Diversity based Relevance Feedback for Time Series Search

Bahaeddin Eravci
Department of Computer Engineering
Bilkent University
Ankara, Turkey
beravci@gmail.com

Hakan Ferhatosmanoglu
Department of Computer Engineering
Bilkent University
Ankara, Turkey
hakan@cs.bilkent.edu.tr

ABSTRACT

We propose a diversity based relevance feedback approach for time series data to improve the accuracy of search results. We first develop the concept of relevance feedback for time series based on dual-tree complex wavelet (CWT) and SAX based approaches. We aim to enhance the search quality by incorporating diversity in the results presented to the user for feedback. We then propose a method which utilizes the representation type as part of the feedback, as opposed to a human choosing based on a preprocessing or training phase. The proposed methods utilize a weighting to handle the relevance feedback of important properties for both single and multiple representation cases. Our experiments on a large variety of time series data sets show that the proposed diversity based relevance feedback improves the retrieval performance. Results confirm that representation feedback incorporates item diversity implicitly and achieves good performance even when using simple nearest neighbor as the retrieval method. To the best of our knowledge, this is the first study on diversification of time series search to improve retrieval accuracy and representation feedback.

1. INTRODUCTION

Time series are encountered frequently in a wide range of applications ranging from finance to healthcare that generate data with a speed that was not possible to this day. Accumulation of such data is gaining momentum with new technologies, such as the decline in the price and the miniaturization of different sensors (pressure, temperature, inertial, etc.). With this data waiting to be translated into knowledge, researchers are attempting to find different ways of extracting information with ever growing interest.

A time series can be defined as a sequence of real numbers with temporal association between elements. The main focal points of mining time series fall into the following categories: pattern recognition, classification, clustering, and summarization. For each mining task, the initial and the fundamental problem has been to identify a good representation of the time series. Many different representations have been proposed to mine information each with a different perspective which fits into different applications and user intents. A family of representations has been studied in time domain while others have incorporated frequency domain properties as well [1]. A measure of similarity is needed to execute most of the related tasks. For this purpose, similarity measures have been proposed which take into account the different nature of the time series with respect to traditional data [5, 12]. Indexing methods for processing queries have been shown a wide interest in the community [7]. Modeling the time series and forecasting the unknown values has also attracted interest [11].

The size and the generation speed of the time series can prohibit users to see the complete data entries. While time series search methods have been widely studied, there is limited work in optimized relevance feedback for such data. To the best of our knowledge, no prior work has considered diversity to improve time-series retrieval and relevance feedback. In this paper, we address these two fundamental challenges in time series. We first explore relevance feedback for retrieval in time series databases. We follow a query by example approach in which the user submits a query reflecting the user intentions. Based on the initial query, a set of items according to a criteria is presented to the user. A challenge is to present a set of results optimized to collect feedback, as opposed to the top matching set, that learns more information about the user intent in the feedback phase. The user evaluates the initial items to enhance the results in the next rounds of retrieval. We utilize effective representations of time, and use diversity between time series data in different rounds of retrieval process to further enhance the user satisfaction.

We also aim to utilize diversity based relevance feedback to identify the most appropriate time series representation for a query or an application. Given the large amount of work in representation methods, finding the right one is an essential and challenging task. We develop a method of feedback in which the initial list is populated using different representations, and the system learns the appropriate one. The user feedback is utilized to converge to the representation which satisfies the user most. The method uses a representation feedback for increasing the total items from the best performing representation for the next rounds of retrieval. This can be useful in dynamic databases where the properties of the system are changing or the user intentions can vary. We note from the experimental results

Proceedings of the VLDB Endowment, Vol. 7, No. 2

that besides the intended use as representation feedback, the method implicitly embodies item diversity as well.

The contributions of this study include the following:

- We identify time series representations for relevance feedback which incorporates various types of global and local information. We use dual-tree complex wavelet transform for similarity/diversity relevance feedback due to its power to identify information localized both in time and frequency domain. We construct relevance feedback for time series by tuning such systems without the need of explicitly defining features like amplitude shift, periodicity, etc.
- The performance of the relevance feedback is enhanced by using diversity between time series in different rounds of the feedback.
- We propose methods that choose suitable representation types according to the user intention. This enables an on the fly learning by exploiting the valuable feedback from the user.
- We perform a rich set of experiments on real time series that provides insights on relevance feedback and diversity for time series databases.

The results of the experiments show 25 point absolute increase in precision on average, with 45 increases on some cases, by the proposed relevance feedback framework. Introducing diversity into relevance feedback increases mean precision by 7% relative to relevance feedback with no transformations and 2% relative to the relevance feedback on NN top-k search using the proposed representation in this paper. Representation feedback method converges to the best performing representation as the relevance feedback session advances and incorporates item diversity implicitly in the process. A weighting algorithm further enhances the performance of the relevance feedback.

2. RELATED WORK

There has been significant work in information retrieval community for relevance feedback (RF) since it was proposed in the 1960s [24, 26, 27]. The first methods have concentrated on relevance feedback query movement in which the query point was moved toward the relevant items. Dimension weighting methods have been proposed for the same objective [13]. There has been use of relevance feedback in the image and multimedia retrieval applications [34, 17, 25]. Lately, researchers state RF problem as a classification problem and propose solutions in the context of machine learning [30, 31].

The problem of combining relevance and diversity for ranking documents has been studied by Carbonell and Goldstein [8] in the context of text retrieval and summarization. They define Maximal Marginal Relevance (MMR) objective function to reduce redundancy while maintaining query relevance in re-ranking retrieved documents. The problem of ambiguity in queries and redundancy in retrieved documents has been studied in [10]. They propose an evaluation framework and emphasize the importance of objective evaluation functions. Chen and Karger propose a retrieval method for maximizing diversity, which assigns negative feedback to the documents that are included in the result list [9]. Studies on using relevance, diversity and density measures to rank documents in information retrieval have found place in the literature lately [33]. Diversifying search results to increase user satisfaction for answering ambiguous web queries has been investigated in [2] and to improve personalized web search in [23]. Graph based diversity measures for spatial and multidimensional data has been proposed in [18]. Methods to find the best representative of a data set based on clustering has been investigated in [21].

Time series data mining research has immense literature on the representation of the time series, similarity measures, indexing methods, and pattern discovery [11]. Besides using geometric distance on coefficients ([1]), dynamic time warping (DTW) is used to identify similarities between time series due to its success in non-aligned data [5, 28]. More recent research has been in mining multivariate time series ([22]) and streaming data [20, 4].

On the contrary to the information retrieval, relevance feedback and diversity have not attracted much attention in time series community yet. Representation of time series with line segments along with weight associated to the related segments and explicit definition of global distortions have been used in time series relevance feedback [14, 15]. We are not aware of any studies using representation feedback for time series retrieval and diversification in such systems.

