, in other words, covers 80 predominantly, oriented toward Western music.

10 Experimentation

Here, we extended the experiment over created and real datasets to reveal the capability of the proposed criteria. The work has been implemented in a MATLAB R2013a using an Intel Pentium 4 processor, 2.99 GHz Windows PC with 2 GB of RAM.

Subsequently, the reduced feature vectors through PCA and LDA are combined into a single vector. However, the fusion vector is made to compute the combination of PCA and LDA as a dominant feature via dataset of cover 80 and cover 30. Besides, to reveal performances of the KD-tree and R-tree indexing by varying retrieval ranking from 10 to 50. Precisely, pick songs randomly from the databases, and experimentation is conducted more than five iterations. The effort outcomes are emphasizing to witness through maximum accuracy obtained in all cases. Given further appreciation and correlated the supremacy of the indexing method, the attempt has been performed on the database under varying sizes from 30 to 70% of the database in various instances (Fig. 2).

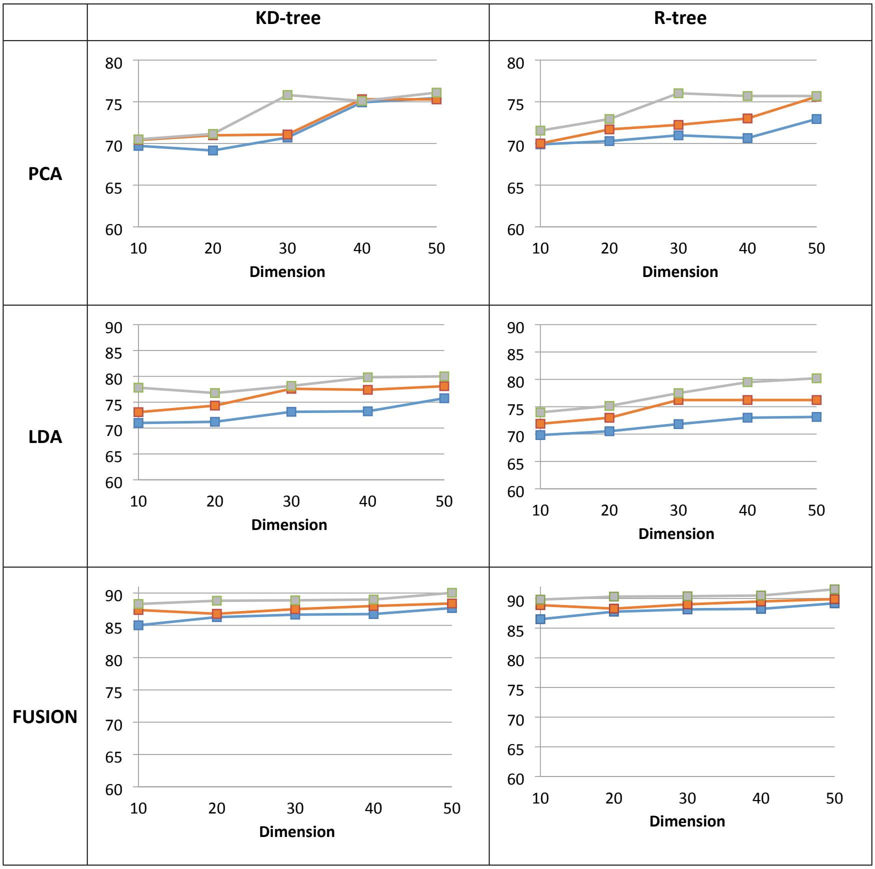

Fig. 2. Accuracy for cover 30 of fusion feature

Moreover, the results obtained for reduced PCA and LDA set 10, 20, 30, 40 and 50 over covers 80 datasets with the appropriate training such as 30, 50 and 70 percentage plots are shown in Figs. 3 and 4. Further, those are tabulated for both ranking and reduction methods with varying training samples. However, analysis of graphical representation can be noticed that the fusion-reduced features with top-ranking retrieval achieve relatively higher accuracy in all cases. Also, the outcomes of the fusion approach have portrayed maximum accuracy when compared to other classifiers due to its computational excellence.

