MUSICAL RETRIEVAL IN P2P NETWORKS UNDER THE WARPING DISTANCE*

Ioannis Karydis, Alexandros Nanopoulos, Apostolos N. Papadopoulos and Yannis Manolopoulos
Department of Informatics, Aristotle University, 54124 Thessaloniki, GREECE
Email: {karydis,alex,apostol,manolopo}@delab.csd.auth.gr

Keywords: Music Information Retrieval, Similarity Searching, Peer-to-peer Networks, Multimedia Databases.

Abstract: Peer-to-peer (P2P) networks present the advantages of increased size of the overall database offered by a the network nodes, fault-tolerance support to peer failure, and workload distribution. Music file storage and exchange has long abandoned the traditional centralised server-client approach for the advantages of P2P networks. In this paper, we examine the problem of searching for similar acoustic data over unstructured decentralised P2P networks. As distance measure, we utilise the time warping. We propose a novel algorithm, which efficiently retrieves similar audio data. The proposed algorithm takes advantage of the absence of overhead in unstructured P2P networks and minimises the required traffic for all operations with the use of an intelligent sampling scheme. Detailed experimental results show the efficiency of the proposed algorithm compared to an existing baseline algorithm.

1 Introduction

The increasing popularity of the availability of music in computer files gives further impulse to the development of digitised music databases as well as to new methods for Music Information Retrieval (MIR) in these collections. Although abundantly used, even nowadays, the traditional metadata (title, composer, performer, genre, date, etc.) of a music object give rather minimal information about the actual content of the music object itself. Their use aims solely in avoiding including musical content in the query. On the other hand, queries based on humming (using a microphone) or on a small piece of musical file, are a more natural approach to MIR. This type of queries lies within the Content-Based MIR (CBMIR). In CBMIR, an actual music piece is required in order to compare its content with the content of the music pieces already available in the database.

As far as the type of the database is concerned, music file storage and exchange has long abandoned the traditional centralised server-client approach for the advantages of the peer-to-peer networks (P2P). Within the advantageous qualities of the P2P networks lies the increased size of the overall database offered by a P2P network, its fault tolerance support to peer failure by other peers and the workload distribution over a network of available CPUs, since CBMIR is computationally highly intensive. Nonetheless, the very advantages of the P2P network are the same parameters that make P2P information retrieval much more complex than the traditional search methods. That is, the lack of central repository for the documents to be retrieved, the large number of documents available and the dynamic character of the network, introduce an increased degree of difficulty in the retrieval process.

Among numerous classifications, P2P networks can be classified based on the control over data location and network topology in *unstructured*, *loosely structured* and *highly structured* (Li and Wu, 2004). Unstructured P2P networks follow no rule in where data is stored while the network topology is arbitrary (e.g., Gnutella). The absence of structure allows for resilience in dynamic environments (peer join/leave) while no guaranties can be given on the retrieval of existing documents. Moving towards increased structure, both the probability of retrieving existing documents and the overhead of handling peer join-leave augment (e.g., Freenet and Chord). Additionally, P2P networks can also be classified according to the number of central directories of document

*Research funded by the IRAKLITOS national program (2003-2005) under the EPEAEK framework

locations in *centralised* (e.g., Napster), *hybrid* (e.g., Kazaa) and *decentralised* (e.g., Chord). Centralised P2P networks are subject to the same drawbacks for which the traditional server-client model was originally abandoned (network failures due to central peer failure, impaired scalability, joining/leaving of peers not easily handled, possible undesirable dominion of controllers). For these reasons, we focus on decentralised unstructured P2P networks, which overcome the aforementioned drawbacks. The absence of structure was selected for the looseness of control over the data location, that is each peer can share its own documents without hosting any documents of other peers due to locality restrains.[2]

Music representation can primarily be separated in two classes: the symbolic representation (MIDI format) and the acoustic representation (audio format - wav, mp3). The focus of this work is on acoustic data, thus a musical piece can be considered as a time series of signal intensity over time. To measure the similarity of two musical pieces, we utilise the Dynamic Time Warping (DTW) method. The main flexibility of the DTW method is its capability to withstand distortion of the comparing series in the time axis. Accordingly, it allows for two locally out of phase time series that are nevertheless similar to align in a non-linear manner. Since different performances of the same musical piece may include locally differentiated tempo, DTW seems a natural choice for this problem (Large and Palmer, 2002). For this reason, it has been recently proposed for shake of MIR in centralised environments (Zhu and Shasha, 2003; Mazzoni and Danneberg, 2001; Jang et al., 2001; Adams et al., 2004).