3. METHOD

3.1 Problem Definition

The main focus of this study is increasing the accuracy of the search problem in time series databases. There is a wide range of applications, such as determining products with similar selling patterns in online commerce, identifying correlated ECG from the past patients to a specific patient for diagnosis, and analyzing seismic waves to identify potential commonalities among events. In such applications, the user poses a query on a database of time series and aims to find "relevant" time series according to the specific domain and user.

Consider a database, $TSDB$, of N time series: $TSDB = \{TS_1, TS_2, ..., TS_N\}$. Each element of $TSDB$, TS_i, is a vector of real numbers which can be of different size, i.e. $TS_i = [TS(1), TS(2), ..., TS(L_i)]$ where L_i is the length of particular TS_i. Given a query, TS_q (not necessarily in $TSDB$), find a result set (subset of $TSDB$) including k time series that will satisfy the expectation of the user. The user is able to give feedback by annotating the result set as relevant or irrelevant. The relevance feedback system uses these cues to increase the performance in terms of user satisfaction.

3.2 Time Series Representation

The capability of representation to decode the user intention is clearly essential for the performance. The appropriate representation depends on the application and user intent. For example, if the user intention is to figure out the time series with a certain periodicity, frequency domain approaches like DFT (Discrete Fourier Transform) would be more successful. An important general property to consider is the shift-invariance of the transform. This allows correct retrieval even if two time series are off in the time scale. Handling of different length time series with ease is another important feature for the representation. For an effective

diverse time-series definition, we seek that the transform should in some manner give the chance of comparing "local" properties of the two time series. Based on these properties, we focus on two different representation methods and approaches: based on Wavelet Transform and based on SAX (Symbolic Aggregate approXimation [19]). On one side, we will illustrate our approaches using these methods; and on another we will evaluate the appropriateness of these successful representations and provide insights on their use for our relevance feedback and diversity study.

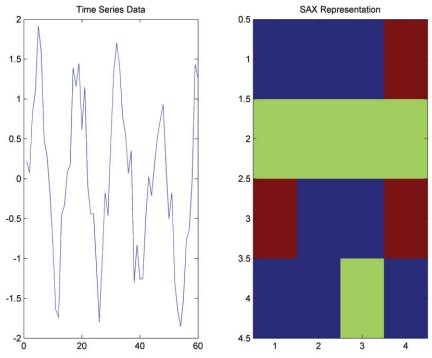

Figure 1: An example SAX bitmap representation

SAX has gained a prominent place in the time series research community due to its success in representation. It transforms the data into a string with a fixed alphabet which gives the chance to exploit different methods already found and used in string manipulation. After the transformation of the string, a method called SAX-bitmap is proposed which turns the string into a bitmap image (matrix) using the different substrings included in the whole string. It has been shown that this method is intuitive and useful in representing the time series and is a perceptually appropriate representation. In our context, we use it as a transformation of the time series to a vector which is then used with different distance measures for retrieval. The method effectively counts the number of different local signatures after transforming the original time series to SAX representation. The level of the representation (L) corresponds to the length of the local patterns in the SAX representation. [19]

The length of the output of the SAX-Bitmap transform is M^L where M is the number of symbols used in the SAX transformation, which is independent of the time series length (L_i). SAX naturally divides the time series into blocks and normalizes the block within itself which inherently extracts local features of the time series. SAX-bitmap method makes use of this transformed string and counts the occurrences of particular substrings. The number of occurrences in the whole time series gives information about the global features as well.

Wavelet Transform (and its variants Discrete WT, Continuous WT, Complex WT, etc.) is a type of time-frequency representation used extensively in time-series domain. The transformed data (scaleogram) provides a good frequency and time localization. CWT is relatively shift-invariant with respect to other flavors of the algorithm. The level of the representation (L) in CWT corresponds to the height of low

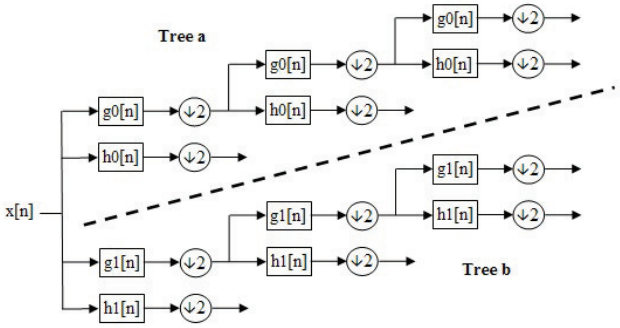

Figure 2: Three level Dual-Tree Complex Wavelet Transform

pass components of CWT which in turn corresponds to different details of the low pass components. The upper part of the tree is the real part of the transform and the lower part of the tree is the complex part given in Figure 2. Because of this nature, the transformation is called the Dual-Tree Complex Wavelet Transform [29]. The magnitude of the complex and real part is used in this paper. The length of the transformed data is independent of the number of levels and is given by $2^{\lceil \log_2 L_i \rceil}$.

CWT has a similar approach but with a different perspective. CWT extracts some low-pass features, i.e., components which are in the lower frequency band and are relatively slowly varying giving an averaged version of the overall series and high pass features, i.e., components which are in the higher frequency band and are relatively fast varying, related to detail and differential information of the series. Down-sampling of the series along the branches allows the transform to extract information from different zooms of the data. As a summary, the branched tree process decomposes the time series into "local patterns" in both time and frequency with different scales. We can see that this decomposition of the time series is suitable for diversity as different subsets of the information provided by the transformation can give a different meaningful perspective to the data.

3.3 Relevance Feedback of Items

Relevance feedback is an essential tool in information retrieval to increase user satisfaction. The user is given a set of relevant items in the first iteration and annotates the relevance of each item. A feedback mechanism is established where items that are more relevant are presented in the next iteration. The basic model is given in Figure 3. Each component of the system shall be explained in the consecutive parts.

We first transform time series into a representation (CWT, FFT, SAX, PCA etc. according to properties of the time series in the database) such that different features are captured. The preprocessing typically involves a normalization (unit-norm, zero-mean, etc.) as necessary. Given an initial time series query (TS_q), the relevant transformation (SAX, CWT or some other) is applied and a transformed query vector, q, is calculated which will be used in the retrieval process. T_i denotes the transformed TS_i according to the transformations explained in the previous sections, i.e., $T_i = \mathcal{F}(TS_i)$.

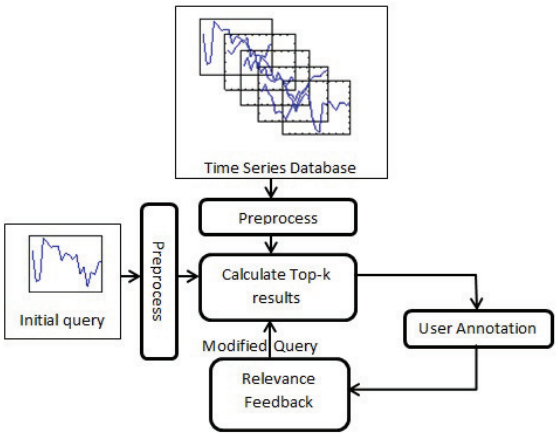

Figure 3: Relevance feedback system

3.3.1 Top-K Diverse Retrieval

Top-K retrieval identifies k time series to be presented to the user who is seeking information relevant to the query, q. The general method used is to find the k-nearest neighbors of q which is a list of time series ranked according to a defined distance function with respect to q. The main assumption in this retrieval process is that the distance to the query is related to the user preference. However, there are always data points close to the query in theoretical sense yet not related to the interest of the user. Moreover, the intent of user can be already ambiguous itself.

