A New Graph-Theoretic Approach to Clustering and Segmentation

Massimiliano Pavan and Marcello Pelillo
Dipartimento di Informatica
Università Ca' Foscari di Venezia
Via Torino 155, 30172 Venezia Mestre, Italy
{mapavan, pelillo}@dsi.unive.it

Abstract

We develop a framework for the image segmentation problem based on a new graph-theoretic formulation of clustering. The approach is motivated by the analogies between the intuitive concept of a cluster and that of a dominant set *of vertices, a novel notion that generalizes that of a maximal complete subgraph to edge-weighted graphs. We also establish a correspondence between dominant sets and the extrema of a quadratic form over the standard simplex, thereby allowing us the use of continuous optimization techniques such as* replicator dynamics *from evolutionary game theory. Such systems are attractive as can be coded in a few lines of any high-level programming language, can easily be implemented in a parallel network of locally interacting units, and offer the advantage of biological plausibility. We present experimental results on real-world images which show the effectiveness of the proposed approach.*

1 Introduction

The segmentation of images is a classic problem in computer vision and pattern recognition [2, 6], and recently there has been an increasing interest in graph-theoretic segmentation algorithms based on clustering (see, e.g., [4, 6, 7, 20, 23, 24, 26]).

An image can be represented as a *similarity* (edge-weighted) graph, where the vertices represent individual pixels, the edges neighborhood relations, and the weights on the edges reflect the similarity between pixel appearances. Graph-theoretic clustering algorithms basically consist of searching for certain combinatorial structures in the similarity graph, such as a minimum spanning tree [27] or a minimum cut [7, 24, 26] and, among these methods, a classic approach to clustering (the "complete-link" algorithm [12]) reduces to a search for a complete subgraph, namely a *clique*.[1] Indeed, some authors [1, 21] argue that the maximal clique is the strictest definition of a cluster. Unfortunately, while the minimum spanning tree and the minimum cut (and 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 [12, 1, 9]. Although such threshold operations can be used to generate a hierarchy of clusters displayed to a user in the form of a *dendogram* [12], in image segmentation applications this approach is infeasible due to the large number of data (pixels) to be clustered. It is therefore of considerable interest to extend the notion of a maximal clique to edge-weighted graphs, thereby allowing the development of a new non-hierarchical (partitional) clustering approach.

To this end, in this paper we propose a new framework for pairwise clustering and image segmentation based on a novel combinatorial concept (that of a *dominant set*) which arises from the study of a continuous formulation of the maximum clique problem, originally due to Motzkin and Straus [16]. Our proposal seems to be a plausible 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, intuitive arguments, and empirical findings make dominant sets reasonable candidates for a new formal definition of a cluster in the context of edge-weighted graphs.

A second contribution of this paper is to establish a correspondence between dominant sets and the extrema of a (continuous) quadratic form over the standard simplex. Interestingly, other well-known approaches to clustering and

[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.

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

segmentation lead to similar (though different) quadratic optimization problems [20, 23, 24]. Computationally, this allows us to find dominant sets (clusters) using straightforward continuous optimization techniques such as *replicator equations*, a class of dynamical systems arising in evolutionary game theory [10, 25]. Such systems, which are intimately related to Hopfield neural networks [11] and relaxation labeling processes [22, 18], are attractive as can be coded in a few lines of any high-level programming language, can easily be implemented in a parallel network of locally interacting units, and offer the advantage of biological plausibility [28].

We apply our clustering methodology to image segmentation. Basically, our approach consists of searching for dominant sets in the similarity graph built upon the image, using the replicator dynamics. Experiments on real-world images show the effectiveness of the proposed framework.

2. Graph-theoretic definition of a cluster

We represent the data to be clustered as an undirected edge-weighted 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 customary, we represent the graph G with the corresponding weighted adjacency (or similarity) matrix, which is the $n \times n$ symmetric matrix $A = (a_{ij})$ defined as:

$$a_{ij} = \begin{cases} w(i,j)\,, & \text{if } (i,j) \in E \\ 0\,, & \text{otherwise.} \end{cases}$$

A common informal definition states that "a cluster is a set of entities which are *alike*, and entities from different clusters are not alike" [12, p. 1]. Hence, a cluster should satisfy two fundamental conditions: (a) it should have high internal homogeneity; (b) 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.