In this paper, we focus on the problem of searching, based on DTW, for similar acoustic data over unstructured decentralised P2P networks. The technical contributions of this paper are summarised as follows:

- The development of a novel algorithm that efficiently retrieves audio data similar to an audio query in an decentralised unstructured P2P network.
- The proposed algorithm takes advantage of the absence of overhead in unstructured P2P networks and efficiently minimises the required traffic for all operations with the use of an intelligent sampling scheme on the lower and upper bounds used. The proposed algorithm has such a design that no false negative results occur.
- The detailed experimental results which show the efficiency of the proposed algorithm, and the performance gains compared to an existing baseline algorithm.

The rest of the paper is organised as follows. Section 2 describes related work. Section 3 provides a complete account of the algorithm proposed in this paper. Subsequently, Section 4 presents and discusses the experimentation and results obtained. Finally, the paper is concluded in Section 5.

2 Background and related work

2.1 Searching methods in unstructured P2P networks

In this section we summarise a number of different searching methods for decentralised unstructured P2P networks. Initially, we examine the Breadth-First Search (BFS) algorithm. In the BFS, a query peer Q propagates the query q to all its neighbor peers. Each peer P receiving the q initially searches its local repository for any documents matching q and then passes on q to all its neighbors. In case a P has a match in its local repository then a *QueryMatch* message is created containing information about the match. The *QueryMatch* messages are then transmitted back, using reversely the path q travelled, to Q. Finally, since more than one *QueryMatch* messages have been received by Q, it can select the peer with best connectivity attributes for direct downloading of the match. It is obvious that the BFS sacrifices performance and network traffic for simplicity and high-hit rates. In order to reduce network traffic, the TTL parameter is used (see Section 2). In a modified version of this algorithm, the Random BFS (RBFS) (Kalogeraki et al., 2002), the query peer Q propagates the query q not to all but at a fraction of its neighbor peers.

In an attempt to rectify the inability of the RBFS to select a path of the network leading to large network segments, the $>$*RES* algorithm was developed (Yang and Garcia-Molina, 2002). In this approach, a node Q propagates the q to k neighboring peers, all of which returned the most results during the last m queries, with k and m being configurable parameters. $>$*RES* can be characterised as quantitative than qualitative, since it does not consider the content of the query.

With *ISM* (Kalogeraki et al., 2002), on the other hand, for each query, a peer propagates the query q to the peers that are more likely to reply the query based on the following two parameters; a profile mechanism and a relevance rank. The profile is is built and maintained by each peer for each of its neighboring peers. The information included in this profile consists of the

[2] We must notice that with the examined framework we refer to applications that support content sharing for legal subscribers (e.g., iTunes). Moreover, it is interesting to notice that the proposed approach can be adopted as a means of identification of illegal sharing, by finding sites that share unregistered content.

t most recent queries with matches and their matches as well as the number of matches the neighboring peer reported. Obviously, the strong point of the *ISM* approach is in environments that show increased degree of document locality.

2.2 MIR in P2P networks

The field of combined CBMIR and P2P networks is definitely very young as the inaugural research paper dates back in 2002 (Wang et al., 2002). Despite, the limited number of works that exist, are presented thereinafter.

In this first attempt, the authors of (Wang et al., 2002) present four P2P models for CBMIR. The four models include all centralised, decentralised and hybrid categories. Accordingly, the authors of (Wang et al., 2002) propose a retrieval acceleration algorithm based on difference in pitch between two tones of music and a result filtering method relying on replication removal techniques. Additionally, the authors propose an architecture of a CBMIR P2P system, that falls within the hybrid category of P2P systems.

Another research based on a hybrid configuration is presented in (Tzanetakis et al., 2004). Therein the authors propose a system that utilises both manually specified attributes (artist, album, title, etc.) and extracted features in order to describe the musical content of a piece. The underlying P2P network is a DHT-based system. In such systems each node is assigned with a region in a virtual address space, while each shared document is associated with a value of this address space. Thus, locating a document requires a key lookup of the node responsible for the key.

