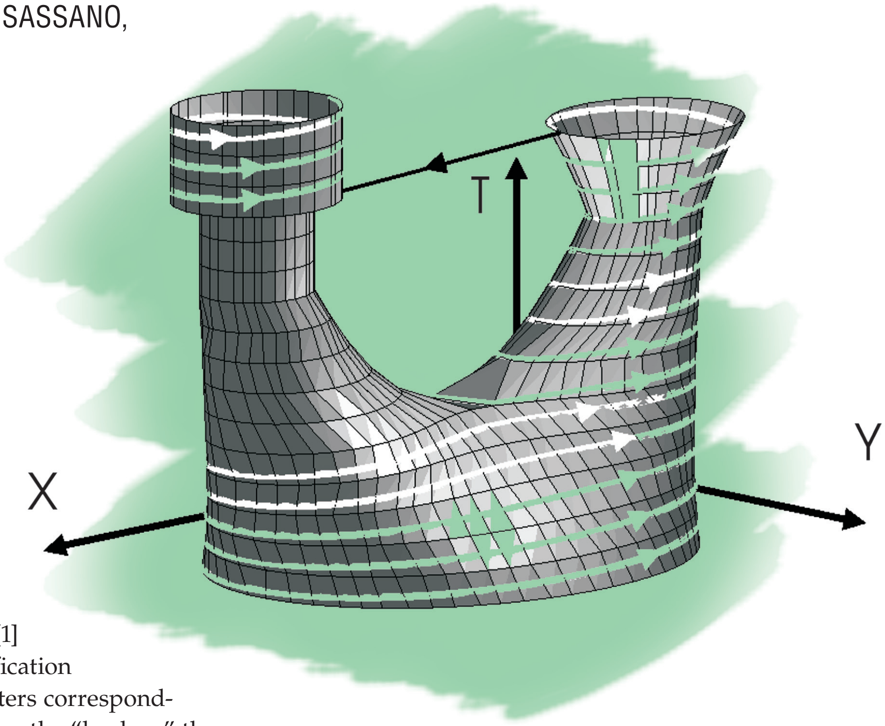
Hamiltonian-Based Clustering

ALGORITHMS FOR STATIC AND DYNAMIC CLUSTERING IN DATA MINING AND IMAGE PROCESSING

DANIELE CASAGRANDE, MARIO SASSANO,
and ALESSANDRO ASTOLFI



The large amount of data available for analysis and management raises the need for defining, determining, and extracting meaningful information from the data. Hence in scientific, engineering, and economics studies, the practice of clustering data arises naturally when sets of data have to be divided into subgroups with the aim of possibly deducting common features for data belonging to the same subgroup. For instance, the innovation scoreboard [1] (see Figure 1) allows for the classification of the countries into four main clusters corresponding to the level of innovation defining the "leaders," the "followers," the "trailing," and the "catching up" countries. Many other disciplines may require or take advantage of a clustering of data, from market research [2] to gene expression analysis [3], from biology to image processing [4]–[7]. Therefore, several clustering techniques have been developed (for details see "Review of Clustering Algorithms").

The easiest way to represent the data to be clustered is by associating each datum with a point in a given space; therefore, clustering can be performed on the basis of the displacement of these points. In fact, clustering algorithms are explicitly or implicitly connected to some definition of proximity measure. Although the Euclidean norm is often used, in particular circumstances an alternative norm may yield better solutions [8].

Digital Object Identifier 10.1109/MCS.2012.2196321
Date of publication: 12 July 2012

Herein a distance-based clustering method is described. While performing the clustering, the algorithm identifies the regions of interest surrounding the points belonging to the same cluster. This idea is developed by making use of the concepts of level function, level set, and level lines; each cluster is obtained by grouping together points belonging to the same connected region of the level set. Moreover, the level function is interpreted as the Hamiltonian function of a Hamiltonian dynamical system. The interesting fact associated with the definition of a level line as the trajectory of a Hamiltonian system is that with the integration of the Hamiltonian system, some particular geometrical features associated with the connected region internal to the level line can be determined. These features, called moments, contain information that is used to develop a clustering procedure for data sets changing on time.

