Dominant Sets and Pairwise Clustering

Massimiliano Pavan and
Marcello Pelillo, *Senior Member*, *IEEE*

Abstract—We develop a new graph-theoretic approach for pairwise data clustering which is motivated by the analogies between the intuitive concept of a cluster and that of a *dominant set* of vertices, a notion introduced here which generalizes that of a maximal complete subgraph to edge-weighted graphs. We establish a correspondence between dominant sets and the extrema of a quadratic form over the standard simplex, thereby allowing the use of straightforward and easily implementable continuous optimization techniques from evolutionary game theory. Numerical examples on various point-set and image segmentation problems confirm the potential of the proposed approach.

Index Terms—Clustering, quadratic optimization, evolutionary game dynamics, image segmentation, perceptual organization.

1 Introduction

Pairwise or proximity-based, data clustering techniques are gaining increasing popularity over traditional central grouping techniques, which are centered around the notion of "feature" (see, e.g., [7], [19], [20], [4]). In many real-world applications, in fact, a feasible feature-based description of objects might be difficult to obtain or inefficient for learning purposes while, on the other hand, it is often possible to obtain a measure of the (dis)similarity between objects. This is the case, for example, when features consist of both continuous and categorical variables or when the objects to be classified are represented in terms of graphs or structural representations.

A classical approach to pairwise clustering uses concepts and algorithms from graph theory [8], [2]. Indeed, it is natural to map the data to be clustered to the nodes of a weighted graph (the so-called similarity graph), with edge weights representing similarity relations. These methods are of significant interest since they cast clustering as pure graph-theoretic problems for which a solid theory and powerful algorithms have been developed. As pointed out in [2], these methods can produce highly intricate clusters, but they rarely optimize an easily specified global cost function. Graph-theoretic algorithms basically consist of searching for certain combinatorial structures in the similarity graph, such as a minimum spanning tree [23] or a minimum cut [22] and, among these methods, a well-known approach (the "complete-link" algorithm [8]) reduces to a search for a complete subgraph, namely, a *clique*.[1] Indeed, some authors [1], [18] argue that the maximal clique is the strictest definition of a cluster. Unfortunately, while the minimum spanning tree and the minimum cut (with variations thereof) are notions that are explicitly defined on edge-weighted graphs, the concept of a maximal clique is defined on unweighted graphs, and it is not clear how to generalize it to the edge-weighted case. As a consequence, maximal-clique-based clustering algorithms typically work on unweighted graphs derived from the similarity graph by means of some threshold operation [8], [1], [6]. Although such threshold operations can be used to generate a hierarchy of clusters displayed to a user in the form of a dendogram [8], in tasks involving a large number of data items, such as image segmentation, this approach is infeasible. It is therefore of considerable interest to extend the notion of a maximal clique to edge-weighted graphs, and this is precisely what we do in this work, which appeared in a preliminary form in [13].

Motivated by the previous arguments, we propose a new approach for pairwise data clustering which is centered around a novel graph-theoretic concept (that of a *dominant set*) arising from the study of a continuous formulation of the maximum clique problem originally due to Motzkin and Straus [11]. Ours is a nontrivial generalization of the notion of a maximal clique in the context of edge-weighted graphs since, in the unweighted case, dominant sets turn out to be equivalent to (strictly) maximal cliques. Formal properties, intuition, and empirical findings make dominant sets reasonable candidates for a new formal definition of a cluster in the context of edge-weighted graphs. A nice feature of our approach is that it naturally provides a principled measure of a cluster's cohesiveness as well as a measure of a vertex participation to each group.

We establish an *exact* correspondence between dominant sets and *local* extrema of a (continuous) quadratic form over the standard simplex. Interestingly, well-known spectral approaches lead to similar (though intrinsically different) quadratic optimization problems [19], [17], [20]. Computationally, this allows us to find dominant sets (clusters) using straightforward continuous optimization techniques known as *replicator dynamics*, a class of dynamical systems arising in evolutionary game theory [21]. Such systems can be coded in a few lines of any high-level programming language, can easily be implemented in a parallel network of locally interacting computational units, and offer the advantage of biological plausibility. Numerical examples on both point data sets as well as image segmentation problems confirm the effectiveness of the proposed approach.

2 Graph-Theoretic Definition of a Cluster

We represent the data to be clustered as an undirected edge-weighted (similarity) graph with no self-loops $G = (V, E, w)$, where $V = \{1, \ldots, n\}$ is the vertex set, $E \subseteq V \times V$ is the edge set, and $w : E \to \mathbb{R}_+^*$ is the (positive) weight function. Vertices in G correspond to data points, edges represent neighborhood relationships, and edge-weights reflect similarity between pairs of linked vertices. As is customary, we represent the graph G with the corresponding weighted adjacency (or similarity) matrix, which is the $n \times n$ symmetric matrix $A = (a_{ij})$, where $a_{ij} = w(i, j)$ if $(i, j) \in E$, and $a_{ij} = 0$ otherwise. Clearly, since there are no self-loops, all the elements on the main diagonal of A are zero.

A common informal definition states that "a cluster is a set of entities which are *alike*, and entities from different clusters are not alike" [8, p. 1]. Hence, a cluster should satisfy two fundamental conditions: 1) it should have high internal homogeneity and 2) there should be high inhomogeneity between the entities in the cluster and those outside. When the entities are represented as an edge-weighted graph, these two conditions amount to saying that the weights on the edges within a cluster should be large, and those on the edges connecting the cluster nodes to the external ones should be small. Clearly, it is not at all obvious what "large" and "small" precisely mean.

