

Efficient Music Genre Retrieval Based on Peer Interest Clustering in P2P Networks

Xianfu Meng¹ Zhenyu Guo¹ Yue Gong¹

¹ School of Electronic and Information Engineering, Dalian University of Technology, Dalian 116024, P. R. China

Abstract

Content-based music retrieval is desirable in Peer-to-Peer (P2P) networks, considering its popularity for users and its ability of semantic search, intensive computing cost raises a barrier to efficiency and scalability though. In this paper, we propose an approach of music genre retrieval based on peer interest clustering. Automatic music feature extraction and adaptive shared music file clustering are described. Peers with similar music genre are clustered together, based on which search mechanism and improvement alternatives are deployed. The results of experiments prove the algorithm increases search performance, including precision and recall while reducing network traffic and peer workload.

Keywords: Music genre classification, Peer interest clustering, Content-based music retrieval, Peer-to-Peer

1. Introduction

P2P file sharing systems have a large number of users scattered all around the world, sharing their data with each other. It is a challenging task to search and locate the data of interest in such systems. Most of existing models only offer keyword search [1] functionality. To achieve semantic search, some mechanisms performing query or keyword expansion [2]-[4] have been investigated as an information retrieval (IR) technique for several decades, however, keyword search may be insufficient in many cases such as multimedia file retrieval, more powerful search mechanisms are needed urgently.

A new technique, named Content-Based Retrieval (CBR), engaged in P2P networks, conquering the keyword-based search problem of the incomplete, incorrect and inaccurate description to multimedia resources. A typical CBR strategy works according to the object's semantic meaning, which requires feature extraction from objects, and based on that constructs feature vectors indices in pursuit of better retrieval performance in further search process.

There have been some techniques that provide CBR functionality. pSearch [5] uses some information

retrieval techniques for content-based text search. It represents documents with feature vectors and then uses these vectors as the coordinates to store the documents in a CAN [6] system. In [7], peers are advertised using their compact data summaries for efficient content-based P2P image retrieval. Ivan Lee et al, in [8] present a system, called DISCOVER for content-based image retrieval over distributed Peer-to-Peer network and achieve retrieval precision of 57% on Limewire [9], a Gnutella-like structure. PRISM [10] proposes a scheme based on reference vectors for distributing and querying multi-dimensional data in P2P systems within a few number of participating peers. [11] is a work towards the vision a multimedia indexing and retrieval system, which would be decentralized and deployed on a large scale. Its merits are low computational cost, incrementality and only require a little amount of information to transit between nodes. Yang [12] contributes a content-based music retrieval algorithm that can be decomposed and parallelized efficiently. It deals with acoustic data and uses an intuitive notion of similarity: two pieces are similar if they are fully or partially based on the same score. Wang [13] presents a method of clustering the peers based on content by their set distance to reduce the query time and redundant messages. Ye [14] demonstrates an audio search engine ASEKS based on keyword spotting technology in the P2P network.

Our design philosophy of music genre retrieval based on peer interest clustering under the Peer-to-Peer framework derives from a number of points. First, music retrieval is a more practical require from most P2P users and usually user's interest concentrates on several music genres, i.e., one whose favorite is rock and roll may not be equally interested in those smooth and peaceful music, like classical. Secondly, from the local shared files, user's interest can be much more veraciously indicated. Even in the situation that one's interests are more than one music genre, still it could be treated discriminately in our consideration. Finally, according to the music genre interest, peers in P2P networks can be separated into different interest groups. Note that some peers may be overlapped, given that a single peer may have various

favorites. Therefore it is reasonable and appropriate to represent some peers with the same interest in one certain music genre by the concept of clustering, and meanwhile it ensures the reduction of visited peers by one query request.