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 simpler and fruitful way. To grasp the intuition behind this idea, consider the graph in Figure 1 and the subgraph induced by the set $S = \{1, 2, 3\}$. Observe that the edges incident

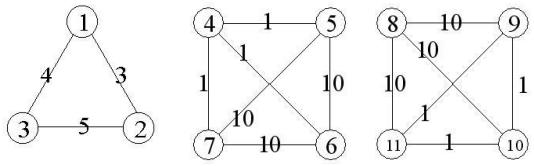

Figure 1. An example edge-weighted graph.

to vertex 1 are the lightest ones (within the subgraph), 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 S, which is captured by the notions introduced below.

Let $S \subseteq V$ be a non-empty subset of vertices and $i \in V$. The *(average) weighted degree* of i w.r.t. S is defined as:

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

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 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 non-empty subset of vertices and $i \in S$. The* weight *of i w.r.t. S is*

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

Moreover, the total weight *of S is defined to be:*

$$\mathrm{W}(S) = \sum_{i \in S} \mathrm{w}_S(i)\,. \tag{4}$$

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. Moreover, the weights so defined respect the intuitive ranking illustrated above. For example, referring again to the graph in Figure 1, it turns out that $\mathrm{w}_{\{1,2,3\}}(1) < \mathrm{w}_{\{1,2,3\}}(2) < \mathrm{w}_{\{1,2,3\}}(3)$.

Proceedings of the 2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'03)
1063-6919/03 $17.00 © 2003 IEEE

Intuitively, $\mathrm{w}_S(i)$ gives us a measure of the overall 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, in the graph of Figure 1 it turns out that $\mathrm{w}_{\{4,5,6,7\}}(4) < 0$ and $\mathrm{w}_{\{8,9,10,11\}}(8) > 0$ and this can be intuitively grasped by looking at the amount of edge-weight associated to vertices 4 and 8: that associated to vertex 4 is significantly smaller than that of subset $\{5,6,7\}$; conversely, that associated to vertex 8 is significantly greater than that of subset $\{9,10,11\}$.

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

Definition 2 *A non-empty subset of vertices $S \subseteq V$ such that $\mathrm{W}(T) > 0$ for any non-empty $T \subseteq S$, 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 non-empty $T \subseteq S$ is a technicality explained in some detail in [17].

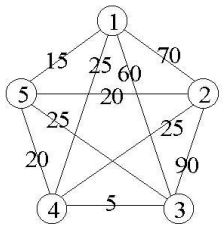

Figure 2. The subset of vertices $\{1,2,3\}$ is dominant.

To illustrate, in the graph of Figure 2 the subset of vertices $\{1,2,3\}$ is dominant, and this may be intuitively 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.

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 i-th column of A_S.

Lemma 1 *Let $S = \{i_1, \ldots, i_m\} \subseteq V$ be a non-empty subset of vertices and, w.l.o.g., assume $i_1 < \cdots < i_m$. Then, we have:*

$$\mathrm{w}_S(i_h) = (-1)^m \det\left({}^hB_S\right), \tag{5}$$

for any $i_h \in S$. Moreover:

$$\mathrm{W}(S) = (-1)^m \det(B_S)\ . \tag{6}$$

Proof: Proceeds by induction and exploits elementary properties of the determinant (see [17] for details). □

An alternative, useful way of computing the $\mathrm{w}_S(i)$'s is given by the following formula (cfr. [17]):

$$\mathrm{w}_S(i) = \sum_{j \in S\setminus\{i\}} (a_{ij} - a_{hj})\, \mathrm{w}_{S\setminus\{i\}}(j) \tag{7}$$

where h is an arbitrary element of $S \setminus \{i\}$ (it can be shown that the sum in (7) does not depend upon the choice of h).

