A graph classification approach using a multi-objective genetic algorithm. Application to symbol recognition

Romain Raveaux[1], Barbu Eugen[2], Hervé Locteau[2], Sébastien Adam[2], Pierre Héroux[2], Eric Trupin[2]

[1] L3I Laboratory – University of La Rochelle, FRANCE
[2] LITIS Labs – University of Rouen, FRANCE
[1] Romain.Raveaux01@univ-lr.fr
[2] {Barbu.Eugen, Herve.Locteau, Sebastien.Adam, Pierre.Heroux, Eric.Trupin}@univ-rouen.fr

Abstract. In this paper, a graph classification approach based on a multi-objective genetic algorithm is presented. The method consists in the learning of sets composed of synthetic graph prototypes which are used for a classification step. These learning graphs are generated by simultaneously maximizing the recognition rate while minimizing the confusion rate. Using such an approach the algorithm provides a range of solutions, the couples (confusion, recognition) which suit to the needs of the system. Experiments are performed on real data sets, representing 10 symbols. These tests demonstrate the interest to produce prototypes instead of finding representatives which simply belong to the data set.

Keywords: graph classification, multi-objective optimization, machine learning, graph dissimilarity measure.

1 Introduction

Graphs are powerful tools to represents structured objects and they have been applied in many fields of computer science. Graphs unify in a single formalism, web pages [1], molecules [2] and graphic symbols [3] since their vertices represent object components while edges represent relations between components. Symbols can be naturally described in a graph model using primitives (vectors, arcs, connected components, loops…) and geometric relations between these primitives (neighborhood, connection, parallelism…). In this context, the pre-segmented symbol recognition question turns into a graph classification problem which involves comparing graphs, i.e., matching graphs to identify their common features [4]. Only error tolerant matching methods can be efficient due to the noise and the shape variability present in graphic documents. The identification phase is to assign a graph describing an unknown symbol to its class using a learning database.

In this paper, a system able to classify graphs representing symbols is described. It uses a learning database to take into account the variability which can occur in symbol image representation. Our algorithm aims at learning sets of graph prototypes to consider the possible distortions. Then, these prototypes are used in a classification step in order to determine the class of an unknown and noisy symbol in a recognition system. Our approach can be decomposed into 3 steps. First, a corpus of noisy symbol images [5], representing N symbol classes with M distorted symbol images per class, is used to extract a set of M graphs per class. Then, from this learning set, a graph based Genetic Algorithm (GA) is applied. Its aim is to generate sets of K graph prototypes for each class. The values to be optimized by the multi-objective GA are the recognition rate and the confusion rate which are obtained in the simulation of a classification algorithm using the selected prototypes as learning samples and a test database. Both steps (prototypes learning and classification) use a dissimilarity measure called graph probing in order to evaluate the similarity between graphs [20]. This measure has been chosen after a comparative study between different approaches. Finally, in a validation step, a classification algorithm is applied using the selected prototype set as learning elements, a validation database and the same dissimilarity measure. The paper is organized as follows: In the second section, the graph probing concept is introduced. Then, the third section presents the genetic algorithm in use, and particularly the specific genetic operators involved. The fourth section presents the application to the symbol recognition problem, the comparative study between the tested dissimilarity measures and the obtained classification results. Finally, a conclusion is given and future works are brought in section 5.

2. Dissimilarity measures

Measures of dissimilarity between complex objects which have a structure (sets, lists, strings, …) are based on the quantity of shared terms. The simplest similarity measure between two objects is the matching coefficient, which is based on the number of common terms. Using this idea as a starting point, dissimilarity measures which take into account the maximal common subgraph (MCS) of two graphs were proposed in [6]. Another method which proposes a metric distance in the universal set of graphs is the edit distance. It represents the minimum-cost sequence of basic editing operations (e.g. insertion or deletion of vertices and edges with associated costs). The graph edit distance and MCS computation are equivalent to each other under a certain cost function associated to edit operations [7]. These distances between graphs have worst case exponential running times. In our application, we use a genetic algorithm which employs intensively computations of dissimilarities between graphs.

