A Game-Theoretic Approach to Hypergraph Clustering

Samuel Rota Bulò and Marcello Pelillo, *Fellow*, *IEEE*

Abstract—Hypergraph clustering refers to the process of extracting maximally coherent groups from a set of objects using high-order (rather than pairwise) similarities. Traditional approaches to this problem are based on the idea of partitioning the input data into a predetermined number of classes, thereby obtaining the clusters as a by-product of the partitioning process. In this paper, we offer a radically different view of the problem. In contrast to the classical approach, we attempt to provide a meaningful formalization of the very notion of a cluster and we show that game theory offers an attractive and unexplored perspective that serves our purpose well. To this end, we formulate the hypergraph clustering problem in terms of a noncooperative multiplayer "clustering game," and show that a natural notion of a cluster turns out to be equivalent to a classical (evolutionary) game-theoretic equilibrium concept. We prove that the problem of finding the equilibria of our clustering game is equivalent to locally optimizing a polynomial function over the standard simplex, and we provide a discrete-time high-order replicator dynamics to perform this optimization, based on the Baum-Eagon inequality. Experiments over synthetic as well as real-world data are presented which show the superiority of our approach over the state of the art.

Index Terms—Hypergraph clustering, evolutionary game theory, polynomial optimization, Baum-Eagon inequality, high-order replicator dynamics

1 Introduction

Clustering is the problem of organizing a set of objects into groups, or *clusters*, in such a way as to have similar objects grouped together and dissimilar ones assigned to different groups, according to some similarity measure (for a recent review, see [1]). The vast majority of approaches to clustering available in the literature assume that object similarities are expressed as pairwise relations, but in some applications, such as, for example, face clustering [2], perceptual grouping [3], and parametric motion segmentation [3], [4], image categorization [5], higher order relations turn out to be more appropriate, and approximating them in terms of pairwise interactions can lead to substantial loss of information. As an illustrative example, taken from [2], consider the problem of grouping a given set of d-dimensional euclidean points into lines. As every pair of data points trivially defines a line, there is no meaningful pairwise measure of similarity for this problem. However, it makes perfect sense to define similarity measures over triplets of points that indicate how close they are to being collinear. Clearly, this example can be generalized to any model fitting problem, where the deviation of a set of points from the model provides a measure of their dissimilarity. The problem of data clustering using high-order similarities is usually referred to as *hypergraph clustering* since we can represent any instance of this problem by means of a hypergraph, where vertices are the objects to be clustered and the (weighted) hyperedges encode high-order similarities.

In the past few years, there has been increasing interest around the hypergraph clustering problem in different application areas such as computer vision, machine learning, and VLSI design. Zien et al. [6] proposed two approaches called "clique expansion" and "star expansion," which transform the similarity hypergraph into an edge-weighted graph whose edge-weights are a function of the hypergraph's original weights, thereby tackling the problem using standard pairwise algorithms. Bolla [7] defined a Laplacian matrix for an unweighted hypergraph and established a link between the spectral properties of this matrix and the hypergraph's minimum cut. Rodrìguez [8] achieved similar results by transforming the hypergraph into a graph according to "clique expansion" and showed a relationship between the spectral properties of a Laplacian of the resulting matrix and the cost of minimum partitions of the hypergraph. Zhou et al. [9] generalized their earlier work on regularization on graphs and defined a hypergraph normalized-cut criterion for a k-partition of the vertices, which can be achieved by finding the second smallest eigenvector of a normalized Laplacian. This approach generalizes the well-known "Normalized cut" pairwise clustering algorithm [10]. In [2], we find another work based on the idea of applying a spectral graph partitioning algorithm on an edge-weighted graph which approximates the original (edge-weighted) hypergraph. It is worth noting that all these approaches, though designed to deal with higher order relations, can easily be reduced to standard pairwise approaches, as shown in [11]. A different formulation is introduced in [4], where the clustering

• *The authors are with the Dipartimento di Scienze Ambientali, Informatica e Statistica, Università Cà Foscari Venezia, via Torino 155, Venezia-Mestre 30172, Italy. E-mail: {srotabul, pelillo}@dais.unive.it.*

Manuscript received 25 Mar. 2011; revised 16 Apr. 2012; accepted 1 Oct. 2012; published online 12 Oct. 2012.
Recommended for acceptance by M. Meila.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2011-03-0177.
Digital Object Identifier no. 10.1109/TPAMI.2012.226.

problem with higher order (supersymmetric) similarities is cast into a nonnegative factorization of the closest hyperstochastic version of the input affinity tensor. Along similar lines, in [3] a pairwise similarity matrix is derived from a factorization of the input affinity tensor, which is then given as input to standard spectral (graph) clustering techniques. Finally, the leading tools in the field of VLSI design are based on two-phase multilevel approaches [12]. In the first phase, a hierarchy of hypergraphs is constructed where the hypergraph at each level is a coarser version of the hypergraph at the previous one according to some measure of homogeneity. In the second phase, starting from a partitioning of the coarsest level, the algorithm works its way down the hierarchy and each level greedily updates the partitioning obtained at the previous one.

All the approaches described above rely on the assumption that the goal of the clustering algorithm is to assign each data point to exactly one label denoting class membership. In so doing, clusters are not modeled and sought directly, but they are obtained as a by-product of a partition of the input data into a predetermined number of classes. This renders these approaches vulnerable to applications where the number of classes is not known in advance or where data is affected by clutter elements which do not belong to any meaningful class. Indeed, there are various applications for which it makes little sense to force all data items to belong to some group, a process which might result either in poorly coherent clusters or in the creation of extra spurious classes. As an example, consider the classical figure/ground separation problem in computer vision which asks for extracting a coherent region (the figure) from a noisy background [13], [14]. It is clear that, due to their intrinsic nature, partitional algorithms have no chance of satisfactorily solving this problem, being, as they are, explicitly designed to partition all the input data, and hence the unstructured clutter items too, into coherent groups. More recently, motivated by practical applications arising in document retrieval and bioinformatics, a conceptually identical problem has attracted some attention within the machine learning community (in the context of pairwise relations) and is generally known under the name of one-class clustering [15], [16]. Further, by adopting a partitional approach, clusters are by definition disjoint sets. However, there are a variety of important applications where this requirement is too restrictive. Examples abound and include, e.g., clustering microarray gene expression data (wherein a gene often participate in more than one process), clustering documents into topic categories, perceptual grouping, and segmentation of images with transparent surfaces. In fact, the importance of dealing with overlapping clusters was recognized long ago [17] and recently there has been renewed interest around this problem [18], [19]. Typically, this is solved by relaxing the constraints imposed by crisp partitions in such a way as to have "soft" boundaries between clusters.

In this paper, following [20], [21], we offer a radically different perspective to the hypergraph clustering problem. Instead of insisting on the idea of determining a partition of the input data, and hence obtaining the clusters as a by-product of the partitioning process, we reverse the terms of the problem and attempt instead to derive a rigorous formulation of the very notion of a cluster. This allows one, in principle, to deal with more general problems where clusters may overlap and/or clutter points may get unassigned. The starting point of our approach is the elementary observation that a "cluster" may be informally defined as a maximally coherent set of data items, i.e., as a subset of the input data C which satisfies both an *internal* criterion (all elements belonging to C should be highly similar to each other) and an *external* one (all elements outside C should be highly dissimilar to the ones inside). In our endeavor to provide a formal definition of the notion of a cluster, we found that game theory offers an elegant and general perspective that serves our purposes well. The basic idea behind our framework is that the hypergraph clustering problem can be considered as a multiplayer noncooperative "clustering game." Within this context, the notion of a cluster turns out to be equivalent to a classical equilibrium concept from (evolutionary) game theory, as the latter reflects both the internal and external cluster conditions alluded to before. We also show that there exists a one-to-one correspondence between these equilibria and the local solutions of a linearly constrained polynomial optimization problem, thereby generalizing the work described in [20]. This characterization allows us to employ a powerful class of dynamical systems to extract our clusters, based on the well-known Baum-Eagon inequality, which generalize classical (pairwise) replicator dynamics [22], [23] from evolutionary game theory to higher order interactions. A distinguishing feature of our approach is that, unlike standard partitional techniques, we do not need to know the number of clusters is advance as we extract them sequentially. Experiments on various hypergraph clustering problems show the superiority of the proposed approach over state-of-the-art techniques.

The paper is organized as follows: In Section 2, we provide a brief introduction to main concepts and results of evolutionary game theory. Next, in Section 3, we formulate the clustering problems as an evolutionary game and provide support to the claim that its equilibria can be considered as a natural formalization of the notion of a cluster. In Section 4, we prove the polynomial optimization characterization of our equilibria and describe an algorithm to find them. In Section 5, we present our experimental results, and Section 6 concludes the paper. Note that a preliminary version of this paper appeared in [24].

2 Notions from Evolutionary Game Theory

According to classical game theory [25], a game of strategy can be formalized as a triplet $\Gamma = (P, S, \pi)$, where $P = \{1, \ldots, k\}$ is a set of $k \geq 2$ "players" (or agents), $S = \{1, \ldots, n\}$ is a set of *pure strategies* (or actions) available to each player, and $\pi : S^k \to \mathbb{R}$ is a *payoff function*, which assigns a utility to each *strategy profile* $\mathbf{s} = (s_1, \ldots, s_k) \in S^k$, which is an (ordered) set of pure strategies played by the different players.[1] A game Γ is *supersymmetric* if its payoff function is

1. We note that although we restrict ourselves to games where all players share the same set of pure strategies and payoff function, in more general settings each agent can well be associated to its own pure strategy set and payoff function.

supersymmetric, i.e., if it is invariant under permutations of the strategy profile. In the sequel we will deal only with such games and therefore we assume π to be supersymmetric.

Evolutionary game theory originated in the early 1970s as an attempt to apply the principles and tools of game theory to biological contexts, with a view to modeling the evolution of animal, as opposed to human, behavior (see the classical work by Smith [26] who pioneered the field). It considers an idealized scenario whereby individuals are repeatedly drawn at random from a large, ideally infinite, population to play a game $\Gamma = (P, S, \pi)$. In contrast to classical game theory, here players are not supposed to behave rationally or to have complete knowledge of the details of the game. They act instead according to an inherited behavioral pattern, or pure strategy, and it is supposed that some evolutionary selection process operates over time on the distribution of behaviors. Here, and in the sequel, an agent with preassigned strategy $j \in S$ will be called *j-strategist*. The state of the population at a given time t can be represented as an n-dimensional vector $\mathbf{x}(t)$, where $x_j(t)$ represents the fraction of j-strategists in the population at time t. Hence, the initial distribution of preassigned strategies in the population is given by $\mathbf{x}(0)$. The set of all possible states describing a population is given by

$$\Delta = \left\{ \mathbf{x} \in \mathbb{R}^n \ : \ \sum_{j \in S} x_j = 1 \text{ and } x_j \geq 0 \text{ for all } j \in S \right\},$$

which is called *standard simplex*. As time passes, the distribution of strategies in the population changes under the effect of a selection mechanism which, by analogy with Darwinian process, aims at spreading the fittest strategies in the population to the detriment of the weakest one which, in turn, will be driven to extinction (we postpone the formalization of one such selection mechanism to Section 4). For notational convenience, we drop the time reference t from a population state and we refer to $\mathbf{x} \in \Delta$ as a population rather than population state. Moreover, we denote by $\sigma(\mathbf{x})$ the *support* of $\mathbf{x} \in \Delta$:

$$\sigma(\mathbf{x}) = \{ j \in S \ : \ x_j > 0 \},$$

which is the set of strategies that are alive in a given population $\mathbf{x}$.

We will find it useful to define the following function $u : \Delta^k \to \mathbb{R}$:

$$u\Big(\mathbf{y}^{(1)}, \ldots, \mathbf{y}^{(k)}\Big) = \sum_{(s_1, \ldots, s_k) \in S^k} \pi(s_1, \ldots, s_k) \prod_{i=1}^{k} y_{s_i}^{(i)}, \quad (1)$$

which is invariant under any permutation of its arguments due to the supersymmetry of the payoff function π. Also, we will use the notations $\mathbf{x}^{[k]}$ as a shortcut for a sequence $(\mathbf{x}, \ldots, \mathbf{x})$ of k identical states $\mathbf{x}$, and $\mathbf{e}^j$ to indicate the n-vector with $x_j = 1$ and zero elsewhere. Now, it is easy to see that the expected payoff earned by a j-strategist ($j \in S$) in a population $\mathbf{x} \in \Delta$ is given by $u(\mathbf{e}^j, \mathbf{x}^{[k-1]})$, while the expected payoff over the entire population is given by $u(\mathbf{x}^{[k]})$.

