Proc. 13th Int. Conf. on Database Systems for Advanced Applications (DASFAA'08), New Delhi, India, 2008.

Index-Supported Similarity Search using Multiple Representations

Johannes Aßfalg, Michael Kats, Hans-Peter Kriegel, Peter Kunath, and Alexey Pryakhin

Institute for Computer Science
Ludwig-Maximilians-University of Munich
Oettingenstr. 67, 80538 Munich, Germany
{assfalg,kriegel,kunath,pryakhin}@dbs.ifi.lmu.de

Abstract. Similarity search in complex databases is of utmost interest in a wide range of application domains. Often, complex objects are described by several representations. The combination of these different representations usually contains more information compared to only one representation. In our work, we introduce the use of an index structure in combination with a negotiation-theory-based approach for deriving a suitable subset of representations for a given query object. This most promising subset of representations is determined in an unsupervised way at query time. We experimentally show how this approach significantly increases the efficiency of the query processing step. At the same time the effectiveness, i.e. the quality of the search results, is equal or even higher compared to standard combination methods.

1 Introduction

Similarity search is an important issue in a broad range of applications like the retrieval of multimedia, biological, spatial, and CAD objects. In order to handle complex domain-specific objects, a feature extraction is typically applied. The feature extraction aims at transforming characteristic object properties into feature values. The extracted feature-values can be interpreted as a vector in a multidimensional vector space called feature space. The most important characteristic of a meaningful feature space is that whenever two of the objects are similar, the associated feature vectors have a small distance according to an appropriate distance function (e.g., the Euclidean distance). Thus, similarity search on complex objects can be naturally translated into a k-nearest neighbor (kNN) query in a feature space. Objects are usually described by several feature spaces in order to capture various object properties. Thus, one of the most promising approaches for effective similarity search in databases is to exploit the properties of multiple feature spaces or representations. Though the effectiveness can be improved by using multiple representations, the efficiency of the multi-represented similarity search should also be addressed. For answering a kNN query, we have to consider all available representations. This can be accomplished in two ways. We can either perform a kNN query on all representations independently and combine the results, or we can combine all feature spaces into a single feature space and perform a kNN query on this combined feature space. Each of these two approaches has its drawbacks. The first approach yields potentially different kNN ranking results for the different representations and it is not obvious how to derive a combined answer. The seconde approach suffers from the well-known "curse of dimensionality". In this paper, we propose a novel approach for efficient, multi-represented similarity search where each representation uses its own index structure. In the first step, our approach performs a pre-selection in order to reduce the number of available representations to a small subset S of the most accurate representations. This reduction is based on the coalitional game theory. Furthermore, our approach allows to calculate such a most promising subset S dynamically (i.e., the subset S is computed depending on the given query object). In contrast to existing approaches (e.g., entropy-based methods [1]) which apply an effective but supervised technique for similarity search with multiple representations, we propose an unsupervised approach. Furthermore we outline an algorithm for answering kNN queries using separate index structures for each representations. Instead of using similarity distances, we follow the idea of negotiation game theory and apply self confidence and so-called payoff values in order to rank multi-represented objects.

2 Related Work

Similarity search based on multiple representations has attracted considerable attention in several research communities. However, to the best of our knowledge, no existing technique dynamically calculates a suitable coalition of representations and supports efficient multi-represented kNN query processing in an unsupervised way. The existing approaches can be grouped into two categories: indexing of multi-represented objects and combining several similarity measures corresponding to different representations.

In [2], the M^2-tree is proposed that combines information from multiple metric spaces within a single index structure. The main drawback of the M^2-tree is that it combines features spaces statically, i.e. independently of the current query object. Furthermore, the combination function has to be known beforehand. In contrast to that, the approach of [3, 4] derives a linear combination of metrics dynamically, i.e. based on a given query. However, all available representations are considered, in contrast to our approach that dynamically selects a small subset of relevant representations.

An overview of combining approaches in information retrieval can be found in [5]. According to [5], the most common way to combine representations is the use of the weighted sum of distances in each representation. To find proper weights for each representation, several approaches were proposed that rely on user feedback. Further approaches to approximate weights employing user feedback are described in [6, 7]. In comparison to our approach, these methods employ global weights and do not use dynamic and unsupervised adjusting.