To give our formal definition of a cluster, we start with the intuitive idea that the assignment of the edge-weights induces, in some way to be described, an assignment of weights on the vertices. This perspective gives us a chance to analyze the assignment of the edge-weights in a fruitful way. Let $S \subseteq V$ be a nonempty subset of vertices and $i \in S$. The *(average) weighted degree* of i with regard to S is defined as:

$$\mathrm{awdeg}_S(i) = \frac{1}{|S|} \sum_{j \in S} a_{ij} \,. \tag{1}$$

1. Recall that a subset of vertices of a graph is said to be a *clique* if all its nodes are mutually adjacent; a *maximal* clique is one which is not contained in any larger clique, whereas a *maximum* clique is one having largest cardinality.

• *The authors are with the Dipartimento di Informatica, Università Ca' Foscari di Venezia, Via Torino 155, 30172, Venezia Mestre, Italy. E-mail: {pavan, pelillo}@dsi.unive.it.*

Manuscript received 16 Nov. 2005; revised 10 Apr. 2006; accepted 6 June 2006; published online 13 Nov. 2006.
Recommended for acceptance by S. Pal.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0624-1105.

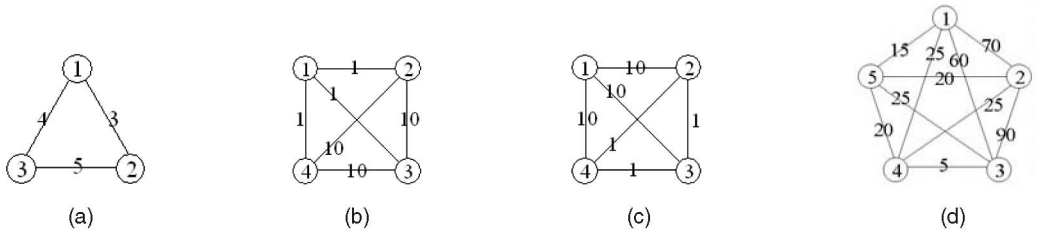

Fig. 1. Four example edge-weighted graphs.

Observe that $\mathrm{awdeg}_{\{i\}}(i) = 0$ for any $i \in V$. Moreover, if $j \notin S$, we define:

$$\phi_S(i,j) = a_{ij} - \mathrm{awdeg}_S(i)\ . \tag{2}$$

Note that $\phi_{\{i\}}(i,j) = a_{ij}$, for all $i,j \in V$ with $i \neq j$. Intuitively, $\phi_S(i,j)$ measures the relative similarity between nodes j and i, with respect to the average similarity between node i and its neighbors in S. Note that $\phi_S(i,j)$ can be either positive or negative.

We are now in a position to formalize the notion of "induction" of node-weights, which is captured by the following recursive definition:

Definition 1. *Let $S \subseteq V$ be a nonempty subset of vertices and $i \in S$. The* weight *of i with regard to S is*

$$\mathrm{w}_S(i) = \begin{cases} 1, & \text{if } |S| = 1 \\ \sum\limits_{j \in S\setminus\{i\}} \phi_{S\setminus\{i\}}(j,i)\mathrm{w}_{S\setminus\{i\}}(j), & \text{otherwise.} \end{cases} \tag{3}$$

Moreover, the total weight *of S is defined to be:* $\mathrm{W}(S) = \sum_{i\in S} \mathrm{w}_S(i)$.

Note that $\mathrm{w}_{\{i,j\}}(i) = \mathrm{w}_{\{i,j\}}(j) = a_{ij}$, for all $i,j \in V$ $(i \neq j)$. Also, observe that $\mathrm{w}_S(i)$ is calculated simply as a function of the weights on the edges of the subgraph induced by S. For example, in Fig. 1a, we have: $\mathrm{w}_{\{1,2,3\}}(3) = \phi_{\{1,2\}}(1,3)\mathrm{w}_{\{1,2\}}(1) + \phi_{\{1,2\}}(2,3)\mathrm{w}_{\{1,2\}}(2) = 18$. Similarly, we obtain $\mathrm{w}_{\{1,2,3\}}(1) = 10$ and $\mathrm{w}_{\{1,2,3\}}(2) = 16$, which yield $\mathrm{W}(\{1,2,3\}) = 44$.

Intuitively, $\mathrm{w}_S(i)$ gives us a measure of the overall (relative) similarity between vertex i and the vertices of $S \setminus \{i\}$ with respect to the overall similarity among the vertices in $S \setminus \{i\}$. For example, for the graph in Fig. 1b, we have $\mathrm{w}_{\{1,2,3,4\}}(1) < 0$, while for that in Fig. 1c we have $\mathrm{w}_{\{1,2,3,4\}}(1) > 0$. This can be explained by considering that in Fig. 1b, vertex 1 is loosely coupled with the remaining vertices, which on their own form a tightly coupled group, whereas in Fig. 1c, exactly the opposite is true. Further, referring again to the graph in Fig. 1a, observe that the edges incident to vertex 1 are the lightest ones, the heaviest ones are incident to vertex 3, and those incident to 2 are the lightest as well as the heaviest ones. This induces a sort of natural ranking among the vertices of the graph, which is indeed captured by the notions introduced above: In fact, we have $\mathrm{w}_{\{1,2,3\}}(1) < \mathrm{w}_{\{1,2,3\}}(2) < \mathrm{w}_{\{1,2,3\}}(3)$.

The following definition represents our formalization of the concept of a cluster in an edge-weighted graph.

