Correct Algorithms for the Comparison of Complex Patterns

Ilaria Bartolini, Paolo Ciaccia, and Marco Patella

DEIS - IEIIT-BO/CNR, University of Bologna, Italy
{ibartolini,pciaccia,mpatella}@deis.unibo.it

Abstract. The comparison of complex patterns, i.e. patterns that are obtained by recursively aggregating other patterns, poses serious challenges because of the different ways the component patterns can be matched. In this paper, we investigate the problem of comparing set of patterns when constraints are imposed on the matching between component patterns. This is motivated by a real world example, namely the retrieval of images in a region-based image retrieval system, where images represent complex patterns that are composed of regions, i.e. base patterns. We present both sequential and index-based exact algorithms for solving the problem, and experimentally evaluate them on a medium-size data set.

1 Introduction

The massive quantity of data produced every day by both industrial and scientific applications poses new challenging requirements to DBMS systems. Such huge amount of data is unlikely to be useful for end users, and automated processing techniques (such as data mining, pattern recognition, and knowledge extraction techniques) are needed in order to reduce such raw data to a compact, manageable, set of knowledge artifacts (e.g. clusters, association rules, time series). A compact and rich in semantics representation of raw data is called a *pattern* [13, 4]. The problem of storing and querying patterns in an effective and efficient way is the focus of so-called *Pattern-Base Management Systems* (PBMSs). Among the several issues that a PBMS has to address (modeling, storage, and retrieval of patterns), we believe that one of the most important operations that should be supported is that of comparison. The comparison between two patterns entails the computation of a score s, $s \in [0, 1]$, assessing their mutual similarity. Given the definition of similarity between patterns, the user may be interested in finding the patterns which are most similar to a given (query) one. More precisely, given a query pattern q and an integer value k, a *best matches query* returns the k patterns having the highest similarity score with respect to q, according to the similarity measurement implemented.

The comparison between *complex* patterns, i.e. patterns obtained by assembling other patterns to obtain a *part-of* hierarchy (for example, a *clustering* pattern is obtained as the composition of *cluster* patterns) is particularly challenging. The similarity score s between two complex patterns is computed starting from the similarity between component patterns, then scores obtained for each sub-pattern are aggregated, using an aggregation logic, to determine the overall similarity of the two patterns [9]. The aggregation logic may be very simple, just an expression combining numerical values, or a more complex one, if constraints and/or transformation costs are to be considered: For example, a suitable "matching" between components patterns might be needed. The aggregation logic can also involve a variety of transformations, each with an associated cost, and the overall similarity score is obtained as the maximum score obtained by applying all the possible transformations to the component patterns.

In this paper, we provide correct sequential and index-based algorithms to solve best matches queries for complex patterns consisting of a set of base patterns when constraints on the matching between component patterns exist. In Section 2 we introduce a (simplified) framework for modeling patterns. Then, Section 3 provides a motivating example, drawn from the world of image retrieval, for the problem of best matches queries. The problem is precisely formalized in Section 4 and both sequential and index-based correct algorithms for its solution are provided in Section 5. Section 6 shows some preliminary results obtained over a real data set and Section 7 concludes the paper.

2 A Model for Patterns

The following framework for modeling patterns is a simplified version of the model proposed in [13, 4], that we adapted to our needs.

A *pattern type* pt specifies the intensional form of patterns, and is represented by a quintuple $pt = (n, ss, ds, ms, f)$, where:

- n is the name of the pattern type.
- The *structure schema* ss describes the structure of the pattern instances of pt, thus defining the space in which patterns can be defined.
- The *source schema* ds defines the schema of the data set from which instances of pt are constructed.
- The *measure schema* ms describes the measures that quantify the quality of the representation of source data achieved by the pattern.
- The *formula* f describes the relationship between source space and pattern space and, thus, carries the semantics of the pattern.