The authors of [8] introduce a technique based on the entropy impurity measure. In comparison to our method, the proposed technique requires a set of labeled objects. An unsupervised technique for the weighted combination of multiple representations was proposed in [9], but this method does not consider efficient data access using index structures. It is furthermore only applicable in combination with summarization which is not a necessary element of general multi-represented similarity search.

3 Adapting Coalitional Game Theory for Similarity Search

3.1 Preliminaries

Comparability of Feature Spaces. Usually, the similarity distance values of different representations do not have a common scale. This problem is called the comparability problem. To overcome this problem, normalization methods are applied. We use the most common Min-Max normalization which calculates the maximum (max) and the minimum (min) of the original distance values. Afterwards, a distance value d is mapped to the normalized distance d_n where $d_n = (d - min)/(max - min)$.

Coalitional Games. We consider the problem of combining similarity information of different representations as a game theory problem, in particular as a so called coalitional game. For a detailed introduction, we refer the reader to [10].

A game is a tuple (N, V), where $N = \{r_1 ... r_n\}$ is the set of players participating in the game. In our approach these players correspond to the available representations, each one trying to suggest its own similarity distance as the best one. V is a function which assigns the so called payoff or gain value to a subset $U \subseteq N$. U is called a coalition.

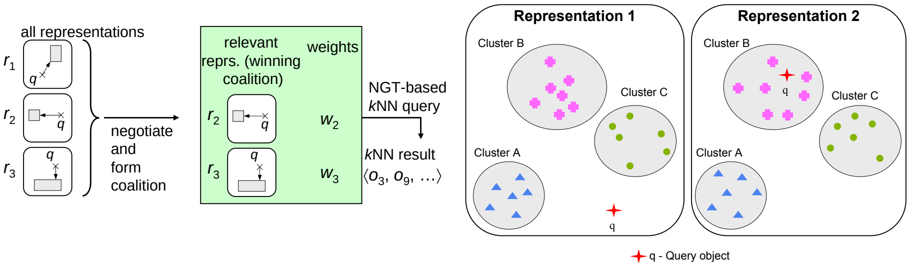
Each player can choose from a predefined set of strategies, each yielding a certain payoff. In our approach each representation can choose between n possible moves: either not to cooperate or to cooperate with one of the $n - 1$ other representations. A representation is more likely to cooperate with another one, if they are similar to each other. A representation is more likely not to cooperate with another representation if either no similar representation is available for a given query or if the representation is very confident of its own similarity distance (cf. Section 3.2).

A game can be described by a $n \times n$ payoff matrix M listing the payoff values for each possible move. The diagonal holds the gain values for the decision not to cooperate while the entry in the i-th row and the j-th column corresponds to the gain for player i to cooperate with player j. Section 3.3 describes how to use M to determine the winning coalition, i.e. to determine a subset of representations that is used to calculate the similarity of database objects for a given query object. The overall idea of our negotiation-game-theory-based approach (NGT) is depicted in Figure 1(a).

3.2 Calculating Gains for Coalitions of Representations

This section describes the calculation of the gain values for the different strategies. Given a query, several influences have to be taken into account.

The first influence is the position of the query object in the different feature spaces. Intuitively, if the feature vector corresponding to the query object is far away from a cluster in a feature space, this might indicate that the current representation considers the query as an outlier. Assuming there are indeed similar objects in the database this means the current representation is less suitable for describing and answering the current query. This idea is illustrated in Figure 1(b), where the first representation might be less useful for the given query q. In order to obtain clustering information we exploit the clustering properties of tree-like index structures. We propose to use an X-tree [11] because it is a common index structure for high-dimensional data and efficiently supports kNN queries. As the directory of a tree-like index structure is a good approximation of the



(a) Overview of our negotiation-based approach for efficient similarity search.

(b) Estimating the suitability of different representations for a given query q based on close-by clusters in the feature spaces.

Fig. 1. Overview and general idea.

underlying data distribution, for representation i we calculate the distance d_i between q and the nearest cluster as the MINDIST [12] between q and a directory node.