Definition 2. *A nonempty subset of vertices $S \subseteq V$ such that $\mathrm{W}(T) > 0$ for any nonempty $T \subseteq S$, is said to be* dominant *if:*

1. $\mathrm{w}_S(i) > 0$, *for all* $i \in S$,
2. $\mathrm{w}_{S\cup\{i\}}(i) < 0$, *for all* $i \notin S$.

The two conditions of the above definition correspond to the two main properties of a cluster: the first regards internal homogeneity, whereas the second regards external inhomogeneity. The condition $\mathrm{W}(T) > 0$ for any nonempty $T \subseteq S$ is a technicality explained in some detail in [12].

To illustrate, in the graph of Fig. 1d, the subset of vertices $\{1,2,3\}$ is dominant, and this may be explained by observing that the edge weights "internal" to that set (60, 70, and 90) are larger than those between internal and external vertices (which are between 5 and 25).

As the example suggests, the main property of a dominant set is that the overall similarity among internal nodes is higher than that between external and internal nodes, and this fact is the motivation of considering a dominant set as a cluster of nodes. Note that, by their own definition, dominant sets are expected to capture highly compact structures. Indeed, it is simple to show that our definition of a dominant set is equivalent to that of a (strictly) maximal clique when applied to unweighted graphs [12]. This means that we have the same concept in the limit of uniform similarity of all objects. This is a further motivation to consider dominant sets as clusters since maximal cliques are a classic formalization of the notion of a cluster [1], [6], [8], [18].

Before concluding this section, we provide a useful characterization of the notions introduced above in terms of determinants. To this end, we need some new notations. If $S \subseteq V$, we denote by A_S the submatrix of A formed by the rows and the columns indexed by the elements of S. Additionally, we define the matrix B_S as:

$$B_S = \begin{pmatrix} 0 & \mathbf{e}^{\mathrm{T}} \\ \mathbf{e} & A_S \end{pmatrix},$$

where $\mathbf{e}$ is a vector of appropriate length consisting of unit entries, and "T" denotes transposition. Assuming $S = \{i_1, \ldots, i_m\}$ with $i_1 < \cdots < i_m$, the matrix ${}^jB_S$ is defined to be:

$${}^jB_S = \begin{pmatrix} 0 & & & & \mathbf{e}^{\mathrm{T}} & & \\ \mathbf{e} & A_S^1 & \cdots & A_S^{j-1} & \mathbf{0} & A_S^{j+1} & \cdots & A_S^m \end{pmatrix},$$

where A_S^i denotes the ith column of A_S.

Lemma 1. *Let $S = \{i_1, \ldots, i_m\} \subseteq V$ be a nonempty subset of vertices and, without loss of generality, assume $i_1 < \cdots < i_m$. Then, we have:*

$$\mathrm{w}_S(i_h) = (-1)^m \det\left({}^hB_S\right), \tag{4}$$

for any $i_h \in S$. Moreover,

$$\mathrm{W}(S) = (-1)^m \det(B_S). \tag{5}$$

Proof. Proceeds by induction and exploits elementary properties of the determinant (see [12], for details). □

An alternative, useful way of computing the $\mathrm{w}_S(i)$s (when $|S| > 1$) is given by the formula:

$$\mathrm{w}_S(i) = \sum_{j \in S\setminus\{i\}} (a_{ij} - a_{hj})\mathrm{w}_{S\setminus\{i\}}(j), \tag{6}$$

where h is an arbitrary element of $S \setminus \{i\}$ (it can be shown [12] that the sum in (6) does not depend upon the choice of h).

3 From Dominant Sets to Local Optima

Consider a similarity graph $G = (V, E, w)$ with n vertices, and its weighted adjacency matrix A. A common way to represent a cluster of vertices (see, e.g., [19], [17], [20]) is to associate a (real-valued) n-dimensional vector to it, where its components express the participation of nodes in the cluster: If a component has a small value, then the corresponding node is weakly associated with the cluster, whereas if it has a large value, the node is strongly

associated with the cluster. Components corresponding to nodes not participating in the cluster are zero.

As pointed out before, a good cluster is one where elements that are strongly associated with it also have large values connecting one another in the similarity matrix. Hence, a natural way of defining the cohesiveness of a cluster is given by the following quadratic form:

$$f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}} A \mathbf{x} \tag{7}$$

and this allows us to formulate the (pairwise) clustering problem as the problem of finding a vector $\mathbf{x}$ that maximizes f. However, note that the objective function is useless without some normalization of the components of $\mathbf{x}$ and, thus, we impose to it simplex (or probability) constraints. This yields the following standard quadratic program, which is a generalization of the so-called Motzkin-Straus program [11]:

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) \\ \text{subject to} & \mathbf{x} \in \Delta, \end{array} \tag{8}$$

where

$$\Delta = \{\mathbf{x} \in \mathbb{R}^n : \mathbf{x} \geq \mathbf{0} \text{ and } \mathbf{e}^{\mathrm{T}}\mathbf{x} = 1\}$$

is the standard simplex of $\mathbb{R}^n$. Thus, within this continuous formulation, a maximally cohesive cluster corresponds to a (local) solution of program (8). It is the purpose of this section to show that this notion of a cluster is intimately related to dominant sets, and that the two notions are indeed two sides of the same coin.

Given a vector $\mathbf{x} \in \Delta$, the *support* of $\mathbf{x}$ is defined as the set of indices corresponding to its nonzero components, that is, $\sigma(\mathbf{x}) = \{i \in V : x_i \neq 0\}$. A point $\mathbf{x} \in \Delta$ satisfies the Karush-Kuhn-Tucker (KKT) conditions for problem (8), i.e., the first-order necessary conditions for local optimality [9], if there exist $n+1$ real constants (Lagrange multipliers) $\mu_1, \ldots, \mu_n$ and λ, with $\mu_i \geq 0$ for all $i = 1 \ldots n$, such that:

$$(A\mathbf{x})_i - \lambda + \mu_i = 0 \tag{9}$$

for all $i = 1 \ldots n$, and $\sum_{i=1}^{n} x_i \mu_i = 0$.