3. From dominant sets to local optima

Given an edge-weighted graph $G = (V, E, w)$ and its weighted adjacency matrix A, consider the following quadratic program (which is a generalization of the so-called Motzkin-Straus program [16]):

$$\begin{array}{ll} \text{maximize} & f(\mathbf{x}) = \frac{1}{2}\mathbf{x}^{\mathrm{T}}A\mathbf{x} \\ \text{subject to} & \mathbf{x} \in \Delta \end{array} \tag{8}$$

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 $\mathbb{R}^n$. Recall that a point $\mathbf{x}^* \in \Delta$ is said to be a *local* solution of program (8) if there exists an $\varepsilon > 0$ such that $f(\mathbf{x}^*) \geq f(\mathbf{x})$ for all $\mathbf{x} \in \Delta$ whose distance from $\mathbf{x}^*$ is less than ε, and if $f(\mathbf{x}^*) = f(\mathbf{x})$ implies $\mathbf{x}^* = \mathbf{x}$, then $\mathbf{x}^*$ is said to be a *strict* local solution.

Given a vector $\mathbf{x} \in \mathbb{R}^n$, the *support* of $\mathbf{x}$ is defined as the set of indices corresponding to its non-zero components, that is:

$$\sigma(\mathbf{x}) = \{i \in V \ : \ x_i \neq 0\} \ . \tag{9}$$

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 [14], 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:

$$(A\mathbf{x})_i - \lambda + \mu_i = 0 \tag{10}$$

for all $i = 1 \ldots n$, and

$$\sum_{i=1}^{n} x_i \mu_i = 0 \ . \tag{11}$$

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:

$$(A\mathbf{x})_i \begin{cases} = & \lambda, \quad \text{if } i \in \sigma(\mathbf{x}) \\ \leq & \lambda, \quad \text{otherwise} \end{cases} \tag{12}$$

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 (12) 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 (12) amount to saying that there exists a real number λ such that:

$$B_\sigma \left(\lambda, x_{i_1}, \ldots, x_{i_m}\right)^{\mathrm{T}} = (1, 0, \ldots, 0)^{\mathrm{T}} \tag{13}$$

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

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

$$x_i^S = \begin{cases} \frac{\mathrm{w}_S(i)}{\mathrm{W}(S)} \ , & \text{if } i \in S \\ 0 \ , & \text{otherwise.} \end{cases} \tag{14}$$

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 (12) if and only if $\mathbf{x} = \mathbf{x}^\sigma$. Moreover, in this case, we have:*

$$\frac{\mathrm{w}_{\sigma \cup \{j\}}(j)}{\mathrm{W}(\sigma)} = (A\mathbf{x})_j - (A\mathbf{x})_i = -\mu_j \tag{15}$$

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

Proof: Note that conditions (13), which are equivalent to the KKT equality conditions in (12), can be regarded as a system of linear equations in the unknowns λ and x_i's ($i \in \sigma$). From Lemma 1, the system has a unique solution since $\det(B_\sigma) \neq 0$. Hence, supposing $\sigma = \{i_1, \ldots, i_m\}$ and, w.l.o.g., $i_1 < \ldots < i_m$, from Cramer's rule and Lemma 1 we have:

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

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 equation (10). Finally, using equation (7), 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 (12) if and only if $\mathbf{x} = \mathbf{x}^\sigma$ (cfr. Lemma 2) and $(A\mathbf{x})_j \leq (A\mathbf{x})_i$ for any $j \notin \sigma$ and $i \in \sigma$, but from (15) 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 intriguing connection between dominant sets and local solutions of program (8).

Theorem 1 *If S is a dominant subset of vertices, then its weighted characteristics 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 [14] can be reformulated in the following way (see [17] 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 non-empty 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 j-th 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 [14], 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 non-empty subset $T \subseteq \sigma$.

Moreover, we have $\mathrm{w}_S(i) > 0$ 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 $\mathrm{w}_{\sigma\cup\{j\}}(j) \leq 0$, for all $j \notin \sigma$. Therefore, the fact that σ is dominant follows trivially from the hypotheses. □

The condition that $\mathrm{w}_{\sigma\cup\{i\}}(i) \neq 0$ 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 (see [17] for details). However, this corresponds to a non-generic situation and thus, in the following, we shall ignore it.