The second influence on the negotiation behavior of different representations is their similarity to each other. More similar representations should more willingly cooperate with each other. As we normalized all distances as described above, we are able to use the next-cluster values d_i as described before to compare different representations. Let in the following μ be the mean value and σ be the standard deviation of the distances d_i. We define the gain for representation r_i not to cooperate as

$$M_{i,i} = (1 - d_i^2) \cdot (1 - |d_i - \mu|^2)$$

The first factor of the product reflects the quality of representation r_i for a given query as described above. The second factor of the formula compares the quality of r_i to the average quality of all representations. Thus, a representation has the largest motivation not to cooperate with another representation if its next-cluster distance is small and at the same time similar to the average next-cluster distance.

The remaining matrix entries $M_{i,j}$, $i \neq j$, indicate the payoff for representation r_i for cooperating with representation r_j. A cooperation with r_j can either increase or decrease the gain for r_i, i.e. $M_{i,j} = M_{i,i} \cdot changeFactor$. The change factor takes the following considerations into account: The smaller the difference between d_i and d_j compared to σ, the more likely is a cooperation. If the difference between d_i and d_j is larger than σ, r_i and r_j are not allowed to cooperate. In order to form stable coalitions we have to prevent the case where r_i decides to cooperate with r_j while at the same time r_j decides to cooperate with r_i. Only a one-directional cooperation is allowed. We favor the representation whose next-cluster distance is nearer to the average μ. This leads to the following definition for non-diagonal payoff matrix elements:

$$M_{i,j} = M_{i,i} \cdot changeFactor = M_{i,i} \cdot \begin{cases} -1 \ \text{ if } |d_i - \mu| < |d_j - \mu| \vee |d_i - d_j| > \sigma \\ (1 + (\sigma - |d_i - d_j|)) * \frac{1+|d_i-\mu|}{1+|d_j-\mu|} \ \text{ else} \end{cases}$$

3.3 Determining the Winning Coalition

After having calculated all payoff values we are now able to determine the winning coalition of representations, i.e. the coalition with the highest gain. The gain of a coalition is the sum of the gain values of the decisions that have led to the coalition. In order to identify the strongest coalition we determine the maximal entry of each row, as this entry indicates the best strategy of the associated representation. Then, for a certain column, we sum up all these maximal values that can be found in this column. This yields the payoff for the coalition consisting of the representations whose best gain values have been summed up.

3.4 Deriving Weights

Aside from limiting the number of representations involved in answering a query, we can use the above generated matrix to calculate weights for the remaining representations. We use these weights to perform high-quality kNN queries very efficiently as described in Section 4. Let C be the coalition. The weight w_i of the coalition member r_i is calculated as the ratio between $M_{i,i}$ and the average non-cooperating gains of all participating representations: $w_i = (M_{i,i} \cdot |C|)/(\sum_{r_j \in C} M_{j,j})$. Intuitively, $M_{i,i}$ reflects how confident representation r_i is about its own quality. So, weights larger than 1 indicate a confidence above the average.

4 Efficient kNN Query Processing on Multiple Representations

In the following, we use the above described weights to calculate a weighted linear combination of these representations. As in the previous section, we are using the principles of the negotiation game theory for combining the representations. In our experiments, we observed that using the weighted sum yields the best results for our NGT-based approach.

A ranking is performed on each of the relevant representations in order to answer a kNN query for a given query object. The following technique bases on the assumption of searching in a tree-like index structure. The kNN query algorithm uses the well-known Hjaltason-Samet ranking algorithm [13] and orders all objects in a single priority queue. The priority queue of the ranking is initialized with the roots of all representations of the best coalition. The ranking priority queue is organized in descending order w.r.t. gain-based value as described below. We propose to calculate the priority of the ranked directory pages dp similar to the representation selection in the previous section. The gain formula has to be adapted as follows: $priority = gain_{dp} * w_i$, where $gain_{dp} = (1 - MinDist(dp)^2)$. In each iteration of the algorithm, the first object is removed from the ranking queue. For each entry of a directory page, we calculate the priority according to the above formula and insert it into the ranking priority queue. In case of a data page, we process all objects within this page. Each retrieved object is added to the result priority queue, where the sum of the weighted gains in all relevant representations is used as priority value. In case an object is already in the queue, its priority value is updated. The gain is calculated as the gain for a directory page, instead of the MINDIST value we use the Euclidean distance value. The idea behind using the sum of the gain values as priority is that objects which have been retrieved in a lot of representations are ranked higher. Furthermore, we also test if the currently retrieved or updated object fulfills a stop condition. We terminate the kNN algorithm if there is no change in the first k objects of our result queue. Once this is the case, it is most likely that no further gain value has enough impact to alter the kNN result queue.