In this paper, we present a music genre interest-based search mechanism in the P2P network. Extracting music features in frequency domain which facilitate music genre classification is the first step. There are nine features, which are mean and standard deviation of entropy, mean of energy ratio, mean and standard deviation of lowest frequency location, mean and standard deviation of brightness, mean of bandwidth and so forth. According to the feature vectors composed by extracted features, music genre-based clustering is completed by an adaptive k-means algorithm on each peer. Each cluster is represented by its centroid vector and standard deviation vector, which is considered as more precise descriptor of music file content. Peers are partitioned into different clusters based on music genre interest accurately and automatically. Peers in the same cluster, sharing music files with the same genre, are linked by “Interest Link” (IL). Lookup with music sample reflecting peer’s interest is forwarded in the domain of cluster preferentially for high hit probability and low search cost. “Non-Interest Link (NIL)” is a contact among peers in different clusters, by which a peer fulfills non-interest retrieval, that is, the fast location of other clusters and efficient reduction of search path. In our design, we name the repository Cluster Information Table or CIT, containing IL and NIL. Moreover, we utilize dynamic CIT update and NIL optimization to achieve better performance, including making sufficient use of neighbor clusters’ information, forwarding query purposively, getting real-time cluster information, supplying more reliable NIL, etc.

The reminder of this paper is organized as follows. In Section 2, we describe the audio feature extraction in favor of music genre classification. The cluster construction is described in Section 3, containing two main parts: genre-based music file clustering and music genre interest-based peer clustering. Music content-based search mechanism and two alternatives to enhance search performance are demonstrated in Section 4. In Section 5, we present experiment results of comparison with two other algorithms and relevant data analysis; then conclude in Section 6.

2. Feature extraction

Feature extraction aims to describe music files on each peer. Considering subjective feature like timbre is apparently not suitable for computer identification of

music genre, objective feature is emphasized. In terms of the ability that a spectrum can indicate the regularity of the frequency component, especially obviously in music spectrums, features in frequency domain are chosen to provide similarity measures between two music files in this paper.

Apart from the technical difficulties, immature artificial intelligence bridging the features to semantic meaning, it is not practical to extract all possible features embedded in one music file due to the computational cost. Our work typically uses a subset of features with the highest semantic impacts and strongest distinguish capacity for different music genres, which are mean and standard deviation of entropy, mean of energy ratio, mean and standard deviation of lowest frequency location, mean and standard deviation of frequency centroid, mean of low frequency centroid (0-5KHz), and mean of bandwidth.

2.1. Short-time Fourier transform (STFT)

Since music signals have spectral characteristics that vary over time, a non-stationary signal model such as STFT, providing simultaneous information about time and frequency of the music, is necessary. STFT uses a sliding window to compute the Fourier transform thereby providing an estimate of the “local frequency” at a given time.

The STFT of a signal $x(n)$ is given by,

$$\text{STFT}(n, f) = \sum_{m=-\infty}^{\infty} x(n+m) \omega(m) e^{-j(2\pi f)m}$$

where $\omega(m)$ is Hanning window function, better at spectral leakage prevention. The two-dimensional time-frequency distribution (TFD) is defined as,

$$\text{TFD}(n, f) = |\text{STFT}(n, f)|^2$$

which represents the energy in the time interval of $[n, n + \Delta n]$ in the frequency band $[f, f + \Delta f]$.

2.2. Entropy

Entropy describes the energy distribution of a signal. The entropy of a signal in time frame n can be calculated as:

$$H(n) = - \sum_{f=0}^{F_m} P_f(\text{TFD}(n, f)) \log_2 P_f(\text{TFD}(n, f))$$

where F_m refers to the maximum frequency and $P_f(\text{TFD}(n, f))$ is defined as:

$$P_f(\text{TFD}(n, f)) = \frac{\text{TFD}(n, f)}{\sum_{f=0}^{F_m} \text{TFD}(n, f)}$$

which represents the percentage of the energy of the signal at time frame n and frequency index f to the sum energy at time frame n .