In the above explained case, as the name nearest neighbor (NN) implies, only the data points in the vicinity of the query point are retrieved. But the database can include time series items very similar to each other giving very limited novel information about the user intentions since q is already known. This will in turn make the relevance feedback less useful and will waste the time and annotation effort of the user.

The user needs to be given somewhat diverse results that are still around the query point. With more diverse choices provided, the successive iterations of the relevance feedback would be expected to better meet the user intentions. In the following analysis, we illustrate our intuition of utilizing diversity using a simple model in this context. For a query, q, we find a top-k list using NN with the last element d distance away from the query. A 1-D Gaussian data distribution for relevant set, $R \sim \mathcal{N}(0, \sigma^2)$ and irrelevant set, $IR \sim \mathcal{N}(\mu, \sigma^2)$ is depicted in Figure 4.

Assuming, there are N relevant and M irrelevant items, we can find the number of relevant ($k1$) and irrelevant items $k2$ in the top-k list with approximations as:

$$\begin{aligned} k_1 &= N.\int_{q-d}^{q+d} R(x)\ \mathrm{d}x \approx R(q).2d \quad if \quad k_1 \ll N \\ k_2 &= M.\int_{q-d}^{q+d} IR(x)\ \mathrm{d}x \approx IR(q).2d \quad if \quad k_2 \ll M \\ k &= k_1 + k_2 \end{aligned} \tag{1}$$

We can then define and calculate precision for the query as:

$$Prec(q) = \frac{k_1}{k_1 + k_2} = \frac{N.R(q)}{N.R(q) + M.IR(q)} \tag{2}$$

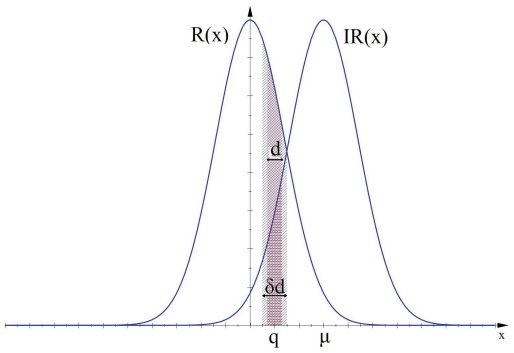

Figure 4: Data distributions used in analysis

This formula illustrates that if R and IR is separable (μ is very large) precision will be high as expected. It also shows that if the query point is near the mean of R than the precision will be high and vice versa. This implicitly states that the better we capture the model of the relevant set, the better the performance will be. We can capture the model of the relevant set using the relevance feedback by modifying the query according to the feedback from the user. Consider a simple model that constructs the new query for the next iteration (q_2) as the average of all the relevant items, i.e.:

$$\begin{aligned} q_2 &= \sum_{i=1}^{k_1} R_i = \int_{q-d}^{q+d} x.R(x)\ \mathrm{d}x \\ &= \sqrt{\frac{\sigma^2}{2.\pi}}[\mathrm{e}^{q-d} - \mathrm{e}^{q+d}] \end{aligned} \tag{3}$$

If we use a diverse retrieval setting around q, we will have a top-k list that spans a larger distance(δd) which is also shown in Figure 4. In this case we get a q_2' from the relevance feedback as:

$$\begin{aligned} q_2' &= \sum_{i=1}^{k_1} R_i = \int_{q-\delta d}^{q+\delta d} x.R(x)\ \mathrm{d}x \\ &= \sqrt{\frac{\sigma^2}{2.\pi}}[\mathrm{e}^{q-\delta d} - \mathrm{e}^{q+\delta d}] \quad \delta > 1 \end{aligned} \tag{4}$$

This ensures $q_2' < q_2$ which increases the precision using Equation 2 we have shown earlier. If the precision is already high which might be the case if R and IR are well separated than the precision increase will not be significant.

We incorporate diversity in top-k retrieval for different iterations of the relevance feedback. We explore two different methods to diversify the top-k results: maximum marginal relevance [8] and cluster based diversity. Maximum Marginal Relevance (MMR) combines the distance of the tested item to the query and the other items already in the relevant set. The distance used is given in Equation 5 and a greedy algorithm is used until a specific number of items is found from the whole set of items. $Dist$ function can be any distance function of choice. When λ is chosen as 1, the $DivDist$ becomes the distance and the result turns to a mere top-k nearest neighbor result. When λ decreases, the importance of the initial query decreases which gives an end result of diverse set of items within itself but are also

related to the query.

$$DivDist(T_q, T_i, R) = \lambda Dist(T_q, T_i) - \frac{1}{|R|}(1-\lambda)\sum_{j=1}^{|R|} Dist(T_i, T_j) \quad (5)$$

The second term of the $DivDist$ involves pairwise comparisons of data points in the database which is independent of the query and is performed repetitively for each query. To decrease the running time of the algorithm, we use a look-up table that stores all the possible pairwise distances calculated once at the beginning for the particular database. This clearly reduces the running times significantly.

The Cluster based diversity (CBD) uses a different approach unlike a formal optimization parameter like the one given in Equation 5 and is similar to the method for finding best representatives of a data set proposed by Liu et al. in [21]. This method retrieves Top-αk elements with a nearest neighbor approach and then clusters the αk elements into k clusters. The data points nearest to the cluster centers or representatives are chosen as the retrieved points shown to the user. The parameter α controls the diversity desired, increasing α increases the diversity of the result set. If α is chosen as 1 then the results are the same as the NN case. We implement a k-means algorithm for the clustering phase in this study. An advantage of this method is that the tuning parameter α is intuitive and predictable.

We note that the performance of using diversity depends on the underlying data distribution. Since the diversification methods are based on some distance definition, the particular meaning of the elements is important for the overall performance of the system. This is especially important in the case of time series, because there is an autocorrelation between the elements. This fact stresses that the transformation of the time series should be inline with general user intentions, be suitable for varying properties of time series acquired from different applications, and should have the power of decomposing time series into meaningful parts which have novel information. These properties have been considered in choosing the suitable representation.

3.3.2 Relevance Feedback for Retrieved Items

After a round of iteration, the user is given the chance to evaluate the results and grade the items presented. One can utilize a variety of approaches for relevance feedback, such as Rocchio's algorithm [24]. According to the user preference, an additional query is formed using the relevant and irrelevant items for successive rounds of relevance feedback. Equation 6 details the procedure where Rel is the set of items graded relevant, $Irrel$ is the set of items graded irrelevant by the user.

$$q_{new} = \frac{1}{|Rel|}\sum_{i=1}^{|Rel|} Rel_i - \frac{1}{|Irrel|}\sum_{i=1}^{|Irrel|} Irrel_i \quad (6)$$

The new query vector is not dependent on the original query but the original query affects the results via Equation 3. The system uses the original query plus the newly formed query vectors in the previous RF stages to calculate the distances. We also implemented a Rocchio algorithm which directly modifies original query at each stage and found that the modified version performs better.