The author in (Yang, 2003) proposed the utilisation of the feature selection and extraction process that is described in (Yang, 2002) for CBMIR in a decentralised unstructured P2P system. The research considers both a replicated database and a general P2P scenario, while special attention is given on the control of the workload produced at queried peers during query time. Each query is divided into two phases, the first of which includes only a subpart of the actual query vectors, in order to distinguish high probability response peers. Accordingly, a peer ranking occurs and the full query vectors are sent to all peers. Given that a peer has free CPU resources, it decides whether to process a query or not based on the ranking that the specific query received. It is obvious that this approach produces large network traffic, since the full query vectors are sent to all peers, instead of the most promising.

2.3 DTW background information

The efficient processing of similarity queries requires the addressing of the following important issues:

- the definition of a meaningful distance measure $D(S, C)$ in order to express the similarity between two time series objects S and C,
- the efficient representation of time series data, and
- the application of an appropriate indexing scheme in order to quickly discard database objects that can not contribute to the final answer.

One of the most fundamental research issues in time series is the definition of meaningful measures towards time series similarity expression. Given two time series S and C the problem is to define a distance measure $D(S, C)$ which expresses the degree of similarity between S and C. One of the most widely used distance measures for time series is the Euclidean distance (L_2 norm), which has the restriction that both series must be of the same length. Given two time series S and C of length N, the Euclidean distance is defined as follows:

$$D_{euclidean} = \sqrt{\sum_{i=1}^{N}(S_i - C_i)^2} \qquad (1)$$

where S_i, C_i are the value of S and C for the i-th time instance. The Euclidean distance has been widely used as a similarity measure in time series literature (Agrawal et al., 1993; Faloutsos et al., 1994; Chan and Fu, 1999; Kontaki and Papadopoulos, 2004), due to its simplicity.

Several alternative distance functions have been proposed in order to allow translation, rotation and scaling invariance. Consider for example the time series depicted in Figure 1. Note that although all time series have the same shape, they will be considered non-similar if the Euclidean distance is used to express similarity. Translation, rotation and scaling invariance are studied in (Agrawal et al., 1995; Yi et al., 2000; Chan and Fu, 1999; Yi and Faloutsos, 2000).

Taking into consideration that the Euclidean distance does not always meet the application's requirements, Dynamic Time Warping (DTW) has been proposed as a more robust similarity measure. DTW can express similarity between two time series even if they are out of phase in the time axis, or they do not have the same length. The DTW distance $D_{DTW}(S, C)$ between time series S and C is essentially a way to map S to C and vice-versa. This process is also known as *alignment* of time series. If S is of length N and C is of length M, then the distance D_{DTW} can be evaluated by using the following method:

1. An $N \times M$ matrix is constructed, where the cell in the i-th row and the j-th column contains the distance $d(S_i, C_j) = (S_i - C_j)^2$.
2. A warping path is defined which is a contiguous set of matrix cells that defines a mapping between elements of S and elements of C.

Although there are many warping paths that map S to C, what is required is to determine the most promising one, by trying to optimise the cumulative distance $\gamma(i,j)$ in each cell of the warping path. Therefore, the following recurrence is defined:

$$\gamma(i,j) = d(S_i, C_j) + min\{\gamma(i-1, j-1), \gamma(i-1, j), \gamma(i, j-1)\} \quad (2)$$

Figure 1 illustrates an example of two time series aligned by means of the Euclidean distance (Figure 1(a)) and by DTW distance (Figure 1(b)). It is evident that the two time series are similar but their phases are different. However, their similarity can not be captured by the Euclidean distance.

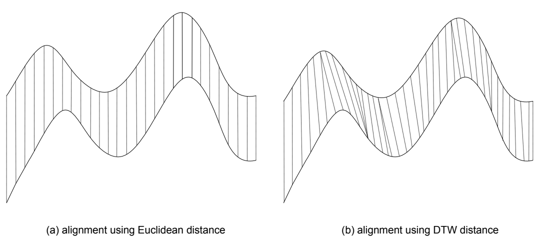


Figure 1: Time series alignment with Euclidean and DTW distances.