2.3. Energy ratio

Energy ratio is a measure of the change rate in the spectral energy over time. Its definition is as follows:

$$ER = \frac{\sum_{f=f_{lower}}^{f=f_{upper}} \text{TFD}(n, f)}{\sum_{f=f_{lower}}^{f=f_{upper}} \text{TFD}(n-1, f)}$$

which represents the energy percentage to the previous time window. Considering the mid and high sub-bands, range from 5KHz to the maximum, contribute little for this metrics, we prefer the calculation in low sub-band, 0-5KHz.

2.4. Brightness

Brightness is also called frequency centroid, which is the weighted midpoint of the energy distribution in a given frame. It is defined by:

$$bt_i(n) = \frac{\sum_{f=0}^{F_m} f \text{TFD}(n, f)}{\sum_{f=0}^{F_m} \text{TFD}(n, f)}$$

Since in most times music signal possesses higher frequency component, frequency centroid is one of the most significant feature for music classification.

2.5. Bandwidth

Bandwidth shows the audio frequency extent. A popular and simple method is to subtract the lowest frequency from the highest frequency. We, however, taking the magnitude-weighted average of the difference between the signal's spectral components and centroid into account, define it as:

$$B(n) = \sqrt{\frac{\sum_{f=0}^{F_m} (f - f_i(n))^2 \text{TFD}(n, f)}{\sum_{f=0}^{F_m} \text{TFD}(n, f)}}$$

Effectively, it shows the spectral shape and the spread of energy relative to the centroid.

3. Cluster construction

Cluster construction could be considered as pretreatment in our design, which includes two main parts: music genre-based file clustering and music genre interest-based peer clustering.

3.1. Music genre-based file clustering

For best clustering performance, high similarity within one class and great diversity among classes, a more sophisticated method, an adaptive k-means algorithm, is employed, which could determine the initial number of clusters and the initial clustering centroid. As a

result, the subjectiveness of k-chosen and blindness of primary cluster division are avoided and the goal that classifies various music genres into different clusters is achieved.

The clustering process of local music files based on music genre is described as follows:

(1) Similarity calculation

Feature matrix $S_{m \times n}$ is comprised by m music files, each of which is represented by an n-dimensional feature vector. The similarity of two feature vectors x and y is calculated as follows:

$$d(x, y) = 1 - \frac{\sum_{i=1}^n x_i y_i}{\sqrt{\sum_{i=1}^n x_i^2} \sqrt{\sum_{i=1}^n y_i^2}}$$

By calculating each group of feature vector, similarity matrix $d_{m \times m}$ is generated as follows:

$$d_{m \times m} = \begin{bmatrix} 0 & d(s_1, s_2) & d(s_1, s_3) & \dots & d(s_1, s_m) \\ & 0 & d(s_2, s_3) & \dots & d(s_2, s_m) \\ & & 0 & \dots & \dots \\ & & & 0 & d(s_{m-1}, s_m) \\ & & & & 0 \end{bmatrix}$$

where s_i is the i^{th} row of $S_{m \times n}$, representing the i^{th} feature vector. If the value of $d(s_i, s_j)$ is smaller than the threshold Λ , defined as mean of all similarity values, the music file represented by s_i is categorized into the same class with all other files represented by s_j . Pre-clustering is accomplished so far.

(2) Pre-clustering result integration

Two problems existed in pre-clustering: one is that a cluster is contained by another one completely; the other is overlapping: one file is considered as the member of more than one cluster. The former needs deletion and the latter requires classifying the file into its nearest cluster. After resolving these two problems, the final number of clusters and the initial centroid are decided, ensuring the rapid convergence of the next step.

(3) Clustering result improvement using traditional k-means

According to the pre-clustering centroid and the decided cluster number, k-means algorithm runs until the clustering results are in constant.

(4) If the music file number of certain cluster is smaller than a given value, cluster deletion and cluster integration to the most similar cluster are triggered to make sure that each cluster's scale is sufficient to complete the further operation.

For a peer P, there are n clusters through music genre-based file clustering, each of which is corresponding to a centroid vector and a standard deviation vector, named μ and σ respectively. The i^{th} cluster in peer P is denoted by $Cluster_{P(i)}$.