The quadratic program we have considered in this section was first analyzed by Motzkin and Straus [16] limited to the case of unweighted graphs, where the matrix A in (8) is a standard (unweighted) adjacency matrix. In this case, it turns out that there exists a strong correspondence between the solutions of the program and the maximal cliques of the (unweighted) graph [8, 19]. Since an unweighted graph can be seen as a special case of an edge-weighted graph, our definition of a dominant set is indeed equivalent to that of a (strictly) maximal clique when applied to unweighted graphs [17]. 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, 9, 12, 21].

By virtue of Theorem 1 dominant sets are in correspondence with (strict) solutions of quadratic program (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 [24] or computing eigenvalues and eigenvectors of the weighted adjacency matrix [20, 23]. In particular, note that we use the same objective function as Sarkar and Boyer [23] (see also [20]), 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 combinatorial interpretation is offered for Sarkar and Boyer's "eigenclusters."

4. Finding dominant sets by replicator dynamics

The main theorem of the previous section provides a tight correspondence between the problem of finding dominant sets in an edge-weighted graph and that of finding solutions of a quadratic program. By virtue of this theoretical result, we can find a dominant set by first localizing a solution of program (8) with an appropriate continuous optimization technique, and then picking up the support set of the solution found. In this sense, we indirectly perform combinatorial optimization via continuous optimization. Here, we provide some details about the continuous optimization method we use to solve problem (8).

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

$$\dot{x}_i(t) = x_i(t)\left[(M\mathbf{x}(t))_i - \mathbf{x}(t)^{\mathrm{T}} M\mathbf{x}(t)\right] , \qquad (16)$$

where a dot signifies derivative w.r.t. time t, and its discrete-time counterpart

$$x_i(t+1) = x_i(t)\frac{(M\mathbf{x}(t))_i}{\mathbf{x}(t)^{\mathrm{T}} M\mathbf{x}(t)} . \qquad (17)$$

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 turns out that their *stationary points*, i.e. the points satisfying $\dot{x}_i(t) = 0$ for (16) or $x_i(t+1) = x_i(t)$ for (17), coincide and are the solutions of the equations

$$x_i[(M\mathbf{x})_i - \mathbf{x}^{\mathrm{T}} M\mathbf{x}] = 0 \qquad (18)$$

for $i = 1 \ldots n$. A stationary point $\mathbf{x}$ is said to be *asymptotically stable* if every solution to (16) or (17) which starts close enough to $\mathbf{x}$, will converge to $\mathbf{x}$ as $t \to \infty$.

Both (16) and (17) are called *replicator equations* in theoretical biology and evolutionary game theory, since they are used to model evolution over time of relative frequencies of interacting, self-replicating entities [10]. The discrete-time dynamical equations turn out to be a special case of a general class of dynamical systems introduced by Baum and Eagon [3] in the context of Markov chains theory. They also represent an instance of the original heuristic Rosenfeld-Hummel-Zucker relaxation labeling algorithm [22], whose dynamical properties have recently been clarified [18] (specifically, it corresponds to the 1-object, n-label case). Moreover, the biological significance

of such processes has been strongly advocated by Zucker et al. [28] who hypothesized that the first 2-3 iterations of the algorithm could be implemented by the pyramidal neurons connecting the striate and the extrastriate cortices.

We are now interested in the dynamical properties of replicator dynamics; it is these properties that will allow us to solve our original combinatorial problem.

Theorem 2 *If $M = M^{\mathrm{T}}$ then the function $\mathbf{x}(t)^{\mathrm{T}} M \mathbf{x}(t)$ is strictly increasing with increasing t along any non-stationary trajectory $\mathbf{x}(t)$ under both continuous-time (16) and discrete-time (17) replicator dynamics. Furthermore, any such trajectory converges to a stationary point. Finally, a vector $\mathbf{x} \in \Delta$ is asymptotically stable under (16) and (17) if and only if $\mathbf{x}$ is a strict local maximizer of $\mathbf{x}^{\mathrm{T}} M \mathbf{x}$ on Δ.*

The previous result is known in mathematical biology as the fundamental theorem of natural selection [10, 25] and, in its original form, traces back to R. A. Fisher [5]. As far as the discrete-time model is concerned, it can be regarded as a straightforward implication of the more general Baum-Eagon theorem [3]. The fact that all trajectories of the replicator dynamics converge to a stationary point has been proven more recently [13, 15].

