Pattern Recognition Letters
journal homepage: www.elsevier.com

Hypergraph Isomorphism Using Association Hypergraphs

Giulia Sandi[a,**], Sebastiano Vascon[a,b], Marcello Pelillo[a,b]

[a]*Dept. of Environmental Sciences, Informatics and Statistics, Ca' Foscari University, Via Torino 155, 30172 Mestre, Venice, Italy*
[b]*European Centre for Living Technology, Ca' Foscari University, Ca' Bottacin Dorsoduro 3911, 30123 Venice, Italy*

ABSTRACT

Association graphs represent a classical tool to deal with the graph matching problem and recently the idea has been generalized to the case of hypergraphs. In this article, the potential of this approach is explored. The proposed framework uses a class of dynamical systems derived from the Baum-Eagon inequality in order to find the maximum (maximal) clique in the association hypergraph, that corresponds to the maximum (maximal) isomorphism between the hypergraphs to be matched. The proposed approach has extensively been tested with experiments on a large synthetic dataset, including hypergraphs of different cardinalities, order and connectivities. In particular the isomorphism version of the problem has been analyzed. The results obtained are impressive in terms of correctness, thus showing that, despite its simplicity, the Baum-Eagon dynamics has an outstanding capacity of finding globally optimal solutions and solving the hypergraph isomorphism problem.

1. Introduction

In recent years hypergraphs (as opposed to graph) have been used to encode structural information in different fields such as computer vision, pattern recognition and machine learning, thanks to the benefits that derive when dealing with relationships among more than two elements, thus encoding a higher pool of information. In Bai et al. (2016) the authors evaluate the complexity traces of different hypergraphs representing images, getting the idea from the work on graphs by Xiao et al. (2009), in order to cluster the data; in Zhang and Hancock (2012) and Zhang et al. (2016) hypergraphs are used for feature selection, relying on the high cardinality of graphs to reflect the functional similarities between more than two features. Moreover, matching this type of structures is of particular interest for solving problems such as, e.g., feature tracking, object recognition, scene registration and shape matching, where high-order relations are commonly used. Various studies have tried to solve this question maximizing the sum of the matching scores, thus transforming it into an optimization problem (see e.g. Duchenne et al. (2011); Lee et al. (2011); Yan et al. (2015) and the references therein).

In Pelillo (1998, 1999) the author focuses on the isomorphism problem using the classical technique of computing the *association graph* between the structures that have to be matched. On this newly built graph, techniques from evolutionary game theory are then applied. Moreover in Pelillo (2003) the use of this approach both to graph isomorphism and to subgraph isomorphism has been widely discussed, highlighting in the obtained results a relation between the complexity class of the analyzed problems and the quality of the obtained results.

Recently, Zhang and Ren (2017) and Hou and Pelillo (2018) tried to apply a similar methodology on uniform hypergraphs, working on the association hypergraph using dynamics inspired from the Baum-Eagon inequality (see Baum and Eagon (1967); Pelillo (1997); Rota Bulò and Pelillo (2013)). In the aforementioned papers, only hypergraphs of cardinality 3 have been taken into consideration, obtaining good results, however with the developed approaches even uniform hypergraphs of larger cardinality can be treated.

Inspired by these recent papers, in this article, which extends the work done in Sandi et al. (2018), we investigate the potential of this technique. In particular, we focus on the simplest case of the matching problem, namely the isomorphism case. Specifically, we carried out a set of experiments using replicator dynamics on a large synthetic dataset, that include 11400 different uniform hypergraphs of varying cardinalities, order and connectivities. The results obtained are impressive: 100% of

**Corresponding author:
e-mail: `giulia.sandi@unive.it` (Giulia Sandi)

the isomorphisms have been successfully returned in all cases, despite the simplicity of the dynamics used.

The article is structured as follows. In section 2 an overview of the state of the art is given; in section 3 we present the definition of association hypergraph together with some fundamental results that are required for using this auxiliary structure in solving the isomorphism problem; in section 4 the Baum-Eagon inequality is introduced, and the related dynamics, both in their linear and exponential form, are described; section 5 illustrates the set-up of the experiments and presents the related results, which proved out to be exceptional in terms of precision; finally, section 6 concludes the article.

2. Related Works

In Zass and Shashua (2008) a probabilistic point of view on the matching problem is offered. The authors employed the Sinkhorn projection to match nodes on graphs and hypergraphs of same sizes, and applied this technique for solving non-rigid image matching. This work has been used in Mirzaalian et al. (2009) to match moles in patients' skin across different scanning time. In Duchenne et al. (2011) the authors proposed a matching algorithm based on high order power iteration, employing an affinity tensor to capture the high order relations among feature tuples, while in Leordeanu et al. (2011) a semi-supervised learning approach that learn the parameters for the hypergraph matching is outlined and the solution is obtained through an Integer Quadratic Program (IQP).

Lee et al. (2011) extended Cho et al. (2010) proposing a hypergraph matching algorithm based on random walk, which was able to embed strong prior on one-to-one correspondences of nodes. Albarelli et al. (2012) use high order matching to classify objects, while in Yan et al. (2015) the hypergraph matching problem has been solved using a discrete formulation that is optimized with a particular adaptive discrete gradient assignment method, that is theoretically guaranteed to converge to a fixed discrete point. In Nguyen et al. (2015) a tensor block coordinate ascent methods for hypergraph matching is proposed, while Jin Chang et al. (2016) used the reweighted random walk approach proposed by Lee et al. (2011) to find correspondences between two dynamic kinematic structures of objects in sequences.