3.2. Music genre interest-based peer clustering

According to the small world theory, it is believed that peers with similar interests, especially in music genres, could easily form interest-based cluster in the distributed loose network model. By clustering, peers not only benefit from their neighbors' recourses, but also could visit peers with similar interests through the direct link, performing more efficient search of priority.

Hence gradual network topology construction in the search process is the principal work in this section. That is linking the peers whose shared music files belong to the same music genre by Interest Link (IL) and linking the others by Non-Interest Link (NIL)

New peer startup is a pivotal part, influencing search performance of whole networks. When a beginner participates into a mature P2P network, it builds ILs and NILs to run the same protocol in order to process its own and others' requests. Instead of flooding its cluster construction request, most prevailing method in informed search such as in [15] and [16], we design the concept of Warmhearted Peer (WP) to help a beginner's cluster construction, which lowers the heavy computing cost on peers affected by the request and tremendous transmit data in networks.

WP is decided according to several metrics, such as computing capacity, on-line time, network condition and so forth, by the negotiation among the accessible peers in the scope limited by TTL. It takes more responsibilities, the most important one of which is as an erudite of various cluster information. It contains entrance addresses of several peers which have quite different interests, in other words, have quite different music genre clusters. When received a cluster construction request, WP calculates the distance between the beginner's cluster and cluster it knows and then forward the request to those peers which may possess the biggest possibility to have the same interests with the beginner.

The interest-based peer clustering mechanism is described as follows:

(1) A beginner sends cluster id, centroid, standard deviation of its each cluster to the WP in order to discover interest-based clusters and participate in them.

(2) WP decomposes the cluster construction request, calculates cluster distances according to the knowledge it retains. Then it forwards the request to peers who are more likely to have the similar interest with the beginner and designate the object cluster id.

(3) The receiver calculates the distance between the cluster with the given id and the beginner's cluster it received with the formula as follows:

$$D(X,Y) = \prod_i^n \frac{1}{\sqrt{2\pi\sigma_{X_i}}} e^{-\frac{(\mu_{X_i} - \mu_{Y_i})^2}{2\sigma_{X_i}^2}} \times \prod_i^n \frac{1}{\sqrt{2\pi\sigma_{Y_i}}} e^{-\frac{(\mu_{Y_i} - \mu_{X_i})^2}{2\sigma_{Y_i}^2}}$$

where $D(X,Y)$ represents the distance of two cluster X and Y ; μ_{X_i}, σ_{X_i} and μ_{Y_i}, σ_{Y_i} are the i^{th} components of centroid vector and standard deviation vector of X and Y respectively. If the value is smaller than a given threshold, these two clusters are assumed as similar clusters.

Given that WP may offer the inaccurate hint, object peer should calculate every cluster its own in order to find similar clusters, if the given cluster id turns to fail. Whether the receiver finds the similar clusters or not, it replies to the beginner.

(4) The beginner, say Peer A, categorizes all replied cluster information by their corresponding one. For example, the set $S\{X | X \approx \text{Cluster}_{A(i)}\}$ contains similar clusters with $\text{Cluster}_{A(i)}$. Peer A ranks clusters in set S by an ascending order of their distance and selects the first n clusters to establish ILs, where n is the given maximum capacity of ILs. Peer A caches some repliers from those replying no similar clusters randomly to build NILs for retrieval among clusters, when it sends lookup that dose not belong to its interests.

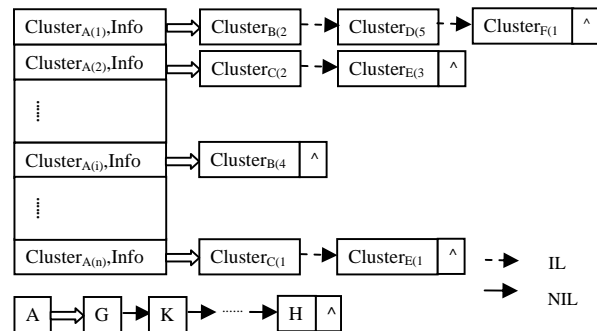


Fig. 1: Cluster information table (CIT).