Hence, we have to find faster algorithms which compute dissimilarities, eventually approximations. In such a context, the graph topology can be approximated considering independently the set of vertices and arcs, for instance, edge matching distance or vertex matching distance [8]. Edge matching distance proposes a cost function for the matching of edges and then derives a minimal weight maximal matching between the edge sets of two graphs. This matching has a worst case complexity of $O(n^3)$, where n is the number of edges of the largest graph.

Another possibility to define a similarity measure is to count the number of occurrences of a set of sub graphs (named fingerprints or probes in different contexts) from each graph and to describe the objects to be compared as vectors [9]. In this setting (named graph probing), the similarity between graphs is the similarity between the two associated vectors. These methods are fast since they can be run in linear time, however, when the distance between two graphs is 0, it does not imply that the two graphs are isomorphic. However, a lower bound relation within a factor of four exists between the graph probing and the edit distance [9]. An experimental comparison between graph probing and other approaches is presented in section 4.

3 The genetic algorithm in use

3.1 Genetic operators dedicated to graphs

Genetic Algorithms (GAs) are adaptive heuristic optimization algorithms based on the evolutionary ideas of natural selection and genetics. The basic concept of GAs is designed to simulate natural processes, necessary for evolution of artificial systems. They represent an intelligent exploitation of a random search within a defined search space to solve a problem. As can be seen on fig 1, after a random initialization of a population of possible solutions, GA's are based on a sequential ordering of four main operators: selection, replication, crossover and mutation. In order to apply genetic algorithms to a given problem, three main stages are necessary: the coding of the problem solutions, the definition of the objective function which attributes a fitness to each individual, and the definition of the genetic operators which promote the exchange of genetic material between individuals. In most existing GA applications, a linear representation of individuals is used. Problem parameters are encoded through a binary or a real string. Crossover is then applied through a single-point or two-point based exchange of genes. Regarding mutation, it is applied through a random modification of a small number of genes chosen randomly.

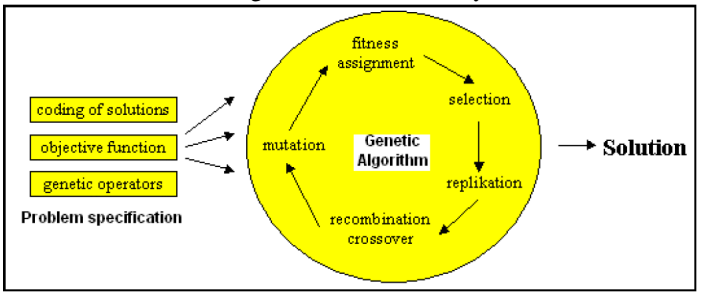

Fig. 1. Overview of a genetic algorithm

In our context of pattern recognition using graph, each individual has to encode a set of graphs (the KxN prototypes). Consequently, the evolution of the individuals through GA implies to revisit classical operators since they have to modify graphs.

Concerning mutation, our operator is based on the six unary edit operations which can be applied to a graph: add or remove a node or an edge and modify a node or an edge label. For each mutation operation, we first decide to apply or not the operator according to a pre-defined rate. If mutation has to be applied, one of the six possibilities is chosen randomly, as well as the new label if the operation is a label modification.

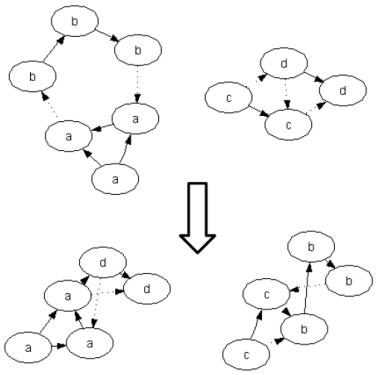

Fig 2: The crossover operator

To perform crossover between individuals (see fig. 2), we first randomly partition the set of nodes of each graph in two subsets (see the label of nodes on figure 2). We call *internal edges*, the edges of the initial graph the nodes of which are in the same subset (continuous lines). At the opposite, edges the nodes of which are in different subsets are called external edges (dotted lines). Then, a node is said to be an output node if it is a source of external edge, and an input node if it is the destination of an external edge. Finally, according to the nature of nodes and edges, fragments are swapped and edges are recombined so that all external edges now point to randomly selected input nodes. Crossover and mutation are combined sequentially as shown in figure 1, after a classical selection process using a fitness based roulette wheel approach.