In light of their dynamical properties, replicator equations naturally suggest themselves as a simple and useful heuristic for finding dominant sets. Indeed, let A denote the weighted adjacency matrix of an edge-weighted graph G. By letting

$$M = A$$

we know that the replicator dynamical systems (16) and (17), starting from an arbitrary initial state, will iteratively maximize the function $\mathbf{x}^{\mathrm{T}} A \mathbf{x}$ over Δ and will eventually be attracted with probability 1 by the nearest asymptotically stable point. By virtue of Theorem 2 this will then correspond to a strict local maximizer of $\mathbf{x}^{\mathrm{T}} A \mathbf{x}$ in Δ and hence, by Theorem 1, to a dominant set.

Since the process cannot leave the boundary of Δ, it is customary to start out the relaxation process from some interior point, a common choice being the barycenter of Δ. This prevents the search from being initially biased in favor of any particular solution.

5. Application to image segmentation

We apply our clustering methodology to the segmentation of intensity and color images. 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. As customary, we define a similarity measure between pixels based on brightness/color proximity. Specifically, in our experiments the similarity between pixels i and j was measured by:

$$w(i,j) = \exp\left(\frac{-\|\mathbf{F}(i) - \mathbf{F}(j)\|_2^2}{\sigma^2}\right)$$

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, v are the HSV values of pixel i, for color segmentation.

In principle, our clustering algorithm consists of iteratively finding a dominant set in the graph using replicator dynamics and then removing it from the graph, until all vertices have been clustered. Indeed, in the experiments reported here, to avoid the formation of small meaningless clusters, we repeated the process until 90% of the pixels were clustered, the remaining ones being assigned to the closest clusters, in terms of brightness/color proximity. In our experiments, we used the discrete-time replicator equations (17). The process was started from the simplex barycenter and stopped after a few iterations (typically, no more than three). To improve the segmentation results, after convergence small isolated regions were incorporated into larger ones (see figure's captions for details).

Figures 3 to 7 show the results obtained with our segmentation algorithm on various natural brightness and color images. The left side of each figure shows the original image and the right one shows the corresponding segmentation, where connected pixels having the same gray level are intended to belong to the same region. On average, the algorithm took only a few seconds to return a segmentation, on a machine equipped with a 750 MHz Intel Pentium III.

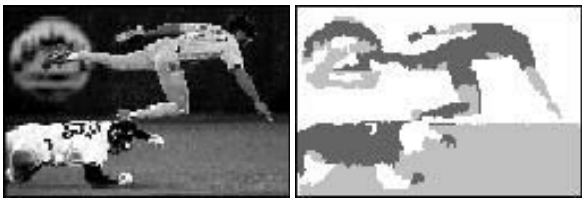

Figure 3. An 83×125 intensity image (left) and its segmentation (right). Parameter setting: $\sigma = 0.14$. Minimal region size of interest is 11.

Figure 3 shows an intensity image taken during a baseball game that has been used originally by Shi and Malik [24] and other authors [4, 7]. As can be seen, unlike other algorithms [24, 7, 4], ours was able to separate the grassy region from the back wall in a nice way. The uniforms of the two players (which have significant variation due to folds in the cloth) are also segmented in a satisfactory way, and smaller but important components such as the

arms and gloves are also correctly segmented. Like other algorithms [24, 7], however, ours did not succeed in distinguishing between the Mets emblem and the left leg of the top player. Also, note that the helmet of the bottom player is incorrectly merged with the back wall. Overall, these results compare well with those presented in [24, 7], and are substantially better than those obtained with an optimally-tuned standard split-and-merge algorithm [7].

Figure 4, which shows a weather radar image, has also been used in [24] as an instance whereby edge-detection-based segmentation would perform poorly. The algorithm was able to correctly partition the image into a background and a foreground, and our results compare well with those reported in [24], where the image was separated into more components.