ILs and NILs are reserved in a repository named "Cluster Information Table" (CIT), whose structure is shown in Figure 1(Peer A's CIT). For example, cluster 2 in Peer B, cluster 5 in Peer D and cluster 1 in Peer F are ILs of cluster 1 in Peer A. Peer G, K and H are NILs of Peer A.

4. P2P music content search with genre interest-based peer clustering

The fundamental idea of our P2P music content search mechanism is query inside the genre interest-based domain using CIT, which is managed and updated in a fully distributed manner. Higher efficiency of music retrieval is guaranteed because of query scope

limitation according to user's interest to music genre. In this section, we show search mechanism and two alternative approaches to enhance the retrieval performance.

4.1. Search mechanism

When a peer joins the P2P network, firstly it constructs a CIT in the way described in Section 3. A searcher, say Peer A, launches a query request given a sample AudioSample, the search process is as follows.

(1) The searcher extracts the AudioSample's features according to the approach presented in Section 2, and obtaining a feature vector p , based on which, calculates the closeness of AudioSample and each local cluster by,

$$C(p, X) = \prod_i^n \frac{1}{\sqrt{2\pi\sigma_x}} e^{-\frac{(p_i - \mu_x)^2}{2\sigma_x^2}}$$

where X is a cluster on the searcher. If $C(p, X_{object})$ is the maximum and greater than the given threshold, AudioSample is viewed as a member of cluster X_{object} .

Then the searcher initializes an interest search, sending its request, including AudioSample's feature vector, object cluster id, to available ILs corresponding to X_{object} .

If there is no eligible $C(p, X)$ or no available ILs, non-interest search is launched. The request is sent to NILs and the object id is set as *null*.

(2) The receiver performs interest matching, if the cluster id in query is not *null*. It calculates the distance between the sample vector and each vector in the designated cluster, and returns the smallest ones.

Otherwise, it performs non-interest matching. First, determining whether AudioSample belongs to any local cluster, if the result is positive, the receiver searches the decided cluster and forwards the message to those corresponding ILs. If the receiver concludes that it has no interest and current hop is still valid, then sends the original request to its NILs.

(3) The searcher gathers search results and ranks them by the ascending order of the vector distance. Then the searcher selects one or more desirable music files, and search process itself is completed.

(4) At the same time, the searcher starts up the CIT update mechanism, an alternative improvement.

4.2. Dynamic CIT update

In our P2P search system, CITs are required to be updated and kept as accurate as possible, for its precision significantly affects the search performance. Considering cluster information might be influenced by many factors, such as the statuses of peers and the

change of shared files, periodic CIT synchronization apparently takes the risk of cluster information incorrectness, and even might lead to search failure and low performance in the full scale of networks. Therefore, dynamic CIT update is more reasonable. It merges update cost with query message spending for search accuracy and network scalability.

In addition, in order to refresh the network, the existed peers would like to know new participators, locating new files and sharing additive computing capacity. Dynamic CIT update could achieve this goal successfully as well.

Dynamic CIT update takes place at following circumstances.

(1) New peer participation

After responding the cluster construction request from a new peer, existed peers update their own CIT according to the new information as well. For scalability, each peer allocates fixed-size IL storage. IL is added and removed from CIT based on their closeness mentioned in Section 3.2. If the new one is more appropriate and the upper limit is reached, it would replace the cluster whose closeness is the smallest. If the storage of ILs is still open, the new cluster is inserted into a corresponding position directly. By this way, existed peers are informed the real-time network condition.

(2) Object cluster deletion

If the lookup indicates that the object cluster visited by IL has disappeared, the searcher deletes the corresponding IL in its CIT.