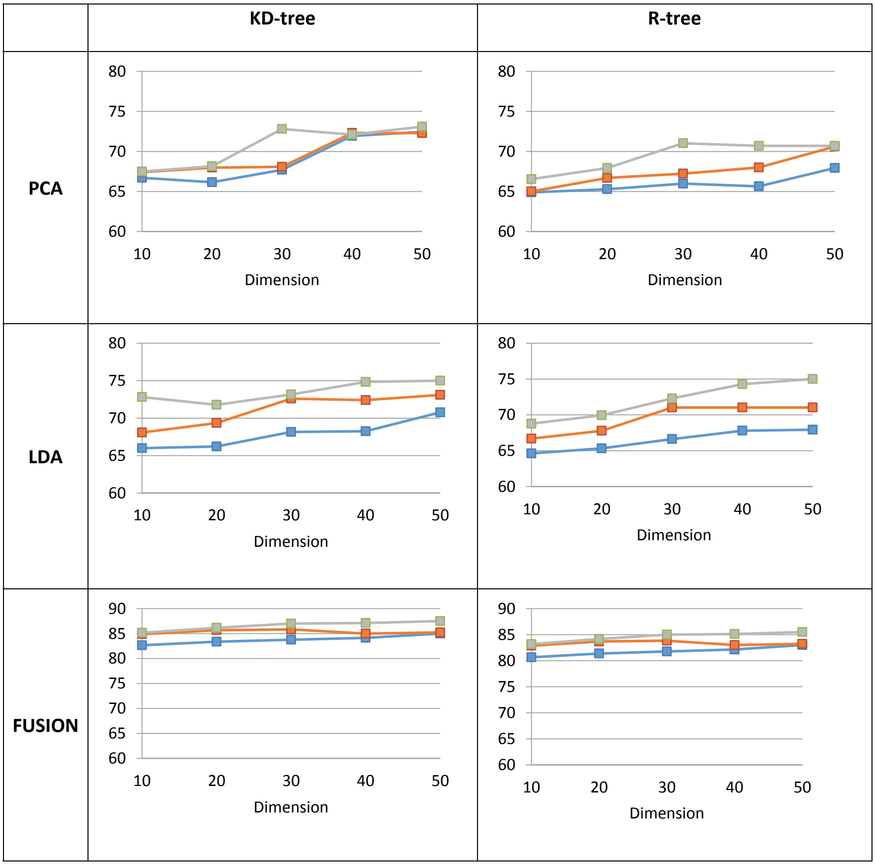

Fig. 3. Accuracy for cover 80 of fusion feature

Further, to enhance the proposed method, the reduced features are fused into one vector using logical OR operation. Thus, a fused vector is called decision-level fusion which contains the actual class from both the vectors. The result of the fusion vectors is shown in Figs. 4 and 5. In addition, to evaluate the proposed method the difference between the maximum and minimum accuracy is calculated and for original vector is shown in Table 1 and for Decision Level Fusion in Table 2.

The outcome of above experiments has to relieve via KD-tree supported for cover 30 and cover 80 are shown convincing performance through empirical display of accuracy. Additionally, the experimentation with R-tree effort perhaps enhances the accuracy when compared to KD-tree. Further the observation of the potentiality of fused the vector of reduced features PCA and LDA are more encourage the accuracy in all the cases of training percentage.

However, accuracy improves as features increases. Also, progress saturates for even after the 50 and 70 dimensions of training. Further, the accuracy spectrum potential difference is 6 in both cases for PCA. In LDA effort, the feature dimension emphasizes