Note that, since both x_i and μ_i are nonnegative for all $i = 1 \ldots n$, the latter condition is equivalent to saying that $i \in \sigma(\mathbf{x})$ implies $\mu_i = 0$. Hence, the KKT conditions can be rewritten as:

$$(A\mathbf{x})_i \begin{cases} = & \lambda, \quad \text{if } i \in \sigma(\mathbf{x}) \\ \leq & \lambda, \quad \text{otherwise} \end{cases} \tag{10}$$

for some real constant λ (indeed, it is immediate to see that $\lambda = \mathbf{x}^{\mathrm{T}} A \mathbf{x}$). A point $\mathbf{x} \in \Delta$ satisfying (10) will be called a *KKT point* throughout.

With the notations introduced at the end of the previous section, note that the KKT equality conditions in (10) amount to saying that there exists a real number λ such that:

$$B_\sigma(\lambda, x_{i_1}, \ldots, x_{i_m})^{\mathrm{T}} = (1, 0, \ldots, 0)^{\mathrm{T}}, \tag{11}$$

where $\sigma = \sigma(\mathbf{x}) = \{i_1, \ldots, i_m\}$ with $i_1 < \cdots < i_m$.

Definition 3. *We say that a nonempty subset of vertices S admits* weighted characteristic vector $\mathbf{x}^S \in \Delta$ *if it has nonnull total weight* $\mathrm{W}(S)$, *in which case, we set:*

$$x_i^S = \begin{cases} \frac{\mathrm{w}_S(i)}{\mathrm{W}(S)}, & \text{if } i \in S \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Note that, by definition, dominant sets always admit a weighted characteristic vector.

The next two results establish useful connections between KKT points of program (8) and weighted characteristic vectors.

Lemma 2. *Let $\sigma = \sigma(\mathbf{x})$ be the support of a vector $\mathbf{x} \in \Delta$ which admits weighted characteristic vector $\mathbf{x}^\sigma$. Then, $\mathbf{x}$ satisfies the KKT equality conditions in (10) if and only if $\mathbf{x} = \mathbf{x}^\sigma$. Moreover, in this case, we have:*

$$\frac{\mathrm{w}_{\sigma\cup\{j\}}(j)}{\mathrm{W}(\sigma)} = (A\mathbf{x})_j - (A\mathbf{x})_i = -\mu_j \tag{13}$$

for all $i \in \sigma$ and $j \notin \sigma$, where the μ_js are the (nonnegative) Lagrange multipliers of program (8).

Proof. Note that conditions (11), which are equivalent to the KKT equality conditions in (10), can be regarded as a system of linear equations in the unknowns λ and x_is ($i \in \sigma$). From Lemma 1, the system has a unique solution since $\det(B_\sigma) \neq 0$. Hence, supposing $\sigma = \{i_1, \ldots, i_m\}$ and, without loss of generality, $i_1 < \ldots < i_m$, from Cramer's rule and Lemma 1, we have:

$$x_{i_h} = \frac{\det({}^h B_\sigma)}{\det(B_\sigma)} = \frac{(-1)^m \mathrm{w}_\sigma(i_h)}{(-1)^m \mathrm{W}(\sigma)} = \frac{\mathrm{w}_\sigma(i_h)}{\mathrm{W}(\sigma)}$$

for any $1 \leq h \leq m$. Therefore, $\mathbf{x} = \mathbf{x}^\sigma$.

The fact that $(A\mathbf{x})_j - (A\mathbf{x})_i = -\mu_j$, for $i \in \sigma$ and $j \notin \sigma$, follows immediately from (9). Finally, using (6), we obtain:

$$\begin{aligned} \frac{\mathrm{w}_{\sigma\cup\{j\}}(j)}{\mathrm{W}(\sigma)} &= \frac{\sum_{h\in\sigma}(a_{jh} - a_{ih})\mathrm{w}_\sigma(h)}{\mathrm{W}(\sigma)} \\ &= \sum_{h\in\sigma} a_{jh} x_h^\sigma - \sum_{h\in\sigma} a_{ih} x_h^\sigma \\ &= (A\mathbf{x}^\sigma)_j - (A\mathbf{x}^\sigma)_i, \end{aligned}$$

which concludes the proof since $\mathbf{x} = \mathbf{x}^\sigma$. □

Proposition 1. *Let $\mathbf{x} \in \Delta$ be a vector whose support $\sigma = \sigma(\mathbf{x})$ has positive total weight $\mathrm{W}(\sigma)$ and, hence, admitting weighted characteristic vector $\mathbf{x}^\sigma$. Then, $\mathbf{x}$ is a KKT point for (8) if and only if the following conditions hold:*

1. $\mathbf{x} = \mathbf{x}^\sigma$,
2. $\mathrm{w}_{\sigma\cup\{j\}}(j) \leq 0$, *for all $j \notin \sigma$.*

Proof. Vector $\mathbf{x}$ satisfies the KKT conditions (10) if and only if $\mathbf{x} = \mathbf{x}^\sigma$ (cf. Lemma 2) and $(A\mathbf{x})_j \leq (A\mathbf{x})_i$ for any $j \notin \sigma$ and $i \in \sigma$, but from (13) the latter condition amounts to saying that $\mathrm{w}_{\sigma\cup\{j\}}(j) \leq 0$, since $\mathrm{W}(\sigma) > 0$. □

The following theorem, which is the main result of this section, establishes an interesting connection between dominant sets and local solutions of program (8).