(3) NIL changing

Peers as the others' NILs should only dispose two kinds of queries: one is the searcher's non-interest search; the other is interest search, launched when the searcher's all corresponding ILs fail. For the latter situation, if the NIL peer obtains the desirable results using local clusters, which means it fails its duty of being a NIL. It is necessary to delete this NIL. If it changes to an eligible IL now, the peer regarding it as NIL before would convert the old NIL into a new IL.

(4) New IL discovery

After all ILs fail, the searcher sends interest search to NILs. Through one or more hops among NIL peers, the desirable results are replied from a responder. If it is outside the current cluster, it is considered as a new discovery of similar interest peer and if the condition we mentioned in (1) is satisfied, it is inserted as a new item in the searcher's CIT

4.3. NIL optimization

An additional improvement to our algorithm is NIL optimization. To avoid the randomness and blindness of choosing NIL peers, an efficient way to evaluate a

better NIL is indispensable. We believe NIL optimization could be a competent way to further enhance the search performance and reducing the data flow and load generated by forwarding query among chosen NIL peers.

An excellent NIL would meet the condition as follows:

- More clusters. Considering straightforwardly more clusters means higher potential hit for a non-interest search.
- More ILs. Possessing more ILs, the NIL peers have a large number of potential candidates to forward the request.
- Stronger capacity. It ensures the functional work of NIL peers in order to decrease the search path and enhance the search efficiency.

NIL optimization is detailed as follows, which consists of NIL rapid construction and NIL maintenance by usefulness.

(1) When the existed peer handles the beginner's cluster construction request, if the beginner's interest is similar with that of itself, the existed peer returns the NIL information it kept to the beginner based on the assumption that the beginner could also feel no interest in the shared content of NIL peers. Due to sharing the information gathered by the sophisticated peers, two benefits are achieved: one is for the network, avoiding the dragging down by the beginner's startup cost; the other is for the beginner, reducing the time of lacking information and stabilizing its search performance.

(2) Each peer continuously keeps track of each CIT's performance and updates its usefulness when new information is learned or network topology is changed. This allows for peers to adapt to dynamic changes and incrementally refine CIT selection, and meanwhile supply more reliable information which the beginner is eager to study.

5. Analysis and experiments

In this paper, we use an event-driven simulator, Peersim, to experiment. The tests place emphasis on precision, recall, network load and peer workload. The number of peers in the network varies from 800 to 1000 to simulate the dynamic network. Other experiment parameters are listed in Table 1.

Experimental data are wav files converted from MP3 files downloaded from Internet by the software of CoolEdit. Representing a music file, each wav file is 5 seconds, mono-channel, 16 bits per sample and sampled at 44.1 kHz. We start by transforming those files into a spectrogram with a window size of 1024 samples which corresponds to about 23 ms at 44.1 kHz. A Hanning window with 50% overlap is used in

STFT. Through the calculation described in Section 2, each file is corresponding to a 9-dimensional feature vector.

Number of Peers	800-1000
Average degree	5
Number of music files assigned to each Peer	70-100
Dimension of extracted feature vector to represent the music	9
TTL value	5

Table 1: Experiment parameters.

In order to compare the performance of our algorithm with other search algorithms, we look at the following metrics:

- Retrieval Precision (RP)

It is the percentage of queries having correct results. We assume that a correct result is that whose music genre is equal to the query sample's music genre and the similarity of them is less than a given threshold.

- Retrieval Recall (RR)

It is the rate of correct results to all files with the same music genre in the network. This metric depicts the usefulness of the algorithm.

- Peer Workload (RW)

It is the average number of messages handled by a peer in a cycle where each peer sends a query request.

- Number of Messages (NoM)

It is the average number of messages generated by one query, which includes initial request, forwarding request and reply request. This metric represents the network load.

- Retrieval Scope (RS)

It is the percentage of nodes visited by one query request. If retrieval scopes are lower, then the number of messages and network cost are comparative lower, and the scalability of the algorithm is better as well.