3.2 The multi-objective optimization concept

When an optimization problem involves more than one objective function, the task of finding one or more optimum solutions is known as multi-objective optimization. Some classical textbooks on this subject have been published, e.g. [10]. We just recall here some essential notions in order to introduce the proposed algorithm. The main difference between single and multi-optimization tasks lies in the requirement of compromises between the various objectives in the multi-optimization case. Even with only two objectives, if they are conflicting, the improvement of one of them leads to a deterioration of the other one. For example, in the context of graph classification, the decrease of the reject rate generally leads to an increase of the confusion rate. Two main approaches are used to overcome this problem in the literature.

The first one consists in the combination of the different objectives into a single one (the simpler way being to use a linear combination of the various objectives), and then to use one of the well-known techniques of single objective optimization (like gradient based methods, simulated annealing or classical genetic algorithm). In such a case, the compromise between the objectives is a priori determined through the choice of the combination rule. The main critic addressed to this approach is the difficulty to choose a priori the compromise. It seems a better idea to postpone this choice after having several candidate solutions at hand. This is the goal of Pareto based method using the notion of dominance between candidate solutions. A solution dominates another one if it is better for all the objectives. This dominance concept is illustrated on figure 3. Two criteria J1 and J2 have to be minimized. The set of non-dominated points that constitutes the Pareto-Front appears as 'O' on the figure, while dominated solutions are drawn as 'X'. Using such a dominance concept, the objective of the optimization algorithm becomes to determine the Pareto front, that is to say the set of non-dominated points. Among the optimization methods that can be used for such a task, genetic algorithms are well-suited because they work on a population of candidate solutions. They have been extensively used in such a context. The most common algorithms are VEGA – Vector Evaluated Genetic Algorithm – [11], MOGA – Multi-Objective Genetic Algorithm –approach [12], SGA – Non-Dominated Sorting Genetic Algorithm – [13], NSGA II [14], PAES – Pareto Archived EvolutionStrategy – [15] and SPEA – Strength Pareto Evolutionary Algorithm – [16]. A good review can be found in [17].

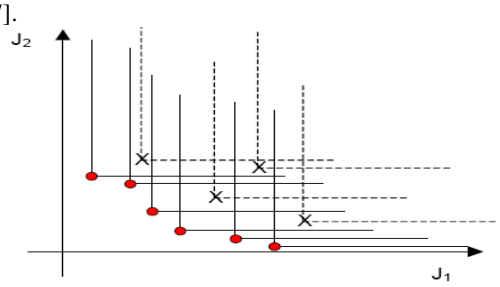

Fig. 3. Illustration of the Pareto Front concept

The proposed genetic algorithm is elitist and steadystate. This means that (i) it manages two populations and (ii) the replacement strategy of individuals in the populations is not made as a whole, but individual per individual. The two populations are a classical population, composed of evolving individuals and an "archive" population composed of the current Pareto Front elements. These two populations are mixed during the iterations of the genetic algorithm. The first population guarantees space exploration while the archive guarantees the exploitation of acquired knowledge and the convergence of the algorithm. This algorithm has been designed in order to be applied to various problems. In the current implementation, the replacement strategy is defined in such a way that the candidate has to be inserted within the archive if no archive element dominates it. In the same time, archive elements dominated by the candidate are eliminated from the archive.

4 Application

In this section, the graph construction step is explained. Then, a comparative study concerning dissimilarity measure is described. It justifies our decision to use graph probing in our context. Finally, the symbol recognition application is presented, results are measured up to another approach and a two objective optimization is performed taking into account the notion of reject.