A fundamental notion in game theory is that of an equilibrium [22]. Intuitively, an evolutionary process reaches an equilibrium $\mathbf{x} \in \Delta$ when every individual in the population obtains the same expected payoff and no strategy can thus prevail upon the other ones. Formally, $\mathbf{x} \in \Delta$ is a *Nash equilibrium* if

$$u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big) \leq u\Big(\mathbf{x}^{[k]}\Big), \qquad \text{for all } j \in S. \quad (2)$$

In other words, at a Nash equilibrium every agent in the population performs at most as well as the overall population expected payoff. Within a population-based setting, however, the notion of a Nash equilibrium turns out to be too weak as it lacks stability under small perturbations. This motivated J. Maynard Smith, in his seminal work [26], to introduce a refinement of the Nash equilibrium concept generally known as an Evolutionary Stable Strategy (ESS). His original work involved pairwise interactions (two-player games), but his notion was later generalized to multiplayer games [27]. Formally, assume that in a population $\mathbf{x} \in \Delta$, a small share ϵ of mutant agents appears whose distribution of strategies is $\mathbf{y} \in \Delta$. The resulting postentry population is then given by $\mathbf{w}_\epsilon = (1 - \epsilon)\mathbf{x} + \epsilon \mathbf{y}$. Biological intuition suggests that evolutionary forces select against mutant individuals if and only if the expected payoff of a mutant agent in the postentry population is lower than that of an individual from the original population, i.e.,

$$u\Big(\mathbf{y}, \mathbf{w}_\epsilon^{[k-1]}\Big) < u\Big(\mathbf{x}, \mathbf{w}_\epsilon^{[k-1]}\Big). \quad (3)$$

Hence, a population $\mathbf{x} \in \Delta$ is said to be *evolutionary stable* if inequality (3) holds for any distribution of mutant agents $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$, granted the population share of mutants ϵ is sufficiently small. It can be shown that an ESS is a refinement of the notion of a Nash equilibrium in the sense that every ESS is necessarily a Nash equilibrium (see [22] for pairwise contests and [27] for k-wise contests).

We end this section with a result which provides a characterization of ESSs that will be instrumental in the discussion that follows.

Lemma 1. *Let $\Gamma = (P, S, \pi)$ be a k-player supersymmetric game. Then, $\mathbf{x} \in \Delta$ is an ESS if and only if for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$ there exists $i \in \{0, \ldots, k-1\}$ such that both conditions*

$$u\Big((\mathbf{y} - \mathbf{x})^{[i+1]}, \mathbf{x}^{[k-1-i]}\Big) < 0, \quad (4)$$

$$u\Big((\mathbf{y} - \mathbf{x})^{[\ell+1]}, \mathbf{x}^{[k-1-\ell]}\Big) = 0, \quad \text{for all } 0 \leq \ell < i, \quad (5)$$

are satisfied.

Proof. We will prove a slightly different statement which, however, implies the result. We will show that for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$, inequality (3) holds if and only if there exists $i \in \{0, \ldots, k-1\}$ such that conditions (4) and (5) are satisfied.

Let $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$. By rewriting $\mathbf{w}_\epsilon = \epsilon\,(\mathbf{y} - \mathbf{x}) + \mathbf{x}$ and by exploiting the multilinearity and supersymmetry of u we have that

$$\begin{aligned} u\Big(\mathbf{y}, \mathbf{w}_\epsilon^{[k-1]}\Big) - u\Big(\mathbf{x}, \mathbf{w}_\epsilon^{[k-1]}\Big) &= u\Big(\mathbf{y} - \mathbf{x}, \mathbf{w}_\epsilon^{[k-1]}\Big) \\ &= \sum_{\ell=0}^{k-1} \binom{k-1}{\ell} \epsilon^\ell u\Big((\mathbf{y} - \mathbf{x})^{[\ell+1]}, \mathbf{x}^{[k-1-\ell]}\Big). \end{aligned} \quad (6)$$

Let i be the index of the first nonzero term of the summation in (6) (or, in other words, assume that (5) holds). Then,

$$u\Big(\mathbf{y}, \mathbf{w}_\epsilon^{[k-1]}\Big) - u\Big(\mathbf{x}, \mathbf{w}_\epsilon^{[k-1]}\Big) = \binom{k-1}{i}\epsilon^i u\Big((\mathbf{y}-\mathbf{x})^{[i+1]}, \mathbf{x}^{[k-1-i]}\Big) + O(\epsilon^{i+1}).$$

Note that this quantity is negative for all sufficiently small values of ϵ if and only if $i < k$ and condition (4) holds, from which the result follows. □

3 The Hypergraph Clustering Game

An instance of the hypergraph clustering problem can be described by an edge-weighted hypergraph [28], which is formally defined as a triplet $H = (V, E, \omega)$, where $V = \{1, \ldots, n\}$ is a finite set of *vertices*, $E \subseteq 2^V \setminus \{\emptyset\}$ is the set of (hyper)edges (here, 2^V is the power set of V), and $\omega : E \to \mathbb{R}_+$ is a real-valued function which assigns a positive weight to each edge. Within our clustering framework, the vertices in H correspond to the objects to be clustered, the edges represent (possibly) high-order neighborhood relationships among objects, and the edge-weights reflect similarity among linked objects. Although hypergraphs may have edges of varying cardinality, in this paper we will focus on a particular class of hypergraphs, called k-graphs, whose edges have fixed cardinality $k \geq 2$ (clearly, if $k = 2$ we get back to the standard notion of a graph). Note that, for simplicity, here we restrict ourselves to positive similarities, although the proposed framework can easily be generalized to deal with negative weights as well.

Given a weighted k-graph $H = (V, E, \omega)$, representing an instance of a hypergraph clustering problem, we cast it into a k-player *(hypergraph) clustering game* $\Gamma = (P, V, \pi)$ where the players' pure strategies correspond to the objects to be clustered and the payoff function π is proportional to the similarity of the objects/strategies $(v_1, \ldots, v_k) \in V^k$ selected by the players:

$$\pi(v_1, \ldots, v_k) = \begin{cases} \frac{1}{k!}\omega(\{v_1, \ldots, v_k\}), & \text{if } \{v_1, \ldots, v_k\} \in E, \\ 0, & \text{otherwise.} \end{cases} \tag{7}$$

Here, the constant of proportionality $1/k!$ has been chosen to simplify later algebraic derivations.

Our clustering game will be played within an evolutionary setting wherein the k players, each of which is assumed to play a preassigned strategy, are repeatedly drawn at random from a large population. Here, given a population $\mathbf{x} \in \Delta$, x_j ($j \in V$) represents the fraction of players that is programmed to select j from the objects to be clustered. A dynamic evolutionary selection process, as the one described in the next section, will then make the population $\mathbf{x}$ evolve, according to a Darwinian survival-of-the-fittest principle, in such a way that, eventually, the better-than-average objects will survive and the others will become extinct. It is clear that the whole dynamical process is driven by the payoff function π, which, in our case, has been defined in (7) precisely to favor the evolution of highly coherent objects. Accordingly, the support $\sigma(\mathbf{x})$ of the converged population $\mathbf{x}$ does represent a cluster, the nonnull components of $\mathbf{x}$ providing a measure of the degree of membership of its elements. Indeed, the expected population payoff $u(\mathbf{x}^{[k]})$ can be regarded as a measure of the cluster's internal coherency in terms of the average similarity of the objects forming the cluster, whereas the expected payoff $u(\mathbf{e}^j, \mathbf{x}^{[k-1]})$ of a player selecting object $j \in V$ in $\mathbf{x}$ measures the average similarity of object j with respect to the cluster.

We claim that, within this setting, the clusters of a hypergraph clustering problem instance can be characterized in terms of the ESSs of the corresponding (evolutionary) clustering game, thereby justifying the following definition.

Definition 1 (ESS-Cluster). *Given an instance of a hypergraph clustering problem $H = (V, E, \omega)$, an* ESS-cluster *of H is an ESS of the corresponding hypergraph clustering game.*

For the sake of simplicity, when it will be clear from context the term ESS-cluster will be used henceforth to refer to either the ESS itself, namely, the membership vector $\mathbf{x} \in \Delta$, or to its support $\sigma(\mathbf{x}) = C \subseteq V$.

The motivation behind the above definition resides in the observation that ESS-clusters do incorporate the two basic properties of a cluster, i.e.:

- *Internal coherency.* Elements belonging to the cluster should have high mutual similarities.
- *External incoherency.* The overall cluster internal coherency decreases by introducing external elements.

The rest of this section is devoted to provide support to this claim.

3.1 Internal Coherency

The internal coherency of an ESS-cluster is a direct consequence of the Nash condition (2), which is satisfied by any ESS. Indeed, if $\mathbf{x} \in \Delta$ is an ESS of a clustering game, then from (2) it follows that every object belonging to the cluster, i.e., every object in $\sigma(\mathbf{x})$, has the same average similarity with respect to the cluster, which in turn corresponds to the cluster's overall average similarity. This is formally stated in the following theorem.

Theorem 1. *Let $H = (V, E, \omega)$ be an instance of a hypergraph clustering problem, and $\Gamma = (P, V, \pi)$ the corresponding clustering game. If $\mathbf{x} \in \Delta$ is an ESS-cluster of H, with support $\sigma(\mathbf{x}) = C$, then*

$$u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big) = u\Big(\mathbf{x}^{[k]}\Big), \qquad \text{for all } j \in C. \tag{8}$$

Proof. Since an ESS is a Nash equilibrium, it follows from (2) and by the multilinearity of u that

$$u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big) \leq u\Big(\mathbf{x}^{[k]}\Big) = \sum_{\ell \in V} u\Big(\mathbf{e}^\ell, \mathbf{x}^{[k-1]}\Big)x_\ell = \sum_{\ell \in \sigma(\mathbf{x})} u\Big(\mathbf{e}^\ell, \mathbf{x}^{[k-1]}\Big)x_\ell \qquad \text{for all } j \in V.$$

Since the right-hand side of this equation is a convex linear combination we have that

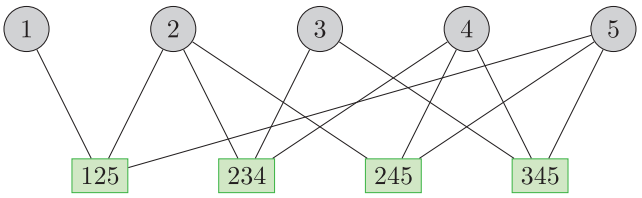

Fig. 1. Example of a three-graph with five nodes (circles) and four edges (rectangles), represented as a bipartite graph. Each edge is connected to the vertices it contains.

$$u\Big(\mathbf{e}^{j^*}, \mathbf{x}^{[k-1]}\Big) \leq \sum_{\ell \in \sigma(\mathbf{x})} u\Big(\mathbf{e}^{\ell}, \mathbf{x}^{[k-1]}\Big) x_\ell \leq u\Big(\mathbf{e}^{j^*}, \mathbf{x}^{[k-1]}\Big) \qquad \text{for all } j \in V,$$

where $j^* \in \arg\max_{j \in \sigma(\mathbf{x})} u(\mathbf{e}^j, \mathbf{x}^{[k-1]})$. Hence,

$$u\Big(\mathbf{e}^{j^*}, \mathbf{x}^{[k-1]}\Big) = u\Big(\mathbf{x}^{[k]}\Big),$$

which holds if and only if (8) holds. □

The internal coherency of an ESS-cluster becomes clearer if we analyze it using a notion from hypergraph theory. Let $H = (V, E, \omega)$ be a (weighted) hypergraph and $C \subseteq V$. We say that C is a *two-cover* of H if for any pair of vertices $\{j, \ell\} \subseteq C$ there exists an edge $e \in E$ such that $\{j, \ell\} \subseteq e \subseteq C$. Note that if H is a graph (i.e., $k = 2$), then two-covers correspond to cliques, namely, to sets of mutually adjacent vertices. To illustrate, in the hypergraph shown in Fig. 1 the sets $\{1, 2, 3, 4, 5\}$ and $\{1, 2, 4, 5\}$ are not two-covers as there is no edge contained in them connecting vertices $\{1, 3\}$ and $\{1, 4\}$, respectively, while the set $\{2, 3, 4, 5\}$ is a two-cover.

