

A Unifying Framework for Relational Structure Matching

Marcello Pelillo

Dipartimento di Matematica Applicata e Informatica
Università “Ca’ Foscari” di Venezia
Via Torino 155, 30173 Venezia Mestre, Italy

Abstract

The matching of relational structures is a problem that pervades computer vision and pattern recognition research. During the past few decades, two radically distinct approaches have been pursued to tackle it. The first views the matching problem as one of explicit search in state-space. The most popular method within this class consists of transforming it in the equivalent problem of finding a large maximal clique in a derived “association graph.” In the second approach, the relational matching problem is viewed as one of energy minimization. In this paper, we provide a unifying framework for relational structure matching which does unify the two existing approaches. The work is centered around a remarkable result proved by Motzkin and Straus which allows us to formulate the maximum clique problem in terms of a continuous optimization problem. We present a class of continuous- and discrete-time “replicator” dynamical systems developed in evolutionary game theory and show how they can naturally be employed to solve our relational matching problem. Experiments are presented which demonstrate the effectiveness of the proposed approach.

1 Introduction

The problem of matching relational structures is of fundamental importance in computer vision and pattern recognition, and a great deal of effort has been devoted over the past decades to devise efficient and robust algorithms for it. Basically, two radically distinct approaches have emerged, a distinction which reflects the well-known dichotomy originated in the Artificial Intelligence field between “symbolic” and “numerical” methods. The first approach views the matching problem as one of explicit search in state-space (see, e.g., [18, 19]). The pioneering work of Ambler *et al.* [1] falls into this class. Their approach is based on the idea that relational structure matching is equivalent to the problem of finding large maximal cliques in the so-called association graph, an auxiliary graph derived from the structures being matched. This framework is attractive because it casts the matching problem in terms of a pure graph-theoretic problem, for which a solid theory and powerful al-

gorithms have been developed [11]. Since its introduction, the association graph technique has been successfully applied to a variety of computer vision problems (e.g., [2, 8]).

In the second approach, the relational matching problem is viewed as one of energy minimization. In this case, an energy (or objective) function is sought whose minimizers correspond to the solutions of the original problem, and a dynamical system, usually embedded into a parallel relaxation network, is used to minimize it [6, 9, 17]. Typically, these methods do not solve the problem exactly, but only in approximation terms. Energy minimization algorithms are attractive because they are amenable to parallel hardware implementation and also offer the advantage of biological plausibility.

In this paper, we develop a new framework for matching relational structures which *does* unify the two approaches just described, thereby inheriting the attractive features of both. The approach is centered around a remarkable result proved by Motzkin and Straus in the mid-1960s, and more recently expanded by many authors [3, 5, 14], which allows us to map the maximum clique problem onto the problem of extremizing a quadratic form over a linearly constrained domain (i.e., the standard simplex in Euclidean space). We present a class of continuous- and discrete-time “replicator” dynamical systems developed in evolutionary game theory as well as other branches of theoretical biology, and show how they can naturally be employed to solve the relational matching problem. Preliminary experiments are presented which confirm the effectiveness of the proposed approach. The technique has recently been applied with success to shape matching problems formulated in terms of “shock” tree matching [15].

2 Relational matching via maximum cliques

A relational structure is a triple $S = (U, \mathcal{P}, \mathcal{R})$, where U is a set of units, $\mathcal{P} = \{P_1, \dots, P_l\}$ is a set of properties, and $\mathcal{R} = \{R_1, \dots, R_k\}$ is a set of (binary) relations over the units. Relational structures are a generalization of traditional graphs, and are indeed often depicted in terms of graphs, the nodes representing the units, and the edges the

relations.