More recently Zhang and Ren (2017) proposed an approach based on game-theory to refining multi-source feature correspondences. Although the optimization methodology is the same of our work (we both optimizes a polynomial with the Baum-Eagon inequality), we differ from this paper in the spirit of the analysis. We are in fact interested in a different problem: the unweighted hypergraph isomorphism. A similar approach has been used in Hou and Pelillo (2018) to perform feature matching across a set of images. Finally Zhang and Wang (2018) use a Markov Chain Monte Carlo method to narrow the search space and solve the hypergraph matching problem.

The vast majority of the works presented here use only small hypergraphs of cardinality 3, due to computational reasons. Even though the cost for solving the hypergraph matching problem increases exponentially with the cardinality and order of the structures to be matched, our method overcome this limitation being able to work with hypergraphs of higher cardinality. In fact we have successfully tested our approach with uniform hypergraphs of cardinality 3, 4 and 5.

3. Hypergraph matching using association hypergraphs

An *unweighted undirected hypergraph* is formally defined by the triplet $H = (V, E, K(H))$, where:

- $V = \{1, \dots, n\}$ represents the finite set of vertices or nodes;
- $E \subseteq 2^V \setminus 0$ represents the finite set of hyperedges or hyperarcs of various cardinalities (where 2^V denotes the powerset of V);
- $K(H) = \{|F| : F \in E\}$ represents the finite set of *edge types* of H, that are all the possible cardinalities of the edges in E.

In this article we focus only on *uniform* hypergraphs, or k-graphs, where all the edges have the same cardinality, and the set of edge cardinalities $K(H)$ is made of one single element $k \geq 2$, hence $E \subseteq V \times \cdots \times V$ for k times. If $k = 2$ we have the classic notion of graph, where only pairwise relations are taken into consideration. The *order* of H is the number of its vertices, while its *size* determines the number of its edges. Given n nodes $i_1, ..., i_n \in V$, they are said to be *adjacent* if there exists an $e \in E$ such that $\{i_1, ..., i_n\} \subseteq e$. The *degree* of a vertex $i \in V$, denoted by $\deg(i)$, is the number of vertices adjacent to it.

Given a k-graph $H = (V, E, k)$, a *clique* is defined as a subset of vertices C such that for all distinct $i_1, ..., i_k \in C$, they are mutually adjacent, that is $\{i_1, ..., i_k\} \in E$. A clique that is not contained in any larger clique is called a *maximal* clique, while the largest clique in the hypergraph is called the *maximum* clique. The cardinality of the maximum clique is called the clique number $\omega(H)$.

In order to ease the lecture of this article, from now on the words hypergraph, k-graph, or graph will be used interchangeably for indicating uniform un-weighted hypergraphs with hyperarcs of cardinality k, when no ambiguity can arise from the context. Similarly the words edge or arc will refer to hyperedge or hyperarc.

Given two hypergraphs $H' = (V', E', k)$ and $H'' = (V'', E'', k)$, an *isomorphism* between them is defined by any bijection $\phi : V' \rightarrow V''$ for which $\{i_1, ..., i_k\} \in E' \Leftrightarrow \{\phi(i_1), \dots, \phi(i_k)\} \in E'', \forall i_1, ..., i_k \in V'$. Two k-graphs are said to be isomorphic if there exists an isomorphism between them.

Usually, when speaking about the graph matching problem we indicate the maximum common sub-graph problem, that aims at understanding if there is a subset of vertices, possibly the largest one, in the two graphs, that are isomorphic. However in this article we are going to focus on the hypergraph isomorphism problem, that is a special case of the more general matching question. We therefore want to understand if two hypergraphs are isomorphic and in case finding the isomorphism itself.

The hypergraph isomorphism problem is easier than the general matching problem (see Garey and Johnson (1979)): the first one is located in the low hierarchy of NP, thus meaning that is not NP-complete nor it has been demonstrated to belong

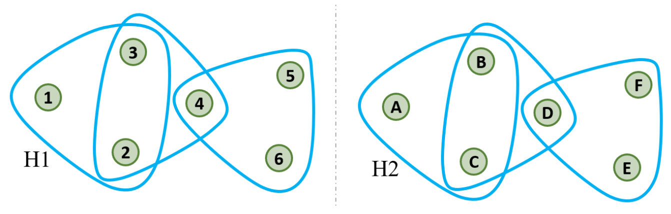

Fig. 1. Two isomorphic and symmetric hypergraphs. There is not a unique isomorphism between these structures. Two possible matches are: (1,A), (2,B), (3,C), (4,D), (5,E), (6, F) and (1,A), (2,C), (3,B), (4,D), (5,F), (6,E), but other matches are possible too. This will result in saddle points when running the optimization process (see Section 5)

to P, while the second one has been demonstrated to be an NP-complete problem.

We can generalize to k-graphs, without too much effort, the notion of *association graph*. This is an handy auxiliary graph structure conceived for solving general graph matching problems, that has been introduced in Barrow and Burstall (1976) and also in Kozen (1978).