The following proposition, which is a weighted counterpart of a result by Frankl and Rödl [29] on unweighted hypergraphs, provides an interesting connection between ESS's and two-covers.

Proposition 1. *Let $H = (V, E, \omega)$ be an instance of a hypergraph clustering problem and $\Gamma = (P, V, \pi)$, the corresponding clustering game. If $\mathbf{x} \in \Delta$ is an ESS-cluster of H, then its support $\sigma(\mathbf{x})$ is a two-cover of H.*

Proof. Let $C = \sigma(\mathbf{x})$ be the support of $\mathbf{x}$ and suppose, by contradiction, that C is not a two-cover. Then, there exist $p, q \in C$ such that for all $e \in E$ either $\{p, q\} \not\subseteq e$ or $e \not\subseteq C$. Let $\mathbf{z}_\epsilon = \epsilon\,(\mathbf{e}^p - \mathbf{e}^q) + \mathbf{x}$, where $0 < \epsilon \leq x_q$. We will show that $u((\mathbf{z}_\epsilon - \mathbf{x})^{[i]}, \mathbf{x}^{[k-i]}) = 0$, for all $i \in \{1, \ldots, k\}$, and this, by Lemma 1, contradicts the fact that $\mathbf{x}$ is an ESS, thereby proving the result.

Indeed, by the multilinearity and the supersymmetry of u and by (7) we have that

$$\begin{aligned} u\Big((\mathbf{z}_\epsilon - \mathbf{x})^{[i]}, \mathbf{x}^{[k-i]}\Big) &= \epsilon^i u\Big((\mathbf{e}^p - \mathbf{e}^q)^{[i]}, \mathbf{x}^{[k-i]}\Big) \\ &= \epsilon^i \sum_{h=0}^{i} \binom{i}{h} (-1)^{i-h} u\Big((\mathbf{e}^p)^{[h]}, (\mathbf{e}^q)^{[i-h]}, \mathbf{x}^{[k-i]}\Big) \\ &= \epsilon^i \sum_{h=0}^{i} \binom{i}{h} (-1)^{i-h} \sum_{(s_1,\ldots,s_{k-i}) \in S^{k-i}} \pi\Big(p^{[h]}, q^{[i-h]}, s_1, \ldots, s_{k-i}\Big) \prod_{\ell=1}^{k-i} x_{s_\ell}, \end{aligned} \tag{9}$$

for all $i \in \{1, \ldots, k\}$. We proceed now by enumerating all possible cases.

Case $i = 1$**:** Since $\mathbf{x}$ is an ESS, it is also a Nash equilibrium. Hence, by (8):

$$u\Big((\mathbf{e}^p - \mathbf{e}^q), \mathbf{x}^{[k-1]}\Big) = 0,$$

and, therefore, $u((\mathbf{z}_\epsilon - \mathbf{x}), \mathbf{x}^{[k-1]}) = 0$.

Case $i > 2$. By (7), we have

$$\pi\Big(p^{[h]}, q^{[i-h]}, s_1, \ldots, s_{k-i}\Big) = 0\,, \quad \text{for all } h \in \{0, \ldots, i\}\,,$$

because $\{p^{[h]}, q^{[i-h]}, s_1, \ldots, s_{k-i}\} \notin E$ and, therefore,

$$u((\mathbf{z}_\epsilon - \mathbf{x})^{[i]}, \mathbf{x}^{[k-i]}) = 0.$$

Case $i = 2$. By (7) and by noting that $\pi(p^{[h]}, q^{[2-h]}, s_1, \ldots, s_{k-2}) = 0$ for $h \in \{0, 2\}$, (9) can be simplified as

$$\begin{aligned} u\Big((\mathbf{z}_\epsilon - \mathbf{x})^{[i]}, \mathbf{x}^{[k-i]}\Big) &= -2\epsilon^2 \sum_{\substack{e \in E \\ \{p,q\} \subseteq e}} \omega(e) \prod_{\ell \in e} x_\ell \\ &= -2\epsilon^2 \sum_{\substack{e \subseteq C \\ \{p,q\} \subseteq e}} \omega(e) \prod_{\ell \in e} x_\ell, \end{aligned}$$

which is zero because, by hypothesis, there is no edge contained in C which also contains both p and q. □

Intuitively, the previous result shows that two objects cannot belong to (the support of) an ESS-cluster if there is no similarity relationship between them within the cluster. This is a minimal property that a cluster should satisfy in order to guarantee some form of internal coherency.

3.2 External Incoherency

In addition to the internal coherency property described above, we now show that ESS-clusters also satisfy a property of external incoherency. This follows, in the first place, from the Nash condition (2) that we already used to prove internal coherency. In fact, according to (2), every object external to an ESS-cluster C has an average similarity with respect to C that cannot exceed the cluster's overall similarity. More formally, if $\mathbf{x} \in \Delta$ is a Nash equilibrium with support $\sigma(\mathbf{x}) = C$, we have

$$u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big) \leq u\Big(\mathbf{x}^{[k]}\Big), \qquad \text{for all } j \notin C.$$

However, the Nash condition alone is not enough, as there may still be cases where the average similarity of an external object equals the cluster's overall similarity, thereby violating the external incoherency criterion. As it turns out, to some extent, this cannot be the case with an ESS, thanks to its additional stability properties.

Theorem 2. *Let $H = (V, E, \omega)$ be an instance of a hypergraph clustering problem and $\Gamma = (P, V, \pi)$ the corresponding clustering game. Then, $\mathbf{x} \in \Delta$ is an ESS-cluster of H if and only if for any $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$ and all sufficiently small positive values of ϵ the following inequality holds:*

$$u\Big(\mathbf{w}_\epsilon^{[k]}\Big) < u\Big(\mathbf{x}^{[k]}\Big),$$

where $\mathbf{w}_\epsilon = (1 - \epsilon)\mathbf{x} + \epsilon\,\mathbf{y}$.

Proof. We will prove a slightly different statement which, in conjunction with Lemma 1, implies the result. We will show that for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$, there exists $i \in \{0, \ldots, k-1\}$ such that conditions (4) and (5) hold if and only if $u(\mathbf{w}_\epsilon^{[k]}) < u(\mathbf{x}^{[k]})$ for all sufficiently small values of ϵ.

Let $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$. By rewriting $\mathbf{w}_\epsilon = \epsilon\,(\mathbf{y} - \mathbf{x}) + \mathbf{x}$ and by exploiting the multilinearity and super-symmetry of u we have

$$u\Big(\mathbf{w}_\epsilon^{[k]}\Big) - u\Big(\mathbf{x}^{[k]}\Big) = \left\{\sum_{\ell=0}^{k} \binom{k}{\ell} \epsilon^\ell u\Big((\mathbf{y}-\mathbf{x})^{[\ell]}, \mathbf{x}^{[k-\ell]}\Big)\right\} - u\Big(\mathbf{x}^{[k]}\Big) = \sum_{\ell=0}^{k-1} \binom{k}{\ell+1} \epsilon^{\ell+1} u\Big((\mathbf{y}-\mathbf{x})^{[\ell+1]}, \mathbf{x}^{[k-1-\ell]}\Big). \tag{10}$$

Let i be the index of the first nonzero term of the summation in (10) or, in other words, assume that (5) holds. Then,

$$u\Big(\mathbf{w}_\epsilon^{[k]}\Big) - u\Big(\mathbf{x}^{[k]}\Big) = \binom{k}{i+1} \epsilon^{i+1} u\Big((\mathbf{y}-\mathbf{x})^{[i+1]}, \mathbf{x}^{[k-1-i]}\Big) + O(\epsilon^{i+2}).$$

Note that this quantity is negative for all sufficiently small values of ϵ if and only if $i < k$ and condition (4) holds, from which the result follows. □

The previous theorem asserts that whenever we try to deviate from an ESS-cluster $\mathbf{x} \in \Delta$, e.g., by adding an external element to its support, the cluster's overall average similarity drops, provided that deviation is not too large. This not only guarantees a form of external incoherency, but also provides support to the claim that the components of $\mathbf{x}$ reflect the degree of cluster membership.

Observe that when the number of players k equals 2, i.e., in the presence of pairwise similarities, our notion of ESS-cluster coincides with that of a dominant set [20], [21], which is a generalization of a maximal clique to the case of edge-weighted graphs. In this case, a stronger notion of external incoherency holds, which asserts that no dominant set can be a subset of another. In the case of higher order similarities, however, there is no theoretical guarantee that the support of an ESS is not contained in that of another one. Indeed, in [27], it is shown that such solution patterns might possibly appear in general games with more than two players (i.e., $k > 2$). To evaluate how often this happens, we conducted a systematic experimental study over all the hypergraphs used in the experiments described in Section 5. Overall, we got 1,510 hypergraphs ranging from 60 to 5,000 vertices. By starting the dynamics described in the next section with 50 randomly chosen initial vectors, we tried to enumerate (at least partially) the ESS-clusters of each hypergraph. Upon convergence, we checked whether the support of any two distinct solutions found had nested ESS-supports. In all 75,500 trials this never happened, thereby suggesting the claim that the existence of nested ESS-clusters appears to not be an issue in practice.

4 Evolution Toward a Cluster

In this section, we address the issue of determining an ESS-cluster for a given instance of a hypergraph clustering problem. Unfortunately, this turns out to be a computationally hard problem [30], [31], but good heuristics do exist. Indeed, we show below that the ESSs of a clustering game are in one-to-one correspondence with (strict) local solutions of a nonlinear optimization problem, thereby allowing the use of standard optimization techniques.

Theorem 3. *Let $H = (V, E, \omega)$ be a hypergraph clustering problem, $\Gamma = (P, V, \pi)$ the corresponding clustering game, and $f(\mathbf{x})$ a function defined as*

$$f(\mathbf{x}) = u\Big(\mathbf{x}^{[k]}\Big) = \sum_{e \in E} \omega(e) \prod_{j \in e} x_j. \tag{11}$$

Nash equilibria of Γ are in one-to-one correspondence with the critical points[2] of $f(\mathbf{x})$ over Δ, while ESSs of Γ are in one-to-one correspondence with strict local maximizers of $f(\mathbf{x})$ over Δ.

Proof. To prove the first part of the theorem, recall that by the definition of critical (KKT) point [32], for all $j \in V$, there exist $\mu_j \geq 0$ and $\lambda \in \mathbb{R}$ such that

$$\nabla f(\mathbf{x})_j + \mu_j - \lambda = \frac{1}{k} u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big) + \mu_j - \lambda = 0 \quad \text{and} \quad \mu_j x_j = 0,$$

where ∇ is the gradient operator. By multiplying both sides of these equations by the corresponding x_j and summing up, it follows that $\lambda = u(\mathbf{x}^{[k]})/k$ and hence, from the nonnegativity of the μ_js, $u(\mathbf{e}^j, \mathbf{x}^{[k-1]}) \leq u(\mathbf{x}^{[k]})$ for all $j \in V$.

To prove the second part, note that by definition a strict local maximizer of f over Δ (say, $\mathbf{x}$) satisfies $u(\mathbf{y}^{[k]}) = f(\mathbf{y}) < f(\mathbf{x}) = u(\mathbf{x}^{[k]})$, for all $\mathbf{y} \in \Delta \setminus \{\mathbf{x}\}$ close enough to $\mathbf{x}$, which is in turn equivalent to the condition of Theorem 2 for all sufficiently small values of ϵ. □

The problem of extracting ESS-clusters can thus be cast into the problem of finding a strict local solutions of (11) in Δ. We will address this optimization task using a well-known result due to Baum and Eagon [33], who introduced a wide class of nonlinear transformations in probability domain. Their result generalizes an earlier one by Blakley [34], who discovered similar properties for certain homogeneous quadratic transformations. The next theorem introduces what is known as the Baum-Eagon inequality.[3]

Theorem 4 (Baum-Eagon). *Let $Q(\mathbf{x})$ be a homogeneous polynomial in the variables x_j with nonnegative coefficients, and let $\mathbf{x} \in \Delta$. Define the mapping $\mathbf{z} = \mathcal{M}(\mathbf{x})$ from Δ to itself as follows:*

$$z_j = x_j \frac{\partial Q(\mathbf{x})}{\partial x_j} \Big/ \sum_{\ell=1}^{n} x_\ell \frac{\partial Q(\mathbf{x})}{\partial x_\ell}, \qquad j = 1, \ldots, n. \tag{12}$$

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

2. A point $\mathbf{x}$ is said to be a critical (or a KKT) point of an optimization problem if it satisfies the first-order necessary conditions for being a solution [32].