Example 1. The pattern type "2-D cluster" can be defined as follows:

$$
\begin{aligned}
n &: \mathsf{2DCluster} \\
ss &: \mathsf{RECORD}(radius\text{: REAL, } center\text{: RECORD}(cx\text{: REAL, } cy\text{: REAL})) \\
ds &: \mathsf{SET}(point\text{: RECORD}(x\text{: REAL, } y\text{: REAL})) \\
ms &: numberOfPoints\text{: INTEGER} \\
f &: (x - cx)^2 + (y - cy)^2 \leq radius^2
\end{aligned}
$$

□

A pattern p, instance of a pattern type pt, is defined as a quintuple $p = (pid, s, d, m, e)$, where:

- pid (pattern identifier) is an unique identifier for p.
- The *structure* s is a value of the structure schema ss.
- The *data set* d conforms to type ds.
- The *measure* m is a value of type ms.
- The *expression* e is obtained by opportunely instantiating the formula f.

Example 2. A cluster of type 2DCluster (see Example 1) can be represented as follows:

$$
\begin{aligned}
pid &: 324 \\
s &: \mathsf{RECORD}(radius\text{: } 2.3, \; center\text{: RECORD}(cx\text{: } 35, \; cy\text{: } 1570)) \\
d &: \text{`SELECT E.age, E.salary} \\
&\quad \text{FROM Employee as E'} \\
m &: numberOfPoints\text{: } 184 \\
e &: (\mathsf{E.age} - 35)^2 + (\mathsf{E.salary} - 1570)^2 \leq 2.3^2
\end{aligned}
$$

□

Note that composition of patterns can be obtained by inserting pattern types in the structure schema of a pattern type pt.

Example 3. A clustering is just a set of clusters (see Example 1) having a measure expressing the *validity* of the obtained clusters:

$$
\begin{aligned}
n &: \mathsf{2DClustering} \\
ss &: clusters\text{: SET(2DCluster)} \\
ds &: \mathsf{SET}(point\text{: RECORD}(x\text{: REAL, } y\text{: REAL})) \\
ms &: validity\text{: INTEGER} \\
f &:
\end{aligned}
$$

□

The similarity between two simple patterns of the same pattern type is computed as a function of the similarity between both the structure and the measure components:

$$sim(p_1, p_2) = f(sim_{struct}(p_1.s, p_2.s), sim_{meas}(p_1.m, p_2.m))$$

where with $p.s$ and $p.m$ we indicate the structure and the measure for pattern p, respectively. If the two patterns have the same structural component, then $sim_{struct}(p_1.s, p_2.s) = 1$, and the measure of similarity naturally corresponds to a comparison of the patterns' measures, e.g. by aggregating differences between each measure [9]. In the general case, however, the patterns to be compared have different structural components, thus a preliminary step is needed to reconcile the two structures to make them comparable. Computing the similarity between complex patterns, i.e. instances of a complex pattern type, in the general case is a two step process:

Matching: Component patterns of a pattern are associated to component patterns of the reference (query) pattern, i.e. by only considering "best" couplings (matches).

Combining: The overall similarity between the two patterns is computed by combining similarity scores corresponding to matched component patterns.

3 Motivating Example

The goal of content-based image retrieval (CBIR) systems is to define a set of properties (*features*) able to effectively characterize the content of images and then to use such features during retrieval in order to provide effective and efficient access to image databases based on content. To increase the effectiveness of image retrieval, in recent times a number of *region-based* image retrieval systems has been presented [5, 12, 1, 14], which "fragment" each image into regions, i.e. sets of pixels sharing common visual characteristics, like color and texture. Similarity between images is then assessed by computing similarity between pairs of regions and combining the results at the image level.

Conceptually, each image is represented as a set of component regions. By considering the model of Section 2, we can represent each region as a simple pattern and the overall image as a set of region patterns. This way, the problem of finding the images that most resemble a given query one can be modeled as a best matches query over the space of image patterns. In particular, the process of similarity assessment between images perfectly fits the matching/combining paradigm introduced in Section 2 (see Figure 1).