Definition 1. *The association hypergraph derived from unweighted uniform hypergraphs $H' = (V', E', k)$ and $H'' = (V'', E'', k)$ is the undirected unweighted hypergraph $H = (V, E, k)$ defined as:*

$$V = V' \times V''$$

and

$$\begin{aligned} E = \{\{(i_1, j_1), ..., (i_k, j_k)\} \in V' \times V'' :\\ i_1 \neq ... \neq i_k, j_1 \neq ... \neq j_k,\\ \{i_1, ..., i_k\} \in E' \Leftrightarrow \{j_1, ..., j_k\} \in E''\}. \end{aligned}$$

Generalizing to hypergraphs an equivalent result achieved by Pelillo (1998, 1999) for graphs, we can demonstrate a one-to-one correspondence between the maximum clique problem and the graph isomorphism problem.

Theorem 1. *Let $H' = (V', E')$ and $H'' = (V'', E'')$ be two hypergraphs of order n and edge cardinality k, and let $H = (V, E)$ be the related association k-graph. Then H' and H'' are isomorphic if and only if $\omega(H) = n$. In this case, any maximum clique of H induces an isomorphism between H' and H'', and vice versa. In general, maximum and maximal common subgraph isomorphisms between H' and H'' are in one-to-one correspondence with maximum and maximal cliques in H, respectively.*

Sketch of proof. Suppose that the two k-graphs are isomorphic, and let ϕ be an isomorphism between them. Then the subset of vertices of H defined as $C_\phi = \{(i, \phi(i)) : \forall i \in V'\}$ is clearly a maximum clique of cardinality n. In reverse, let C be an n-vertex maximum clique of H, and for each $(i, h) \in C$ define $\phi(i) = h$. Then it is easy to see that ϕ is an isomorphism between H' and H'' because of the way the association k-graph is constructed. The proof for the general case is analogous.

With this result we can move our original problem of finding a matching between graphs to the problem of finding a maximum clique in the related association graph. Obviously this does not reduce the complexity of our problem since, again, the clique decision problem is NP-complete (see Karp (1972)), however it gives us the possibility to use one of the existing approaches that are known to work well in solving the latter problem.

Let's consider the arbitrary undirected hypergraph of order n $H = (V, E, k)$, and let S_n indicates the standard simplex of $\mathbb{R}^n$:

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

Given C a subset of vertices of H, $\mathbf{x}^c$ indicates its characteristic vector, and it corresponds to the point in S_n described by:

$$x_i^c = \begin{cases} 1/|C|, & \text{if } i \in C \\ 0, & \text{otherwise} \end{cases}$$

where $|C|$ indicates the cardinality of C.

Let's now consider the *Lagrangian* of H, which is the polynomial function defined as:

$$f(\mathbf{x}) = \sum_{e \in E} \prod_{i \in e} x_i \qquad (1)$$

A point $\mathbf{x}^* \in S_n$ is said to be a global maximizer of f in S_n if $f(\mathbf{x}^*) \geq f(\mathbf{x}), \forall\, \mathbf{x} \in S_n$. On the other hand it is said to be a local maximizer if $\exists$ an $\epsilon > 0$ such that $f(\mathbf{x}^*) \geq f(\mathbf{x})\ \forall\, \mathbf{x} \in S_n$ whose distance from $\mathbf{x}^*$ is smaller than ϵ. Moreover, $\mathbf{x}^*$ is said to be a strict local maximizer if $f(\mathbf{x}^*) = f(\mathbf{x})$ implies $\mathbf{x}^* = \mathbf{x}$.

When dealing with graphs (namely, $k = 2$), the Motzkin-Straus theorem (see Motzkin and Straus (1965)) provides a notable relation between local (global) maximizers of function (1) in S_n and maximal (maximum) cliques of the graph itself. In particular, it claims that a subset C of vertices of a graph G, is a maximum clique if and only if a global maximizer of the Lagrangian of the graph G in the standard simplex S_n is in the form of the characteristic vector $\mathbf{x}^C$. This result interestingly gives the opportunity to move from a discrete to a continuous formulation of our problem.

The formulation of $f(\mathbf{x})$ as the Lagrangian of the hypergraph is the same used in Zhang and Ren (2017), Hou and Pelillo (2018) and Sandi et al. (2018), and is derived by directly extending the Motzkin-Straus theorem on graphs. Given the good results on the aforementioned papers, in our experiments we are going to adopt this function. However different formulations are possible, for example the ones that generalize the Motzkin-Straus theorem, together with Bomze's formulation, to both uniform and non-uniform hypergraphs, presented in Rota Bulò and Pelillo (2009) and Peng et al. (2016) respectively.

4. Finding isomorphisms using the Baum-Eagon dynamics

Given the definitions and results just presented, the problem of finding an isomorphisms can be reduced to solving the following (linearly constrained) polynomial optimization problem:

$$\begin{aligned} \text{maximize} \quad & f(\mathbf{x}) = \sum_{e \in E} \prod_{i \in e} x_i \\ \text{subject to} \quad & \mathbf{x} \in S_n \end{aligned} \qquad (2)$$

Applying a result achieved by Baum and Eagon (1967) we obtain an easy but powerful tool for optimizing this function.

