
Genetic Approximate Matching of Attributed Relational Graphs

Thomas Bärecke
Marcin Detyniecki

Laboratoire d'Informatique de Paris 6, 104 avenue du Président Kennedy, 75016 Paris, France

THOMAS.BAERECKE@LIP6.FR
MARCIN.DETYNIECKI@LIP6.FR

Stefano Berretti
Alberto Del Bimbo

Dipartimento di Sistemi e Informatica, University of Firenze, via S. Marta 3, 50139 Firenze, Italy

BERRETTI@DSI.UNIFI.IT
DELBIMBO@DSI.UNIFI.IT

1. Introduction

Image segmentation algorithms identify meaningful spatial entities for content-based image retrieval. One or several visual features are extracted for each entity. Based on the feature vectors of the spatial entities and their mutual relationships, attributed relational graphs (ARG) can effectively model entire images. The image retrieval process in an ARG context requires efficient methods to compare these graph models. Each comparison involves the resolution of a general inexact (sub-)graph matching problem. Inexact graph matching already has a long tradition in the domain of pattern recognition (Conte et al., 2004). The variety of existing methods can be classified as either exhaustive state-space search approaches, which guarantee the optimal solution, or approximate methods. The last ones trade the global optimality in for a complexity reduction by accepting sub-optimal solutions. They are usually based on the optimization of an objective function. Although our primary research interest lies in content-based image retrieval, this paper focuses on the general comparison of the two classes. We compare a state-of-the-art exhaustive tree search algorithm (Berretti et al., 2001) and a prototype based on a genetic approach. Both methods are universally applicable, i.e. they do not impose neither any constraints nor any preprocessing steps on the graphs. In the following, we summarize both approaches, before presenting the experimental results and drawing conclusions.

2. Exhaustive tree search algorithm

The search is performed incrementally by repeatedly growing a partial assignment of the vertices of one (sub-)graph S to the vertices of the other graph G . The solution space is organized as a tree, where the k^{th} level contains all partial assignments of the first k enti-

ties of S . The A^* algorithm (Ullmann, 1976) performs a depth-first search which always extends the partial interpretation towards the local optimum, and which backtracks when the scored distance of the current assignment runs over a maximum acceptable threshold. When the inspection reaches a complete interpretation, a match under the threshold is found. At this point, the global optimum is not guaranteed, but the obtained scored distance implies a stricter threshold for acceptable distance that is then used to efficiently extend the search until the global optimum is found. Berretti et al's algorithm extends the A^* scheme using an admissible heuristic to increase the cost of the current partial assignment with a lower boundary of the future cost that will be spent in the extension to a complete match. The extension cost estimate is obtained using the solution of a bipartite matching problem, which is solvable in polynomial time. Based on this value, partial assignments that can not lead to a final match with acceptable similarity can be discarded earlier, thus accelerating the convergence of the algorithm.

3. Genetic Algorithm approach

Genetic algorithms (GAs) are nonlinear optimization methods inspired from the natural evolution process. They operate on a set of candidate solutions and achieve gradual improvements by iteratively selecting fitter individuals and arbitrarily altering and recombining the existing ones. GAs can explore large heterogeneous search spaces efficiently but non-exhaustively. Thus, finding the globally optimal solution is not guaranteed. The implementation of a GA requires the choice of numerous parameters. Among them of crucial importance are the coding of the individuals, the fitness function, appropriate operators and the selection strategy. We applied permutation encoding and

used the sum of the distances of node and edge attributes as error measure. We implemented tournament selection, which can be directly based on an error measure instead of a fitness function. A standard swap operator is used for mutation. Our cross-over operator is similar to the position-based crossover proposed by Syswerda (1991) for the schedule optimization problem. However, we do not impose the order from the second parent onto the child but try to place as many alleles as possible on the same locus as in either of the parents.

4. Experimental results

We compare the algorithms using a set of randomly attributed complete graphs. A subgraph S of size 10 is extracted from each of them based on a random permutation of the original nodes. We perform a matching of every original graph G against each of the subgraphs. Therefore, the best scoring assignment does not necessarily correspond, for any pair, to an actual match. We measure the execution time of both algorithms and the error of the GA. The score computed by the exhaustive approach serves as benchmark for accuracy of the genetic approach. Fig. 1 shows our preliminary accuracy results for two different graph sizes.

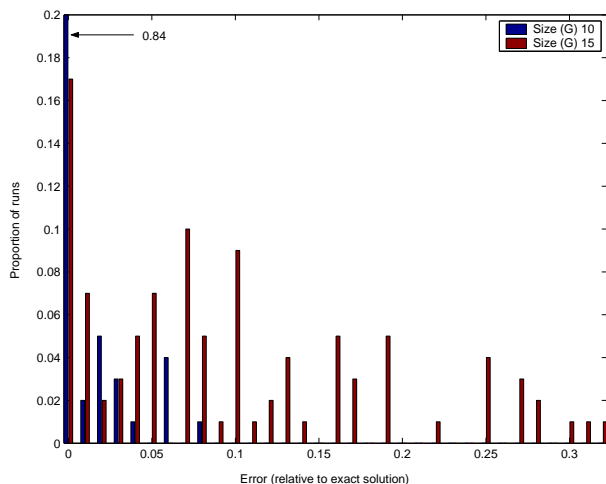


Figure 1. Results. Each bar pair corresponds to a one percent interval (e.g. $[0\%;1\%)$) of relative distance to the globally optimal solution. The bar length indicates the number of runs falling into the interval.

We notice that the GA is always able to find an exact match given that one exists. If not, solutions of the GA are close to the global optimum, even if their relative error increases with the graph size. In terms of execution time, the A^* -based approach outperforms the

genetic method when an exact solution exists. However, in general, the genetic algorithm is significantly faster (about 60 times for graphs with 10 nodes) and bigger graphs further amplify the effect.

5. Conclusions and future work

As shown by the preliminary experiments our approach provides decent solutions in a very short time. Moreover, genetic algorithms can return a complete candidate solution *at any time*, its quality increasing with the time (i.e. number of generations). Thus, this method is particularly interesting in applications with time constraints. It also provides a better scaling capability than exhaustive search methods, despite of the obvious fact that it will show increasing divergence relative to the global optimum. The principal drawback compared to exhaustive methods is the missing guarantee for finding the globally best scoring mapping. Thus, in applications where accuracy is crucial, exact methods are obviously the better option. If the maximum acceptable structural difference is sufficiently small, tree search combined with a look-ahead strategy offers an extremely good performance. However, the error-correcting (sub-)graph isomorphism problem imposes that a matching between two graphs can be accepted even if there are structural differences.

We are planning to use our algorithm in the problem of finding the best match for a given two-dimensional facial image in a database of three-dimensional face models. We will have to evaluate each model using the best scoring assignment of the entities of the 2D image. This implies the need of extremely quick graph matching algorithms that will perform in a noisy data space.

References

- Berretti, S., Bimbo, A. D., & Vicario, E. (2001). Efficient matching and indexing of graph models in content-based retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, *23*, 1089–1105.
- Conte, D., Foggia, P., Sansone, C., & Vento, M. (2004). Thirty years of graph matching in pattern recognition. *Int. Journal of Pattern Recognition and Artificial Intelligence*, *18*, 265–298.
- Syswerda, G. (1991). Schedule optimization using genetic algorithms. In L. Davis (Ed.), *Handbook of genetic algorithms*, 332–349. New York: Van Nostrand Reinhold.
- Ullmann, J. R. (1976). An algorithm for subgraph isomorphism. *J. ACM*, *23*, 31–42.