Algorithm 1 High-level algorithm for relevance feedback system

```
Initialize parameter k // number of items to retrieve
Initialize parameter NumberOfIterations
q_1 is given as the initial query
TSDB is given as the time series database
// Parameters for MMR
λ = [λ_1, λ_2, ..., λ_NumberOfIterations]
// Parameters for CBD
α = [α_1, α_2, ..., α_NumberOfIterations]
for i = 1 → NumberOfIterations do
  // Find Top-k results
  if Nearest Neighbor then
    R = Top-K(q_1,...,q_i,TSDB)
  else if MMR then
    R = Top-K_MMR(q_1,...,q_i,TSDB,λ_i)
  else if CBD then
    R = Top-K_CBD(q_1,...,q_i,TSDB,α_i)
  end if
  // Let user grade the retrieval results
  (Rel,Irrel) = UserGrade(R)
  // Expand query points via relevance feedback
  q_{i+1} = Relevance_Feedback(Rel,Irrel)
end for
```

For the successive iterations, a distance is calculated with respect to all the query points of the previous iterations which is detailed in Equation 7.

$$Dist(q_1, q_2, ..., q_N, T_{test}) = \frac{1}{N}\sum_{i=1}^{N} Dist(q_i, T_{test}) \quad (7)$$

where N is the RF iteration number

The high level algorithm for relevance feedback is shown in Algorithm 1.

3.3.3 Learning Weights of Time Series using Feedback

The transformations aim to retain as much information as possible in the original time series so that the available information for the subsequent tasks are not reduced but are represented in a more discriminative fashion. Dimension reduction can be applied subsequently to the transformed time series if the related problem can efficiently be solved by focusing on a limited number of properties of the time series.

As we want to learn the user intent and have no assumption about it, we need to utilize generic local and global properties which can have meanings for different user intentions. Since the framework is built on general principles, each user intention may only be associated with a subset of these properties. However, this subset of interests is not known a priori and should be learned according to the user interactions with the system. For this purpose, we implement a simple learning step based on a linear model of the properties introduced by representation by modifying similar approaches that have been used in information retrieval community for relevance feedback [13]. The user intention is modeled as a linear combination of the properties of the time series which weighs each property according to the relevance

of the respective property for his/her objective:

$$\text{User Intention} = \sum_{j=1}^{N^F} \beta^j . T^j$$

where N^F: $Numberoffeatures$

and $\sum_{j=1}^{N^F} \beta^j = 1$

Algorithm 2 Estimation and update of β parameters

INITIALIZATION
T_i^j denotes the jth feature of time series i (or its representation)
σ^j denotes the standard deviation for jth feature on all of time series
N^F: Number of features
for $j = 1 \rightarrow N^F$ **do**
  σ^j = standard deviation over all values of $T_1^j, T_2^j, \ldots, T_N^j$
end for
for $j = 1 \rightarrow N^F$ **do**
  $\beta^j = \frac{1}{N^F}$
end for
ESTIMATION
for Each RF iteration **do**
  if $|Rel| > 3$ **then**
    Calculate standard deviation differences
    for $j = 1 \rightarrow N^F$ **do**
      $\hat{\sigma}^j$ = standard deviation over $T_i \in (Rel)$
      $\Delta\sigma^j = \frac{\sigma^j}{\hat{\sigma}^j}$
    end for
    Normalize
    $\Delta\sigma^j = \frac{\Delta\sigma^j}{\sum_j \Delta\sigma^j} \quad \forall j : 1, \ldots, N^F$
    Update
    $\beta^j = \frac{\Delta\sigma^j + \beta^j}{2} \quad \forall j : 1, \ldots, N^F$
  end if
end for

Estimating the β parameters from the user feedback can be done in various ways. We use an approach based on the comparison of the standard deviations of the particular feature. If a particular feature is favored by the user, we expect to see some consistent values in that particular field. We attribute an importance for each property based on the decrease with respect to the total database in the standard deviation of the particular feature. We simply use this value as an estimate for β parameters. The Algorithm 2 explains the details of the estimation and the update process for β parameters.

We use the β parameters calculated at the previous RF iteration for weighting the similarity measures of top-k ranking to incorporate the user preferences in the present ranking of the time series. This weighting method can be used in all of the previously explained diverse retrieval methods by introducing the related parameters when the similarities are calculated.

3.4 Representation Feedback

As expected and observed from the experimental results with item diversity, the representation has significant effects on the performance. This may be caused by either the properties of the data such that the meaningful and useful clusters may not be separated in one representation with respect to other. This can partially be solved via doing experiments with different representation against some performance parameter and choosing the representation accordingly. But this approach would fail when the data properties change as the entries to the database change dynamically. The second main cause is that a particular representation may not be able to represent the user intention as well as some other representation. Different users' intentions can vary even if using the same time series database. For example, some users might be interested in time domain features while others might be looking for a frequency domain feature. A universal time series representation appropriate for all the possible application areas and user intentions is simply not possible. Hence, we naturally see different time series representation proposed in the research community for different cases and applications.

To help overcome the representation choice problem, one can feedback the system with both related items and related representation(s). We investigate two different methods for representation feedback. The first method partitions the top-k list according to the different representations available and tries to converge to the best performing representation. The second method concatenates different representation vectors and uses the weighting structure explained in the previous sections to learn the best performing parts of a variety of representations. The high level flow of the representation feedback algorithm is given in Algorithm 3.

3.4.1 Representation Feedback via Top-k List Partitioning

In this method, we construct the top-k list as composed of different representations and use the annotation to converge to the representation which satisfies the user most. This is besides the traditional use of modifying query for the next iteration. An advantage of fusing different time series representation is that each representation can explain a different property of data which can not be seen when one of them is considered. This property can ensure a rich list of top-k elements with diversity between items implicitly caused by the retrieval process enhancing the relevance feedback performance.

The method splits the k value for the system into different k_i values (where $\sum_{i=1}^{Number\ of\ Representations} k_i = k$), each designating the number of elements to be chosen from each of the different representations. An equal distribution can be used in the first iteration. If a prior knowledge is present about the user, this value can be used as well. The system, in the long run, can make use of the feedbacks of all the users using the system to tune the initial k_i values.

Top-k elements from each of the representations available are retrieved using any of the NN, MMR or CBD methods. k_i items from each of the top-k list from different representations are chosen and presented as a total of top-k time series to the enquirer.