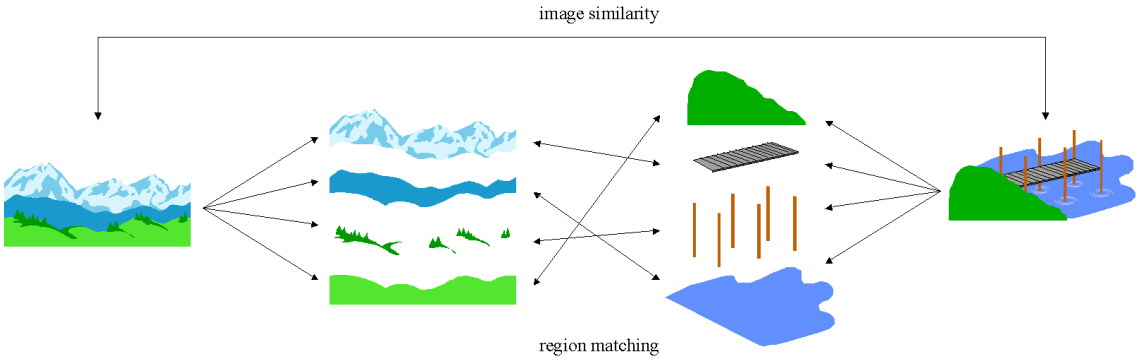


Fig. 1. In region-based systems, similarity between images is assessed by taking into account similarity between matched regions.

Region matching algorithms have only recently emerged as a need for region-based CBIR systems. Existing systems [5, 12, 14], however, use naïve heuristic matching algorithms when associating regions of the images being compared, thus obtaining incorrect results.[1]

The criterion used to assess the similarity between two regions vary from system to system. For example, the WINDSURF system [1] segments images into sets of pixels that are homogeneous for color and texture by using the Discrete Wavelet Transform (DWT, [7]) and a fuzzy c-means algorithm. Each region is then represented as an elliptical cluster in the HSV space and the similarity between regions is computed by taking into account both differences in the color and textures descriptors (the pattern structure) by way of the Bhattacharyya distance and in their relative size (the pattern measure). For more details, see [1].

4 The Problem of Optimal Matching

Given a reference (query) complex pattern cp_q, composed of a set of patterns $\{p_{q_1}, \ldots, p_{q_n}\}$, and a pattern p_s, also composed of a set of patterns $\{p_{s_1}, \ldots, p_{s_m}\}$, the problem of *optimal matching* consists in associating (matching) each pattern p_{q_i} of cp_q to a pattern $p_{s_j} = \Gamma_s(p_{q_i})$ of cp_s (possibly, no pattern is associated to p_{q_i}, i.e. $\Gamma_s(p_{q_i}) = \emptyset$) such that the overall similarity score between patterns cp_q and cp_s, $sim(cp_q, cp_s)$, is maximized. Similarity between base patterns is assessed by way of the $sim_{base}(p_{q_i}, p_{s_j})$ function. Every $\Gamma_s()$ has to satisfy the following constraint: Two patterns of cp_q cannot be associated to the same pattern of cp_s, therefore if $p_{q_i} \neq p_{q_j}$ and $\Gamma(p_{q_i}) = \Gamma(p_{q_j})$, it is $\Gamma(p_{q_i}) = \Gamma(p_{q_j}) = \emptyset$. Similarity between complex patterns is computed by taking into account similarity between associated base patterns using a monotonic function PM_{sim}, i.e. $sim(cp_q, cp_s) = PM_{sim}(sim_{base}(p_{q_1}, \Gamma_s(p_{q_1})), \ldots, sim_{base}(p_{q_n}, \Gamma_s(p_{q_n})))$. The only requirement for the function PM_{sim} is that it has to be a monotonic increasing function, that is if $s_i \leq s'_i, i \in \{1, n\}$, then it is $PM_{sim}(s_1, \ldots, s_i, \ldots, s_n) \leq PM_{sim}(s_1, \ldots, s'_i, \ldots, s_n)$. This is intuitive, since better matches between base patterns can only increase the overall similarity score between corresponding complex patterns. Moreover, for the sake of simplicity, in the following we will assume that PM_{sim} is a commutative function. The optimal matching between regions, i.e. that for which $sim(cp_q, cp_s)$ is maximum, will be denoted as Γ_s^{opt}.