3. Indeed, the original Baum-Eagon inequality is more general than presented here as it deals with a maximization problem over a product of simplices.

Although this result applies to homogeneous polynomials, in a subsequent paper Baum and Sell [35] proved that Theorem 4 still holds in the case of arbitrary, nonhomogeneous polynomials and further extended the result by showing that $\mathcal{M}$ increases Q homotopically, which means that for all $0 \leq \eta \leq 1$, $Q(\eta\mathcal{M}(\mathbf{x}) + (1-\eta)\mathbf{x}) \geq Q(\mathbf{x})$ with equality if and only if $\mathcal{M}(\mathbf{x}) = \mathbf{x}$.

Another way of looking at Theorem 4 is from the standpoint of dynamical systems theory [36], [37]. The nonlinear operator $\mathcal{M}$ defines, in fact, a discrete dynamical system and it is therefore of particular interest to study how it behaves in the vicinity of its equilibrium points. In the theory of dynamical systems, this is formalized by the concept of stability. An equilibrium point $\mathbf{x}$ is said to be *stable* if, whenever started sufficiently close to $\mathbf{x}$, the system will remain near to $\mathbf{x}$ for all future times. A stronger property, which is even more desirable, is that the equilibrium point $\mathbf{x}$ be *asymptotically stable*, meaning that $\mathbf{x}$ is stable and in addition is a *local attractor*, i.e., when initiated close to $\mathbf{x}$, the system tends toward $\mathbf{x}$ as time increases. One of the most fundamental tools for establishing the stability of a given equilibrium point is known as Lyapunov's direct method. It involves seeking a so-called *Lyapunov* function, i.e., a continuous real-valued function defined in state space which is nondecreasing along any trajectory. Of particular interest are *strict* Lyapunov functions which are, instead, strictly increasing along nonconstant trajectories. Accordingly, Theorem 4 essentially states that the polynomial Q is a Lyapunov function for the discrete-time dynamical system defined by $\mathcal{M}$.

The Baum-Eagon inequality therefore provides an effective iterative means for maximizing polynomial functions in probability domains, and in fact it has served as the basis for various statistical estimation techniques developed within the theory of probabilistic functions of Markov chains [38]. It has also been employed for studying the dynamical properties of relaxation labeling processes [39]. Note that, even in the presence of negative coefficients, it is still possible to use the Baum-Eagon theorem, and hence the corresponding dynamical system by applying a simple transformation to the original polynomial which does preserve the original solutions. This could be useful, for example, when the edge-weights in the hypergraph encode both similarity and dissimilarity information.

Now, let us go back to our clustering problem. Note that the function f defined in (11) is precisely a homogeneous polynomial with nonnegative coefficients and hence the Baum-Eagon theorem applies. In this case, we have

$$\frac{\partial f(\mathbf{x})}{\partial x_j} = \frac{1}{k} u\Big(\mathbf{e}^j, \mathbf{x}^{[k-1]}\Big), \qquad j = 1 \ldots n,$$

which yields

$$\sum_{\ell=1}^{n} x_\ell \frac{\partial f(\mathbf{x})}{\partial x_\ell} = \frac{1}{k} u\Big(\mathbf{x}^{[k]}\Big)$$

so that the proposed discrete-time dynamics to extract an ESS-cluster takes the following form:

$$x_j(t+1) = x_j(t) \frac{u\Big(\mathbf{e}^j, \mathbf{x}(t)^{[k-1]}\Big)}{u\Big(\mathbf{x}(t)^{[k]}\Big)}, \qquad j = 1 \ldots n. \tag{13}$$

This dynamic can be given a natural evolutionary interpretation, and in fact generalizes a classical formalization of natural selection processes in two-player evolutionary game theory [22], [23], known as "replicator dynamics." To see this, recall that $u(\mathbf{e}^j, \mathbf{x}^{[k-1]})$ represents the expected payoff of an i-strategiest in population $\mathbf{x}$, while $u(\mathbf{x}^{[k]})$ represents the expected payoff over the entire population. Hence, during the evolution of (13), better-than-average strategies, i.e., those satisfying $u(\mathbf{e}^j, \mathbf{x}^{[k-1]}) > u(\mathbf{x}^{[k]})$, will spread in the population while the others will become extinct, therefore giving rise to a Darwinian selection process.

From Theorem 4, we can assert that f is a strict Lyapunov function for this dynamical system and this, in conjunction with the fact that every ESS-cluster is a strict local maximizer of f in Δ, proves the following theorem which is an obvious consequence of Lyapunov's theorem of asymptotically stability [36], [37].

Theorem 5. *A point $\mathbf{x} \in \Delta$ is an ESS-cluster of an instance of a hypergraph clustering problem if and only if it is an asymptotically stable equilibrium point (and, hence, a local attractor) for the nonlinear dynamics (13).*

In practical applications, without heuristic information about the optimal solution, it is customary to start out the dynamics from the barycenter of the simplex, i.e., from the vector $\mathbf{x}(0) = (\frac{1}{n}, \ldots, \frac{1}{n})^\top \in \Delta$, which is the uniform distribution over the set of vertices V. This choice ensures that no particular solution is favored. Moreover, the dynamics (13) satisfies the invariant property $\sigma(\mathbf{x}(t)) \subseteq \sigma(\mathbf{x}(0))$ for any time $t > 0$. Hence, in order to allow any vertex $i \in V$ to potentially take part of a solution, we need to select an initial state $\mathbf{x}(0)$ with full support, i.e., $\sigma(\mathbf{x}(0)) = V$. In particular, if the numerator of (13) is positive for all $j \in \sigma(\mathbf{x}(0))$, then $\sigma(\mathbf{x}(t)) = \sigma(\mathbf{x}(0))$ for all *finite* values of $t \geq 0$ and only asymptotically might we possibly have $\sigma(\mathbf{x}^*) \subseteq \sigma(\mathbf{x}(0))$, $\mathbf{x}^*$ being the limit point of the trajectory, namely, $\mathbf{x}^* = \lim_{t\to\infty} \mathbf{x}(t)$. This fact suggests that given a solution $\mathbf{x}(T)$ obtained after $T < \infty$ steps of (13), we need to threshold its components in order to get the support of the corresponding ESS-cluster. Observe also that the components of an ESS-cluster $\mathbf{x}$ provide information about the degree of membership of each element to the cluster (which could be useful, e.g., to extract a representative of the cluster found).

Unlike standard partitional techniques, our approach involves extracting one cluster at a time, much in the same spirit as [20], [40], [41]. Depending on the application at hand, one might want to obtain either overlapping or nonoverlapping clusters. In the latter case, a simple, yet effective "peel-off" strategy, which has also been used in the experiments reported below, can be as follows: 1) Find an ESS-cluster with dynamics (13), 2) remove its vertices from the hypergraph, 3) reiterate on the remaining vertices. Alternatively, in order to extract overlapping groups one needs to enumerate the ESS-clusters. In this paper, we do not address this issue, but we mention that a possible strategy to accomplish this has been proposed in [42], although restricted to the standard pairwise case.

Finally, as pointed out in [35], note that our dynamics (13) contrasts sharply with gradient methods, for which an

increase in the objective function is guaranteed only when infinitesimal steps are taken, and determining the optimal step size entails computing higher order derivatives. We add that performing gradient ascent in Δ requires some projection operator to ensure that the constraints not be violated, and this might cause some problems for points lying on the boundary [43], [44]. In (13), instead, a computationally simple row normalization is required. Overall, the complexity of finding an ESS-cluster with our algorithm turns out to be $O(\rho|E|)$, where $|E|$ is the number of edges of the hypergraph and ρ is the average number of iteration needed to converge. In the experiment reported below, ρ never exceeded 100. More efficient algorithms to extract ESS-clusters can well be developed, e.g., along the lines suggested in [45] and [46] for quadratic optimization.

5 Experiments

To test the effectiveness of the proposed approach, we conducted experiments on synthetic as well as real-world data. We compared our approach against two of the most powerful hypergraph clustering algorithms available in the literature, namely, the Clique Averaging algorithm (CAVERAGE) of Agarwal et al. [2], and the Supersymmetric Nonnegative Tensor Factorization (SNTF) of Shashua et al. [4]. Note that in [2] CAVERAGE was shown to outperform consistently several existing hypergraph clustering techniques such as Clique Expansion combined with Normalized cuts [47], Gibson's Algorithm under sum and product model [48], the two-phase multilevel algorithm kHMeTiS [12], and therefore we decided not to include them in our experimental comparisons. Note also that, unlike CAVERAGE, which resorts to a pairwise approximation of the high-order similarity function, SNTF works directly on the hypergraph as we do.

Since both CAVERAGE and SNTF, in contrast to our method, require as a parameter the number of clusters K, we run them with values of $K \in \{K^* - 1, K^*, K^* + 1\}$, where K^* denotes the correct number of clusters. As, in practical application, the optimal number of clusters is not known in advance, this allowed us to assess the robustness of the approaches in the presence of under and overestimation of the correct number of clusters.[4] As concerns the other free parameters of all competing algorithms, they were optimally tuned using a small validation set which consisted of a set of labeled observations sampled from the same distribution as that used in the testing phase. As for our algorithm, we used the peel-off strategy described in the previous section. The quality of the clusterings found by the different algorithms was evaluated in terms of classification error with minimum-cost bipartite matching except for the experiment in Section 5.3, where a different evaluation protocol has been adopted (see the description in the section).

We run all the experiments on an AMD Sempron 3 Ghz computer equipped with a 4 Gb RAM. In the case of CAVERAGE and SNTF, we used the original codes as provided by the authors (Matlab and C++ implementations, respectively). For our algorithm, we used a nonoptimized Matlab implementation. As for running time, we report that our algorithm typically took 100 seconds or so to converge, CAVERAGE was an order of magnitude faster, while SNTF was an order of magnitude slower. This is indeed to be expected as CAVERAGE, unlike our algorithm and SNTF, transforms the original hypergraph into a graph at the outset, thereby greatly reducing the complexity of the problem. On the other hand, like our algorithm, SNTF does not resort to any graph approximation, but, by optimizing a single variable at a time, it has a substantially larger computational complexity.

5.1 Line Clustering

Here, we consider the problem of clustering lines in spaces of dimension greater than two, i.e., given a set of points in $\mathbb{R}^d$, the task is to extract subsets of collinear points. This is a typical example where classical pairwise approaches cannot work because any pair of points defines a straight line, and hence higher order similarity relations are needed (see, e.g., [2]). An obvious ternary similarity measure for this clustering problem can be defined as follows: Given a triplet of points $\{i, j, k\}$ and its best fitting line ℓ, we calculate the mean distance $d(i, j, k)$ between each point and ℓ, and then we obtain a similarity function using a standard Gaussian kernel: $\omega(\{i, j, k\}) = \exp(-\beta d(i, j, k)^2)$, where β is a properly tuned precision parameter.

In order to assess the robustness of the competing approaches to both local and global perturbations, we conducted two kinds of experiments. In the first set of experiments, we generated a few lines (from 3 to 5) in a 5D space $[-2, 2]^5$. Each line consisted of 20 points, which were locally perturbed using a varying amount of Gaussian noise, namely, from $\sigma = 0$ to $\sigma = 0.08$ (see Fig. 2a for a specific example). Figs. 2b, 2c, and 2d show the results obtained by the competing algorithms in terms of classification error with three, four, and five lines, respectively, as a function of the noise level. Each plot shows the average performance obtained over 30 randomly generated instances together with the corresponding standard deviations.