1066-033X/12/$31.00©2012IEEE

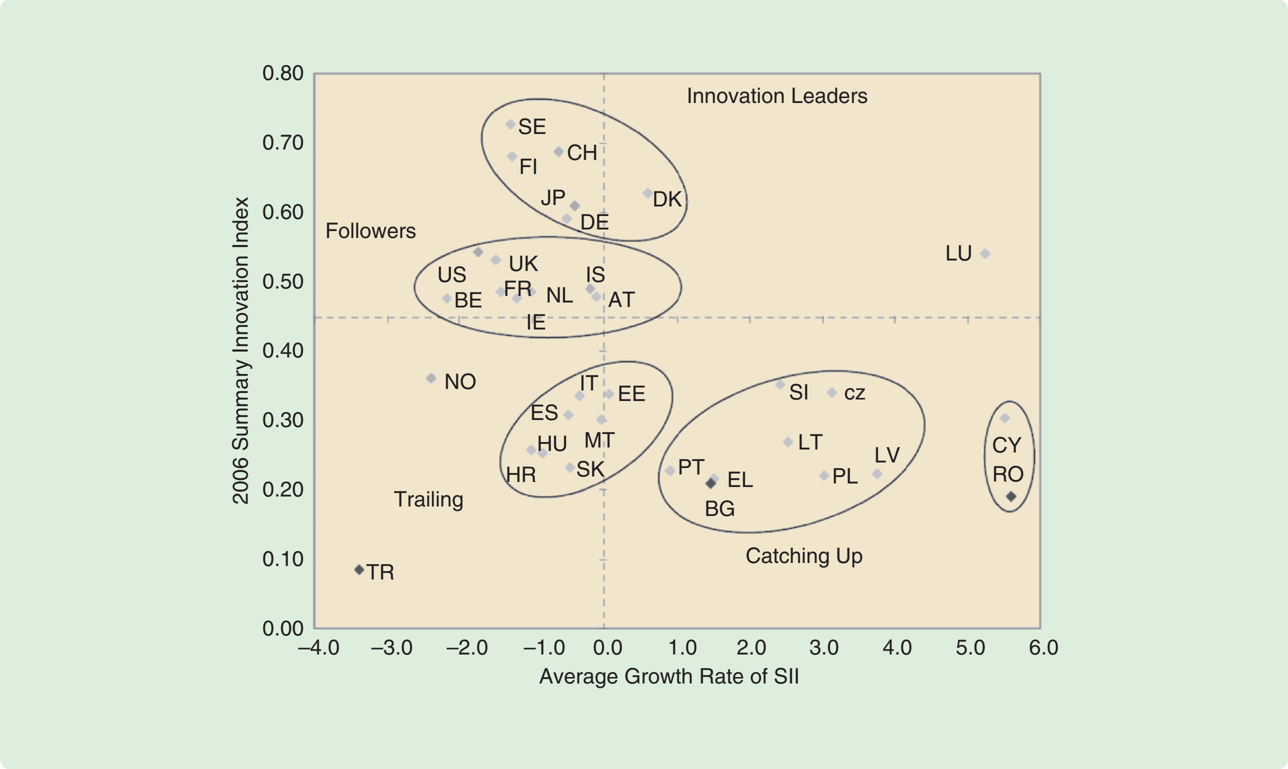


FIGURE 1 Innovation scoreboard of EU countries. EU countries are represented on a Cartesian plane according to their summary innovation index (vertical axis) and their annual growth rate (horizontal axis). Four main clusters can be identified.

The article is organized as follows. After describing the two-dimensional Hamiltonian-based clustering algorithm, the discrete-time dynamic clustering is developed, together with the case of the time-dependent Hamiltonian function. Subsequently, the n-dimensional version of the method is presented. Finally, examples and applications of the methods are given, and open problems and future developments are described.