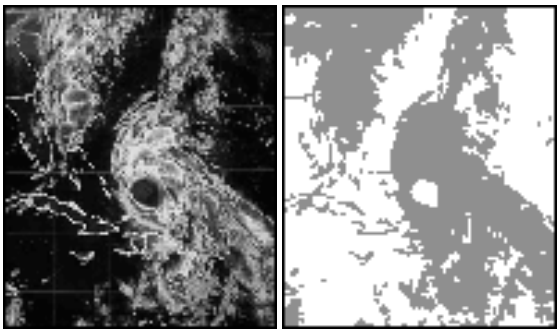

Figure 4. A 115×97 intensity image (left) and its segmentation (right). Parameter settings: $\sigma = 0.1$. Minimal region size of interest is 2.

Figure 5 shows the facade of a building. Again, the segmentation produced by our algorithm is of good quality. Specifically, the sky, the windows, the banners, and the wall (except from some noisy components around the central and right window on the first level) are all correctly segmented. Note also that the shadow of the balcony, the central door, and the window on the left-hand side are erroneously put in a unique component.

Figures 6 and 7 show results on color images taken from the COREL database. Figure 6 shows the image of a spaceman. Note how the segmentation is very clean despite substantial color variation. Essentially, the algorithm found three large regions: a large component for the black sky, another one for the earth, and a third one for the astronaut. Note also that some spurious regions were also found in the earth and in the man areas.

Finally, Figure 7 (used also in [4]) shows an image of the Eiffel tower at night. Here, the algorithm was able to partition the image into meaningful components. It found a large component for the sky and, within the tower, it distinguished between the bright and the dark areas. However, note that the dark area in the bottom part of the image was incorrectly merged with the lower dark part of the tower. The quality of our segmentation looks comparable to the one reported by Felzenszwalb and Huttenlocher in [4]: they were able to separate the lower dark area from the tower, but failed in clearly distinguishing the dark from the bright tower regions.

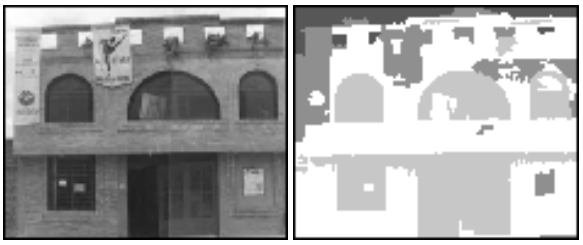

Figure 5. A 94×115 intensity image (left) and its segmentation (right). Parameter setting: $\sigma = 0.15$. Minimal region size of interest is 11.

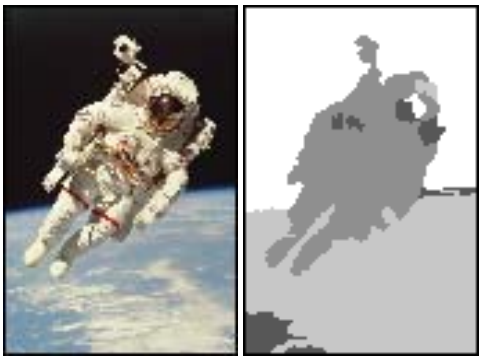

Figure 6. An 125×83 color image (left) and its segmentation (right). Parameter setting: $\sigma = 0.11$. Minimal region size of interest is 16.

6. 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 clustering as well as image segmentation problems. We have also established an intriguing 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 for intensity and color image seg-

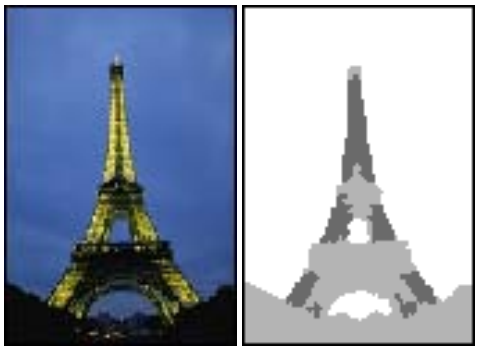

Figure 7. A 121×82 color image (left) and its segmentation (right). Parameter settings: $\sigma = 0.325$. Minimal region size of interest is 10.

mentation. The framework, however, is general and can be applied in a variety of computer vision and pattern recognition domains such as, for example, texture segmentation, perceptual grouping, and the unsupervised organization of an image database. All this will be the subject of future work.