4.1 Graph data set construction

Our data are made of graphs corresponding to a corpus of 180 noisy symbol images, generated from 10 ideal models proposed in a symbol recognition contest (GREC workshop). In a first step, considering the symbol binary image, we extract both black and white connected components. These connected components are automatically labeled with a partitional clustering algorithm [18] applied on a set of features called Zernike moments [4]. Using these labeled items, a graph is built. Each connected component represents an attributed vertex in this graph. Then, edges are built using the following rule: two vertices are linked with an undirected and unlabeled edge if one of the nodes is one of the h nearest neighbors of the other node in the corresponding image. The two values h and c, concerning respectively, the cluster number found by the clustering algorithm and the number of significant neighbors, are issued from a comparative study. An example of the association between two symbol images and the corresponding graphs is illustrated in fig 4.

4.2 Test on dissimilarity distances

In order to choose the best dissimilarity measure in the context of our application, a study has been led concerning the correlation values between the dissimilarity measures proposed in section 2.
Two experiments compose this study. First, we have computed Pearson correlation coefficients (cor) between the different dissimilarity measures. Results are presented on the first line of table 1. The second experiment concerns the correlation between a userdefined ground truth order (or partial order) and the order calculated using the distance between representations. Such a correlation has to be as high as possible since our objective is the classification of graphs. This correlation can be measured using the Kendall rank correlation coefficient (tau) [19]. Using these values, we can select a graph representation and a dissimilarity measure which satisfies both running time constraints and high correlation with the groundtruth of our application. The obtained values, associated with the corresponding run time complexity, point out the trade-off to be made between the quality (agreement with the ground truth) of a similarity measure and its run time complexity. Since our application is quite demanding of dissimilarity measures, graph probing seems more suitable, showing a better trade-off: meaningful and operating in linear time.

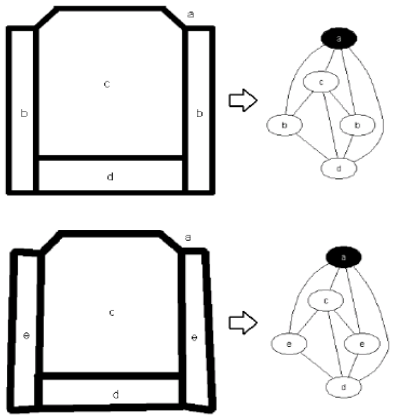

Fig 4: From symbols to Graphs

Table 1. Correlation between the edit distance: (ED), the edge matching distance(EMD), graph probing(GP) and Ground Truth (GT).

	ED	GP	EMD
ED using cor	1	0.58	0.63
ED using tau	1	0.53	0.63
GT using tau	0.699	0.622	0.657

4.3 Classification experiments: Mono-Objective

In a first step, with an aim of comparison, we focus on a mono-objective problem which will be extended in the next par to multi-objective. Under such conditions, the learning algorithm consists in the generation of K graph prototypes per symbol class for a group of N classes.

These prototypes are produced by a graph based GA, the aim of which is to find the near optimal solution of the recognition problem using the selected prototypes. In such a context, each individual in our GA is a vector containing K graphs per class, that is to say K feasible solutions (prototypes) for a given class. Hence, an individual is composed of KxN graphs. For the initialization of the population, each graph of each individual is selected randomly from the initial graph corpus. The fitness (the suitability) of each individual is quantified thanks to the classification rate obtained using the corresponding prototypes and a test database. The classification is processed by a 1-NN classifier using the graph probing distance. Then, using the operators described in section 3, the GA iterates, in order to optimize the classification rate. The stopping criterion is the generation number. At the end of the GA, a classification step is applied on a validation database in order to evaluate the quality of the selected prototypes. The obtained results are compared with an approach which also finds K representatives in a set of objects, described only by their reciprocal matrix distance.

This approach which minimizes (unsquared) distances from objects to representatives is called Partition Around Medoids (PAM) [18]. PAM gives us its K best prototypes for a given class. Using these prototypes and the graph probing distance, we can compute the recognition rate. Table 2 gives the comparative results for K=1,2,3 prototypes per class, and a group of 10 classes. One can also note that using only the ideal models as learning set, a 1NN classification using graph probing provides a 88,28% recognition rate. All these results show the interests of the prototype selection using GA combined with the graph probing approach. According to us, the main reason is that the learning application creates NxK synthetic elements thanks to the genetic operators in order to obtain the best representation of a particular class. Hence, our range of possibilities is not limited to the graphs constituting the class.