Following the user feedback, k_i values are updated according to the performance of the related representation. The initial and the update of the k_i are given in Equation 8. The relevance feedback for the time series items via creating new query points is also performed according to details given in

Algorithm 3 High-level algorithm for representation feedback system

```
r is given as the number of representations
Initialize parameter NumberOfIterations
q_1 is given as the initial query in time domain
TSDB_r is given as the time series database with representation r
if Representation Feedback via Weighting then
  TSDB = Concatenation the representations (TSDB_r)
  Initialize β weights
  Initialize parameter k // number of items to retrieve
else if Representation Feedback via Partitioning then
  Initialize parameter k_i for i : 1...r
end if
for i = 1 → NumberOfIterations do
  // Find Top-k results using any alternative method
  if Representation Feedback via Weighting then
    R = Top-K(q_1,...,q_i,TSDB,β)
  else if Representation Feedback via Partitioning then
    R = ∅
    for j = 1 → r do
      R = R ∪ Top-K(q_1^j,...,q_i^j,TSDB_j,k_j)
    end for
  end if
  // Let user grade the retrieval results
  (Rel,Irrel) = UserGrade(R)
  // Expand query points via relevance feedback
  if Representation Feedback via Weighting then
    q_{i+1} = Relevance_Feedback(Rel,Irrel)
  else if Representation Feedback via Partitioning then
    R = ∅
    for j = 1 → r do
      q_{i+1}^j = Relevance_Feedback(Rel,Irrel)
    end for
  end if
  // Update representation feedback parameters
  if Representation Feedback via Weighting then
    β = UpdateWeights(β,Rel,Irrel)
  else if Representation Feedback via Partitioning then
    k_i = UpdateK(k_i,Rel,Irrel)
  end if
end for
```

previous sections.

$$\begin{aligned} &Initialization: \\ &k_i = k/r \text{ where } r \text{ is number of representations} \\ &Update: \\ &k_i = \frac{Number\ of\ relevant\ items\ from\ i}{Number\ of\ relevant\ items} \quad \forall i \leq r \end{aligned} \tag{8}$$

3.4.2 Representation Feedback via Weighting

This method extends the weighting in Section 3.3.3 for representation feedback. We concatenate the vectors of each representation to form a vector consisting of different properties of the time series each depending on the perspective and expressive power of the representation. This aggregate representation is processed by the retrieval engine according to the previously explained procedures.

By using the aggregate representation, we introduce a new level of information to the system which enables flexibility for representing the user intentions. We can think of each of these representations as new hyper-vectors to map the user intentions more thoroughly according to the linear user intention model. The increase in the span of possible mappings depends on the chosen representations. The set of representations should be chosen such that the linear combination of the vectors includes more of the user intent. This can be achieved by representations with different perspectives and each extracting novel information that are desired to be uncorrelated.

4. EXPERIMENTS

We performed an extensive number of experiments on real data, whose details are given in the consecutive parts, to validate the performance of the proposed methods and provide insights on diversity and relevance feedback for time series databases.

4.1 Dataset

We used the UCR Time series data sets for our experiments [16]. The data sets on the website are partitioned into training and test for supervised learning purposes. Since our application is unsupervised we have lumped the two parts together into a single database increasing the size of the data sets. The data sets used are listed in the Appendix. The listing in the Appendix is used as the numbering in the figures of the next sections.

The number of classes in the data sets varies from 2 to 50. The sizes of the data sets vary from 56 to 9236. The lengths of the time series in the data sets vary from 24 to 1024. We experimented on 29 of the data sets available on the web site. The data sets we used are diverse enough to give a good understanding of the performance of the proposed methods under different constraints.

4.2 Experimental Setting

First, the time series were transformed into SAX Bitmap and CWT. The SAX bitmap parameter was $N = L_i$, $n = \lceil \frac{N}{5} \rceil$ with an alphabet of four. This means that we did not use the sliding window when transforming to SAX and for each subseries of length 5, SAX procedure produced an output symbol out of a possible of four symbols. While transforming the SAX series to SAX Bitmap we counted the number of occurrences of all possible L permutations of the four symbols in the SAX series to complete the transformation ending with a vector of length 4^L. We chose three values for $L \in 1, 2, 3$ in the experiments. The values for N and n should be optimized for different data to increase the performance. Since our main aim is not to find the best representation but to enhance RF via diversification, we did not try to optimize the parameters with respect to the data and used the same parameters for each data set for a controlled experimentation.

For the Complex Wavelet Transformation we utilized the Dual-Tree CWT implementation given in [6]. We used 5 different levels, $L \in 1, 2, 3, 4, 5$ for experimenting. We used both the complex and real parts by taking the absolute value of the CWT coefficients.

Because of the immense load of results, we present the results for SAX with $L = 3$ and CWT with $L = 5$, which were overall the best performing representations in our experimental setting for all data sets. We run the same methods

on the original time series (TS) without any modification to see the effectiveness of the representations used.

We also performed experiments to see the effects of the representation feedback in which we used CWT with $L = 5$, SAX with $L = 3$ and the original time series as different representations.

In the experiments, we explored 5 different methods of top-k retrieval:

1. nearest neighbor (NN)
2. MMR with $\lambda = [0.5, 1, 1]$ $(MMR(\lambda_1))$
3. MMR with $\lambda = [0.5, 0.75, 1]$ $(MMR(\lambda_2))$
4. CBD with $\alpha = [3, 1, 1]$ $(CBD(\alpha_1))$
5. CBD with $\alpha = [3, 2, 1]$ $(CBD(\alpha_2))$

In the algorithmic configuration explained above, we try to see the effect of diversification in the total performance of the RF system. We investigate how the level of diversification in different iterations affects the system performance. We note that $MMR(\lambda_2)$ and $CBD(\alpha_2)$ cases decrease the diversity in a more graceful way. But $MMR(\lambda_1)$ and $CBD(\alpha_1)$ go directly to NN form after the first iteration. We did not try to optimize the parameters (λ and α) of the diversification schemes as the values present themselves as mere intuitive estimates.

We implemented a unit normalization method for each dataset and used cosine distance for all the experiments.

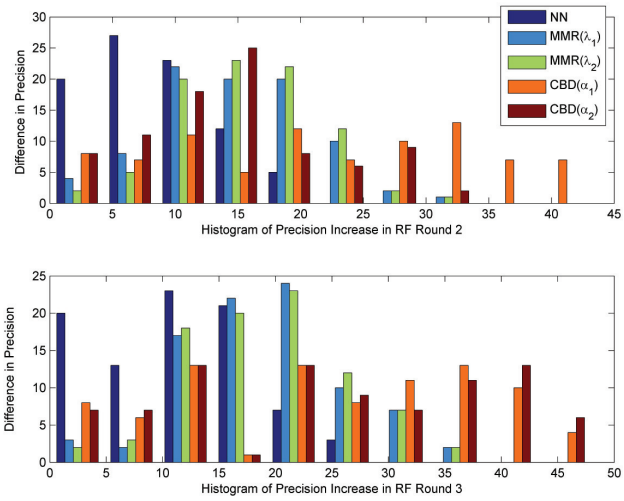

Figure 5: Absolute increase in performance with relevance feedback

We implemented the method given in [15] to compare our algorithms. This method represents the time series with piecewise linear approximation (PLA-RF) and uses a weight for each piece to show importance of the related part of the series when calculating the distances to query. These weights are modified in each iteration of feedback according to the annotations of the user.