The most important disadvantage of the DTW method is that it does not satisfy the triangular inequality, which is a desirable property for constructing efficient indexing schemes and pruning the search space. Moreover, the calculation of $D_{DTW}(S,C)$ is significantly more CPU intensive than the calculation of $D_{Euclidean}(S,C)$. Therefore, an interesting direction for performance improvement is the definition of a lower bound, in order to take advantage of indexing schemes and avoid the computation of DTW when there is a guarantee that the two time series are not similar. In this work we utilise the lower bound proposed in (Keogh and Ratanamahatana, 2004) which is termed LB_Keogh and it is defined as follows:

$$\text{LB\_Keogh(S,C)} = \sqrt{\sum_{i=1}^{N} \begin{cases} (C_i - U_i)^2, & \text{if } C_i > U_i \\ (C_i - L_i)^2, & \text{if } C_i < L_i \\ 0, & otherwise \end{cases}} \quad (3)$$

where U and L is the upper and lower bound respectively for the time series S. Essentially, for each i, the upper bound guarantees that $U_i \geq S_i$ and the lower bound guarantees that $L_i \leq S_i$. In (Keogh and Ratanamahatana, 2004) it has been proven that $LB\_Keogh(S,C) \leq D_{DTW}(S,C)$, and therefore the distance measure $LB\_Keogh(S,C)$ can be effectively used for pruning, resulting in considerably less number of D_{DWT} computations.

3 Proposed method

3.1 Overview

As explained, P2P searching algorithms are based on the following scheme: the node that poses the query examines its contents and finds documents that satisfy the query. Then, it selects a subset of its peers and propagates the query to them. Each peer in its turn examines its contents to find qualifying documents, and then propagates the query to a subset of its peers. To avoid the involvement of a prohibitively large number of nodes, the propagation of queries is restrained by a MaxHop parameter, which determines the number of peers a query should be forwarded. (The MaxHop parameter is equivalently called Time To Leave (TTL).)

In the context examined in this paper, each query searches for similar music documents (i.e., query by content and not by metadata, like title, artist, etc.), where similarity is measured through DTW. To speedup searching, we use lower bounds (LB) that have been developed for DTW, the LB_Keogh in particular (Keogh, 2002). As mentioned, LB_Keogh is based on a bounding envelope, which is defined by the U and L sequences (see Section 2.3). Therefore, in this context, the information that is propagated between nodes comprises the U and L sequences. A node that receives these sequences, computes the LB value between its documents and the envelope. When a LB value is smaller than the user-specified similarity threshold, then the actual query sequence is propagated to this node[3] and the actual DTW distance is computed between the query and the corresponding document.

The queries we consider constitute music phrases, that is, excerpts of the music documents that are a type of units of music information[4]. This holds especially in the context of query by humming, where users tend to hum a piece that is (i) relatively short, (ii) well identified and separated within a song. The identification of phrases can be done following the methodology presented in (Zhu and Shasha, 2003). In particular, a transcription algorithm (Klapuri, 2004) can produce the pitch information of the acoustic sequence. Time intervals, corresponding to phrases in the pitch information, are detected in between the time instances that silence exists (the same time intervals produce the phrases in the corresponding acoustic sequence). In summary, we are interested in finding music documents that contain phrases similar to the query se-

[3] The query can be directly propagated from the node that initially posed the query, since the currently visited node always knows the address of this initial node.

[4] A minimum-length portion of the musical piece that is meaningfully independent and complete within a piece of music.

quence. Similarity through DTW is suitable in this context, since the properties of DTW help in alleviating errors that humming produces.

An important observation is that acoustic data tend to be very large. Although queries are music phrases (i.e., parts of the music sequences), the number of elements in a phrase of even few seconds can be several hundred thousands. The length of the U and L sequences is equal to the length of the query sequence. This means that a straightforward approach, which directly propagates U and L sequences between nodes, will result into an extremely large traffic over the P2P network. Moreover, when the length of the envelope's sequences are large, the computation of LB in each node can become rather costly. This violates the need of a P2P network to burden the participating nodes as little as possible. Notice that the aforementioned requirements are not present in other contexts, like the searching of similar text documents over a P2P network, where queries consist of up to some tenths of terms.

We propose a two-fold scheme which significantly reduces the traffic over the P2P network when querying music documents by content. The scheme works as follows:

- It reduces the length of the envelope's sequence by sampling them. However, plain sampling can be ineffective, since it leads to underestimation of LB. For this reason, we describe a novel sampling method to reduce the length of the sequences without significantly affecting the computation of LB. Additionally, we are interested in not introducing false-negatives due to the use of sampling.