Table 2. Global classification rate.

K	1	2	3
PAM	91,42%	94,28%	96,66%
GA	95,29%	96,47%	98,23%

4.3 Multi-Objective experiments:

In another step, we add one more objective to the problem: the confusion rate minimization. Therefore, from now, the classifier has distance rejection capability, the capacity not to take decision in case of ambiguity. Consequently, the relation between the recognition rate and the confusion rate is defined as follow:

Confusion rate = 1 – (Recognition rate + Reject rate).

Hence, the problem becomes to find all dominant solutions, the L couples(confusion, recognition), where one couple represents a set of KxN graphs, K prototypes per class for N classes. We perform our multi-objective algorithm on the learning data in order to discover the Pareto Front and finally the found solutions are evaluated on the test database (Fig 5).

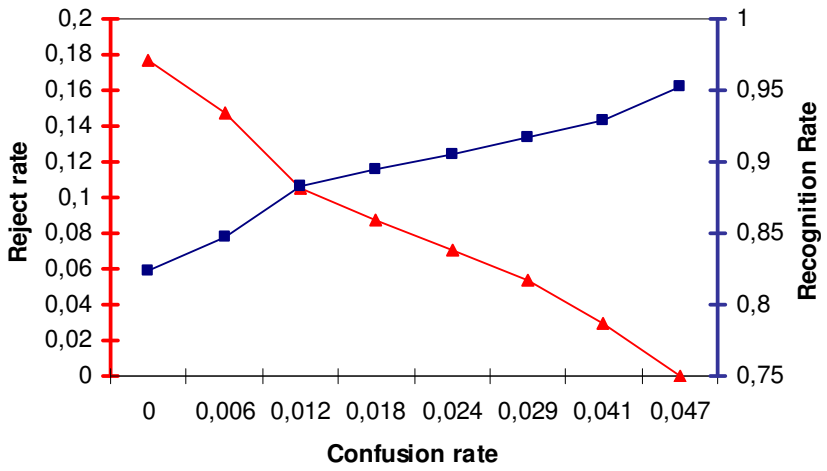

Figure 5: Example for K=1, Pareto Front of two criteria: [Confusion, Recognition]. The left axis corresponds to the reject rate curve (triangles) and the axis on the right goes with recognition rate (squares).

The Figure 5 illustrates clearly the tradeoff to be made between Confusion rate and Recognition rate. Among the range of possibilities found by the method, we spot two specific cases: (i)A confusion rate equal to zero has been found but in against part, the classifier has rejected a lot. (ii)On the other hand, a high recognition rate has been discovered too but this solution implies some classification errors. This choice depends on what are the system needs. The advantage of providing a range of solutions corresponds completely with an adaptive system which will see its constraints changed dynamically, and at any time, the most suitable prototype set could be picked up among the heap of dominant solutions. In such a case, no need to relearn new graph prototypes, since each solution represents an adaptation of the learning. The way of reaching many objectives at once, gives to our method a wider field of action and even if we have performed only a two objective optimization during our experiments, we precise that the approach can be generalized to other criteria.

5 Conclusion

In this paper, a graph classification algorithm has been proposed with an application to symbol recognition. The approach is based on the learning of graph prototypes using multi-objective genetic algorithm and a fast dissimilarity measure called graph probing. This measure has been judged more efficient from the computation speed point of view. The obtained results, compared with a classification using PAM to select prototype, have shown the interest to generate synthetic prototypes through the use of genetic operators rather than finding them among the elements defining the classes.

In addition, the reject is integrated to the method in terms of multi-optimization without including a priori knowledge. A wide range of solutions is provided to fulfill the system needs. Our further works concern different points. The first of them consists in enriching the symbol description as graph, for example through the use of contour vectorization results. A second one consists in testing the approach on a more important database. Another one consists in comparing the approach with the use of graph kernel SVM. And finally, we are investigating some studies to increase the dissimilarity measure relevance. In this direction, we improve the information extracted from graphs, in taking into account in each probe the neighborhood notion. The idea is to give a more significant interpretation of graph topologies in increasing the sub-graph sizes used as probes. In such a context, each probe is a vision of graphs for a certain level of neighborhood.