Our experiments aim to evaluate the satisfaction of a user who wants to find the time series that are in the same class as the query. The elements in Top-10 which are in the same class as the query are considered relevant for the feedback system and the performance of the overall system is defined as the precision value based on the classes of the retrieved time series. The experiments were performed on a leave-one-out basis. Each time series in the database is selected as a query and relevance feedback system is executed with the related parameters using the database excluding the query itself. The class of the elements in Top-10 list is recorded for each query and iteration. The precision (in percentage) for the query is calculated using the recorded information for different iterations of the relevance feedback. The averaged precision over all the queries in the database is considered as the final performance parameter.

$$Prec(T_q) = \frac{1}{10}\sum_{i=1}^{10}\delta(i) * 100 \quad (9)$$

$$AvePrec = \frac{1}{N}\sum_{\forall T_q} Prec(T_q) \quad (10)$$

where $\delta(i) = \begin{cases} 1 & \text{if class of } T_q \text{ is equal to class of } R_i \\ 0 & \text{otherwise} \end{cases}$

Table 1: Average Increase (absolute) in Precision

RF Round	2	3
NN	8.35	11.31
MMR(λ_1)	14.80	19.00
MMR(λ_2)	15.63	19.25
CBD(α_1)	21.49	24.46
CBD(α_2)	14.55	25.02
PLA-RF	3.82	4.7

4.3 Experimental Results and Discussions

4.3.1 *Experimental Results for Item Diversity*

The experimental results for item diversity are given in Figure 6 for all the data sets. Each row in the figure corresponds to one of five retrieval methods explained in the previous section and each row corresponds to the representation (CWT, SAX and unmodified time series (TS)) used in the experiment. In each individual graph, the average precision in different RF iterations is plotted with the data set number given in x-axis. We present an aggregate result here to summarize the results.

We calculated the difference in precision between the different rounds and the first round of RF for a particular representation, method and data set. The histogram of the resulting data is illustrated in Figure 5. The average increases are provided in Table 1 to depict the performance increase with the use of RF with different methods. We performed a t-test (paired and unknown variance) between the average values given in the table and a zero mean distribution to verify the statistical significance of the improvement. The p-values, in the range of 10^{-15}, are much smaller than 0.05 which is considered as a threshold for significance. RF with the configurations given in this study works in all cases without any dependence of data type or data representation. We notice that it provides significant benefits for time series retrieval with 45 point absolute increases in some cases in terms of precision. The figures show that introducing diversity at some level increases the retrieval performance. We note that the proposed methods outperform the state of the art. The experiments produced large amount of results given

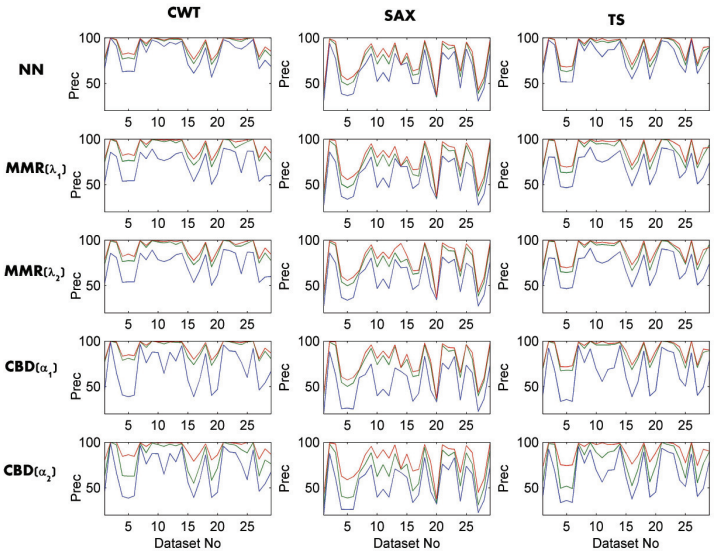

Figure 6: Performance with relevance feedback for the datasets

the large number of parameters, e.g., time series data type, the representation, the method used in the retrieval process, the parameters of the retrieval process. For illustration, we considered the time series without any transformation as the reference representation and NN method as the reference retrieval method.

For each of the RF round and each of the data set, the performance results are normalized to a total 100 with respect to the base case for that data set and RF round. Figure 7 shows the normalized results averaged over all the data sets. In nearly all data sets, CWT based approach outperformed

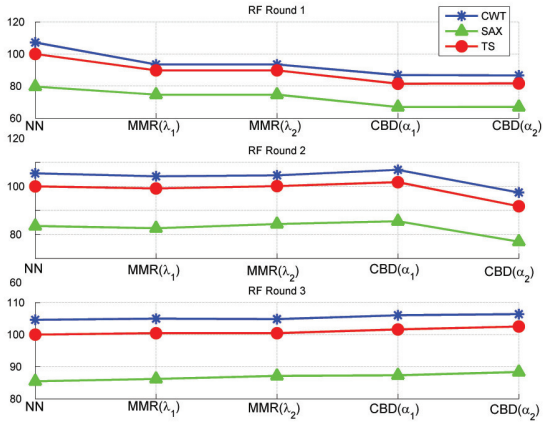

Figure 7: Normalized performances of different methods and representations

both our versions of SAX and the time series without any transformation (TS). We note that SAX and CWT parameters were not optimized and different results may be achieved by optimization of the related transformations. We did not perform such analysis since it would diverge us from the main contributions of the study. However, CWT performed well consistently with no need of parameter optimization.

As expected, NN achieves the best performance in the first iteration of RF. However, providing diversity in the first iteration leads to a jump in performance and beats NN method in nearly all the cases. Best performing method, CBD(α1), has put 7% (p-value < 0.05) performance increase over the reference case and 2% (p-value < 0.05) over the case which uses NN method over CWT. The diversity increases its effect further in the third iteration of RF where NN is outperformed in even more cases with similar performance advancements in average precision. In terms of number of times each method tops the performance chart with respect to different cases; we see that CBD(α_1) performs best in second iteration and CBD(α_2) in the third iteration. This also points out the enhancement in performance due to increasing diversity if number of iterations increases. MMR(λ_2) and the two CBD methods perform better than NN with regards to the increase in average precision. CBD outperforms MMR in each case.

We investigated the effect of the data set purity on the performance. We calculate the purity value of each data set by using a simple clustering scheme. We consider the mean of each class (centroid) as the representative for the related class and we assign each data to the class of the nearest representative. The classification is compared to the ground truth and the average over all data is considered as the purity of the data set. The method actually checks the purity of each Voronoi cell formed by the centroids of the classes. This parameter which is in the range $0-1$ can also describe the separability of the classes. We plot the normalized precision described in the previous paragraphs against the purity of the related dataset and the corresponding linear fit in Figure 8. Positive effect of diversity is reduced in the cases where classes are already pure and separable. Effect of diversity increases when the classes are more interleaved which is the harder case in terms of system performance.