- It uses (whenever possible) a compact representation of the sampled sequences of the envelope. The representation comprises a kind of compression for the sequences, but it does not burden the nodes of the P2P network with the cost of decompression. If the latter is not undesirable, further compression can be achieved through the use of existing methods. We do not explore this direction, since it does not affect the relative performance of the proposed scheme against the plain one that directly propagates the envelope (i.e., the performance of both methods will be equally improved).

In the following we describe the aforementioned issues in more detail. We have to notice that, for simplicity, we use the BFS algorithm as a basis for searching over the P2P network. The examination of the proposed scheme in more advanced searching algorithms (e.g., ISM, >RES) is actually a matter of current research. In a larger version of this work, we will include a comparison that considers such searching algorithms as well.

3.2 Sampling and representation

Let the considered phrase length be equal to N. The length of each query Q, and therefore of its upper (U) and lower (L) sequences, will also be equal to N. We would like to sample U and L, so as to obtain two sequences U' and L', each of length $M \ll N$. Initially, we assume that uniform sampling is performed. In this case, we simply select each time the $(i \times N/M)$-th element of U and L, where $1 \leq i \leq M$. When we compute the LB_Keogh between the query sequence Q and a data sequence, we consider each phrase C of length N in Q. Each phrase has to be sampled in the same way as U and L. This leads to a sampled phrase C'. Therefore, we get a lower-bound measure LB', given as:

$$LB' = \sqrt{\sum_{i=1}^{M} \begin{cases} (C'_i - U'_i)^2, & \text{if } C'_i > U'_i \\ (C'_i - L'_i)^2, & \text{if } C'_i < L'_i \\ 0, & otherwise \end{cases}} \quad (4)$$

In the aforementioned equation, the third case (i.e., when $L'_i \leq C_i \leq U'_i$) does not contribute in the computation of LB'. The problem of uniform sampling is that, as it selects elements without following any particular criterion, it tends to select many elements from U and L that result to this third case. Therefore, LB' may become a significantly bad underestimation of LB that would have been computed if sampling was not used. The underestimation of the lower-bound value will result to an increase in false-alarms, thus incurring high traffic.

To overcome this problem, we propose an alternative sampling method. We sample U and L separately. Initially, we store the elements of U in ascending order. In U' we select the first M elements of this ordering. Respectively, we sort L in descending order and we select the first M elements in L'. The intuition is that the selection of the smallest M values of U, helps in increasing the number of occurrences of the first case (i.e., when $C'_i > U'_i$), since the smallest the value of U'_i is, the more expected is to have a C'_i larger than it. An analogous reasoning holds for the sampling of L'.

It is easy to see the following:

Lemma 1 *The sampling of U and L does not produce any false negatives.*

Proof. While computing LB', due to sampling, the first and second cases of Equation 4 occur less times than while computing LB (i.e., without sampling). Therefore, $LB' \leq LB$. Since $LB \leq D$ (where D the actual distance computed with DTW), we have: $LB' \leq D$. Thus, no false negatives are produced. □

The separate sampling of U and L presents the requirement of having to store the positions from which elements are being selected in U' and L'. If the positions are stored explicitly, then this doubles the amount of information kept ($2M$ numbers for storing U' and L' and additional $2M$ numbers for storing the positions of selected elements). Since this information is propagated during querying, traffic is increased. For this reason we propose an alternative representation. To represent U', we use a bitmap of length N (the phrase length). Each bit corresponds to an element in U. If an element is selected in the sample U', then its bit is set to 1, otherwise it is set to 0. Therefore, the combination of the bitmap and the M values that are selected in U' are used to represent U'. The same is applied for L'. This representation is efficient: the space required for U' is $M + \lceil N/8 \rceil$ bytes.[5] The plain representation requires $5M$ bytes (since it requires only one integer, i.e., 4 bytes, to store the position of each selected element). Thus, the proposed method is advantageous when $N < 32M$, i.e., for sample larger than about 3% (our experiments show that samples with size 10% are the best choice).