HAMILTONIAN-BASED ALGORITHM

This section describes a clustering algorithm based on the notions of level function, level set, and level lines.

Definition 1

A *level function* is a continuous function $H:\mathbb{K}\to\mathbb{R}$, with compact support $\mathbb{K}\subset\mathbb{R}^2$. A *(super) level set* is a set $\mathcal{L}\subset\mathbb{K}$ such that, for all $\boldsymbol{\xi}\in\mathcal{L}$, $H(\boldsymbol{\xi})$ is larger than a given value. More precisely, the level set corresponding to the level H_r, is the set $\{\boldsymbol{\xi}\in\mathbb{K}:H(\boldsymbol{\xi})\geq H_r\}$. In general, a level set may be the disjoint union of more than one connected set, hence the following definitions are useful. A connected set $\mathcal{R}\subset\mathcal{L}$ is a *maximal connected component (MCC)* of the level set $\mathcal{L}$ if for each connected set $\mathcal{R}_0\neq\mathcal{R}$ such that $\mathcal{R}\subset\mathcal{R}_0$, there exists $\boldsymbol{\xi}_0\in\mathcal{R}_0\backslash\mathcal{R}$ such that $\boldsymbol{\xi}_0\notin\mathcal{L}$. The boundary of each maximal connected component of the level set is a *level line*.

Consider Figure 2, and let the level set $\mathcal{L}$ be denoted by the shaded areas. Then the set $\mathcal{R}$, delimited by the dotted line, is not an MCC. In fact the set $\mathcal{R}_0$, denoted by the dashed line, is such that $\mathcal{R}_0\neq\mathcal{R}$ and $\mathcal{R}\subset\mathcal{R}_0$ but all the points in $\mathcal{R}_0\backslash\mathcal{R}$ belong to $\mathcal{L}$. On the other hand, the sets $\mathcal{R}_1$ and $\mathcal{R}_2$ are MCCs of $\mathcal{L}$. In particular, $\mathcal{L}=\mathcal{R}_1\cup\mathcal{R}_2$.

The graph of the function H is a surface in $\mathbb{R}^3$ that may exhibit local maxima or local minima. Given a value H_r, included between the highest and the lowest quotes of the graph, all the points of the graph can be connected having the same quote H_r. The set of these points results in a number of lines in $\mathbb{R}^3$; the projection of each of these

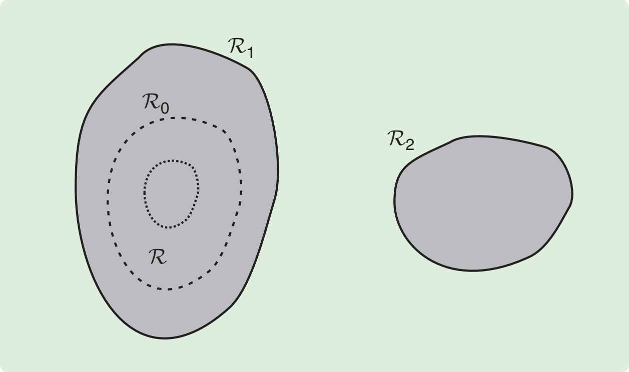


FIGURE 2 Level sets and maximal connected components. A level set $\mathcal{L}$ (shaded area) in general consists of more than one maximal connected component. In this figure it consists of two maximal connected components (MCCs) ($\mathcal{R}_1$ and $\mathcal{R}_2$). $\mathcal{R}$, delimited by the dotted line, is not an MCC of $\mathcal{L}$; in fact $\mathcal{R}\subsetneq\mathcal{R}_0\subset\mathcal{R}_1$.

Review of Clustering Algorithms

Clustering algorithms are divided into the following principal categories.