We compare the performance of the following three search algorithms.

(1) BFS: Query is forwarded in a flooding way.

(2) Peer Clustering (PC) [17]: all files in one peer are regarded as one interest cluster. Peers keep information of other nodes that have similar interest and search inside the interest domain.

(3) Music Genre Retrieval (MGR): Our algorithm described above.

Figure 2 shows the music clustering precision and retrieval precision of some music genres picked up randomly from the experimental data using MGR algorithm. Apparently, MCP is considerable high, for the pretreatment works well, that is, extracted music features are helpful to classify music genre and the adaptive file clustering algorithm is satisfying. Using MGR, retrieval precision of most music genres

achieves approximate 80%. It seems the retrieval precision of some music genre is lower, such as classical and jazz, however two reasons could explain for that. One is unobvious music features; the other is smaller number of those kinds of music files in the network.

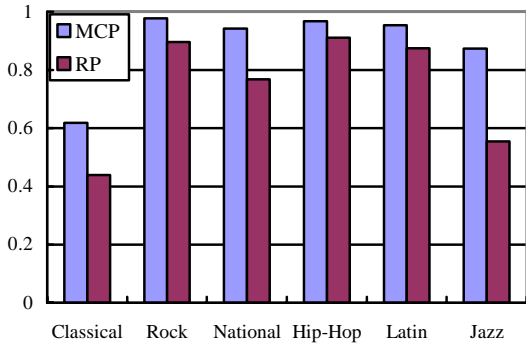


Fig. 2: Music clustering precision (MCP) and retrieval precision (RP) using MGR.

Table 2 shows the comparison results of these three algorithms, including four measure values in average, under the same simulation setting. MGR outperforms both BFS and PC. Using MGR, the outstanding advantage is the highest retrieval precision exceeding 74% and reasonable retrieval recall, approximate 63%, while the peer workload and number of message is reduced deeply: PW is reduced by 74.4% compared to BFS and by 30% compared to PC. Compared to BFS and PC, Number of Message is reduced to 857, 89.3% and 59.8% reduction respectively.

	RP	RR	PW	NoM
MGR	74.86%	63.60%	371	857
BFS	56.63%	64.70%	647	1622
PC	58.72%	53.37%	530	1369

Table 2: Performance comparison.

To evaluate retrieval performance in the dynamic network, we test retrieval recall in the environment where shared music files change and peers join or leave the network frequently. In Figure 3, with the knowledge accumulation while more queries are issued, RR increases steadily and exceeds that of BFS to 72%. Since dynamic CIT update removes the limitation of update periodically and keeps notice of the real-time information concerning both shared resources and peers, MGR avoids the RR shaking happened in PC successfully.

Figure 4 plots retrieval scope of these three algorithms. While the number of queries increasing peers in the network keep more information of similar

peers who share the same interest in certain music genre. Therefore, retrieval is limited within the interest domain, and RR of MGR is much lower than the other two. When the network becomes steady, it is near 35%. Meanwhile, the slope representing decrease trend is more obvious, which means the potential scalability of networks is improved.

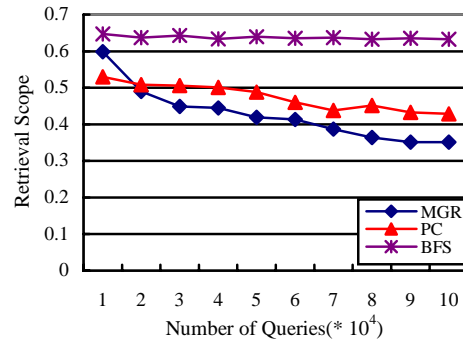


Fig. 3: Retrieval recall versus number of queries issued.