In the first place, note that our algorithm performs essentially as well as the best performing parameterization of SNTF on all instances with a level of noise not exceeding 0.04. As for CAVERAGE, note that even using the correct number of clusters $K = K^*$, its performances gradually deteriorate as the number of lines is increased. In all cases, both SNTF and CAVERAGE are systematically outperformed by our algorithm when they are run with a nonoptimal value of K. We also observe that when $K = K^* - 1$, the error of CAVERAGE and SNTF is expected to decrease significantly as we increase K^*, e.g., when we use five instead of four lines, while this does not happen, thereby suggesting that they do not achieve the best possible result here. Further, as expected, the influence of local noise on their performance is typically negligible. Indeed, this makes intuitively sense as, once they stick to a partition of the original input data, it is unlikely that the result will change drastically under moderate local perturbations. On the other hand, our approach appears to be slightly more vulnerable to local perturbations as

4. Note that running any clustering algorithm with $K < K^*$ prevents it from achieving perfect results. However, we think that the experiments presented with $K = K^* - 1$ do indeed provide some interesting information concerning the algorithms' behavior.

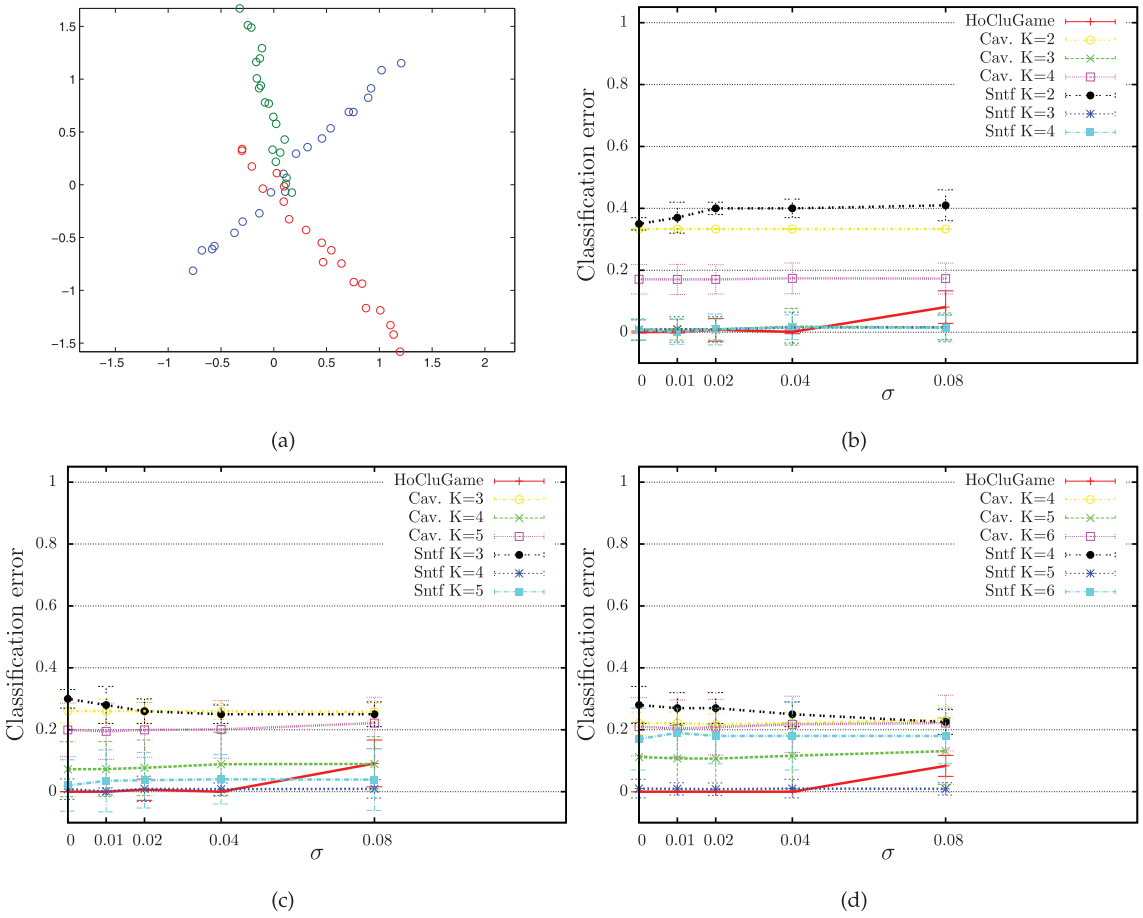

Fig. 2. Results of clustering three, four, and five lines perturbed locally with increasing levels of Gaussian local noise ($\sigma = 0, 0.01, 0.02, 0.04, 0.08$). (a) Example of three 5D lines (projected in 2D), perturbed with $\sigma = 0.04$. (b) Three lines. (c) Four lines. (d) Five lines.

points deviating too much from a cluster's average collinearity will get excluded, by construction, as they undermine internal coherency.

The second series of experiments aimed at assessing robustness to clutter (global noise). To this end, we randomly generated a few lines (in our experiments, from 2 to 4) in the 5D hypercube $[-2, 2]^5$, and then added from 0 to 40 clutter points uniformly drawn from the hypercube (see Fig. 3a for a specific example). In order to make the setting more realistic, we also slightly perturbed the original lines using a local Gaussian noise with standard deviation 0.01. As in the previous set of experiments, each generated line consisted of 20 points.

Figs. 3b, 3c, and 3d show the results obtained by all algorithms as a function of clutter. As can be clearly seen, our algorithm substantially outperformed both CAVERAGE and SNTF even when they were fed by the correct number of clusters K^*, and it worked almost perfectly irrespective of the clutter level. Note also that both competitors achieved better performances when $K > K^*$, and this is intuitively clear as the only way to get rid of clutter points is to group them into additional (garbage) clusters. Nevertheless, due to the intrinsic unstructured nature of clutter points, they typically did not get assigned to the garbage class, but, instead, were associated to the original groups, thereby making the performance of CAVERAGE and SNTF poorer and poorer as clutter increases.

5.2 Plane Clustering

In this section, we take a step further and increase the complexity of the previous series of experiments by clustering 3D points into planes rather than lines. Since any three points uniquely identify a plane, here we need to use (at least) a quaternary similarity measure. In our experiments, we adopted essentially the same measure used above, namely, we measured the deviation from coplanarity of four points $\{i, j, k, \ell\}$ by using their mean distance $d(i, j, k, \ell)$ from the best fitting plane, and then transformed this into a similarity measure using a Gaussian kernel.

In a first series of experiments we assessed the robustness of the different approaches to local noise by randomly generating datasets with two, three, and four planes in the cube $[-2, 2]^3$. This time, to make the problem more challenging, we forced the generated planes to pass through the origin, thereby increasing the possibility of classification errors (especially around the origin). Each plane consisted of 15 points, each of which was perturbed according to increasing levels of Gaussian noise, from $\sigma = 0$ to $\sigma = 0.08$. Fig. 4a shows an example with three planes.

Figs. 4b, 4c, and 4d report the classification error obtained by the three algorithms on 10 different trials as a function of noise level (with corresponding standard deviations). From the results obtained, it is clear that this problem is harder than the previous one. While our

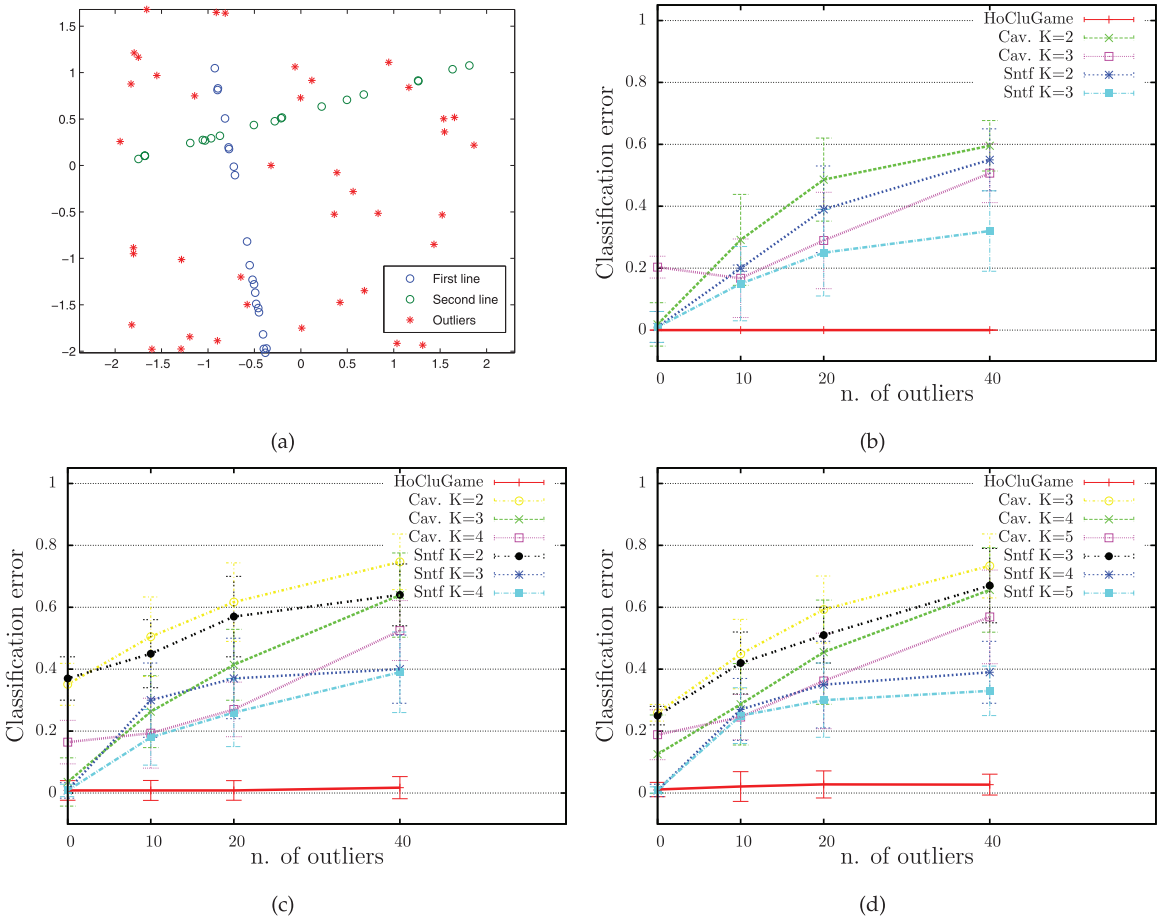

Fig. 3. Results of clustering two, three, and four lines with an increasing number of clutter points $(0, 10, 20, 40)$. (a) Example of two 5D lines (projected in 2D) with 40 clutter points. (b) Two lines. (c) Three lines. (d) Four lines.

approach is outperformed in the two-planes experiment by both CAVERAGE and SNTF, especially in the presence of substantial noise, it compares favorably as the level of complexity increases, exhibiting dramatically better results with four planes. As in the previous set of experiments, note that when the parameter K is less than the correct one (K^*), then both CAVERAGE and SNTF perform poorly.

As before, with the second series of experiments we aimed at assessing robustness against global noise. After generating (slightly perturbed) sets of 15 coplanar points in $[-2, 2]^3$, we added up to 40 uniformly generated clutter points (see Fig. 5 for an example). The results obtained using two, three, and four planes are reported in Figs. 3b, 3c, and 3d. As in the line clustering experiment, our approach dramatically outperforms the competitors due to their intrinsic inability to get rid of clutter. Also, unlike the previous experiment, our algorithm appears to be less sensitive to the level of clutter than CAVERAGE and SNTF.

5.3 Model-Based 3D Point-Pattern Matching

We present here a different type of experiment, which highlights the advantages of our approach over the existing partition-based ones. We consider the problem of finding in a scene (possibly multiple) copies of a reference 3D model subject to a similarity transformation (i.e., rescaling + rotation + translation). Here, both the model and the scene are represented as clouds of 3D points. Motivated by the approach described in [49] (which deals with pairwise relations only), here we tackle this problem from a hypergraph clustering perspective.