Theorem 1. *If S is a dominant subset of vertices, then its weighted characteristic vector $\mathbf{x}^S$ is a strict local solution of program (8).*

Conversely, if $\mathbf{x}^*$ is a strict local solution of program (8) then its support $\sigma = \sigma(\mathbf{x}^*)$ is a dominant set, provided that $\mathrm{w}_{\sigma\cup\{i\}}(i) \neq 0$ for all $i \notin \sigma$.

Proof. First, we note that the well-known bordered Hessian test from nonlinear programming [9] can be reformulated in the following way (see [12] for details): Given a subset of m vertices $Q \subseteq V$, A_Q is negative definite in the subspace $\{\mathbf{y} \in \mathbb{R}^m : \sum_{i=1}^m y_i = 0\}$ if and only if $\mathrm{W}(T) > 0$ for any nonempty subset $T \subseteq Q$.

Now, let S be a dominant set. Then, from Proposition 1, it follows that $\mathbf{x}^S$ is a KKT point for (8). Moreover, by Lemma 2, we have that the jth nonnegative Lagrange multiplier μ_j ($j \notin S$) is positive if and only if $\mathrm{w}_{S\cup\{j\}}(j) < 0$. Therefore, the second-order sufficient conditions for local optimality [9], together with the bordered Hessian test, imply that $\mathbf{x}^S$ is a strict local solution for program (8).

Conversely, suppose that $\mathbf{x}^*$ is a strict local solution of (8), and let $\sigma = \sigma(\mathbf{x}^*)$ be its support. After some algebra, it follows that the submatrix A_σ is negative definite in the subspace $\{\mathbf{y} \in \mathbb{R}^m : \sum_{i=1}^m y_i = 0\}$, where $m = |\sigma|$. Hence, from the bordered Hessian test, we have $\mathrm{W}(T) > 0$ for any nonempty subset $T \subseteq \sigma$.

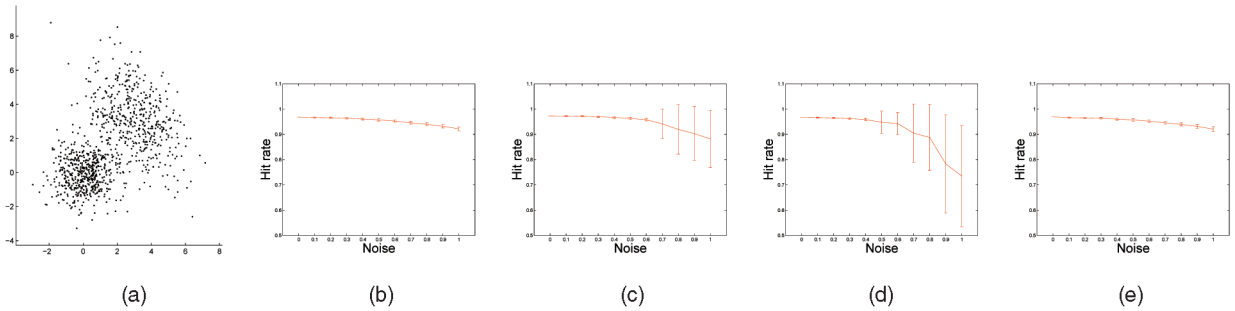

Fig. 2. Evaluating the robustness of the peeling strategy against random perturbations. (a) Original 1,000-point data set obtained from a mixture of two Gaussian distributions. (b), (c), (d), and (e) Classification accuracy obtained with (b) K-means, (c) NCut, (d) DBSCAN, and (e) dominant sets, as a function of noise.

Moreover, we have $w_S(i) > 0$ for all $i \in S$. This follows directly from Lemma 2 (in fact, $\mathbf{x}^*$ is a KKT point) and the definition of weighted characteristic vector. Finally, Proposition 1 states that $\mathbf{x}^* = \mathbf{x}^\sigma$ and $w_{\sigma\cup\{j\}}(j) \leq 0$, for all $j \notin \sigma$. Therefore, the fact that σ is dominant follows trivially from the hypotheses. □

The condition that $w_{\sigma\cup\{i\}}(i) \neq 0$ for all $i \notin \sigma$ is a technicality due to the presence of "spurious" solutions in (8), namely, solutions whose support does not admit a weighted characteristic vector. However, this corresponds to a nongeneric situation and, thus, in the following, we shall ignore it.

By virtue of Theorem 1, dominant sets are in correspondence with (strict local) solutions of the quadratic program shown in (8). This is interesting because, recently, other quadratic programming formulations have been proposed for clustering and segmentation, though motivated by the different idea of finding cuts in a similarity graph [20] or computing eigenvalues and eigenvectors of the weighted adjacency matrix [17], [19]. In particular, note that we use the same objective function as Sarkar and Boyer [19] (see also [17]), which provides a measure of the cohesiveness of a cluster. However, we differ from them in the feasible region, namely, we look for solutions in the standard simplex, whereas they consider the sphere. This is important as the components of the weighted characteristic vectors give us a measure of the participation of the corresponding vertices in the cluster. Hence, in contrast to Sarkar and Boyer's approach, we automatically avoid the nuisance of dealing with negative components, which are meaningless. Note also that no exact combinatorial interpretation is offered for Sarkar and Boyer's "eigenclusters."