Consider two relational structures $S' = (U', \mathcal{P}', \mathcal{R}')$ and $S'' = (U'', \mathcal{P}'', \mathcal{R}'')$. A pair of units (u', u'') , one from S' and the other from S'' , is said to be *good* if all properties that hold for u' hold for u'' as well, and vice versa, that is if

$$P'_i(u') \Leftrightarrow P''_i(u'')$$

for all $i = 1 \dots l$, where $P'_i \in \mathcal{P}'$ and $P''_i \in \mathcal{P}''$. Similarly, two good pairs (u', u'') and (v', v'') , with $u' \neq v'$ and $u'' \neq v''$, are said to be *compatible* if

$$R'_j(u', v') \Leftrightarrow R''_j(u'', v'') \text{ and } R'_j(v', u') \Leftrightarrow R''_j(v'', u'')$$

for all $j = 1 \dots k$, where $R'_j \in \mathcal{R}'$ and $R''_j \in \mathcal{R}''$. A *match* between S' and S'' is any relation $\mu \subseteq U' \times U''$ such that all its assignments are good and mutually compatible. A match is *maximal* if it is not included in any other match, and is *maximum* if it has largest cardinality. The relational structure matching problem is just the problem of finding a maximum match between two relational structures. When the relational structures being matched are graphs the problem becomes the (double) subgraph isomorphism problem, which is known to be *NP*-complete [4].

Ambler *et al.* [1] introduced the notion of association graph as a useful auxiliary structure for matching relational structures. The association graph of two relational structures S' and S'' is the graph $G = (V, E)$ defined as

$$V = \{(u', u'') \in U' \times U'' : (u', u'') \text{ is good}\}$$

and

$$E = \{((u', u''), (v', v'')) \in V \times V : (u', u'') \text{ and } (v', v'') \text{ are compatible}\}.$$

Given an arbitrary undirected graph G , a subset of vertices is called a *clique* if all its vertices are pairwise connected. A clique is said to be *maximal* if it is not contained in a larger clique, and *maximum* if it is the largest clique in the graph. It is clear that, given the way we have constructed the association graph, the notions of match, maximal match, and maximum match turn out to coincide with those of clique, maximal clique, and maximum clique of the association graph, respectively. The problem of matching two relational structures is therefore equivalent to the maximum clique problem, one of the earliest combinatorial problems proven to be *NP*-complete, and certainly one of the best studied [11].

3 Continuous formulation of MAX-CLIQUE

Let $G = (V, E)$ be an undirected graph, where $V = \{1, \dots, n\}$ is the set of vertices and $E \subseteq V \times V$ is the set of edges, and let $A = (a_{ij})$ denote its adjacency matrix, which is the $n \times n$ matrix defined as follows: $a_{ij} = 1$ if $(i, j) \in E$,

and $a_{ij} = 0$ otherwise. Let S_n denote the standard simplex of the n -dimensional Euclidean space \mathbb{R}^n :

$$S_n = \left\{ \mathbf{x} \in \mathbb{R}^n : x_i \geq 0, i = 1 \dots n, \text{ and } \sum_{i=1}^n x_i = 1 \right\}.$$

Given a subset of vertices C of graph G , we will denote by \mathbf{x}^c its *characteristic vector* which is the point in S_n defined as $x_i^c = 1/|C|$ if $i \in C$, $x_i^c = 0$ otherwise, where $|C|$ denotes the cardinality of C .

Now, consider the following quadratic function:

$$f(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} \quad (1)$$

where “ T ” denotes transposition. The Motzkin-Straus theorem [10] establishes a remarkable connection between global (local) maximizers of f in S_n and maximum (maximal) cliques of G . Specifically, it states that a subset of vertices C of a graph G is a maximum clique if and only if its characteristic vector \mathbf{x}^c is a global maximizer of the function f in S_n . A similar relationship holds between (strict) local maximizers and maximal cliques [5, 14]. This result has an intriguing computational significance in that it allows us to shift from the discrete to the continuous domain in an elegant manner.