Their work introduces a class of non-linear transformations in the standard simplex, and proves a key result, generalizing a previous one introduced by Blakley (1964) on related characteristics for specific homogeneous quadratic transformations. The Baum-Eagon inequality is presented in the following theorem.

Theorem 2. (Baum and Eagon (1967)) *Let $Q(\mathbf{x})$ be a homogeneous polynomial in the variables x_j with non-negative coefficients, and let $\mathbf{x} \in \Delta$. Define the mapping $z = \mathcal{M}(\mathbf{x})$ from Δ to itself as follow:*

$$z_j = x_j \frac{\partial Q(\mathbf{x})}{\partial x_j} \Big/ \sum_{l=1}^{n} x_l \frac{\partial Q(\mathbf{x})}{\partial x_l}, \quad j = 1, ...n. \tag{3}$$

Then $Q(\mathcal{M}(\mathbf{x})) > Q(\mathbf{x})$ unless $\mathcal{M}(\mathbf{x}) = \mathbf{x}$.

This theorem defines a continuous mapping that is generally known as a *growth transformation*. It is worth noticing that, when defining the mapping $\mathcal{M}$, only first-order derivatives are involved. Nevertheless $\mathcal{M}$ is able, with a finite number of steps, to increase the polynomial Q, thus strongly differentiating itself from classical gradient methods, that need instead to estimate high-order derivatives for determining the size of the infinitesimal steps to be taken. Moreover the use of gradient descend methods induce some problems for the points on the boundary, since some projection operator are performed. By contrast, a simple normalization on rows has to be performed when using results of theorem 2.

Therefore the Baum-Eagon inequality provides a powerful tool and has been widely used for maximizing polynomials functions in S_n. In particular they are a main component in various statistical estimation methodologies developed within the theory of probabilistic functions of Markov Chains (see Baum et al. (1970)), and they are used to analyze the dynamical properties of relaxation labelling processes (see Pelillo (1997)). Theorem 2 can be employed to optimize our problem in equation (2), since we have to maximize the function f that is indeed an uniform polynomial with non-negative coefficients, with the constraint of having $\mathbf{x}$ laying within the standard simplex.

The following discrete time dynamic can be defined from the Baum-Eagon inequality:

$$x_j(t+1) = x_j(t) \frac{\delta_j(\mathbf{x})}{\sum_{i=1}^{n} x_i \delta_i(\mathbf{x})}, \quad j = 1...n, \tag{4}$$

where for readability reasons we have defined $\delta_j(\mathbf{x}) = \partial f(\mathbf{x}) / \partial x_j$.

At time 0, the dynamic in equation 4 starts from a random point within the standard simplex S_n, then, one step after the other, the state vector is updated until convergence. At the end of the iterations the resulting state vector will have the form of a characteristic vector, that can be thresholded against a small amount close to zero thus returning only the subset of nodes in the association graph that are part of the (maximum) clique. Even though we have no theoretical guarantee that the discrete time dynamic returns a global maximum of the function, and not a local maximum, we will see in section 5 that the basins of attraction of the global optimum are quite large, and the use of this dynamic derived from the Baum-Eagon inequality always returned the maximum clique in the association graph, in all the performed experiments, and never incurred in local solutions.

An exponential version of the proposed dynamic can be defined, in order to obtain a faster convergence:

$$x_j(t+1) = x_j(t) \frac{e^{\kappa \delta_j(\mathrm{x})}}{\sum_{i=1}^{n} x_i(t) e^{\kappa \delta_i(\mathrm{x})}}, \quad j = 1...n. \tag{5}$$

Using this exponential dynamic might decrease the time needed in the optimization step, however it comes with a price: we have in fact to tune the newly introduced parameter κ. In setting this parameter we have to be careful to speed-up the process without loosing the correctness of the results. In section 5 some remarks are made on how the value of κ influences the search for the global optimum.

5. Experiments

In this section we are first going to provide a description of how we produced the dataset and how we carried out the experiments. Then the obtained results are analyzed.

5.1. Experimental set-up

The proposed approach has been systematically tested on random hypergraphs of different cardinalities, sizes and connectivities, in order to evaluate its efficacy and to understand if the results heavily differs, depending of the various combination of parameters used in constructing the graphs.

The hypergraph isomorphism problem has been inspected with a specifically designed dataset considering hypergraphs of cardinality 3, 4 and 5. For each cardinality the dataset is made of graphs of different orders, in particular: for $k = 3$ we have $n = \{25, 50, 75, 100\}$, for $k = 4$ we have $n = \{25, 50, 75\}$, while for for $k = 5$ we have $n = \{25, 50\}$. For each possible cardinality-order pair, 15 different connectivity rates are tested in the interval [0.01%, 0.99%] (see Table 1). Moreover, all the different densities have been evaluated on 100 different hypergraphs, except for the combination of parameters $k = 4, n = 75$ and $k = 5, n = 50$ that for complexity reasons have been tested on 30 different hypergraphs. The dataset is therefore made of a total of 11400 experiments.

Table 1. Specification of the different parameters used to create the dataset.

Arcs Cardinality k	3				4			5	
Order	25	50	75	100	25	50	75	25	50
Connectivity Rate	0.01, 0.03, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 0.95, 0.97, 0.99								

The disparity in the considered orders and number of experiments among the different cardinalities is due only to complexity reasons. In fact the proposed framework can be used for larger experiments both in terms of cardinality and order, as long as there is enough time and memory available.