The quadratic program we have considered in this section was first analyzed by Motzkin and Straus [11] limited to the case of unweighted graphs, where the matrix A in (8) is a standard 0/1 adjacency matrix. In this case, it turns out that there exists a correspondence between local/global solutions of the program and maximal/maximum cliques of the (unweighted) graph [5], [16]. Since, in unweighted graphs, dominant sets turn out to be equivalent to (strictly) maximal cliques [12], Theorem 1 can be considered as a step toward generalizing the Motzkin-Straus theorem to edge-weighted graphs (see [5] for a generalization involving vertex-weighted graphs).

A straightforward way to find (local) solutions of the program shown in (8) is given by the so-called *replicator dynamics*, a class of continuous and discrete-time dynamical systems arising in evolutionary game theory [21] which are also intimately related to relaxation labeling processes. In our simulations, we used the following model:

$$x_i(t+1) = x_i(t)\frac{(A\mathbf{x})_i}{\mathbf{x}(t)^{\mathrm{T}}A\mathbf{x}(t)} \tag{14}$$

for $i = 1 \ldots n$, which corresponds to the discrete-time version of first-order replicator equations (see, e.g., [21]). It is readily seen that the simplex Δ is invariant under these dynamics, which means that every trajectory starting in Δ will remain in Δ for all future times. Moreover, it can be proven that, since A is symmetric, the objective function $f(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}A\mathbf{x}$ is strictly increasing along any nonconstant trajectory of (14), and its asymptotically stable points are in one-to-one correspondence to strict local solutions of (8) [21]. These, in turn, correspond to dominant sets for the similarity matrix A.

4 Numerical Examples

A simple, yet effective strategy to obtain a hard partition of the input data into coherent groups, is as follows: 1) Find a dominant set (i.e., a cluster), 2) remove the vertices in the cluster from the similarity graph, and 3) reiterate on the remaining vertices. Thus, the algorithm iteratively peels off clusters (dominant sets) and, at each iteration, it determines one by finding a local solution of the quadratic program shown in (8). Clearly, as we proceed, the graphs wherein dominant sets are looked for become smaller and smaller, and this makes the algorithm particularly efficient. In principle, we should keep peeling off clusters until all data have been covered, but in applications involving large and noisy data sets, such as, for example, image segmentation, this makes little sense. In these cases, a better strategy is to stop the algorithm prematurely when *most* of the data points have been classified and then assign the unprocessed ones to the "nearest" cluster according to some distance criterion. Typically, these unassigned items are few noisy and peripheral points that cannot be naturally grouped into the major clusters.

In order to understand the behavior of this *peeling* strategy and to evaluate its robustness against perturbations of the similarity values, we conducted the following experiment on the two-Gaussian data set shown in Fig. 2a. Each point in the original set was randomly perturbed by adding a normally distributed noise, with increasing values of the standard deviation σ. For each value of σ, 100 different data sets were constructed, on each of which we ran our peeling clustering technique. For the sake of comparison, we also ran on the same data K-means [8], Normalized Cut (NCut) [20], and DBSCAN [10]. Fig. 2 shows the classification accuracy of the four algorithms as a function of noise. As can be seen, K-means and our peeling strategy exhibit essentially the same robust behavior, whereas NCut and DBSCAN turn out to be more sensitive to noise.

In many computer vision problems, one would like to extract structure from cluttered background. This is the case, for example, with figure/ground separation and perceptual grouping. In such cases, standard algorithms such as K-means or graph partitioning techniques are not expected to work well, due to their insisting on partitioning *all* the input data and, hence, the unstructured clutter points too, into coherent groups. Our approach, on the contrary, appears to be particularly suited for such applications since it allows one to extract as many clusters as desired, while leaving the remaining points (namely, the clutter) ungrouped.

To illustrate this point, consider the data set shown in Fig. 3a, containing a dense central cluster of random points (the "figure"), surrounded by equally distributed clutter points (the "background"). As expected, on these data, both K-means and NCut failed as they both split the central group in two pieces, whereas our peeling algorithm, as well as DBSCAN, produced accurate results.

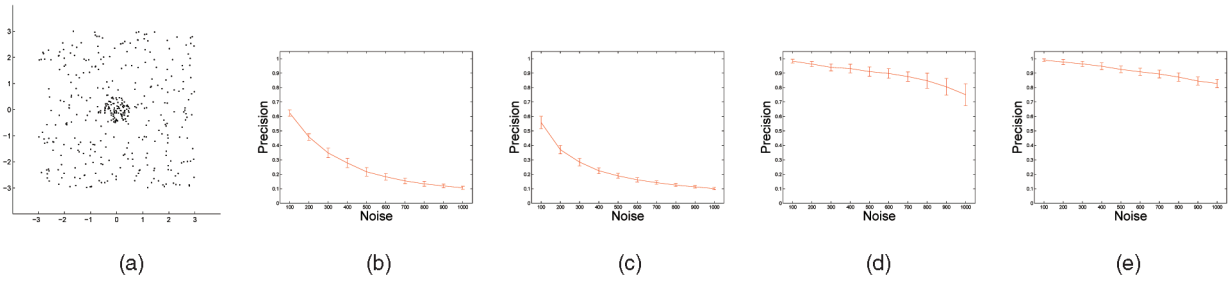

Fig. 3. An example of separating structure from background clutter. (a) Original 400-point data set (100 points for the central, dense group, and 300 for the background). (b), (c), (d), and (e) Precision curves obtained with (b) K-means, (c) NCut, (d) DBSCAN, and (e) dominant sets, as a function of clutter.