One drawback associated with the original Motzkin-Straus formulation relates to the existence of spurious solutions, i.e., maximizers of f which are not in the form of characteristic vectors [14]. In principle, spurious solutions represent a problem since, while providing information about the *size* of the maximum clique, do not allow us to extract the vertices comprising the clique. Fortunately, there is straightforward solution to this problem which has recently been introduced and studied by Bomze [3]. Consider the following regularized version of function f :

$$\hat{f}(\mathbf{x}) = \mathbf{x}^T A \mathbf{x} + \frac{1}{2} \mathbf{x}^T \mathbf{x}. \quad (2)$$

The following is the spurious-free counterpart of the original Motzkin-Straus theorem (see [3] for proof).

Theorem 3.1 *Let C be a subset of vertices of a graph G , and let \mathbf{x}^c be its characteristic vector. Then the following statements hold:*

- (a) C is a maximum clique of G if and only if \mathbf{x}^c is a global maximizer of \hat{f} over the simplex S_n . Its size is then given by $|C| = 1/2(1 \Leftrightarrow f(\mathbf{x}^c))$.
- (b) C is a maximal clique of G if and only if \mathbf{x}^c is a local maximizer of \hat{f} in S_n .
- (c) All local (and hence global) maximizers of \hat{f} over S_n are strict.

Unlike the Motzkin-Straus formulation, the previous result guarantees that *all* maximizers of \hat{f} on S_n are strict, and are characteristic vectors of maximal/maximum cliques in the graph. In an exact sense, therefore, a one-to-one correspondence exists between maximal cliques and local maximizers of \hat{f} in S_n on the one hand, and maximum cliques and global maximizers on the other hand.

4 Replicator equations for matching

Let W be a non-negative $n \times n$ matrix, and consider the following dynamical system:

$$\frac{d}{dt}x_i(t) = x_i(t) \left(\pi_i(t) \Leftrightarrow \sum_{j=1}^n x_j(t) \pi_j(t) \right), \quad i = 1 \dots n \quad (3)$$

where $\pi_i(t) = \sum_{j=1}^n w_{ij} x_j(t)$, $i = 1 \dots n$, and its discrete-time counterpart:

$$x_i(t+1) = \frac{x_i(t) \pi_i(t)}{\sum_{j=1}^n x_j(t) \pi_j(t)}, \quad i = 1 \dots n. \quad (4)$$

It is readily seen that the simplex S_n is invariant under these dynamics, which means that every trajectory starting in S_n will remain in S_n for all future times.

Both (3) and (4) are called *replicator equations* in theoretical biology, since they are used to model evolution over time of relative frequencies of interacting, self-replicating entities [7].

Theorem 4.1 *If W is symmetric, then the quadratic polynomial $F(\mathbf{x}) = \mathbf{x}^T W \mathbf{x}$ is strictly increasing along any non-constant trajectory of both continuous-time (3) and discrete-time (4) replicator equations. Moreover, a vector $\mathbf{x} \in S_n$ is asymptotically stable under (3) and (4) if and only if \mathbf{x} is a strict local maximizer of F on S_n .*

The previous result is known in mathematical biology as the Fundamental Theorem of Natural Selection [3, 7].

We now have all the ingredients to explicitly describe the proposed structure matching algorithm. Let S' and S'' be two relational structures, and let A denote the adjacency matrix of the corresponding association graph. As seen in Section 2, once we have constructed the association graph the relational structure matching problem is just the problem of searching for a large maximal clique in it.

By letting

$$W = A + \frac{1}{2}I,$$

where I is the identity matrix, we know from the previous section that the replicator dynamical system, starting from an arbitrary initial state, will iteratively maximize the function \hat{f} defined in (2) and will eventually converge to a strict local maximizer, which in turn will correspond to the

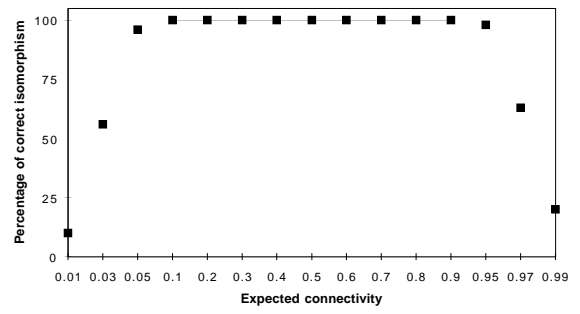


Figure 1: Percentage of correct isomorphisms obtained using replicator equations, as a function of the expected connectivity.