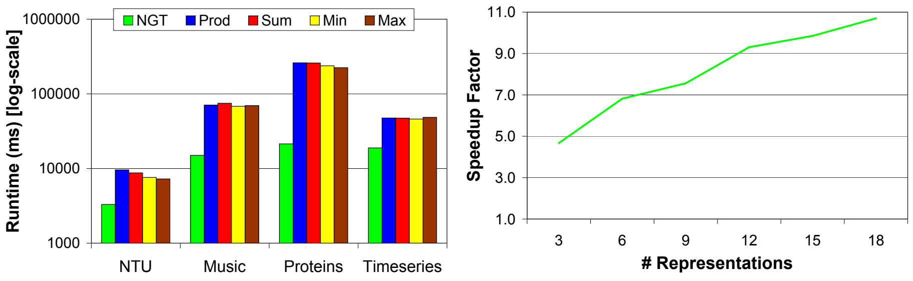
5 Experimental Evaluation

We performed our evaluation on four different datasets. Please note that each dataset is described by three or more representations because our NGT approach is only applicable if at least three representations are available. As mentioned in Section 3, all datasets are organized in an X-tree. Unless noted otherwise, we conducted 30 kNN queries with randomly chosen query objects and averaged the results.

The **NTU** dataset is based on a subset of the NTU 3D Model Benchmark [14] and consists of 549 objects in 46 classes. We extracted three representations for this dataset with an average dimensionality of 80. The **Music** dataset contains 516 songs taken from 15 different music genres. We generated 6 different feature representations per song with an average dimensionality of 500. The **Proteins** dataset consists of 2465 objects taken from the SWISS-PROT [15] protein database. We derived 18 feature representations with an average dimensionality of 20. The last dataset is the synthetic CBF **Timeseries** dataset [16] for which we calculated 9 different feature representations with an average dimensionality of 20.

5.1 Efficiency Evaluation

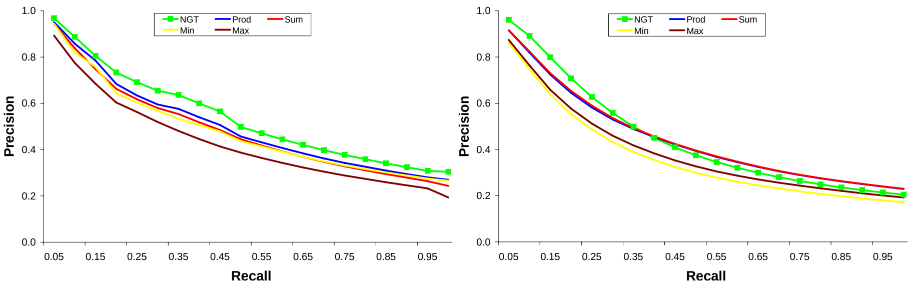
At first, we turned our attention to the efficiency of our proposed approach. We compared our approach with the standard combination rules SUM, PROD, MIN, and MAX.



(a) Runtime of NGT approach vs. standard combination rules.

(b) Speedup factor of NGT vs. Prod.

Fig. 2. Efficiency evaluation.



(a) NTU dataset

(b) Proteins dataset

Fig. 3. Quality of NGT approach vs. standard combination rules.

Figure 2(a) depicts the runtime results of a 5NN query for all four datasets. Because our approach executes the kNN query only for a subset of the available representations, it yields a significant speedup on all datasets. Even when only three different representations are available (*NTU*), our NGT approach achieves a speedup of about factor 3. For the *Proteins* dataset, we executed another experiment in order to evaluate the relationship between the runtime and the available representations (see Figure 2(b)). We started with three representations and added another three representations in every step. As already demonstrated in the first evaluation, our NGT approach shows an advantage in runtime even for three representations. This advantage increases when more representations are available up to a factor of 10.7 for 18 representations. We observed the NGT approach does not favor a particular subset of representations. Instead, the best coalition is always chosen dynamically from all available representations.