3.3 The similarity searching algorithm

As previously explained, the similarity searching algorithm is on the basis of breadth-first-search over the nodes of the P2P network. The algorithm that uses the proposed sampling and representation methods, is denoted as BFSS (breadth-first-search with sampling). The pseudo-code for BFSS is given in Figure 2. Each time, the current node n is considered. A TTL value denotes how many valid hops are remaining for n, whereas T_s is the user-defined similarity threshold. It is assumed that sequences U' and L' carry also the associated bitmaps.

Evidently, the movement of the actual query sequence from the node that commenced the query to the currently visited node, increases the traffic (not being sampled, the query sequence has rather large length). For this reason, it is important not to have a large number of false-alarms.

The algorithm that does not use sampling (denoted as BFS) may produce less false-alarms. However, between each pair of peers it has to propagate U and L sequences, with length equal to the one of the query sequence. Therefore, it is clear that there is a trade-off between the number of additional false-alarms produced due to sampling and the gains in traffic from propagating sampled (i.e., smaller) envelopes. This trade-off is examined through the experimental results in the following section.

[5] Each element in an acoustic sequence is in the range 0-255, thus it requires one byte.

Procedure BFSS(Node n, **int** TTL, Sequence U',
    Sequence L', **float** T_s)
begin
1. **foreach** data sequence D in n
2.   **foreach** phrase C of D
3. $l = LB'(C, U', L')$
4.   **if** $l < T_s$
5.   get query sequence
6.   compute actual DTW distance, D,
                   between phrase C and query sequence
7.   **if** $D \leq T_s$
8.   include C in answer set
9. **if** TTL > 0
10.   **foreach** peer p of n that has not been visited yet
11.   BFSS(p, TTL-1, U', L', T_s)
end

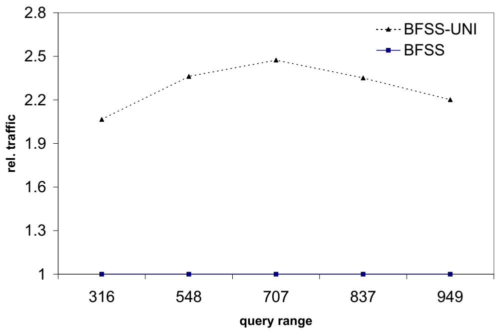
Figure 2: The BFSS algorithm.

4 Experimental results

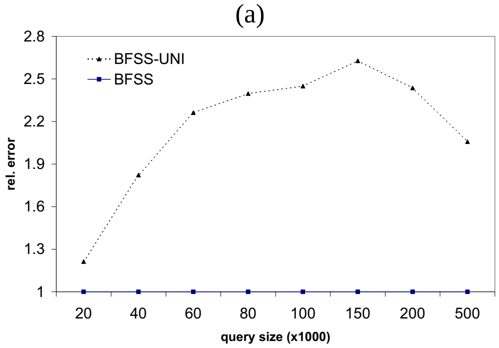
The performance of the considered similarity searching algorithms was compared through simulation. The P2P network had 100 nodes and the average number of neighbors for each node was a random variable with average value equal to 7 (this kind of topology is called logarithmic). We used 500 real acoustic sequences, which correspond to various pop songs. Each song was sampled at 11 KHz and the average duration was about 5 minutes. To represent the fact that music songs (especially popular ones) are shared among several nodes, we replicated each sequence. The number of replications for each sequence was randomly variable with average value equal to 10.

The evaluation metric is the average traffic (measured in MB) that each query incurs. The parameters we examine are: the sample size, query size (length of query sequence), query range (the user-defined threshold for similarity), and TTL value (max allowed number of hops).

In our first experiment, we focused on BFSS and compared the proposed sampling method (this method is denoted as BFSS) against uniform sampling (this method is denoted as BFSS-UNI). The results are depicted in Figure 3. Figure 3a illustrates the relative traffic between BFSS and BFSS-UNI (i.e., the traffic of the latter is normalised w.r.t. the traffic of the former) against the query range. As shown, BFSS-UNI incurs about twice the traffic that BFSS does. As already explained, this is due to the fact that uniform sampling produces a bad underestimation of the lower bound value. This can be further understood when examining the discrepancy, denoted as error, between the bounds produced by BFSS and BFSS-UNI, and the actual bound produced by LB_Keogh. The relative error between BFSS and BFSS-UNI (i.e., the latter is normalised w.r.t. the former) is given in Figure 3b, against the query size. The error of BFSS-UNI ranges between 1.3 times the error of BFSS (for smaller queries) and 2.8 times (for medium sized queries).