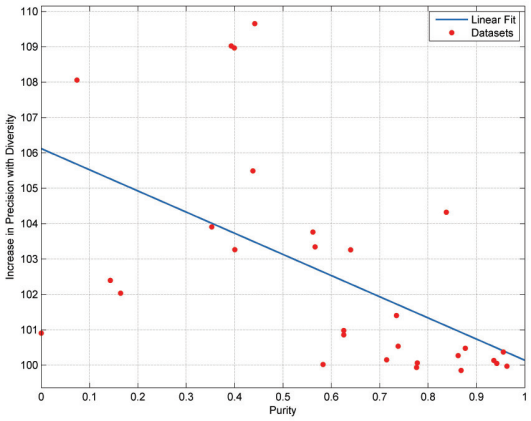

Figure 8: Normalized performances of different datasets versus purity of dataset

4.3.2 *Results for Representation Feedback via Top-k List Partitioning*

The experimental results for both item diversity and representation feedback are summarized in the following. The averaged results which are normalized with the performance NN method in the first round of RF are given in Figure 9. An important observation is that NN performs better or equal to the methods incorporating item diversity in addition to representation feedback. This is due to the fact that each of the representations already gives some diverse results and this enhances the RF process without further diversity. This shows that diversity of items have been achieved implicitly by the process of using more than one representation. Figure

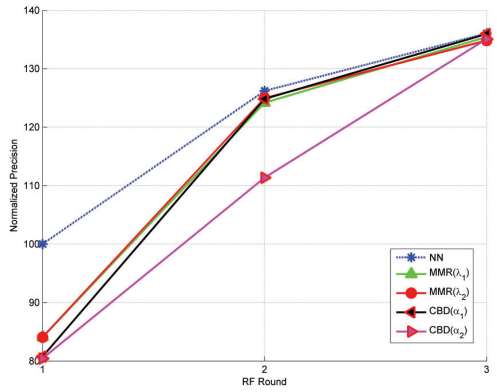

Figure 9: Normalized performances of method with representation feedback

10 shows the results for representation feedback in comparison with the best performing method found in the item diversity experiments. The figure illustrates that our aim is achieved and as the RF iterations increase the RF system converges to the best performing representation. We used NN method for comparison, since it performed just as good as the other methods in the case of representation feedback.

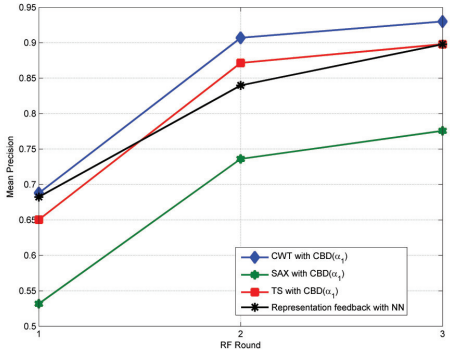

Figure 10: Comparison of representation feedback with item diversity

4.3.3 *Results for Learning Important Properties with RF*

We experimented on the learning of weights for important features using user feedback as well. For clarity we performed experiments on three randomly chosen data sets: ECG200, fish and synthetic control. We performed NN and the $CBD(\alpha_1)$ diverse retrieval methods to see how diversity influences these experimental cases as well. The results are given in Figure 11.

The importance learning algorithm does not seem to suit the SAX representation. In the second round of RF, the performance of the retrieval system falls behind the initial round of retrieval. This may be because SAX-bitmap features may well be correlated with each other or SAX representation is not suitable for weighted similarity measures.

In most of the cases when SAX is not considered, weighting scheme can provide significant performance gains. Moreover, we see that even when using the diversity retrieval engines the weighting method can provide enhancements.

We have chosen CWT and TS representation and used the representation feedback via weighting approach. The results show that combining different representations can enhance the performance of the RF system. An interesting result can be seen in the synthetic control data set where the aggregated representation (CWT+TS) has performed considerably better than the individual representations. This can also be an example to our previous assertion that if the span of the chosen representations starts to explain different parts of user intention the performance increase can be more satisfactory.

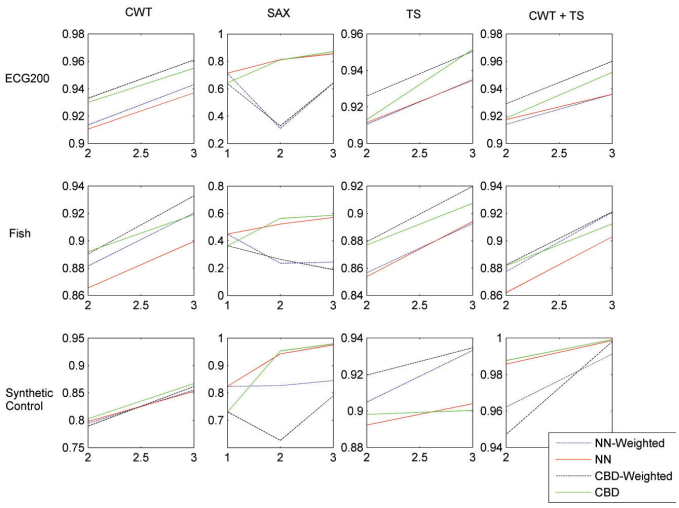

Figure 11: Performance of RF with weighting

5. CONCLUSION

In this paper we studied relevance feedback and diversity enhancing the established representations in time series applications. While diversity and relevance feedback have been successfully explored in information retrieval and text mining, they have not attracted enough attention from time series databases and data mining community. The experimental results showed that regardless of the representation, relevance feedback, even with a simple model, increases the retrieval accuracy. This is valid even in just one iteration. This confirms the potential of relevance feedback in the domain of time series. Furthermore, we showed that diversity in the first iteration of the relevance feedback increases the performance further in many of the cases. Cluster based diversity with the given parameters has performed best in our experimental setting in terms of precision. The tuning of the system parameters can lead to even higher increases in the performance. Diversity performs better in non-pure and non-separable data cases which are usually the challenging cases in the performance of retrieval systems.

We also developed two methods which feedback the suitable representation type for the next iteration. The first method partitions the top-k list and diversifies the items which in turn increases the performance even if a simple nearest neighbor retrieval is used. The other method augments different representations and enables the feedback via estimated importance parameters. The results show that this method can be used for better mappings of the user intentions in the time series relevance feedback framework.

The presented results and analysis can serve as a basis for new approaches for diversification of time series data. During our exploration of the topic, we experimented different potential approaches for diversification of time series. We adapted matching based similarity (e.g., k-n match [32]) and STFT (short time fourier transform [3]) for time series diversification. However, we did not include their discussions in this paper as the proposed approaches produced better results than these possible alternatives.

6. ACKNOWLEDGMENTS

This study was funded in part by The Scientific and Technological Research Council of Turkey (TUBITAK) under grant EEEAG 111E217. We thank the anonymous reviewers for their helpful recommendations and insights.