$$sim(cp_q, cp_s) = \max\ PM_{sim}(s_{i_1 j_1}, \ldots, s_{i_{|\mathcal{H}|} j_{|\mathcal{H}|}}), \\ (i_h j_h), (i_l j_l) \in \mathcal{H}, (i_h j_h) \neq (i_l j_l) \quad (1)$$

$$\mathcal{H} = \{(i,j) | x_{ij} = 1\} \quad (2)$$

$$\sum_{j=1}^{m} x_{ij} \leq 1\ (i = 1, \ldots, n), \quad (3)$$

$$\sum_{i=1}^{n} x_{ij} \leq 1\ (j = 1, \ldots, m), \quad (4)$$

$$x_{ij} \in \{0, 1\}\ (i = 1, \ldots, n)(j = 1, \ldots, m) \quad (5)$$

Equation 1 means that to determine the overall score $sim(cp_q, cp_s)$ we have to consider only the matches $\Gamma_s()$ in $\mathcal{H}$ (Equation 2). Equation 3 (Equation 4) expresses the constraint that at most one pattern p_{s_j} of cp_s (resp. p_{q_i} of cp_q) can be assigned to a pattern p_{q_i} of cp_q (resp. p_{s_j} of cp_s).

Definition 1 (Correct matching). *A set of x_{ij} values that satisfies the constraints expressed by Equations 3, 4, and 5 is called a* correct *matching.*

Definition 2 (Complete matching). *A correct matching for which it is $\sum_{j=1}^{m} x_{ij} = 1, (i = 1, \ldots, n)$ (i.e. each pattern of cp_q is associated to a pattern of cp_s) is called a* complete *matching.*

[1] As an example, suppose that an user asks for an image containing two tigers: If a database image contains a single tiger, it is *not* correct to associate both query regions to the single "tiger" region of the DB image, since, in this case, information about the number of query regions is lost.

It should be noted that *any* correct matching for a pattern cp_s having a number of patterns lower than that of cp_q is obviously not complete.

Definition 3 (Optimal matching). *The correct matching that maximizes the function expressed by Equation 1 is called the* optimal *(or* exact*) matching, and will be denoted as* $\Gamma_s^{opt}()$.

5 Solving the Problem

A typical form of the scoring function PM_{sim} is that of a sum (this is indeed the case, save for a constant scale factor, for the image retrieval systems WALRUS [12] and WINDSURF [1]), leading to a re-formulation of Equation 1 as follows:

$$sim(cp_q, cp_s) = \max \sum_{i=1}^{n} \sum_{j=1}^{m} s_{ij} \cdot x_{ij} \tag{6}$$

The generalized assignment problem, in this case, takes the form of the well known Assignment Problem (AP), one of the most popular topics in combinatorial optimization. To resolve it, we can apply the Hungarian Algorithm [11] to the matrix $\{s_{ij}\}$ of similarity scores between regions. Sequential evaluation of a best matches query is performed by way of a simple algorithm that computes the optimal matching between the query pattern and all the searched patterns and returns the k patterns for which the highest similarity score is obtained [3]. Of course, the sequential algorithm requires to compute the similarity between the base patterns of the query and all the indexed base patterns. Moreover, the matching problem has to be solved for all the searched complex patterns.

In order to obtain a complexity sub-linear in the data set size, we describe an index-based algorithm that speeds up the evaluation of best matches queries by reducing the number of *candidate patterns*, i.e. patterns on which the optimal region matching problem has to be solved.

In order to use an index to speed-up the search, we suppose that the similarity between base patterns is computed by way of a distance between pattern features (this is the case, for example, for most of the region-based CBIR systems, see Section 3). In this case, a distance-based access method (DBAM), like the M-tree [6], can be used to index base patterns according to their respective (dis-)similarity. Such index structures are able to efficiently answer k nearest neighbor queries, as well as to perform a *sorted access* to the data, i.e. to output objects one by one in increasing order of distance with respect to a query [10].

To retrieve best matches for query patterns, we run a sorted access to the indexed patterns for each base pattern in the query. The $\mathcal{A}_0^{WS}$ algorithm shown in Figure 2 is able to return the correct result for a best matches query by only solving the matching problem for the *candidate set*, i.e. for those patterns having at least one base pattern that has been returned by a sorted access [3, 2]. The *random access* phase consists in computing those similarity scores s_{ij} between query patterns and patterns of candidates not returned in the X^i result sets.