The dataset has been built in the following way: one hypergraph for each experiment (from now on H_1) has been randomly generated adopting the random graph model proposed by Gilbert (1959). Given the following parameters: cardinality k, order n, connectivity rate p, each k-graph is created with the algorithm:

1. start from n isolated nodes;
2. select a k-tuple of nodes, and generate a random number between 0 and 1. If the random number is smaller than p then an edge connecting the selected nodes is added, otherwise the nodes are left disconnected;

3. repeat step 2 for each of the possible $\binom{n}{k}$ edges.

Once H_1 has been built, the second graph H_2 is generated starting from H_1 and randomly perturbing the IDs of its nodes. The k-graphs here created are ready to be matched. Using random graphs to test the framework has two main advantages. First, a wide variety of parameters combination can be extensively tested, since we are not bound to any specific application. In this way even structures that might be uncommon for some application, but still of interest for others, are explored. Second, we can create an experimental system that is easily reproducible, and can therefore be used to compare our approach with other algorithms.

When running the experiments, an immediate problem arises during the creation of the association hypergraph. Since the set of its vertices contains all the possible association of nodes, the order and size increases exponentially with the order of the structures to be matched. For this reason some pruning to the possible couples of nodes, as far as it is possible without removing any information, has been developed. In fact we can easily understand that not all the pair of nodes have sense. In particular, since we are trying to understand if the two hypergraphs are identical, there is no use in pairing nodes with different degree. Therefore, the vertex set of the association graph is constructed as follow:

$$V = \{(i, j) \in V' \times V'' : deg(i) = deg(j)\}$$

and the edge set has been defined as in definition 1. When the two graphs are isomorphic, theorem 1 continues to hold, since the isomorphisms preserves the degree property of vertices. However, using this simple heuristic, we greatly decrease the number of nodes in the association graph, and the same goes for the number of arcs, greatly simplifying the optimization task. In fact, after pruning, in some parameters combination we keep less than the 5% of the possible coupled nodes.

Once the association graph is ready, the optimization process can start. Each experiment has been performed with the linear Baum-Eagon dynamic (Eq. (4)) and with its exponential version (Eq. (5)) with the κ parameter ranging in $\{10, 25, 50\}$. The algorithm was started in the barycentre of the simplex and stopped when either the distance of two subsequent points was smaller than a given threshold, set to 10^{-5}, or when a maximum number of time-steps, equal to 600, has been processed. When the algorithm stops, we check if a clique has been found: in the negative case, the final point is randomly perturbed and the algorithm is started again, using as initial point the randomly perturbed $\mathbf{x}$.

Perturbing is needed in order to avoid *saddle points*, that are solutions in which the algorithm is not able to return a clique because there are some symmetries in the graphs to be matched. This results in more than one possible isomorphism, and the algorithm does not have any way of choosing one solution with respect to the other. This problem arises because the algorithm starts the search for the barycentre of the simplex.

Figure 2 shows all the steps done by the algorithm to solve the symmetric isomorphism problem of figure 1. A perturbation is clearly visible at step 26. After the perturbation the symmetry is broken and the algorithm is able to "choose" the correct pairings, thus reaching the global optimum and finding the maximum clique. The perturbation is repeated for at most 4 times, then the algorithm is stopped independently from the fact that it has returned a clique or not.

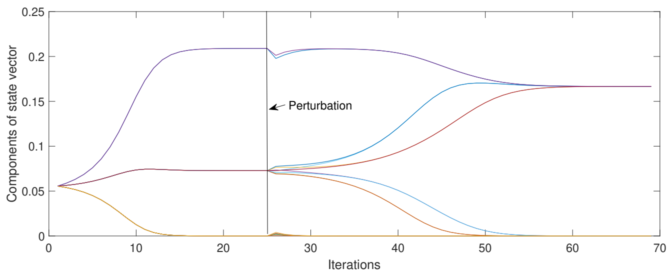

Fig. 2. Evolution through time of the components of the state vector $\mathbf{x}(t)$ from the hypergraphs in figure 1 using the linear dynamic in Eq. 4. A perturbation can be seen at iteration 26, after which the algorithm is able to choose which associations to keep which to discard.

Once the optimization is finished we have to understand if the returned state vector represents a correct isomorphism. In order to automatically check the results, the vector is thresholded against a small value close to 0 (the value 10^{-5} has been used), thus selecting only some couples. Then we verify if the returned couples represent a clique in the association hypergraph, and, in case they do, we verify if the clique number is equal to the number of nodes of the original structures. In fact, from theorem 1, there is the guarantee of finding the isomorphism only if the returned couples represent a maximum clique. Since H_1 and H_2 are of the same order by construction (otherwise we wouldn't have any isomorphism), there can not be an isomorphism concerning a number of nodes that is larger than the size of the vertex set of these two graphs. All the experiments have been run on a workstation equipped with an Intel Core i7-6800K at 3.40GHz with 128GB of RAM, and the code for the framework has been implemented in C.

5.2. Experimental results

In terms of correctness, the proposed framework, has returned impressive results: as we can see in figure 3, all the isomorphisms have been properly found, with the algorithm returning 100% of the nodes exactly coupled, for all the 11400 experiments, independently of the dynamic that has been used and independently of the cardinality k, the number of nodes n and the connectivity rate p.