(a)

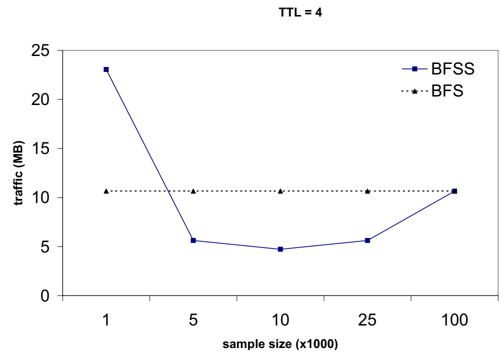


(b)

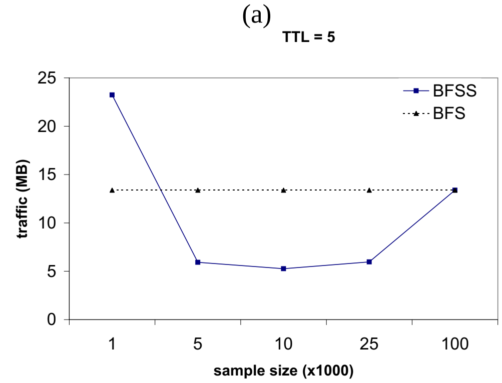
Figure 3: BFSS vs. BFSS-UNI (a) Relative traffic. (b) Relative error w.r.t. actual LB_Keogh value.

We now move on to compare BFSS against BFS (i.e., the method that does not use any sampling at all). For BFSS we examined several sample sizes. The results are depicted in Figure 4, whereas BFS has a constant value, as it does not use sampling. In Figure 4a, TTL was set to 4, query size was 100,000, and query range was set to 0 (i.e., exact match). As shown, for very small samples (with 1,000 elements), BFS performs better. This is expected, since the use of a very small sample affects BFSS by resulting to a large number of false alarms (due to bad underestimation of lower bound values), which increase traffic. However, by increasing the sample size, BFSS becomes better and clearly outperforms BFS. It is interesting to notice that the best performance is for sample size equal to 10,000 (i.e., 10% of the original query size). Finally, for large sample sizes, both methods converge to the same traffic. Analogous results are obtained for the case where TTL is set to 5. It also worths noticing that the traffic of BFS is significantly more increased than the traffic of BFSS does, compared to the case when TTL was 4.

Next, we compared BFSS against BFS for varying query size and query range. Figure 5a illustrates the results for the former case. The size of sample for BFSS was set each time to 10% of query size, TTL was set to 4 and query range was set to 0. As shown, BFSS clearly outperforms BFS in all cases, except for rather small queries. Again, for very very small queries, the resulting sample is very small and many false-alarms are produced. Finally, Figure 5b depicts the results for the latter case (varying query range). Query size was set to 100,000, sample size was 25%, and TTL was set to 5. BFSS clearly compares favorably with BFS.



(a)

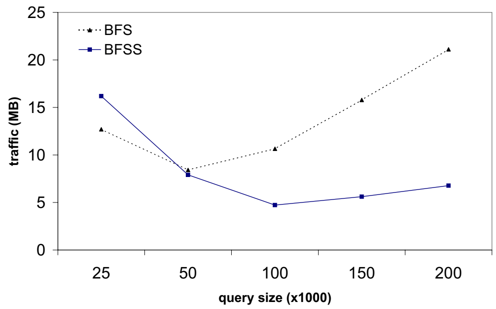


(b)

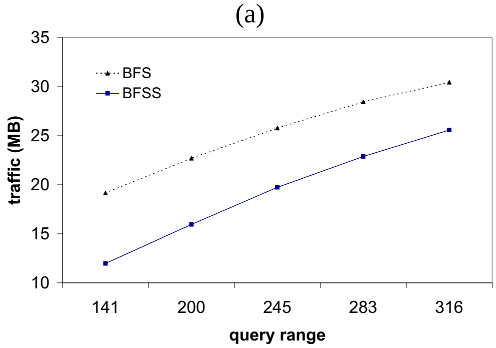
Figure 4: BFSS vs. BFS (a) Traffic (in MB) when TTL=4. (b) Traffic (in MB) when TTL=5.