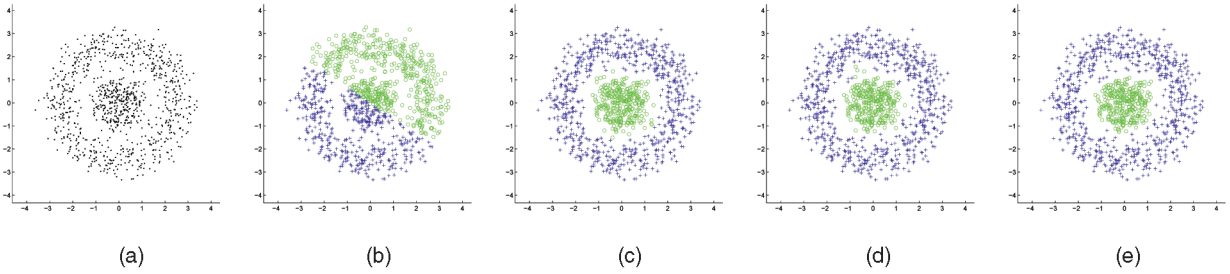

Fig. 4. A clustering example with structurally different groups. (a) Original 1,000-point data set. (b), (c), (d), and (e) Results obtained with (b) K-mean, (c) NCut, (d) DBSCAN, and (e) dominant sets.

In order to study the robustness of the approach against random noise in the background, we let the level of clutter vary, starting from 100 to 1,000 points. Note that DBSCAN automatically distinguishes between noise (i.e., ground) and nonnoise (i.e., figure) points, whereas, for K-means and NCut, the decision as to label a given group as figure and the other as ground was taken using a simple majority rule (using the ground truth). As for our approach, we simply declared figure the (first) dominant set found and background the remaining points. For each noise level, we calculated the precision of the four algorithms, as the percentage of true figure points among the number of points classified as figure. Fig. 3 shows the behavior of the precision curves as a function of noise for the four algorithms. As it turns out, ours substantially outperforms both K-means and NCut, and performs slightly better than DBSCAN.

A third experiment was done on the 1,000-point data set shown in Fig. 4a, which has become a standard benchmark for pairwise clustering techniques. The main feature of this data set is that it contains two structurally different (noisy) clusters, one being compact, and the other having an elongated structure. Here, K-means produces totally wrong results, as shown in Fig. 4b. A direct application of our algorithm to the similarity matrix whose entries are taken to be inversely proportional to the Euclidean distances would yield an oversegmentation of the external ring (the central disc being separated correctly). Indeed, this is not surprising due to the intrinsic feature of dominant sets of capturing compact groups. To avoid this phenomenon, we used the path-based (dis)similarity measure recently proposed by Fischer and Buhmann in [3], which stresses connectedness of data points via mediating elements. The results obtained are shown in Fig. 4e and are similar to those produced by NCut and DBSCAN: All three algorithms were able to separate correctly the data into two classes.

As for the computational time, we remark that in the three series of experiments all algorithms (except K-means which was by far the fastest) typically took a few seconds to converge, with our algorithm being two to three time faster than both NCut and DBSCAN.

Finally, we apply our clustering framework to the image segmentation problem. The image to be segmented is represented as an edge-weighted undirected graph, where vertices correspond to individual pixels and the edge-weights reflect the "similarity" between pairs of vertices. In our experiments, the similarity between pixels i and j was measured by $w(i,j) = \exp(-\|\mathbf{F}(i) - \mathbf{F}(j)\|_2^2/\sigma^2)$, where σ is a positive real number which affects the decreasing rate of w, and $\mathbf{F}(i)$ is defined as the intensity value at node i, normalized to a real number in the interval [0, 1], for segmenting brightness images, and as $\mathbf{F}(i) = [v, vs\sin(h), vs\cos(h)](i)$, where h, s, and v are the HSV values of pixel i, for color segmentation.

For the sake of comparison, we also ran NCut on the same images.[2] The results presented here were obtained after a careful tuning of its parameters. To get cleaner segmentations for both algorithms, connected components whose area was around 0.1 percent of that of the whole image were incorporated into larger adjacent regions using a straightforward spatial proximity criterion. We remark that only 2-3 percent of the pixels in the whole images were involved in this operation, which means that the overall quality of the segmentations cannot be credited to this postprocessing. Fig. 5 shows the results obtained with our segmentation algorithm and NCut on various natural brightness and color images (typical image size is 90×120 pixels). On average, our algorithm (and NCut too) took only a few seconds to return a segmentation on a machine equipped with a 2 GHz Intel Pentium IV. As can be seen, the dominant-set segmentations are substantially cleaner than those obtained with NCut, which typically tends to produce oversegmented results.

5 Conclusions

We have introduced the notion of a *dominant set* of vertices in an edge-weighted graph and have shown how this concept can be relevant in pairwise data clustering. We have established a connection between the (combinatorial) problem of finding dominant sets and (continuous) quadratic programming, and this allows the use of straightforward dynamics from evolutionary game theory to determine them. Experimentally, we have demonstrated the potential of our approach on various point-set and image segmentation examples. Extensions of the approach presented here involving hierarchical data partitioning and out-of-sample extensions of dominant-set clusters can be found in [14], [15], respectively. We are currently working toward providing a massive experimental evaluation of our approach on (high-resolution) image and spatio-temporal video segmentation.