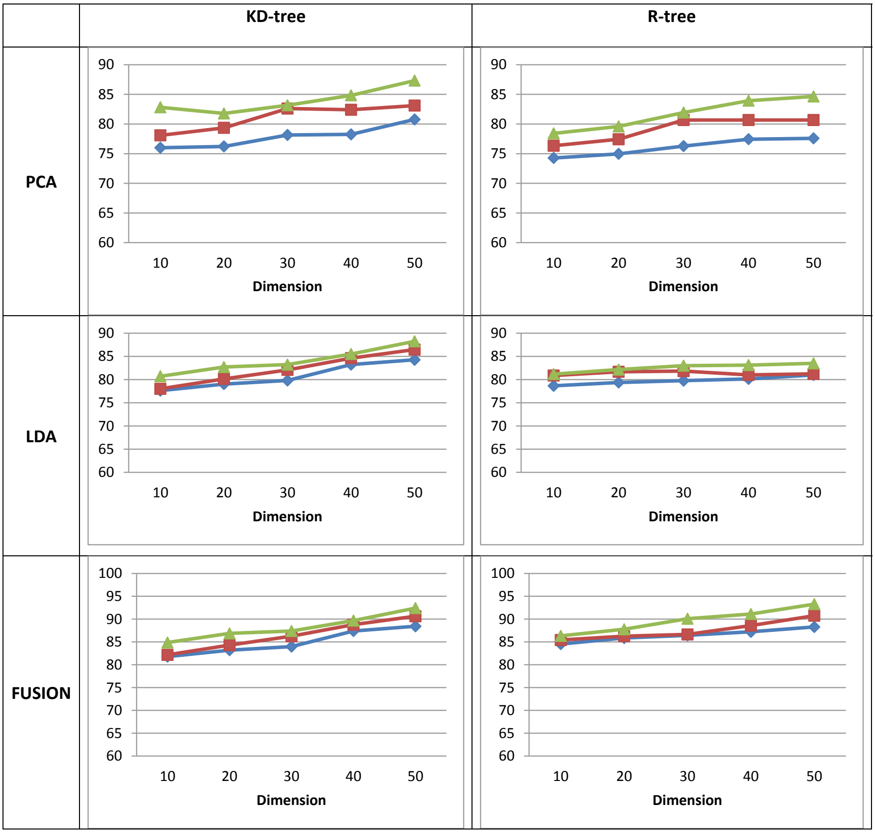

Fig. 4. Accuracy for cover 30 of decision fusion

the snail phase improvement in accuracy for various dimensions and saturated after 40 and hardly a matter of training impact. On the other hand, accuracy patch spread value is approximated empirically 11.32, and the smaller value is 4. However, the smaller value indicates the feature fusion and decision fusion which are predominantly support the improvement of accuracy as shown in Tables 1 and 2.

11 Conclusion

The new framework of cover song retrieval of viewing the scores as features, performing feature normalization and training, has led to a sizable increase in performance compared to existing cover song retrieval through exhibiting the high score. Moreover, feature standardization has made cover song retrieval move from just determining high scores to a general retrieval system while using supervised training offers further increases in performance. Despite these gains, there still exists a space for improvement. In this work, we also compare the reduction strategy with retrieval indexing as KD- and R-trees, respectively. Summarily, here observe that when an individual reduced vector is provided