Correctness of $\mathcal{A}_0^{WS}$ (the proof can be found in [3]) is independent of the specific PM_{sim} function used to combine scores into similarity between patterns, since it only relies on the monotonicity of PM_{sim}.

It can be noted that sorted and random access phases of $\mathcal{A}_0^{WS}$ somewhat resemble those of Fagin's $\mathcal{A}_0$ algorithm [8], the major difference being that $\mathcal{A}_0$ does not deal with the issue of correct matching, thus, if applied, it could report non-correct results.

6 Experimental results

Preliminary experimentation of proposed techniques has been performed on the WINDSURF system, using a sample medium-size data set consisting of about 2000 real-life images from the *IMSI-PHOTOS* CD-ROM.[2] The over 8000 obtained regions were indexed using an M-tree [6]. The query

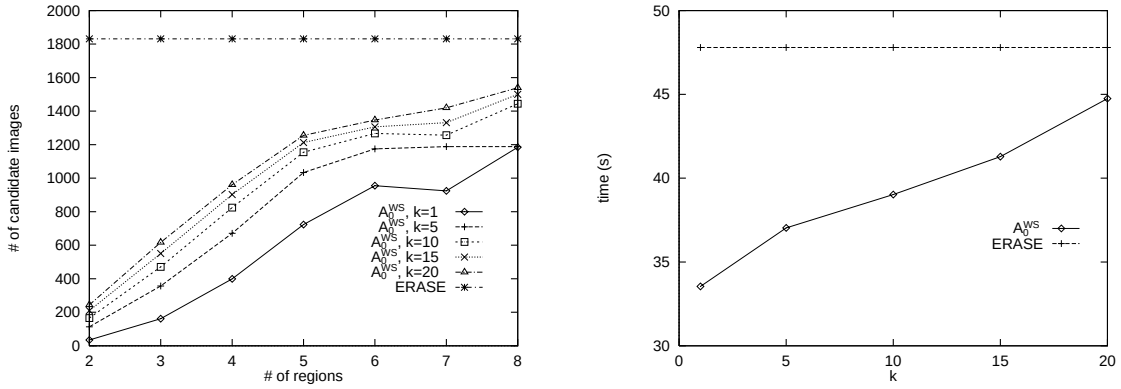
[2] IMSI MasterPhotos 50,000: `http://www.imsisoft.com`.

```
A_0^WS(cp_q: query, k: integer, T: DBAM)
{ ∀ base pattern p_{q_i} of cp_q, open a sorted access index scan on T
    and insert results in the set X^i;
  stop the sorted accesses when there are at least k patterns for which
    a complete assignment exists, considering only base patterns in ∪_i X^i;
  ∀ pattern cp_s having base patterns in ∪_i X^i,
    ∀ pair p_{q_i}, p_{s_j}
      if p_{s_j} ∉ X^i compute score s_ij = sim_base(p_{q_i}, p_{s_j}); (random access)
    compute the optimal assignment; (combining phase)
  return the k patterns having the highest overall scores sim(cp_q, cp_s); }
```

Fig. 2. The $\mathcal{A}_0^{WS}$ algorithm.

workload consists in about one hundred randomly chosen images not included in the data set. All experiments were performed on a Pentium II 450 MHz PC with 64MB of main memory running Windows NT 4.0.

The experiments we present concern the efficiency of the $\mathcal{A}_0^{WS}$ index-based algorithm as compared to that of the sequential algorithm. In Figure 3 (a) we compare the average number of candidate images, i.e. the images on which the Hungarian algorithm has to be applied, as a function of the number of query regions, for different values of k. Of course, the sequential algorithm (the horizontal line labeled ERASE, for Exact Region Assignment SEquential algorithm [3]) would lead to a number of candidate images equal to the number of images in the data set, whereas for $\mathcal{A}_0^{WS}$ this number depends both on k and on the number of query regions. As the graph shows, $\mathcal{A}_0^{WS}$ does well in reducing the number of candidate images. Clearly, its performance degrades as the number n of query regions increases, since the complexity of finding k objects in the intersection of n sets augments with n. This is also confirmed by Figure 3 (b), where query response times are shown for the case $n = 3$.



(a) (b)

Fig. 3. Average number of candidate images vs. number of query regions (a), and response time vs. k ($n = 3$) (b).