The mean CPU time needed to run the optimization, considering the best performing dynamic for each set of parameters, is shown in figure 4. As we can see all the curves have the same behaviour, showing that the algorithm is extremely slow when dealing with very sparse or very dense graphs, while it is way faster when solving matching between graphs with mean density. This is due to the fact that the association hypergraphs in the extreme connectivity rates are way bigger and denser than the ones in the median connectivity rates, thus giving birth to more complex polynomials to be optimized.

With no surprise, we see that dealing with smaller hypergraphs results in shorter execution times, nevertheless the behaviour of the curves according to all the other parameters is exactly the same independently on the cardinality or on the order of the original hypergraphs. We can also notice than, as it could be expected, keeping the cardinality k fixed, execution

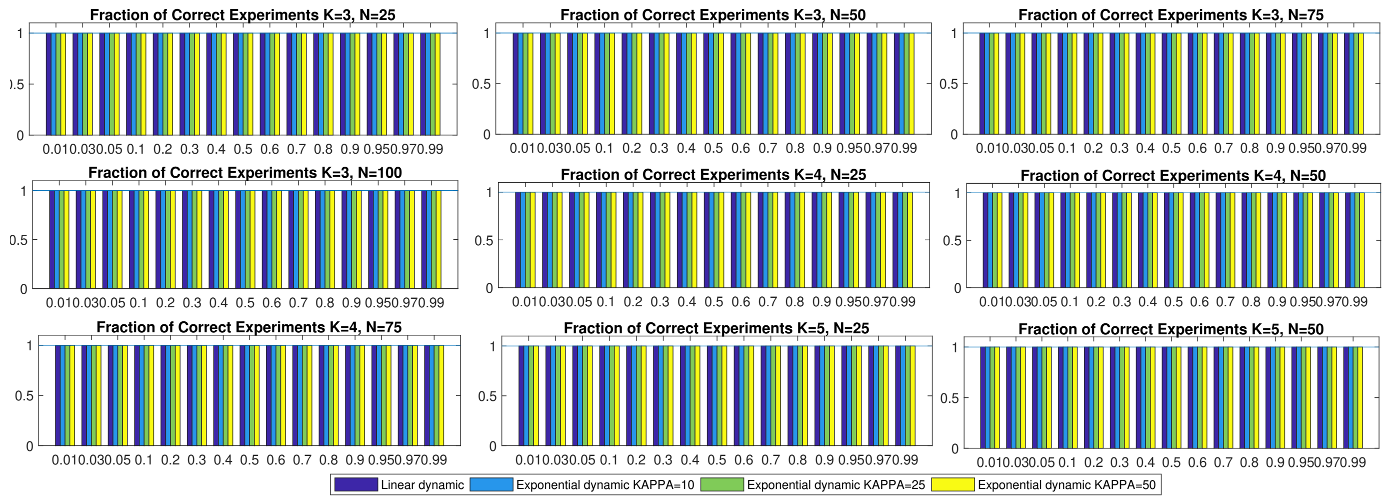

Fig. 3. Image has been changed Fraction of correct results for all the different experiments tested. 100% of all the performed experiments, independently of the different combination of parameters, have returned the correct isomorphism in all the dynamic used. On the x-axis we have the different connectivity rates, on y-axis we have the fraction of correct results.

takes longer for hypergraphs of bigger order; vice versa, keeping the order n fixed, the higher the cardinality the longer the running time. In general, the exponential dynamic always out-

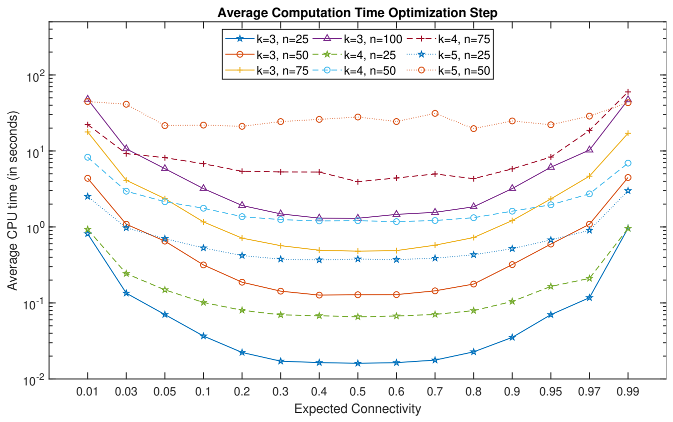

Fig. 4. Mean CPU timeneeded to run the optimization algorithm for finding isomorphism on hypergraphs with different cardinalities and orders. Only the best performing dynamic is drawn for each combination of parameters. Y-axes are in logarithmic scale.

performs the linear one, thus resulting to be really attractive, from a computational point of view. However it introduces a parameter to be tuned, the κ in equation 5. The value for κ has been searched initially in a logarithmic scale $\{1, 10, 100\}$ and further refined in the proximity of a good value. With $\kappa = 100$ we obtained a strongly oscillatory trend in the elements of the state vector that prevented the dynamic to converge (see figure 5). On the other hand, with $\kappa = 1$ the dynamic was nearly as slow as the linear one, thus not giving any incentive in using the exponential form. With $\kappa = 10$ we achieved a faster and convergent behaviour, hence we refined the space for this parameter between 10 and 100.