References

1. A. Schenker, M. Last, H. Bunke and A. Kandel. "Classification of web documents using a graph model", In Proceedings of the 7th International Conference on Document Analysis and Recognition (ICDAR), 2003, pp 240-244.
2. King, R. D., Sternberg, M. J. E., Srinivasan, and Muggleton, S. H., (1995). Knowledge discovery in a database mutagenetic chemicals. In proceedings of the workshop "Statistics, machine leaning, discovery in databases" at the ECML-95.
3. Cordela, L.P., Vento, M., Symbol recognition in documents : a collection of techniques?. International Journal on Document Analysis and Recognition, Vol. 3(2), 2000, pp. 73 – 88.
4. A. Khotazad and Y.H. Hong, "Invariant image recognition by Zernike Moments", PAMI, Vol 12, No 5, May 1990, pp 489-497.
5. E. Valveny and Ph. Dosch. "Symbol Recognition Contest: A Synthesis". Lecture Notes in Computer Science, N° 3088, 2004, pp. 368-385.

6 H. Bunke and K. Shearer, "A graph distance metric based on the maximal common subgraph". Pattern Recogn. Lett., 19, 1998, pp 255-259

7 H. Bunke, "On a relation between graph edit distance and maximum common subgraph", Pattern Recogn. Lett., 18, 1997, pp 689-694.

8 H.P. Kriegel and S. Schönauer, "Similarity Search in Structured Data", Lecture Notes in Computer Science, N° 2737, 2003, pp. 309-319.

9 D. P. Lopresti and G.T. . Wilfong, "A fast technique for comparing graph representations with applications to performance evaluation", International Journal on Document Analysis and Recognition, 6, 2003, pp 219-229.

10 K. Deb, "Multi-Objective optimization using Evolutionary algorithms", Wiley, London, 2001.

11 J.D. Schaffer and J.J. Grefenstette, "Multiobjective learning via genetic algorithms", In Proceedings of the 9th international joint conference on artificial intelligence, Los Angeles, California, pp 593-595, 1985.

12 C.M. Fonseca, P.J. Fleming, "Genetic algorithm for multi-objective optimization: formulation, discussion and generalization", In Stephanie editor, Proceedings of the fifth international conference on genetic algorithm, San Mateo, California, pp 416-423, 1993.

13 N. Srinivas, K. Deb, "Multiobjective optimization using nondominated sorting in genetic algorithm", Evolutionary Computation 2, 1994, pp 221-248.

14 K. Deb, S. Agrawal, A. Pratab and T. Meyarivan, "A fast and elitist multi-objective genetic algorithm: NSGA-II", IEEE Transactions on Evolutionary Computation 6, 2000, pp 182-197.

15 J.D. Knowles, D.W. Corne, “Approximating the nondominated front using the Pareto archived evolution strategy”, Evolutionary computation 8, 2000, pp 149-172.
16 E. Zitzler, L. Thiele, “Multiobjective evolutionary algorithms : a comparative study and the strength pareto approach”, IEEE Transactions on Evolutionary Computation 3, 1999, pp 257-271.
17 C. A. Coello Coello, “A short tutorial on Evolutionary Multiobjective Optimisation”, In Eckart Zitzler, Kalyanmoy Deb, Lothar Thiele, Carlos A. Coello Coello and David Corne (editors), First International Conference on Evolutionary Multi-Criterion Optimization, Lecture Notes in Computer Science, . Springer-Verlag n° 1993, pp 21-40, 2001.
18 L. Kaufman and P.J. Rousseeuw, “Finding groups in data”, John Wiley & Sons, Inc., New York, 1990
19 M. G. Kendall, “Rank Correlation Methods”, Hafner Publishing Co., New York, 1955.
20 Sébastien Sorlin, Christine Solnon, “Reactive Tabu Search for Measuring Graph Similarity”. 172-182 Graph-Based Representations in Pattern Recognition, 5th IAPR InternationalWorkshop, GbRPR 2005