7 Conclusions

In this work we have investigated the problem of correct resolution of best matches queries for complex patterns, obtained as sets of base patterns. In particular, an index-based algorithm ($\mathcal{A}_0^{WS}$) has been presented which computes the optimal matching between the base patterns, in order to

maximize the overall similarity score between complex patterns, under the condition that only one-to-one matches exist. Preliminary experiments conducted over a region-based image retrieval system have shown that our approach is indeed very effective with respect to alternative retrieval strategies. In the future we plan to investigate how to solve the problem when different kind of constraints or aggregation logics exist, and also to devise algorithms for dealing with multiple levels of aggregation (i.e. when the composition hierarchy has more than just one level, as was the case considered in this work).

References

1. Stefania Ardizzoni, Ilaria Bartolini, and Marco Patella. Windsurf: Region-based image retrieval using wavelets. In *Proceedings of the 1st International Workshop on Similarity Search (IWOSS'99)*, pages 167–173, Florence, Italy, September 1999. IEEE Computer Society.
2. Ilaria Bartolini, Paolo Ciaccia, and Marco Patella. A sound algorithm for region-based image retrieval using an index. In *Proceedings of the 4th International Workshop on Query Processing and Multimedia Issues in Distributed Systems (QPMIDS 2000)*, pages 930–934, London/Greenwich, UK, September 2000.
3. Ilaria Bartolini and Marco Patella. Correct and efficient evaluation of region-based image search. In *Atti dell'Ottavo Convegno Nazionale SEBD*, pages 289–302, L'Aquila, Italy, June 2000.
4. Elisa Bertino, Barbara Catania, Matteo Golfarelli, Stefano Rizzi, Manolis Terrovitis, Panos Vassiliadis, and Michalis Vazirgiannis. A preliminary proposal for the panda logical model. Technical Report PANDA-UNIMI-2003-001, The PANDA Consortium, February 2003.
5. Chad Carson, Megan Thomas, Serge Belongie, Joseph M. Hellerstein, and Jitendra Malik. Blobworld: A system for region-based image indexing and retrieval. In *Proceedings of the 3rd International Conference on Visual Information Systems VISUAL'99*, pages 509–516, Amsterdam, The Netherlands, June 1999. `http://elib.cs.berkeley.edu/photos/blobworld/`.
6. Paolo Ciaccia, Marco Patella, and Pavel Zezula. M-tree: An efficient access method for similarity search in metric spaces. In *Proceedings of the 23rd International Conference on Very Large Data Bases (VLDB'97)*, pages 426–435, Athens, Greece, August 1997. Morgan Kaufmann.
7. Ingrid Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1992.
8. Ronald Fagin. Combining fuzzy information from multiple systems. In *Proceedings of the 15th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'96)*, pages 216–226, Montreal, Canada, June 1996. ACM Press.
9. Venkatesh Ganti, Johannes Gehrke, Raghu Ramakrishnan, and Wei-Yin Loh. A framework for measuring changes in data characteristics. In *Proceedings of the 18th ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems (PODS'99)*, pages 126–137, Philadelphia, PA, May 1999. ACM Press.
10. Gísli R. Hjaltason and Hanan Samet. Distance browsing in spatial databases. *ACM Transactions on Database Systems*, 24(2):265–318, June 1999.
11. Harold W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistic Quarterly*, 2:83–97, 1955.
12. Apostol Natsev, Rajeev Rastogi, and Kyuseok Shim. WALRUS: A similarity retrieval algorithm for image databases. In *Proceedings 1999 ACM SIGMOD International Conference on Management of Data*, pages 396–405, Philadelphia, PA, June 1999. ACM Press.
13. Stefano Rizzi, Barbara Catania, Matteo Golfarelli, Maria Halkidi, Manolis Terrovitis, Panos Vassiliadis, Michalis Vazirgiannis, and Euripides Vrachnos. Towards a logical model for patterns. Submitted, 2003.
14. James Ze Wang, Jia Li, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive Integrated Matching for Picture LIbraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, September 2001. `http://wang.ist.psu.edu/cgi-bin/zwang/regionsearch_show.cgi`.