Three different values of κ (10, 25, 50) have been used on all experiments. In nearly every parameters configuration, the best performing dynamic that has been inserted in figure 4 is always the exponential with $\kappa = 50$.

The only exception is when we have $k = 3$ and $n = 25$. In this case the graphs to be matched, and therefore the related association hypergraph, are really small, and the exponential

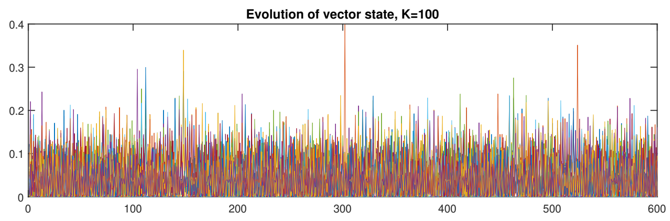

Fig. 5. Evolution of the state vector when matching two hypergraphs with $k = 3$, $n = 75$ and $p = 0.7$ using the exponential dynamic with $\kappa = 100$. The value of κ is too big and the dynamic is not able to converge.

dynamic with $\kappa = 50$ oscillates too much. The best performing dynamic becomes therefore the exponential one with $\kappa = 25$.

Figure 6 shows the evolution through time of the state vector in one of the experiments with $k = 3$ and $n = 25$, with all the different dynamics.The first thing that is clearly visible is that all the dynamics have the same behaviour, perturbing exactly one time the state vector before finding the correct isomorphism. As we could have expected, the linear dynamic is the slowest one, while all the exponential dynamics are faster. However we can see that the value $\kappa = 50$ is too high: even though the correct isomorphism is always found, this dynamic oscillates too much and it needs more iterations in order to converge to a final solution, thus requiring more time than all the other dynamics. This shows that the value of κ is strictly dependent on the problem under examination, and a single value cannot be used on all the experiments. Thus, in a real setting with a specific problem, the parameters of that problem have to be analyzed and evaluated, and the value of κ has to be tuned according to those specific values.

6. Conclusions

In this paper, we tackled the hypergraph isomorphism problem by generalizing the notion of association graph to the higher-order case of uniform hypergraphs. Given two uniform k-graphs, we proved that maximal cliques in their association hypergraph, are in one-to-one correspondence with their possible isomorphisms. Furthermore, in our extensive experimentation, we successfully employed the simple Baum-Eagon inequality and its exponential variant to find maximum cliques.

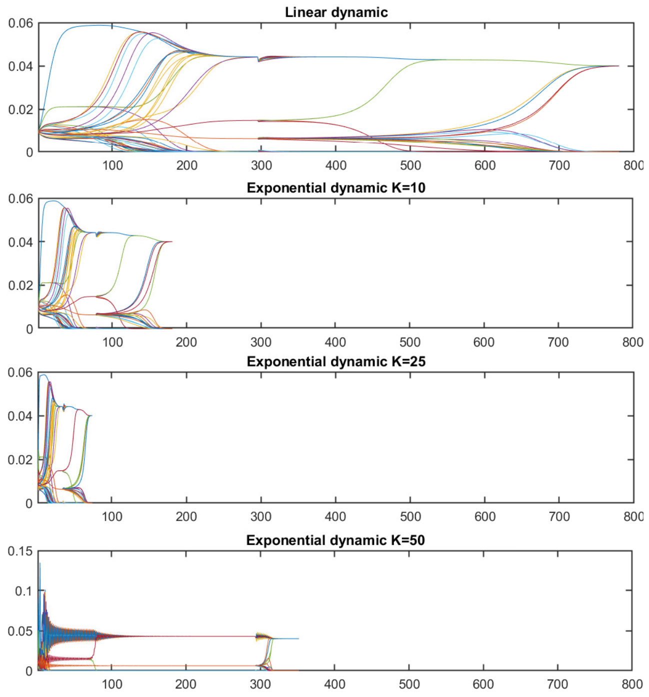

Fig. 6. Comparison of the evolution of the state vector for one experiment with cardinality $k = 3$, order $n = 25$ and connectivity $p = 0.99$. Even though all the dynamics converge to the correct isomorphism after exactly one perturbation, the number of iteration needed to stop the algorithm is different. It is worth noticing that the value $\kappa = 50$ is too large for the exponential dynamic, since it oscillates a lot, thus needing a lot of iterations to converge.

We reported impressive results: in 11400 experiments on randomly generated hypergraphs of different cardinalities, orders and connectivities we have always obtained 100% of correct isomorphisms, emphasizing the ability of the proposed dynamical system to escape from local optima. Alongside the experimentation, we have shown that for the extreme connectivity rates (sparse and dense graphs) very large association hypergraph are generated needing more resources to run the optimization. In such cases, the use of a sparsification technique and the exponential dynamics might lower the time and space needed to perform the experiments, but at the price of setting the parameter κ. In our future work we plan to explore the more general and complex task of sub-hypergraph isomorphism, and to investigate the regularized formulation proposed in Rota Bulò and Pelillo (2009).

References

Albarelli, A., Bergamasco, F., Rossi, L., Vascon, S., Torsello, A., 2012. A stable graph-based representation for object recognition through high-order matching, in: Proc. 21th International Conference on Pattern Recognition (ICPR), pp. 3341–3344.