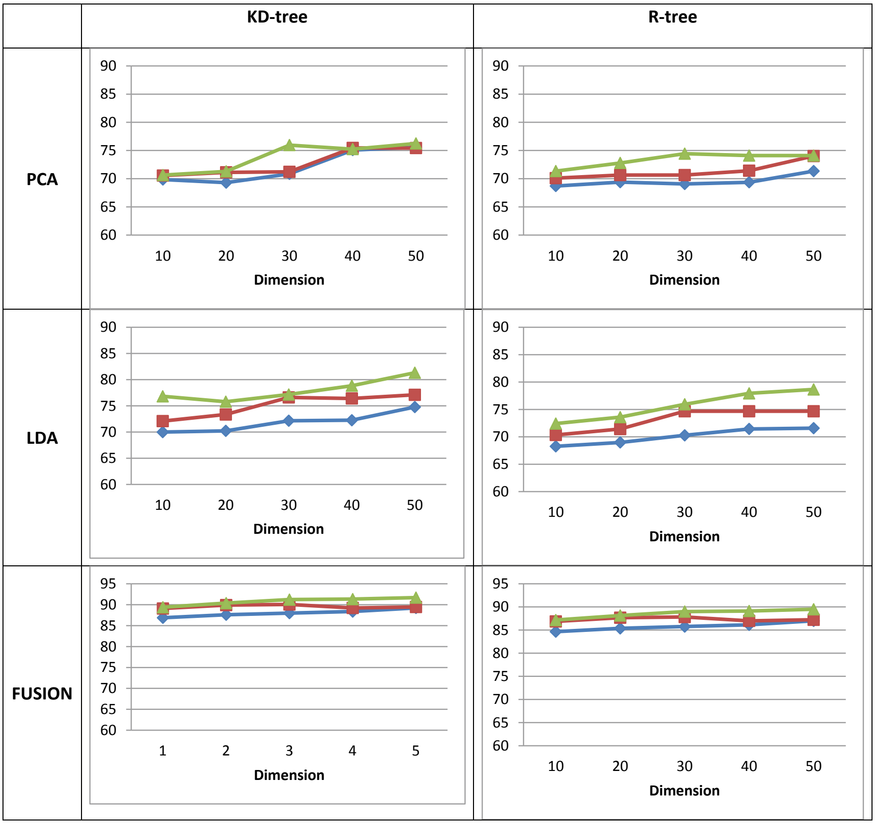

Fig. 5. Accuracy for cover 80 decision fusion

Table 1. Approximated accuracy between the lowest and highest feature fusion

Dataset	Cover 30		Cover 80	
Process/Indexing	KD-tree	R-tree	KD-tree	R-tree
PCA	9.01	10.39	6.94	6.07
LDA	11.32	10.05	9.10	10.39
Fusion	4.86	5.20	4.85	4.85

to indexing is not enough when compared to the fusion of reduced features. Moreover, the emphasizing feature fusion and decision fusion enhances the accuracy empirically for varying the training samples. The fusion vector achieves 95% as maximum accuracy.

Table 2. Approximated accuracy between the lowest and highest decision feature

Dataset	Cover 30		Cover 80	
Process/Indexing	KD-tree	R-tree	KD-tree	R-tree
PCA	5.25	6.01	6.11	4.48
LDA	11.12	7.16	8.45	9.1
Fusion	10.44	8.14	4	4.87

References

1. Balen JV, Bountouridis D, Wiering F, Veltkamp R (2014) Cognition-inspired descriptors for scalable cover song retrieval. In: ISMIR'14, pp 379–384
2. Martin B, Brown DG, Hanna P, Ferraro P (2012) BLAST for audio sequences alignment: a fast scalable cover identification tool. In: ISMIR'12, pp 529–534
3. Khadkevich M, Omologo M (2013) Large-scale cover song identification using chord profiles. In: ISMIR'13, pp 233–238
4. Serrà J, Gómez E, Herrera P, Serra X (2008) Chroma binary similarity and local alignment applied to cover song identification. IEEE Trans Audio Speech Lang Process 16:1138–1151
5. Lu Y, Cabrera JE (2012) Large scale similar song retrieval using beat-aligned chroma patch codebook with location verification. In: SIGMAP'12, pp 208–214
6. Humphrey EJ, Nieto O, Bello JP (2013) Data-driven and discriminative projections for large-scale cover song identification. In: ISMIR'13, pp 149–154
7. Salamon J, Serrà J, Gómez E (2013) Tonal representations for music retrieval: From version identification to query-by-humming. In: Int J Multimedia Inf Retrieval. Special Issue on Hybrid Music Information Retrieval 2: 45–58
8. Degani A, Dalai M, Leonardi R, Migliorati P (2013) A heuristic for distance fusion in cover song identification. In: WIAMIS'13, pp 1–4
9. Bartsch MA, Wakefield GH (2001) To catch a chorus: using chroma-based representations for audio thumbnailing. In: Proceedings of IEEE workshop on applications of signal processing to audio and acoustics, Mohonk, New York
10. Maddage NC, Xu C, Kankanhalli MS, Shao X (2004) Content-based music structure analysis with applications to music semantics understanding. In: Proceedings of ACM multimedia, New York, NY, pp 112–119
11. Abe T, Honda M (2006) Sinusoidal model based on instantaneous frequency attractors. IEEE Trans Audio Speech Lang Proc 14(4): 1292–1300
12. Tsai T, Prätzlich T, Muller M (2016) Known-artist live song id: a hash print approach. In: International society for music information retrieval conference (2016)
13. Aly M, Munich M, Perona P (2011) Indexing in large scale image collections: scaling properties and benchmark. In: WACV
14. Gong J, Ke S (2011) 3D spatial query implementation method based on R-tree. In: Proceedings of remote sensing, environment and transportation engineering (RSETE), pp 2828–2831
15. Ellis DPW, Cotton CV (2007) The 2007 LabRosa cover song detection system. Music Inf Retrieval Eval Exch