characteristic vector of a maximal clique of the association graph. The solution found will therefore provide a maximal match between the two relational structures. Clearly, in theory there is no guarantee that the converged solution will be a *global* maximizer of \hat{f} , and therefore that it will induce a maximum match. Previous experimental work done on the maximum clique problem [12], and also the results presented in the next section, however, suggest that the basins of attraction of global maximizers are quite large, and very frequently the algorithm converges to one of them.

5 Experimental results

To assess the effectiveness of the proposed approach, we conducted a series of experiments on the graph isomorphism problem, a problem which still resists any computational complexity characterization. Despite decades of active research, in fact, no polynomial-time algorithm for it has yet been found. At the same time, while clearly belonging to NP , no proof has been provided that it is NP -complete [4].

In the experiments reported here, the discrete-time replicator equation (4) was used. The algorithm was started from the barycenter of the simplex and it was stopped when either a maximal clique was found or the distance between two successive points was smaller than a fixed threshold, which was set to 10^{-17} . In the latter case the converged vector was randomly perturbed, and the algorithm restarted from the perturbed point. Because of the one-to-one correspondence between local maximizers and maximal cliques, this situation corresponds to convergence to a saddle point. All the experiments were run on a Sparc20.

Undirected 100-vertex random graphs were generated with expected connectivities ranging from 1% to 99%. For each connectivity value, 100 graphs were produced and each of them had its vertices randomly permuted so as to obtain a pair of isomorphic graphs. Due to the high computational time required, in the 1% and 99% cases the algorithm was tested on 10 pairs, instead of 100. Overall, therefore, 1320 pairs of isomorphic graphs were used. Each pair

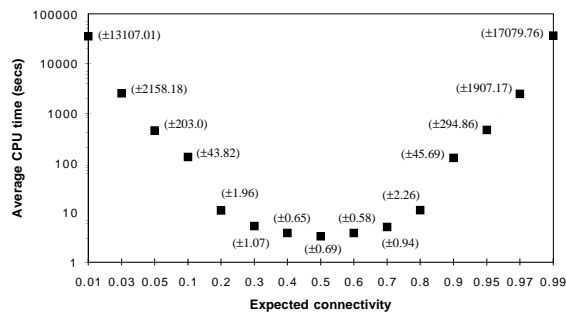


Figure 2: Average CPU time taken by replicator equations, as a function of the expected connectivity. The vertical axis is in logarithmic scale, and the numbers in parentheses represent the standard deviation.

was given as input to the replicator model and, after convergence, a success was recorded when the cardinality of the returned clique was equal to the size of the graphs given as input. Because of the stopping criterion employed, this guarantees that a maximum clique, and therefore a maximum match, was found. The proportion of successes as a function of the expected connectivities is plotted in Fig. 1, whereas Fig. 2 shows the average time taken by the algorithm to converge.

These results are significantly superior to those reported by Simić [20] who obtained poor results at connectivities less than 40% even on smaller graphs (i.e., up to 75 vertices). They also compare favorably with the results obtained more recently by Rangarajan *et al.* [16] on 100-vertex random graphs for connectivities up to 50%. Specifically, at 1% and 3% connectivities they report a percentage of correct isomorphisms of about 30% and 0%, respectively. Using our approach we obtained, on the same kind of graphs, a percentage of success of 80% and 11%, respectively. Rangarajan and Mjolsness [17] also ran experiments on 100-vertex random graphs with various connectivities, using a powerful Lagrangian relaxation network. Except for a few instances, they always obtained a correct solution. The computational time required by their model, however, turns out to largely exceed ours. As an example, the average time taken by their algorithm to match two 100-vertex 50%-connectivity graphs was about 30 minutes on an SGI workstation. As shown in Fig. 2, we obtained identical results in about 3 seconds. However, for very sparse/dense graphs our algorithm becomes slow. In [13], we present an exponential dynamics which turns out to be dramatically faster and even more accurate than the model presented here.