5 Conclusions

We have presented a novel algorithm, which efficiently retrieves similar audio data. The proposed algorithm takes advantage of the absence of overhead in unstructured P2P networks and minimises the required traffic for all operations with the use of an intelligent sampling scheme. Additionally, the algorithm has such a design that no false negative results occur. Detailed comparative evaluation to an already existing algorithm showed significantly reduction in the traffic produced by a query.



(a)



(b)

Figure 5: BFSS vs. BFS w.r.t. (a) query size, (b) query range.

Further work is oriented towards the examination of the proposed scheme in more advanced searching algorithms (e.g., ISM, >RES).

REFERENCES

Adams, N. H., Bartsch, M. A., Shifrin, J. B., and Wakefield, G. H. (2004). Time series alignment for music information retrieval. In *Proceedings 5th ISMIR*.

Agrawal, R., Faloutsos, C., and Swami, A. (1993). Efficient similarity search in sequence databases. In *Proceedings 4th FODO*, pages 69–84.

Agrawal, R., Lin, K. I., Sawhney, H. S., and Swim, K. (1995). Fast similarity search in the presence of noise, scaling, and translation in time-series databases. In *Proceedings 21st VLDB*.

Chan, K. and Fu, A. W. (1999). Efficient time series matching by wavelets. In *Proceedings 15th ICDE*, pages 126–133.

Faloutsos, C., Ranganathan, M., and Manolopoulos, Y. (1994). Fast subsequence matching in time-series databases. In *Proceedings ACM SIGMOD*, pages 419–429.

Jang, J. S. R., Lee, H. R., and Chen, J. C. (2001). Super mbox: An efficient/effective content-based music retrieval system. In *Proceedings 9th ACM MM*, pages 636–637.

Kalogeraki, V., Gunopulos, D., and Zeinalipour-Yazti, D. (2002). A local search mechanism for peer-to-peer networks. In *Proceedings 11th CIKM*, pages 300–307.

Keogh, E. (2002). Exact indexing of dynamic time warping. In *Proceedings 28th VLDB*, pages 406–417.

Keogh, E. and Ratanamahatana, A. N. (2004). Exact indexing of dynamic time warping. *Knowledge and Information Systems*.

Klapuri, A. (2004). Automatic music transcription as we know it today. *Journal of New Music Research*.

Kontaki, M. and Papadopoulos, A. N. (2004). Similarity search in streaming time sequences. In *Proceedings 16th SSDBM*.

Large, E. and Palmer, C. (2002). Perceiving temporal regularity in music. *Cognitive Science*, 26(1):1–37.

Li, X. and Wu, J. (2004). Searching techniques in peer-to-peer networks. In *Handbook of Theoretical and Algorithmic Aspects of Ad Hoc, Sensor, and Peer-to-Peer Networks*. CRC Press.

Mazzoni, D. and Danneberg, R. B. (2001). Melody matching directly from audio. In *Proceedings 2nd ISMIR*, pages 17–18.

Tzanetakis, G., Gao, J., and Steenkiste, P. (2004). A scalable peer-to-peer system for music information retrieval. *Computer Music Journal*, 28(2):24–33.

Wang, C., Li, J., and Shi, S. (2002). A kind of content-based music information retrieval method in a peer-to-peer environment. In *Proceedings 2nd ISMIR*, pages 178–186.

Yang, B. and Garcia-Molina, H. (2002). Improving search in peer-to-peer networks. In *Proceedings 10th ACM MM*, pages 5–15.

Yang, C. (2002). Efficient acoustic index for music retrieval with various degrees of similarity. In *Proceedings 10th ACM MM*, pages 584–591.

Yang, C. (2003). Peer-to-peer architecture for content-based music retrieval on acoustic data. In *Proceedings 12th WWW*, pages 376–383.

Yi, B. K. and Faloutsos, C. (2000). Fast time sequence indexing for arbitrary lp norms. In *Proceedings 26th VLDB*.

Yi, B. K., Jagadish, H. V., and Faloutsos, C. (2000). Efficient retrieval of similar time sequences under time wrapping. In *Proceedings 16th ICDE*, pages 201–208.

Zhu, Y. and Shasha, D. (2003). Warping indexes with envelope transforms for query by humming. In *Proceedings ACM SIGMOD*, pages 181–192.