Let $\mathcal{M}$ be a set of 3D points representing the model to be found and let $\mathcal{S}$ be a set of 3D points representing the scene. We denote by $\mathcal{A}$ the set of all possible pointwise correspondences between model and scene points, i.e., $\mathcal{A} = \mathcal{M} \times \mathcal{S}$. Given a set of three correspondences $e = \{(\mathbf{m}_1, \mathbf{s}_1), (\mathbf{m}_2, \mathbf{s}_2), (\mathbf{m}_3, \mathbf{s}_3)\} \subseteq \mathcal{A}$, we compute the similarity transformation T which minimizes the least-squares error $d(e) = \sum_{i=1}^{|e|} \|T(\mathbf{m}_i) - \mathbf{s}_i\|^2$ using the Horn method [50]. Consider now the hypergraph $H = (\mathcal{A}, \mathcal{E}, \phi)$ where the set of vertices is given by the set of correspondences $\mathcal{A}$, the set of hyperedges $\mathcal{E}$ consists of subsets of $\mathcal{A}$ of cardinality 3, and $\phi(e)$ is the edge-weight function defined as $\phi(e) = \exp(-\beta d(e)^2)$, where $\beta > 0$ is a precision parameter. Intuitively, the function $\phi(e)$ can be regarded as a *compatibility function* encoding the likelihood of the correspondences in e to be related by the same similarity transformation. According to our framework, an ESS-cluster C of H is a subset of correspondences in $\mathcal{A}$ exhibiting both internal coherence and external incoherence. Therefore, all correspondences in C are mutually highly compatible. This, by definition of ϕ, implies that all correspondences in C are related by the same similarity transformation between the model and the scene. Hence, C is a good candidate for being a potential match (i.e., a set of

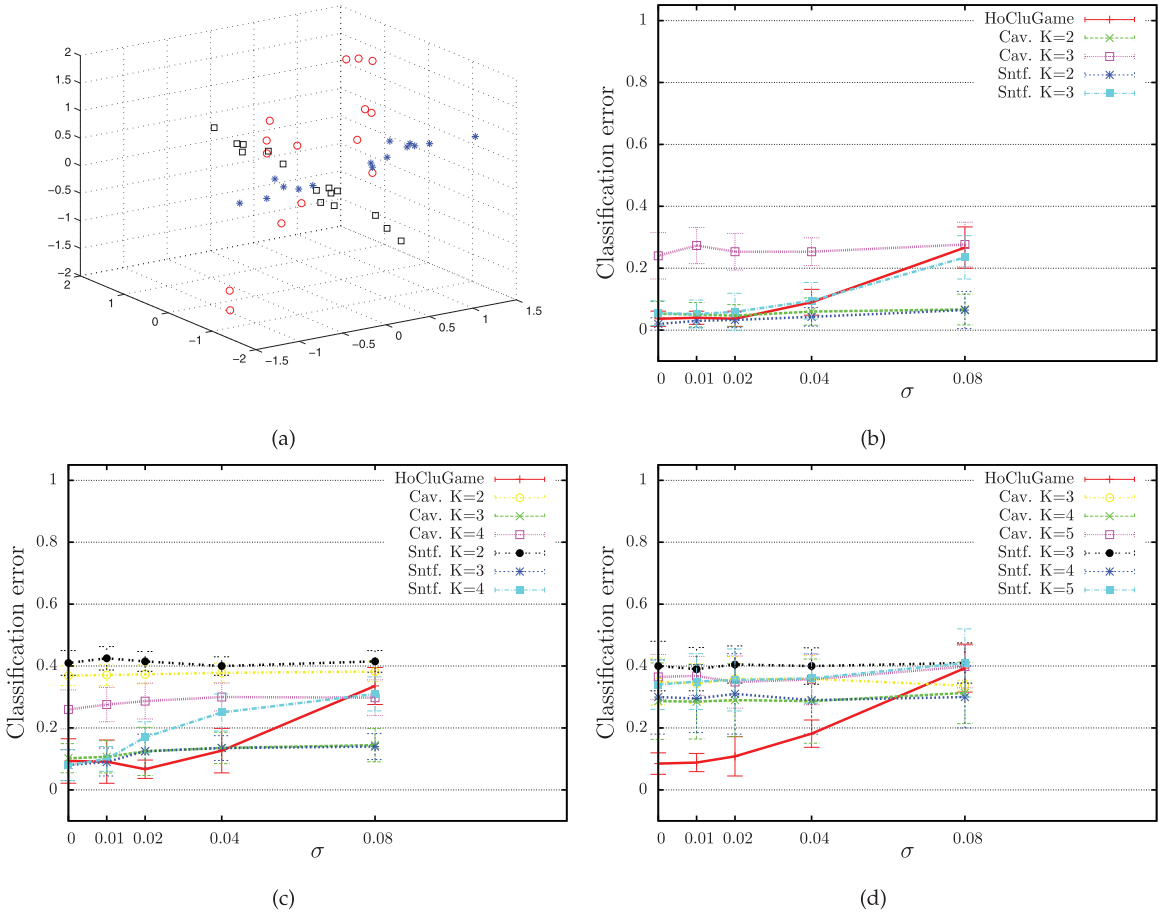

Fig. 4. Results of clustering three, four, and five planes through the origin perturbed with increasing levels of Gaussian local noise ($\sigma = 0, 0.01, 0.02, 0.04, 0.08$). (a) Example of three 3D planes, perturbed with $\sigma = 0.01$. (b) Two planes. (c) Three planes. (d) Four planes.

correspondences) providing a detection of the model in the scene, which is invariant to a similarity transformation. This motivates the use of our game-theoretic approach in order to address this matching problem from a clustering perspective. Note that this problem is particularly challenging, for only a small fraction of the correspondences in $\mathcal{A}$ will be part of a solution, the rest being outliers. Indeed, for example, if we consider a scene $\mathcal{S}$ containing any number of distinct instances of a model $\mathcal{M}$, then only a small share of *at most* $|\mathcal{M}|^{-1}$ correspondences appearing in $\mathcal{A}$ do belong to the solution.

We tested our approach on different artificial datasets. Each dataset is characterized by a reference model consisting of 30 random 3D points and a scene which contains up to three instances of the model. Each model instance in the scene is equivalent to the original one modulo a random similarity transformation. Moreover, a random subset of points of each copy of the model has been dropped (0-20 percent of points) in order to introduce structural noise. Consistently with the previous experiments, we considered two types of settings to assess the robustness of the algorithms to local and global noise. In the first set, we employed a Gaussian perturbation of the points in the scene, whereas in the second one 3D points (clutter points) were randomly added to the scene. For each different combination of number of model instances, noise type and noise level we generated 20 random datasets. We refer to Figs. 6a and 7a for examples of datasets with three model instances affected by local and global noise, respectively.

The hypergraphs we got in the various experimental settings were considerably large. Indeed, the number of vertices (i.e., potential correspondences) varied between 1,000 and 5,000 and the number of edges (i.e., triplets of correspondences) ranged approximately between 10^8 and 10^{10}. In order to reduce the size of the edge set, we adopted a sampling strategy aimed at efficiently excluding triplets that cannot belong to a good match. This allowed us to limit the number of edges to a maximum number of 25,000 edges.

The evaluation protocol used to assess the quality of the results is given as follows: First, we clustered the hypergraphs, thus obtaining a set of potential matches. Then, by means of the Horn method, we estimated a similarity transformation from the pointwise correspondences in each cluster. This yielded a set of m transformations $\{T_j\}_{j=1}^{m}$, which were used to determine the correspondences between the scene points and the model points according to the projection error. Specifically, let $d_{ijt} = \|T_t(\mathbf{m}_i) - \mathbf{s}_j\|$ be the distance between scene point $\mathbf{s}_j$ and model point $\mathbf{m}_j$ mapped according to transformation T_j and consider $(j^*, t^*) \in \arg\min_{(j,t)} d_{ijt}$. Then we decided to leave scene point $\mathbf{s}_i$ unassigned if $d_{ij^*t^*} > \tau$ (i.e., the point did not belong to the model) for some fixed threshold $\tau > 0$, while it was assigned to point $\mathbf{m}_{j^*}$ otherwise. Let $\mathcal{R} \subseteq \mathcal{A}$ be the set of

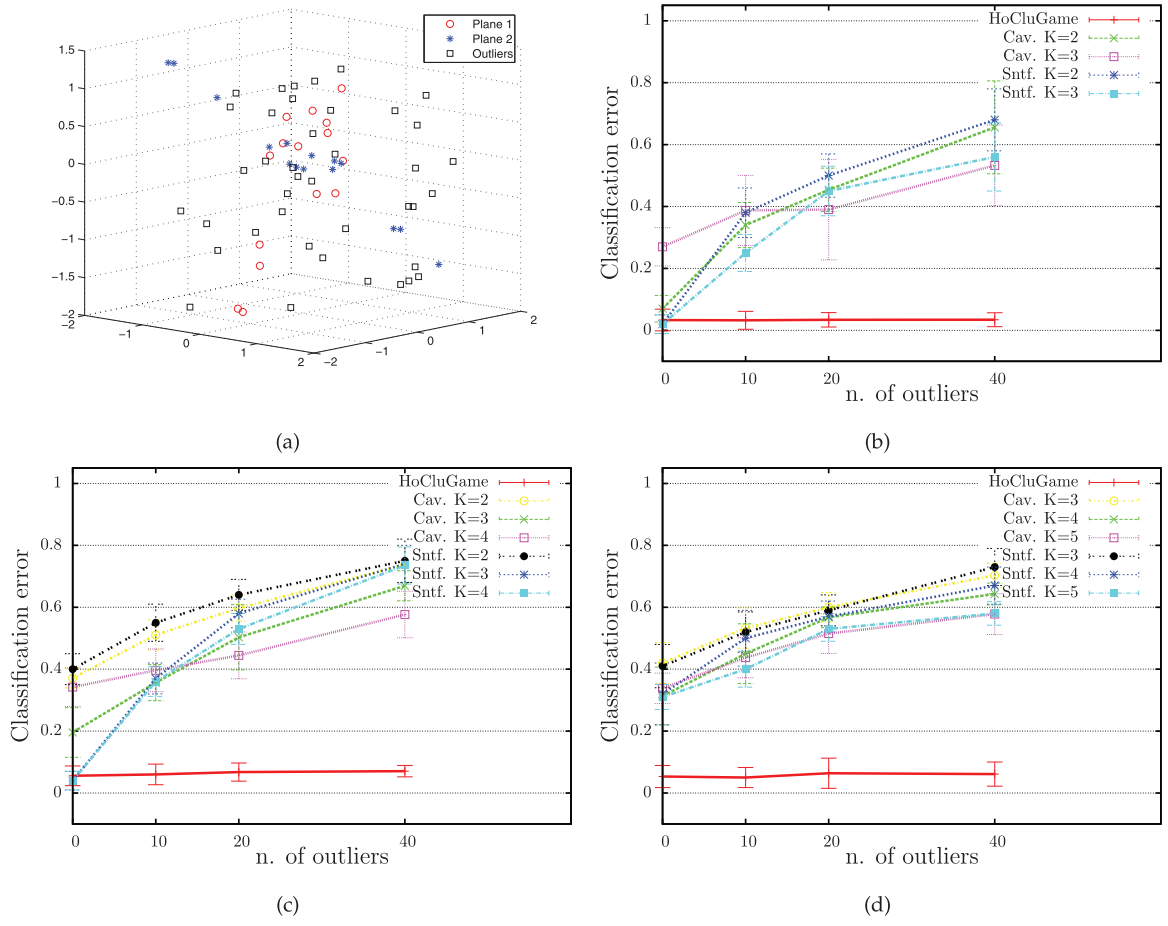

Fig. 5. Results of clustering two, three, and four planes through the origin with an increasing number of outliers $(0, 10, 20, 40)$. (a) Example of two 3D planes with 40 outliers. (b) Two planes. (c) Three planes. (d) Four planes.