We note that all the algorithms mentioned above do incorporate sophisticated annealing mechanisms to escape from local minima. By contrast, in the presented work no attempt was made to avoid such solutions.

Future work is aimed at applying the proposed methodology to more general matching problems.

Acknowledgments. This work has been done at the Department of Computer Science, Yale University, and funded by CNR, Italy. The author would like to thank I. M. Bomze, A. Rangarajan, K. Siddiqi, and S. W. Zucker for many stimulating discussions.

References

- [1] A. P. Ambler *et al.*, “A versatile computer-controlled assembly system,” in *Proc. 3rd IJCAI*, Stanford, CA, 1973.
- [2] R. C. Bolles and R. A. Cain, “Recognizing and locating partially visible objects: The locus-feature-focus method,” *Int. J. Robotics Res.* 1:57–82, 1982.
- [3] I. M. Bomze, “Evolution towards the maximum clique,” *J. Global Optim.* 10:143–164, 1997.
- [4] M. R. Garey and D. S. Johnson, *Computers and Intractability*. Freeman, San Francisco, CA, 1979.
- [5] L. Gibbons *et al.*, “Continuous characterizations of the maximum clique problem,” *Math. Op. Res.* 22:754–768, 1997.
- [6] S. Gold and A. Rangarajan, “A graduated assignment algorithm for graph matching,” *IEEE Trans. Pattern Anal. Machine Intell.* 18:377–388, 1996.
- [7] J. Hofbauer and K. Sigmund, *The Theory of Evolution and Dynamical Systems*. Cambridge University Press, Cambridge, UK, 1988.
- [8] R. Horaud and T. Skordas, “Stereo correspondence through feature grouping and maximal cliques,” *IEEE Trans. Pattern Anal. Machine Intell.* 11:1168–1180, 1989.
- [9] S. Z. Li, “Matching: Invariant to translations, rotations, and scale changes,” *Pattern Recognition* 25:583–594, 1992.
- [10] T. S. Motzkin and E. G. Straus, “Maxima for graphs and a new proof of a theorem of Turán,” *Canad. J. Math.* 17:533–540, 1965.
- [11] P. M. Pardalos and J. Xue, “The maximum clique problem,” *J. Global Optim.* 4:301–328, 1994.
- [12] M. Pelillo, “Relaxation labeling networks for the maximum clique problem,” *J. Artif. Neural Net.* 2:313–328, 1995.
- [13] M. Pelillo, “Replicator equations, maximal cliques, and graph isomorphism,” *Tech. Rep. TR-1149*, Yale University, Dept. Computer Science, 1998.
- [14] M. Pelillo and A. Jagota, “Feasible and infeasible maxima in a quadratic program for maximum clique,” *J. Artif. Neural Net.* 2:411–420, 1995.
- [15] M. Pelillo, K. Siddiqi, and S. W. Zucker, “Matching hierarchical structures using association graphs,” in *Proc. EC-CV’98*, Freiburg, Germany, 1998.
- [16] A. Rangarajan, S. Gold, and E. Mjolsness, “A novel optimizing network architecture with applications,” *Neural Comput.* 8:1041–1060, 1996.
- [17] A. Rangarajan and E. Mjolsness, “A Lagrangian relaxation network for graph matching,” *IEEE Trans. Neural Net.* 7:1365–1381, 1996.
- [18] L. G. Shapiro and R. M. Haralick, “Structural descriptions and inexact matching,” *IEEE Trans. Pattern Anal. Machine Intell.* 3:504–519, 1981.
- [19] W.-H. Tsai and K.-S. Fu, “Subgraph error-correcting isomorphisms for syntactic pattern recognition,” *IEEE Trans. Syst. Man Cybern.* 13:48–62, 1983.
- [20] P. D. Simić, “Constrained nets for graph matching and other quadratic assignment problems,” *Neural Comput.* 3:268–281, 1991.