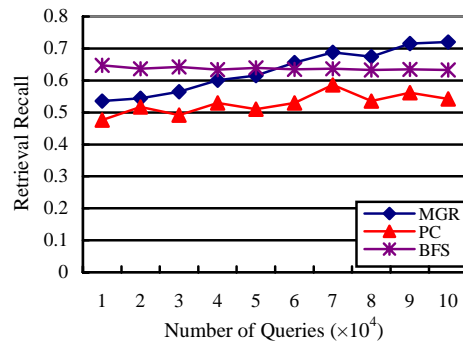


Fig. 4: Retrieval scope versus number of queries issued.

6. Conclusions

In this paper, we present an approach for content-based music genre retrieval over distributed P2P networks. The precise music genre feature extraction in frequency domain and adaptive k-means clustering in each peer accurately classify the music genre according to users' predilection. The CIT is constructed in each peer to contact the clusters among the possible similar interest peers, which increases the search precision and recall remarkably. We also present two alternatives: dynamic CIT update handling the change of network and peer interests; NIL optimization benefiting new peer startup and efficient search among different interest clusters. The experiments prove that the goal of high precision, recall and low network cost, peer workload is achieved successfully.

References

- [1] <http://gnutella.wego.com>.
- [2] M. Mitra, A. Singhal, and C. Buckley, Improving automatic query expansion. *Proc. ACM SIGIR'98*, pp. 206-214, 1998.
- [3] W. Hersh, S. Price and L. Donohoe, Assessing thesaurus-based query expansion using the UMLS metathesaurus. *Proc. the 2000 Annual AMIA, Fall Symposium*, pp. 344-348, 2000.
- [4] K. Nakauchi, Y. Ishikawa, et al, Peer-to-peer keyword search using keyword relationship. *3rd International Symposium on Cluster Computing and the Grid*, pp. 359-367, 2003.
- [5] C. Tang, Z. Xu, and S. Dwarkadas, Peer-to-peer information retrieval using self-organizing semantic overlay networks. *SIGCOMM*, pp. 175-186, 2003.
- [6] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Schenker, A scalable content-addressable network. *SIGCOMM*, pp 161-172, 2001.
- [7] W. Muller and A. Henrich, Fast retrieval of high-dimensional feature vectors in P2P networks using compact peer data summaries. *MIR*, pp. 79-86, 2003.
- [8] I. Lee and L. Guan, A distributed content-based image retrieval system over the peer-to-peer network. *ICUE Proceeding*, 2003.
- [9] <http://www.limewire.org>.
- [10] O.D. Sahin, A. Gulbeden, F. Emekci, D. Agrawal and A.E. Abbadi, PRISM: Indexing multi-dimensional data in P2P networks using reference vectors. *Proc. of the 13th Annual ACM Intl. Conf. on Multimedia*, pp. 946-955, 2005.
- [11] A. Nikseresht and M. Gelgon, Low-cost distributed learning of a Gaussian mixture model for multimedia content-based indexing on a peer-to-peer network. *Proceedings of the 7th ACM SIGMM International Workshop*, pp. 41-48, 2005.
- [12] C. Yang, Peer-to-peer architecture for content-based music retrieval on acoustic data. *Proceedings of the 12th International Conference*, pp. 376-383, 2003.
- [13] J. Wang and S. Yang, Content-based clustered P2P search model depending on set distance. *WI-IATW'06*, pp. 471-476, 2006.
- [14] R. Ye and Y. Yang, ASEKS: A P2P audio search engine based on keyword spotting. *Proceedings of the Eighth IEEE International Symposium on Multimedia*, pp. 615-620, 2006.
- [15] K. Sripanidkulchai, B. Maggs, and H. Zhang, Efficient content location using interest-based locality in peer-to-peer systems. *INFOCOM*, pp. 2166-2176, 2003.
- [16] D. Menascé and L. Kanchanapalli, Probabilistic scalable P2P resource location services. *SIGMETRICS Perf. Eval. Review*, pp. 48-58, 2002.
- [17] I. King, C.H. NG, and K.C. Sia, Distributed content-based visual information retrieval system on peer-to-peer networks. *ACM Transactions on Information Systems (TOIS)*, pp. 477-501, 2004.