Bai, L., Escolano, F., Hancock, E., 2016. Depth-based hypergraph complexity traces from directed line graphs. Pattern Recognition 54.

Barrow, H.G., Burstall, R.M., 1976. Subgraph isomorphism, matching relational structures and maximal cliques. Inf. Process. Lett. 4, 83–84.

Baum, L.E., Eagon, J.A., 1967. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. Bulletin of the American Mathematical Society 73, 360–363.

Baum, L.E., Petrie, T., Soules, G., Weiss, N., 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains. The annals of mathematical statistics 41, 164–171.

Blakley, G.R., 1964. Homogeneous nonnegative symmetric quadratic transformations. Bulletin of the American Mathematical Society 70, 712–715.

Cho, M., Lee, J., Lee, K.M., 2010. Reweighted random walks for graph matching, in: European Conference on Computer Vision (ECCV), Springer. pp. 492–505.

Duchenne, O., Bach, F., Kweon, I., Ponce, J., 2011. A tensor-based algorithm for high-order graph matching. IEEE Transactions on Pattern Analysis and Machine Intelligence 33, 2383–2395.

Garey, M.R., Johnson, D.S., 1979. Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman & Co.

Gilbert, E.N., 1959. Random graphs. The Annals of Mathematical Statistics 30, 1141–1144.

Hou, J., Pelillo, M., 2018. A game-theoretic hyper-graph matching algorithm, in: Proc. 24th International Conference on Pattern Recognition (ICPR), pp. 1012–1017.

Jin Chang, H., Fischer, T., Petit, M., Zambelli, M., Demiris, Y., 2016. Kinematic structure correspondences via hypergraph matching, in: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 4216–4225.

Karp, R.M., 1972. Reducibility among combinatorial problems, in: Complexity of computer computations. Springer, pp. 85–103.

Kozen, D., 1978. A clique problem equivalent to graph isomorphism. ACM SIGACT News 10, 50–52.

Lee, J., Cho, M., Lee, K.M., 2011. Hyper-graph matching via reweighted random walks, in: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1633–1640.

Leordeanu, M., Zanfir, A., Sminchisescu, C., 2011. Semi-supervised learning and optimization for hypergraph matching, in: International Conference on Computer Vision (ICCV), pp. 2274–2281.

Mirzaalian, H., Hamarneh, G., Lee, T.K., 2009. A graph-based approach to skin mole matching incorporating template-normalized coordinates, in: Conference on Computer Vision and Pattern Recognition. (CVPR), pp. 2152–2159.

Motzkin, T.S., Straus, E.G., 1965. Maxima for graphs and a new proof of a theorem of Turán. Canad. J. Math. 17, 533–540.

Nguyen, Q., Gautier, A., Hein, M., 2015. A flexible tensor block coordinate ascent scheme for hypergraph matching, in: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 5270–5278.

Pelillo, M., 1997. The dynamics of nonlinear relaxation labeling processes. Journal of Mathematical Imaging and Vision 7, 309–323.

Pelillo, M., 1998. A unifying framework for relational structure matching, in: Proc. 14th International Conference on Pattern Recognition, (ICPR), pp. 1316–1319.

Pelillo, M., 1999. Replicator equations, maximal cliques, and graph isomorphism. Neural computation 11, 1933–1955.

Pelillo, M., 2003. Computational complexity and the elusiveness of global optima. Limitations and future trends in neural computation. NATO science series 186, 71–94.

Peng, Y., Peng, H., Tang, Q., Zhao, C., 2016. An extension of the motzkin-straus theorem to non-uniform hypergraphs and its applications. Discrete Applied Mathematics 200, 170–175.

Rota Bulò, S., Pelillo, M., 2009. A generalization of the motzkin–straus theorem to hypergraphs. Optimization Letters 3, 287–295.

Rota Bulò, S., Pelillo, M., 2013. A game-theoretic approach to hypergraph clustering. IEEE Transactions on Pattern Analysis and Machine Intelligence 35, 1312–1327.

Sandi, G., Vascon, S., Pelillo, M., 2018. On association graph techniques for hypergraph matching, in: Structural, Syntactic, and Statistical Pattern Recognition - Joint IAPR International Workshop, (S+SSPR), Beijing, China, pp. 481–490.

Xiao, B., Hancock, E.R., Wilson, R.C., 2009. Graph characteristics from the heat kernel trace. Pattern Recognition 42, 2589–2606.

Yan, J., Zhang, C., Zha, H., Liu, W., Yang, X., Chu, S.M., 2015. Discrete hypergraph matching, in: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1520–1528.

Zass, R., Shashua, A., 2008. Probabilistic graph and hypergraph matching, in: Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1–8.

Zhang, H., Ren, P., 2017. Game theoretic hypergraph matching for multi-source image correspondences. Pattern Recognition Letters 87, 87–95.

Zhang, R., Wang, W., 2018. Second- and high-order graph matching for correspondence problems. IEEE Transactions on Circuits and Systems for Video Technology 28, 2978–2992.

Zhang, Z., Bai, L., Liang, Y., Hancock, E., 2016. Joint hypergraph learning and sparse regression for feature selection. Pattern Recognition 63.

Zhang, Z., Hancock, E., 2012. Hypergraph based information-theoretic feature selection. Pattern Recognition Letters 33.