References

[1] J. G. Auguston and J. Minker. An analysis of some graph theoretical clustering techniques. *J. ACM*, 17(4):571–588, 1970.

[2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice-Hall, Englewood Cliffs, NJ, 1982.

[3] 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. Amer. Math. Soc.*, 73:360–363, 1967.

[4] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. Preprint available at http://www.cs.cornell.edu/∼dph/papers/seg-revised.pdf.

[5] R. A. Fisher. *The Genetical Theory of Natural Selection*. Oxford University Press, London, UK, 1930.

[6] D. Forsyth and J. Ponce. *Computer Vision: A Modern Approach*. Prentice-Hall, Englewood Cliffs, NJ, 2002.

[7] 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 Anal. Machine Intell.*, 23(10):1053–1074, 2001.

[8] L. E. Gibbons, D. W. Hearn, P. M. Pardalos, and M. V. Ramana. Continuous characterizations of the maximum clique problem. *Math. Oper. Res.*, 22:754–768, 1997.

[9] C. C. Gotlieb and S. Kumar. Semantic clustering of index terms. *J. ACM*, 15(4):493–513, 1968.

[10] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics*. Cambridge University Press, Cambridge, UK, 1998.

[11] J. J. Hopfield and D. W. Tank. Neural computation of decisions in optimization problems. *Biol. Cybern.*, 52:141–152, 1985.

[12] A. K. Jain and R. C. Dubes. *Algorithms for Clustering Data*. Prentice Hall, Englewood Cliffs, NJ, 1988.

[13] V. Losert and E. Akin. Dynamics of games and genes: Discrete versus continuous time. *J. Math. Biol.*, 17:241–251, 1983.

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

[15] Y. Lyubich, G. D. Maistrowskii, and Y. G. Ol'khovskii. Selection-induced convergence to equilibrium in a single-locus autosomal population. *Problems of Information Transmission*, 16:66–75, 1980.

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

[17] M. Pavan, M. Pelillo, and E. Jabara. On the combinatorics of standard quadratic optimization. Forthcoming, 2003.

[18] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *J. Math. Imaging Vision*, 7(4):309–323, 1997.

[19] M. Pelillo and A. Jagota. Feasible and infeasible maxima in a quadratic program for maximum clique. *J. Artif. Neural Networks*, 2:411–420, 1995.

[20] P. Perona and W. Freeman. A factorization approach to grouping. In H. Burkhardt and B. Neumann, editors, *Computer Vision—ECCV'98*, pages 655–670. Springer-Verlag, Berlin, 1998.

[21] V. V. Raghavan and C. T. Yu. A comparison of the stability characteristics of some graph theoretic clustering methods. *IEEE Trans. Pattern Anal. Machine Intell.*, 3:393–402, 1981.

[22] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *IEEE Trans. Syst. Man & Cybern.*, 6:420–433, 1976.

[23] S. Sarkar and K. L. Boyer. Quantitative measures of change based on feature organization: Eigenvalues and eigenvectors. *Computer Vision and Image Understanding*, 71(1):110–136, 1998.

[24] J. Shi and J. Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 22(8):888–905, 2000.

[25] J. W. Weibull. *Evolutionary Game Theory*. MIT Press, Cambridge, MA, 1995.

[26] Z. Wu and R. Leahy. An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation. *IEEE Trans. Pattern Anal. Machine Intell.*, 15(11):1101–1113, 1993.

[27] C. T. Zahn. Graph-theoretic methods for detecting and describing gestalt clusters. *IEEE Trans. Comput.*, 20:68–86, 1971.

[28] S. W. Zucker, A. Dobbins, and L. Iverson. Two stages of curve detection suggest two styles of visual computation. *Neural Computation*, 1:68–81, 1989.