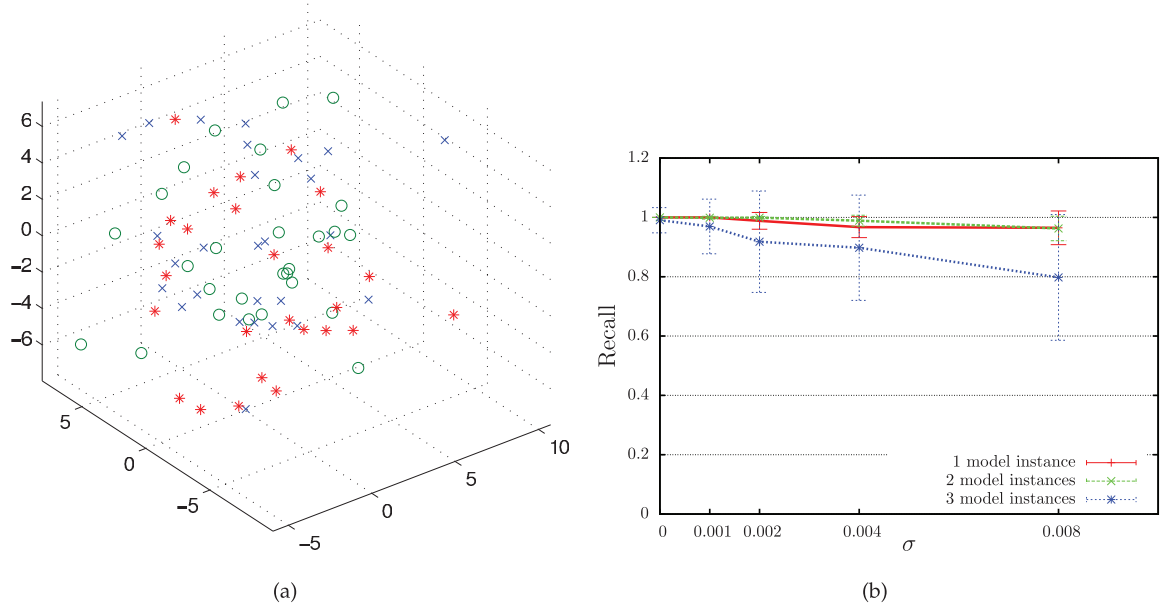

Fig. 6. Results of the experiments on model-based 3D point-pattern matching with one, two, and three model instances perturbed with increasing levels of Gaussian local noise ($\sigma = 0, 0.001, 0.002, 0.004, 0.008$). (a) Example of a model-based 3D point-pattern matching problem instance with three model instances perturbed with $\sigma = 0.004$. (b) Results obtained by our approach. Note that CAVERAGE and SNTF, which do not appear in the plots, obtained recall below 10 percent.

assignments obtained according to this procedure and let $\mathcal{G} \subseteq \mathcal{A}$ be the set of ground-truth assignments of scene points to model points. We evaluated the quality of the obtained result in terms of the share of ground-truth assignments that have been correctly recovered (recall), i.e., $|\mathcal{G} \cap \mathcal{R}|/|\mathcal{G}|$.

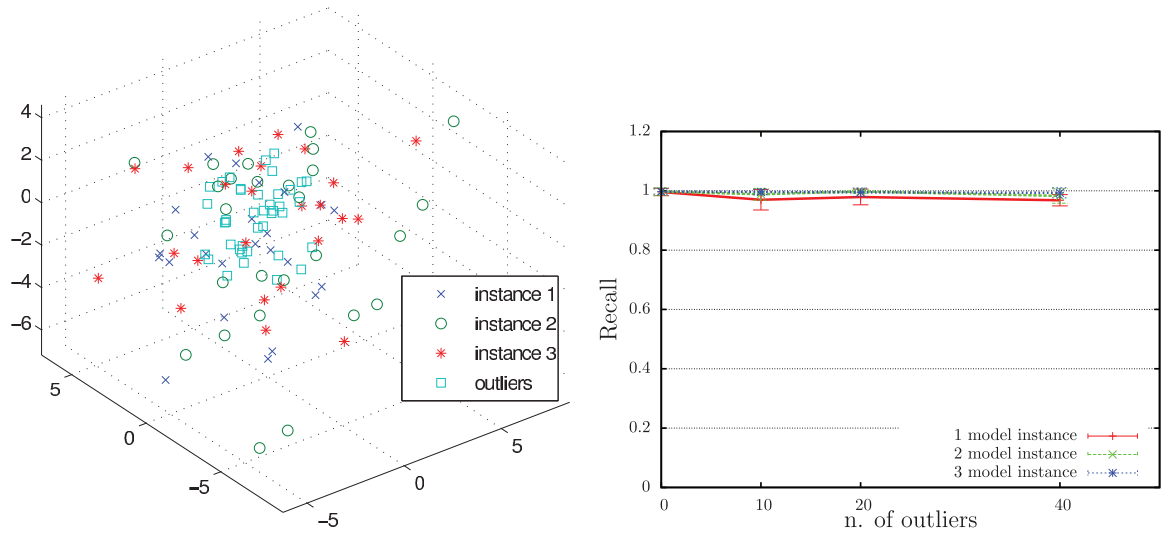

Fig. 7. Results of the experiments on model-based 3D point-pattern matching with one, two, and three object instances with a level of Gaussian local noise of $\sigma = 0.001$ and increasing number of clutter points $(0, 10, 20, 40)$. (a) Example of a model-based 3D point-pattern matching problem instance with 40 clutter points. (b) Results obtained by our approach. Note that CAVERAGE and SNTF, which do not appear in the plots, obtained recall below 10 percent.

In the following, we report only the results obtained by our approach because the competitors CAVERAGE and SNTF were unable to provide a meaningful solution (recall below 10 percent). In fact, this is not surprising because the structure of the clustering problem arising from this application is characterized by an amount of wrong correspondences in $\mathcal{A}$ (outliers) that is considerably larger than the number of correct correspondences and, as demonstrated also by the previous series of experiments, partition-based approaches like CAVERAGE and SNTF are highly sensitive to outliers. As a consequence, the similarity transformations computed from the noisy clusters found by these approaches did not correspond to any mapping between the model and the scene. Our approach, on the other hand, was very robust to this kind of global noise and was thus able to perform considerably well also in challenging situations like the ones addressed for this experiment.

In Fig. 6, we report the average recall and standard deviations obtained by our approach on the experiments with increasing level of local Gaussian noise ($\sigma = 0, 0.001, 0.002, 0.004, 0.008$). As experienced in previous sections, our approach achieves good scores, which slightly drop at increasing levels of noise. Indeed, larger perturbations of the points in the scene prevent the clustering approach from finding a correct similarity transformation and therefore some points in the scene are erroneously considered as clutter points. We also note that the drop in the performance is sharper in case of datasets with three model instances. This is due to the fact that three models in the scene lead to a higher density of points and, hence, wrong assignments enforced by the local noise are more likely to happen. Additionally, the edge sampling procedure mentioned above lead to less accurate hypergraph representations in case of datasets with a large number of points.

In Fig. 7, we report the results obtained by our approach with a fixed level of $\sigma = 0.001$ local Gaussian noise and with an increasing level of global noise, expressed in terms of 0, 10, 20, 40 clutter points. The obtained results confirm the robustness of our approach to clutter points. Indeed, independently from the noise level and the number of model instances, we achieve an almost constant performance between 97-100 percent.

5.4 Illuminant-Invariant Face Clustering

In [51], it has been shown that images of a Lambertian object illuminated by a point light source lie in a 3D subspace. According to this result, if we assume that four images of a face form the columns of a matrix, then $d = s_4^2/(s_1^2 + \cdots + s_4^2)$ provides us with a measure of dissimilarity, s_i being the ith singular value of this matrix. Following [2], we used this dissimilarity measure for clustering faces in high-dimensional space. We tested our algorithm and its competitors over the Yale Face Database B and its extended version [52], [53], which contained faces of 38 individuals under 64 different illumination conditions. Specifically, we considered subsets of faces from four and five randomly drawn individuals (10 faces per individual), with and without outlier faces. The case with outliers consisted in 10 additional faces taken from as many random individuals. For each such combination, we created 10 different subsets (see Fig. 8 for an example with four individuals and outlier faces). Similarly to the case of line clustering, we run both CAVERAGE and SNTF with values of $K \in \{K^* - 1, K^*, K^* + 1\}$, where K^* is the correct number of individuals.

Table 1 reports the results obtained by the three approaches in terms of classification error (mean and standard deviation). The results are consistent with those obtained in the case of line and plane clustering with the exception of SNTF, which performed worse than the other

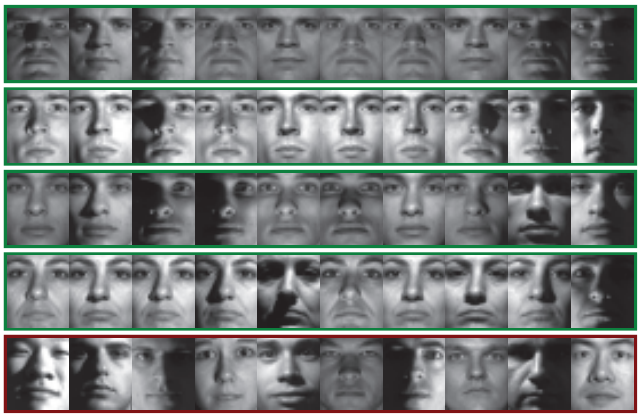

Fig. 8. Example of dataset for illuminant-invariante face clustering with four individuals (first four rows) and 10 outlier faces (last row).

TABLE 1
Experiments on Illuminant-Invariant Face Clustering

n. of classes:	4		5	
n. of outliers:	0	10	0	10
CAVERAGE K=3	0.26±0.09	0.40±0.10	-	-
CAVERAGE K=4	**0.03±0.04**	0.24±0.07	0.21±0.11	0.65±0.12
CAVERAGE K=5	0.13±0.05	0.12±0.05	0.07±0.07	0.41±0.09
CAVERAGE K=6	-	-	0.13±0.08	0.37±0.11
SNTF K=3	0.29±0.10	0.39±0.09	-	-
SNTF K=4	0.14±0.06	0.26±0.09	0.28±0.11	0.51±0.12
SNTF K=5	0.19±0.09	0.25±0.13	0.11±0.09	0.43±0.11
SNTF K=6	-	-	0.14±0.09	0.39±0.13
HoCluGame	0.06±0.03	**0.07±0.02**	**0.06±0.02**	**0.07±0.03**

We report the average classification error and the corresponding standard deviation.

approaches. On the other hand, our algorithm and (the optimal-tuned) CAVERAGE performed comparably well within the no-outlier setting, while our approach dramatically outperformed the other algorithms in the cases comprising outliers.

It would be interesting to address within our framework the following application which was suggested by one of the reviewers: Given a query image I, find the (ESS-)cluster(s) which contains I. To this end, a simple approach could be to start the Baum-Eagon dynamics from a neighborhood of a given image, namely, from an initial distribution which peaks on a target image instead of starting from the uniform distribution (the simplex barycenter), and then test whether, upon convergence, the algorithm eventually extracts a cluster which contains the target image. However, we note that by using the procedure suggested above there is no theoretical guarantee that the support of the converged solution will contain the query image, especially when the basins of attraction of other ESS-clusters (not containing it) are large enough to attract the dynamics somewhere else. We performed several experiments to see what happens in practice and the results confirmed our original concern. A different strategy would consist of adding extra constraints to the original optimization problem in order to force the query image to belong to the support of the solution found. The problem here is that by modifying the feasible set of the optimization problem we most likely change the structure of the solution space, in which case there is no guarantee that the solutions of the modified problem are also solutions of the original one (i.e., ESS-clusters). This observation was in fact confirmed empirically on our image dataset. Hence, it appears that the problem cannot be dealt with using simple-minded arguments but requires instead a more formal treatment and possibly a nontrivial modification of our approach. This would go well beyond the scope of the present paper and we plan to address this issue in our future work.

6 Conclusion

In this paper, we offered a game-theoretic perspective to the hypergraph clustering problem. Within our framework the clustering problem is viewed as a multiplayer noncooperative game, and classical equilibrium notions from evolutionary game theory turn out to provide a natural formalization of the very notion of a cluster. We showed that the problem of finding these equilibria (clusters) is equivalent to solving a polynomial optimization problem with linear constraints, which we solve using high-order replicator dynamics based on the Baum-Eagon inequality. An advantage of our approach over traditional techniques is independence from the number of clusters, which is indeed an intrinsic characteristic of the input data, and robustness against clutter, which is especially useful when solving figure-ground-like grouping or one-class clustering problems. We also mention, as a potential positive feature of the proposed approach, the possibility of finding overlapping clusters (e.g., along the lines presented in [42]), although in this paper we have not explicitly dealt with this problem. The experimental results show the superiority of our approach over the state of the art in terms of quality of solution. We are currently studying alternatives to the replicator dynamics in order to improve efficiency (e.g., [45]). We finally note that, inspired by our work in [24], in a recent paper a parameterized version of our framework was introduced which allows one to control the minimum cluster size [54].