- *Hierarchical algorithms* (single and complete link algorithms, CURE [S1], BIRCH [S2], partial least squares [S3], and CHAMELEON [S4]). A measure of similarity and dissimilarity between objects must be defined together with a linkage criterion. Hierarchical clustering algorithms organize data into a hierarchical structure according to the proximity matrix, which contains all the pairwise dissimilarities or similarities between the points in the data set. The results are usually depicted by a binary tree or dendrogram. Finally the clusters are obtained by considering the dendrogram at some fixed level.
  - — *Agglomerative algorithms.* Each data point belongs to a separate cluster, and a series of merge operations is executed until all objects belong to the same group.
  - — *Divisive algorithms*. The entire data set belongs to a cluster, and a procedure successively divides it until all clusters are singleton clusters.
- *Partitional algorithms* (k-means, k-medoid clustering [S5]–[S8], k-means with Fisher discriminant analysis [S9], CLARANS [S10]). A single partition of the data is obtained. Partitional methods are suitable for large data sets. The partitional techniques usually produce clusters by optimizing a criterion function. The number of clusters must be fixed a priori.
- *Artificial neural networks* (Kohonen nets [S11], ellipsoid-ART with Mahalanobis distance [S12], [S13]). Two approaches may be distinguished, supervised or unsupervised neural networks. In the former a deterministic mechanism is designed to adjust (training phase) the connection weights for the neurons. The latter consist of a single layer of neurons known as a Kohonen layer and are usually self-organizing networks.
- *Density-based algorithms* (ADACLUS [S14], DBSCAN [S15], DENCLUE [S16]). Density-based algorithms identify clusters as dense regions of objects in the data space separated by regions of low density. The method allows to determine clusters or arbitrary shapes. A density threshold parameter must be fixed a priori.
- *Graph theory-based algorithms* (AMOEBA [S17]). Nodes of a weighted graph describe objects in the feature space while edges quantify the similarity between separate objects. The clustering is performed by determining maximally complete subgraphs (cliques).
- *Kernel-based algorithms* (support vector machine [S18], [S19]). The data points are nonlinearly mapped into a higher dimensional feature space where the data points can be linearly separated.

REFERENCES

[S1] S. Guha, R. Rajeev, and K. Shim, "CURE: An efficient clustering algorithm for large databases," *Inform. Syst*., vol. 26, no. 1, pp. 35–58, 2001.

[S2] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," in *Proc. 1996 ACM SIGMOD Int. Conf. Management of Data,* Montreal, 1996, pp. 103–114.

[S3] J. L. Liu, Y. Bai, J. Kang, and N. An, "A new approach to hierarchical clustering using partial least squares," in *Proc. Int. Conf. Machine Learning and Cybernetics*, 2006, pp. 1125–1131.

[S4] G. Karypis, E. H. Han, and V. Kumar, "CHAMELEON: A hierarchical clustering algorithm using dynamic modeling," *IEEE Comput*., vol. 32, no. 8, pp. 68–75, 1999.

[S5] A. K. Jain, M. N. Murty, and P. J. Flynn, "Data clustering: A review," *ACM Comput. Surveys*, vol. 31, no. 3, pp. 264–323, 1999.

[S6] J. McQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, 1967, pp. 281–297.

[S7] Z. Huang, "Extension to the k-means algorithm for clustering data sets with categorical values," *Data Mining Knowl. Discov*., vol. 2, no. 3, pp. 283–304, 1998.

[S8] K. Krishna and M. M. Narasimha Murty, "Genetic K-means algorithm," *IEEE Trans. Syst., Man, Cybern. B*, vol. 29, no. 3, pp. 433–439, 1999.

[S9] D. A. Clausi, "K-means iterative Fisher (KIF) unsupervised clustering algorithm applied to image texture segmentation," *Pattern Recognit*., vol. 35, no. 9, pp. 1959–1972, 2002.