2. We used Shi's implementation, which can be downloaded from: http://www.hid.ri.cmu.edu/Hid/software_ncutPublic.html.

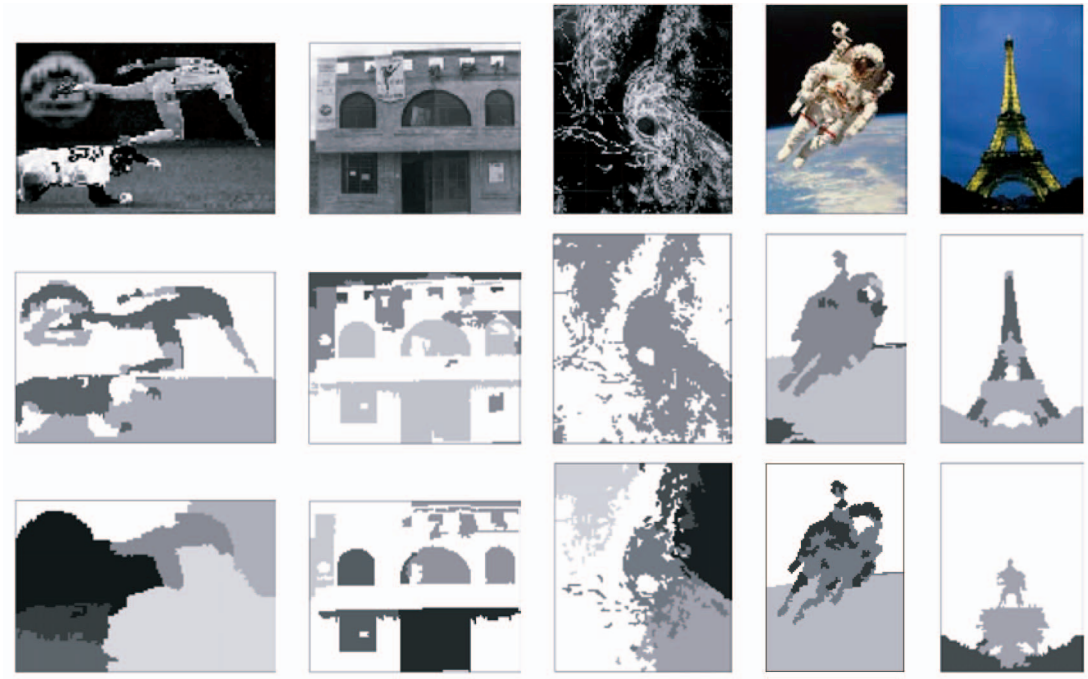

Fig. 5. Image segmentation results. Top row: Original images (from left to right: three brightness and two color images). Middle row: Segmentations obtained with dominant sets. Bottom row: Segmentations obtained with NCut.

REFERENCES

[1] J.G. Auguston and J. Minker, "An Analysis of Some Graph Theoretical Clustering Techniques," *J. ACM,* vol. 17, no. 4, pp. 571-588, 1970.

[2] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification.* J. Wiley & Sons, 2000.

[3] B. Fischer and J.M. Buhmann, "Path-Based Clustering for Grouping Smooth Curves and Texture Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 25, no. 4, pp. 513-518, Apr. 2003.

[4] Y. Gdalyahu, D. Weinshall, and M. Werman, "Self-Organization in Vision: Stochastic Clustering for Image Segmentation, Perceptual Grouping, and Image Database Organization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 10, pp. 1053-1074, Oct. 2001.

[5] L.E. Gibbons, D.W. Hearn, P.M. Pardalos, and M.V. Ramana, "Continuous Characterizations of the Maximum Clique Problem," *Math. Operations Research,* vol. 22, pp. 754-768, 1997.

[6] C.C. Gotlieb and S. Kumar, "Semantic Clustering of Index Terms," *J. ACM,* vol. 15, no. 4, pp. 493-513, 1968.

[7] T. Hofmann and J. Buhmann, "Pairwise Data Clustering by Deterministic Annealing," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 1-14, Jan. 1997.

[8] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.

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

[10] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise," *Proc. Second Int'l Conf. Knowledge Discovery and Data Mining,* pp. 226-231, 1996.

[11] T.S. Motzkin and E.G. Straus, "Maxima for Graphs and a New Proof of a Theorem of Turán," *Canadian J. Math.,* vol. 17, pp. 533-540, 1965.

[12] M. Pavan, "A New Graph-Theoretic Approach to Clustering, with Applications to Computer Vision," PhD thesis, Università Ca' Foscari di Venezia, Italy, 2004.

[13] M. Pavan and M. Pelillo, "A New Graph-Theoretic Approach to Clustering and Segmentation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 145-152, 2003.

[14] M. Pavan and M. Pelillo, "Dominant Sets and Hierarhical Clustering," *Proc. IEEE Int'l Conf. Computer Vision,* vol. 1, pp. 362-369, 2003.

[15] M. Pavan and M. Pelillo, "Efficient Out-of-Sample Extension of Dominant-Set Clusters," *Advances in Neural Information Processing Systems 17,* L.K. Saul, Y. Weiss, and L. Bottou, eds., pp. 1057-1064, 2005.

[16] M. Pelillo and A. Jagota, "Feasible and Infeasible Maxima in a Quadratic Program for Maximum Clique," *J. Artificial Neural Networks,* vol. 2, pp. 411-420, 1995.

[17] P. Perona and W. Freeman, "A Factorization Approach to Grouping," *Proc. European Conf. Computer Vision,* H. Burkhardt and B. Neumann, eds., pp. 655-670, 1998.

[18] V.V. Raghavan and C.T. Yu, "A Comparison of the Stability Characteristics of Some Graph Theoretic Clustering Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 3, pp. 393-402, 1981.

[19] 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.

[20] 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.

[21] J.W. Weibull, *Evolutionary Game Theory.* MIT Press, 1995.

[22] Z. Wu and R. Leahy, "An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1101-1113, Nov. 1993.

[23] C.T. Zahn, "Graph-Theoretic Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computers,* vol. 20, pp. 68-86, 1971.