Acknowledgments

The authors acknowledge financial support from the FET programme within EU FP7 under the SIMBAD project (contract 213250). They thank Sameer Agarwal and Ron Zass for providing them with the code of their algorithms, and Ilya Safro for pointing out some useful references. They would also like to thank the anonymous reviewers and the associate editor for their useful comments.

References

[1] A.K. Jain, "Data Clustering: 50 Years Beyond K-Means," *Pattern Recognition Letters*, vol. 31, no. 8, pp. 651-666, 2010.
[2] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond Pairwise Clustering," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 838-845, 2005.
[3] V.M. Govindu, "A Tensor Decomposition for Geometric Grouping and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1150-1157, 2005.
[4] A. Shashua, R. Zass, and T. Hazan, "Multi-Way Clustering Using Super-Symmetric Non-Negative Tensor Factorization," *Proc. European Conf. Computer Vision*, vol. 3954, pp. 595-608, 2006.

[5] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D.N. Metaxas, "Unsupervised Image Categorization by Hypergraph Partition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 6, pp. 1266-1273, June 2011.

[6] J.Y. Zien, M.D.F. Schlag, and P.K. Chan, "Multilevel Spectral Hypergraph Partitioning with Arbitrary Vertex Sizes," *IEEE Trans. Computer-Aided Design of Integration Circuits and Systems,* vol. 18, no. 9, pp. 1389-1399, Sept. 1999.

[7] M. Bolla, "Spectral, Euclidean Representations and Clusterings of Hypergraphs," *Discrete Math.,* vol. 117, pp. 19-39, 1993.

[8] J. Rodrìguez, "On the Laplacian Spectrum and Walk-Regular Hypergraphs," *Linear and Multilinear Algebra,* vol. 51, pp. 285-297, 2003.

[9] D. Zhou, J. Huang, and B. Schölkopf, "Learning with Hypergraphs: Clustering, Classification, Embedding," *Proc. Advances in Neural Information Processing Systems,* vol. 19, pp. 1601-1608, 2006.

[10] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.

[11] S. Agarwal, K. Branson, and S. Belongie, "Higher Order Learning with Graphs," *Proc. Int'l Conf. Machine Learning,* vol. 148, pp. 17-24, 2006.

[12] G. Karypis and V. Kumar, "Multilevel K-Way Hypergraph Partitioning," *VLSI Design,* vol. 11, no. 3, pp. 285-300, 2000.

[13] L. Herault and R. Horaud, "Figure-Ground Discrimination: A Combinatorial Optimization Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 9, pp. 899-914, Sept. 1993.

[14] A. Shashua and S. Ullman, "Structural Saliency: The Detection of Globally Salient Features Using a Locally Connected Network," *Proc. IEEE Int'l Conf. Computer Vision,* 1988.

[15] G. Gupta and J. Ghosh, "Robust One-Class Clustering Using Hybrid Global and Local Search," *Proc. Int'l Conf. Machine Learning,* 2005.

[16] K. Crammer, P.P. Talukdar, and F. Pereira, "A Rate-Distortion One-Class Model and Its Applications to Clustering," *Proc. Int'l Conf. Machine Learning),* 2008.

[17] N. Jardine and R. Sibson, "The Construction of Hierarchic and Non-Hierarchic Classifications," *Computer J.,* vol. 11, pp. 177-184, 1968.

[18] A. Banerjee, C. Krumpelman, S. Basu, R.J. Mooney, and J. Ghosh, "Model-Based Overlapping Clustering," *Proc. Int'l Conf. Knowledge Discovery and Data Mining,* pp. 532-537, 2005.

[19] K. Heller and Z. Ghahramani, "A Nonparametric Bayesian Approach to Modeling Overlapping Clusters," *Proc. Int'l Conf. AI and Statistics,* 2007.

[20] M. Pavan and M. Pelillo, "Dominant Sets and Pairwise Clustering," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 1, pp. 167-172, Jan. 2007.

[21] A. Torsello, S. Rota Bulò, and M. Pelillo, "Grouping with Asymmetric Affinities: A Game-Theoretic Perspective," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 292-299, 2006.

[22] J.W. Weibull, *Evolutionary Game Theory.* Cambridge Univ. Press, 1995.

[23] J. Hofbauer and K. Sigmund, *Evolutionary Games and Population Dynamics.* Cambridge Univ. Press, 1998.

[24] S. Rota Bulò and M. Pelillo, "A Game-Theoretic Approach to Hypergraph Clustering," *Proc. Advances in Neural Information Processing Systems,* vol. 22, pp. 1571-1579, 2009.

[25] D. Fudenberg and J. Tirole, *Game Theory.* MIT Press, 1991.

[26] J. Maynard Smith, *Evolution and the Theory of Games.* Cambridge Univ. Press, 1982.

[27] M. Broom, C. Cannings, and G.T. Vickers, "Multi-Player Matrix Games," *Bull. Math. Biology,* vol. 59, no. 5, pp. 931-952, 1997.

[28] C. Berge, *Hypergraphs: Combinatorics of Finite Sets.* North-Holland, 1989.

[29] P. Frankl and V. Rödl, "Hypergraphs Do Not Jump," *Combinatorica,* vol. 4, pp. 149-159, 1984.

[30] K. Etessami and A. Lochbihler, "The Computational Complexity of Evolutionarily Stable Strategies," *Int'l J. Game Theory,* vol. 37, no. 1, pp. 93-113, 2007.

[31] N. Nisan, "A Note on the Computational Hardness of Evolutionary Stable Strategies," *Electronic Colloquium on Computational Complexity,* 2006.

[32] D.G. Luenberger, *Linear and Nonlinear Programming.* Addison Wesley, 1984.

[33] L.E. Baum and J.A. Eagon, "An Inequality with Applications to Statistical Estimation for Probabilistic Functions of Markov Processes and to a Model for Ecology," *Bull. Am. Math. Soc.,* vol. 73, pp. 360-363, 1967.

[34] G.R. Blakley, "Homogeneous Nonnegative Symmetric Quadratic Transformations," *Bull. Am. Math. Soc.,* vol. 70, pp. 712-715, 1964.

[35] L.E. Baum and G.R. Sell, "Growth Transformations for Functions on Manifolds," *Pacific J. Math.,* vol. 27, pp. 221-227, 1968.

[36] D.G. Luenberger, *Introduction to Dynamic Systems.* Wiley, 1979.

[37] J.P. LaSalle, *The Stability and Control of Discrete Processes.* Springer-Verlag, 1986.

[38] L.E. Baum, T. Petrie, G. Soules, and N. Weiss, "A Maximization Technique Occurring in the Statistical Analysis of Probabilistic Functions of Markov Chains," *Ann. Math. Statistics,* vol. 41, pp. 164-171, 1970.

[39] M. Pelillo, "The Dynamics of Nonlinear Relaxation Labeling Processes," *J. Math. Imaging and Vision,* vol. 7, no. 4, pp. 309-323, 1997.

[40] P. Perona and W. Freeman, "A Factorization Approach to Grouping," *Proc. European Conf. Computer Vision,* pp. 655-670, 1998.

[41] S. Sarkar and K.L. Boyer, "Quantitative Measures of Change Based on Feature Organization: Eigenvalues and Eigenvectors," *Computer Vision and Image Understanding,* vol. 71, no. 1, pp. 110-136, 1998.

[42] A. Torsello, S. Rota Bulò, and M. Pelillo, "Beyond Partitions: Allowing Overlapping Groups in Pairwise Clustering," *Proc. Int'l Conf. Pattern Recognition,* 2008.

[43] O.D. Faugeras and M. Berthod, "Improving Consistency and Reducing Ambiguity in Stochastic Labeling: An Optimization Approach," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, no. 4, pp. 412-424, Apr. 1981.

[44] J.L. Mohammed, R.A. Hummel, and S.W. Zucker, "A Gradient Projection Algorithm for Relaxation Labeling Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 5, no. 3, pp. 330-332, May 1983.

[45] S. Rota Bulò, M. Pelillo, and I.M. Bomze, "Graph-Based Quadratic Optimization: A Fast Evolutionary Approach," *Computer Vision and Image Understanding,* vol. 115, pp. 984-995, 2011.

[46] M. Pelillo and A. Torsello, "Payoff-Monotonic Game Dynamics and the Maximum Clique Problem," *Neural Computation,* vol. 18, no. 5, pp. 1215-1258, 2006.

[47] T. Hu and K. Moerder, "Multiterminal Flows in Hypergraphs," *Proc. VLSI Circuit Layout: Theory and Design,* pp. 87-93, 1985.

[48] D. Gibson, J.M. Kleinberg, and P. Raghavan, "Clustering Categorical Data: An Approach Based on Dynamical Systems," *Proc. 24th Int'l Conf. Very Large Data Bases,* pp. 311-322, 1998.

[49] A. Albarelli, A. Torsello, S. Rota Bulò, and M. Pelillo, "Matching as a Non-Cooperative Game," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

[50] K.P. Horn, "Closed-Form Solution of Absolute Orientation Using Unit Quaternions," *J. Optical Soc. of Am. A,* vol. 4, pp. 629-642, 1987.

[51] P. Belhumeur and D. Kriegman, "What Is the Set of Images of an Object under All Possible Lighting Conditions," *Int'l J. Computer Vision,* vol. 28, no. 3, pp. 245-260, 1998.

[52] A.S. Georghiades, P.N. Belhumeur, and D.J. Kriegman, "From Few to Many: Illumination Cone Models for Face Recognition under Variable Lighting and Pose," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 6, pp. 643-660, June 2001.

[53] K.C. Lee, J. Ho, and D. Kriegman, "Acquiring Linear Subspaces for Face Recognition under Variable Lighting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 27, no. 5, pp. 684-698, May 2005.

[54] H. Liu, L.J. Latecki, and S. Yan, "Robust Clustering as Ensembles of Affinity Relations," *Proc. Advances in Neural Information Processing Systems,* vol. 23, pp. 1414-1422, 2010.

Samuel Rota Bulò received the PhD degree in computer science from the University of Venice, Italy, in 2009 and is currently a postdoctoral researcher at the same institution. His main research interests include the areas of computer vision and pattern recognition with particular emphasis on discrete and continuous optimization methods, graph theory, and game theory. Additional research interests include the field of stochastic modelling. He has more than 20 publications in well-recognized conferences and top-level journals mainly in the areas of computer vision and pattern recognition. He held research visiting positions at the following institutions: IST—Technical University of Lisbon, University of Vienna, Graz University of Technology, University of York, United Kingdom, and Microsoft Research Cambridge, United Kingdom.

Marcello Pelillo joined the faculty of the University of Bari, Italy, as an assistant professor of computer science in 1991. Since 1995, he has been with the University of Venice, Italy, where he is currently a full professor of computer science. He leads the Computer Vision and Pattern Recognition Group and served from 2004 to 2010 as the chair of the board of studies of the Computer Science School. He held visiting research positions at Yale University, University College London, McGill University, the University of Vienna, York University, United Kingdom, and the National ICT Australia (NICTA). He has published more than 130 technical papers in refereed journals, handbooks, and conference proceedings in the areas of computer vision, pattern recognition, and neural computation. He serves (or has served) on the editorial board for the journals *IEEE Transactions on Pattern Analysis and Machine Intelligence* and *Pattern Recognition*, and is regularly on the program committees of the major international conferences and workshops of his fields. In 1997, he co-established a new series of international conferences devoted to energy minimization methods in computer vision and pattern recognition (EMMCVPR) which has now reached the eighth edition. He is (or has been) scientific coordinator of several research projects, including SIMBAD, an EU-FP7 project devoted to similarity-based pattern analysis and recognition. He is a fellow of the IEEE and the IAPR.