5.2 Effectiveness Evaluation

In this section, we demonstrate the effectiveness of our NGT approach. Figure 3 depicts the quality of our NGT approach and the four standard combiners for the NTU and the Proteins dataset. This figure displays precision-recall plots, which were obtained by performing kNN queries where $k = |dataset|$. For all datasets, the effectiveness of the NGT approach is either comparable or even better than the traditional combination rules. On the *NTU* dataset, our NGT technique is able to outperform all other combiners for all recall values. For the other datasets, the result quality of the NGT method is comparable to the standard combination rules.

6 Conclusions and Outlook

In this paper, we presented a novel approach that is able to efficiently process similarity queries using multi-represented objects. The main idea of our approach is to select a small subset of representations and to perform a subsequent similarity search using only this small subset. In order to determine a suitable subset of representations, a novel method is proposed that is based on coalitional negotiation game theory. All available representations start a negotiation process for a given query in order to form coalitions. The goal of each representation is to contribute its own similarity measure in the coalition. The negotiation is based on the assumed usefulness of a certain representation for the given query. The negotiation step selects only a suitable subset of representations that is used to answer the query. Furthermore we introduced an efficient kNN query algorithm which operates on the selected representations. We demonstrated the efficiency of the proposed approach on four datasets. Furthermore, our novel approach yields a comparable or even higher effectiveness on all considered datasets. As future work we plan to turn our attention to further aspects of the game theory. For example, it seems promising to use a mix of coalitional and behavioristic negotiation theory to determine suitable coalitions.

References

1. Bustos, B., Keim, D., Saupe, D., Schreck, T., Vranic, D.: Automatic selection and combination of descriptors for effective 3d similarity search. In: Proc. ICME. (2004)
2. Ciaccia, P., Patella, M.: The M^2-tree: Processing complex multi-feature queries with just one index. In: DELOS Workshop: Information Seeking, Searching and Querying in Digital Libraries. (2000)
3. Bustos, B., Keim, D., Schreck, T.: A pivot-based index structure for combination of feature vectors. In: ACM Symposium on Applied Computing. (2005)
4. Bustos, B., Skopal, T.: Dynamic similarity search in multi-metric spaces. In: Proc. MIR. (2006)
5. Croft, W.B.: Advances in Information Retrieval: Recent Research from the CIIR. Kluwer Academic Publishers (2000)
6. Chua, T.S., Low, W.C., Chu, C.X.: Relevance feedback techniques for color-based image retrieval. In: Proc. MMM. (1998)
7. Rui, Y., Huang, T.S., Mehrotra, S.: Content-based image retrieval with relevance feedback in mars. In: Proc. ICIP. (1997)
8. Bustos, B., Keim, D.A., Saupe, D., Schreck, T., Vranic, D.V.: Using entropy impurity for improved 3d object similarity search. In: Proc. ICME. (2004)
9. Kriegel, H.P., Kröger, P., Kunath, P., Pryakhin, A.: Effective similarity search in multimedia databases representations. In: Proc. MMM. (2006)
10. von Neumann, J., Morgenstern, O.: Theory of games and economic behavior (2004)
11. Berchtold, S., Keim, D.A., Kriegel, H.P.: The X-Tree: An index structure for high-dimensional data. In: Proc. VLDB. (1996)
12. Roussopoulos, N., S.Kelley, Vincent, F.: Nearest neighbor queries. In: Proc. SIGMOD. (1995)
13. Hjaltason, G., Samet, H.: (Incremental similarity search in multimedia databases)
14. Chen, D.Y., Tian, X.P., Shen, Y.T., Ouhyoung, M.: On visual similarity based 3d model retrieval. EUROGRAPHICS (2003)
15. Boeckmann, B., Bairoch, A., Apweiler, R., Blatter, M.C., Estreicher, A., Gasteiger, E., Martin, M., Michoud, K., O'Donovan, C., Phan, I., Pilbout, S., Schneider, M.: ”The SWISS-PROT Protein Knowledgebase and its Supplement TrEMBL in 2003”. Nucleic Acid Research (2003)
16. Saito, N.: Local feature extraction and its application using a library of bases. PhD thesis, Yale University, New Haven, Connecticut (1994)