7. REFERENCES

[1] R. Agrawal, C. Faloutsos, and A. N. Swami. Efficient similarity search in sequence databases. In *Proceedings of the 4th International Conference on Foundations of Data Organization and Algorithms*, FODO '93, pages 69–84, 1993.

[2] R. Agrawal, S. Gollapudi, A. Halverson, and S. Ieong. Diversifying search results. In *Proceedings of the Second ACM International Conference on Web Search and Data Mining*, WSDM '09, pages 5–14, 2009.

[3] J. Allen. Short term spectral analysis, synthesis, and modification by discrete fourier transform. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 25(3):235–238, 1977.

[4] F. Altiparmak, E. Tuncel, and H. Ferhatosmanoglu. Incremental maintenance of online summaries over multiple streams. *IEEE Transactions on Knowledge and Data Engineering*, 20(2):216–229, 2008.

[5] D. J. Berndt and J. Clifford. Using Dynamic Time Warping to Find Patterns in Time Series. In *Knowledge Discovery and Data Mining*, pages 359–370, 1994.

[6] S. Cai and K. Li. Dual-tree complex wavelet transform matlab code.

[7] A. Camerra, T. Palpanas, J. Shieh, and E. J. Keogh. isax 2.0: Indexing and mining one billion time series. In *Proceedings of the The 10th IEEE International Conference on Data Mining (ICDM 2010)*, pages 58–67, 2010.

[8] J. Carbonell and J. Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.

[9] H. Chen and D. R. Karger. Less is more: probabilistic models for retrieving fewer relevant documents. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 429–436, 2006.

[10] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '08, pages 659–666, 2008.

[11] T.-C. Fu. A review on time series data mining. *Engineering Applications of Artificial Intelligence*, 24(1):164 – 181, 2011.

[12] M. Gavrilov, D. Anguelov, P. Indyk, and R. Motwani. Mining the stock market (extended abstract): which measure is best? In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 487–496, 2000.

[13] Y. Ishikawa, R. Subramanya, and C. Faloutsos. Mindreader: Querying databases through multiple examples. In *VLDB 98, Proceedings of 24rd International Conference on Very Large Data Bases, August 24-27, 1998, New York City, New York, USA*, pages 218–227, 1998.

[14] E. Keogh and M. Pazzani. An enhanced representation of time series which allows fast and accurate classification, clustering and relevance feedback. In *Proceedings of 4th KDD*, 1998.

[15] E. Keogh and M. J. Pazzani. Relevance feedback retrieval of time series data. In *Proceedings Of The 22 Th Annual International ACM-SIGIR Conference On Research And Development In Information Retrieval*, pages 183–190, 1999.

[16] E. Keogh, Q. Zhu, B. Hu, Y. Hao, X. Xi, L. Wei, and C. A. Ratanamahatana. The ucr time series classification/clustering homepage, 2011.

[17] M. L. Kherfi, D. Ziou, and A. Bernardi. Combining positive and negative examples in relevance feedback for content-based image retrieval. *J. Visual Communication and Image Representation*, 14(4):428–457, 2003.

[18] O. Kucuktunc and H. Ferhatosmanoglu. l-diverse nearest neighbors browsing for multidimensional data. *IEEE Transactions on Knowledge and Data Engineering*, 25(3):481–493, 2013.

[19] N. Kumar, N. Lolla, E. Keogh, S. Lonardi, and C. A. Ratanamahatana. Time-series bitmaps: a practical visualization tool for working with large time series databases. In *SIAM 2005 Data Mining Conference*, pages 531–535. SIAM, 2005.

[20] X. Lian and L. Chen. Efficient similarity search over future stream time series. *IEEE Transactions on Knowledge and Data Engineering*, 20(1):40–54, 2008.

[21] B. Liu and H. V. Jagadish. Using trees to depict a forest. *PVLDB*, 2(1):133–144, 2009.

[22] D. Minnen, C. Isbell, I. Essa, and T. Starner. Detecting subdimensional motifs: An efficient algorithm for generalized multivariate pattern discovery. *Data Mining, IEEE International Conference on*, 0:601–606, 2007.

[23] F. Radlinski and S. Dumais. Improving personalized web search using result diversification. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '06, pages 691–692, 2006.

[24] J. Rocchio. *Relevance feedback in information retrieval. In: The SMART Retrieval System Experiments in Automatic Document Processing.*, pages 313–323. Englewood Cliffs, N.J.: Prentice Hall, 1971.

[25] Y. Rui, T. S. Huang, S. Mehrotra, and M. Ortega. A relevance feedback architecture for content-based multimedia information retrieval systems. In *Content-Based Access of Image and Video Libraries, 1997. Proceedings. IEEE Workshop on*, pages 82–89. IEEE, 1997.

[26] G. Salton, editor. *The SMART Retrieval System Experiments in Automatic Document Processing.* Englewood Cliffs: Prentice-Hall, 1971.

[27] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[28] S. Salvador and P. Chan. Toward accurate dynamic time warping in linear time and space. *Intell. Data Anal.*, 11(5):561–580.

[29] I. Selesnick, R. Baraniuk, and N. Kingsbury. The dual-tree complex wavelet transform. *Signal Processing Magazine, IEEE*, 22(6):123 – 151, nov. 2005.

[30] Z. Su, H. Zhang, S. Li, and S. Ma. Relevance feedback in content-based image retrieval: Bayesian framework, feature subspaces, and progressive learning. *Image Processing, IEEE Transactions on*, 12(8):924–937, 2003.

[31] S. Tong and E. Chang. Support vector machine active learning for image retrieval. In *Proceedings of the ninth ACM international conference on Multimedia*, MULTIMEDIA '01, pages 107–118, 2001.

[32] A. K. Tung, R. Zhang, N. Koudas, and B. C. Ooi. Similarity search: a matching based approach. In *Proceedings of the 32nd international conference on Very large data bases*, pages 631–642. VLDB Endowment, 2006.

[33] Z. Xu, R. Akella, and Y. Zhang. Incorporating diversity and density in active learning for relevance feedback. In *Proceedings of the 29th European conference on IR research*, ECIR'07, pages 246–257, 2007.

[34] X. S. Zhou and T. S. Huang. Relevance feedback in image retrieval: A comprehensive review. *Multimedia systems*, 8(6):536–544, 2003.

APPENDIX

The data sets used in the experimental verification are as follows: 50words, CBF, Coffee, Cricket_X, Cricket_Y, Cricket_Z, Diatom Size Reduction, ECG200, ECG FiveDays, FaceAll, FaceFour, FacesUCR, Gun_Point, Italy Power Demand, Lighting2, Lighting7, Medical Images, MoteStrain, OSULeaf, OliveOil, SonyAIBO Robot Surface, SonyAIBO Robot SurfaceII, StarLight Curves, SwedishLeaf, Trace, TwoLeadECG, Words Synonyms, fish, synthetic control. The numbering of the datasets is according to the above listing. Information about the data sets are given in [16].