[S10] R. Ng and J. Han, "Efficient and effective clustering method for spatial data mining," in *Proc. 20th Very Large Data Base Conf*., 1994, pp. 144–155.

[S11] N. R. Pal, J. C. Bezdek, and E. C. K. Tsao, "Generalized clustering networks and Kohonen's self-organizing scheme," *IEEE Trans. Neural Netw*., vol. 4, no. 4, pp. 549–557, 1993.

[S12] G. C. Anagnostopoulos and M. Georgiopoulos, "Ellipsoid ART and ARTMAP for incremental clustering and classification," in *Proc. Int. Joint Conf. Neural Networks*, 2001, pp. 1221–1226.

[S13] R. Xu and D. Wunsch, "Survey of clustering algorithms," *IEEE Trans. Neural Netw*., vol. 16, no. 3, pp. 645–678, 2005.

[S14] G. V. Nosovskiy, D. Liu, and O. Sourina, "Automatic clustering and boundary detection algorithm based on adaptive influence function," *Pattern Recognit*., vol. 41, no. 9, pp. 2757–2776, 2008.

[S15] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowledge Discovery and Data Mining*, 1996, pp. 226–231.

[S16] A. Hinneburg and D. A. Keim, "An efficient approach to clustering in large multimedia databases with noise," in *Proc. 4th Int. Conf. Knowledge Discovery and Data Mining*, 1998, pp. 58–65.

[S17] V. Estivill-Castro and I. Lee, "AMOEBA: hierarchical clustering based on spatial proximity using Delaunay diagram," in *Proc. 9th Int. Symp. Spatial Data Handling*, 2000, pp. 26–41.

[S18] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learn*., vol. 20, no. 3, pp. 273–297, 1994.

[S19] A. Ben-Hur, D. Horn, H. T. Siegelmann, and V. Vapnik, "Support vector clustering," *J. Machine Learn. Res*., vol. 2, pp. 125–137, 2001.

lines onto $\mathbb{R}^2$ is a level line corresponding to the level H_r. If a level line is closed and all the internal points belong to the level set, then the set of the internal points is a maximal connected component of the level set. Moreover, if the reference H_r is suitably chosen, the maximal connected components are disjoint and are the projections of pieces of the surface around a local maxima. Now, suppose that the set $\mathcal{P}$ of the data to be clustered is a subset of $\mathbb{R}^2$. The notion of level function may be exploited for clustering purposes. The result is achieved constructing a level function such that the images of data points that share common features lie around the same local maxima. In this case it is possible to find a suitable reference level such that each maximal connected component corresponds to a cluster.

Suppose that $\mathcal{P}$ has N elements. A way to construct the function H is to associate a function $H_i : \mathbb{R}^2 \to \mathbb{R}$ with each datum point $\boldsymbol{\xi}_i \in \mathcal{P}$, $i = 1,\dots,N$. This function represents the information associated with the position of $\boldsymbol{\xi}_i$. The functions H_i, $i = 1,\dots,n$ are then combined to define the overall level function H. The final aim of a clustering method is to merge the information associated with the single data into the information associated with the cluster. As a consequence, the functions H_i need to have their maximum value in $\boldsymbol{\xi}_i$, where the information is maximal, and a decreasing value when increasing the distance from $\boldsymbol{\xi}_i$. Some examples are the cone function, the hat function, and the Gaussian function; see a one-dimensional version in Figure 3.

Since the overall function $H(\boldsymbol{\xi})$ is expected to maintain the entire information provided by each value $H_i(\boldsymbol{\xi})$, then a possible choice is to define H as the p-norm of the vector of all the H_is, i.e.,

$$H(\boldsymbol{\xi}) = \|(H_1(\boldsymbol{\xi}), H_2(\boldsymbol{\xi}), \dots, H_N(\boldsymbol{\xi}))\|_p = \sqrt[p]{\sum_{i=1}^{N} H_i^p(\boldsymbol{\xi})}, \quad (1)$$

with $p = 1,\dots,\infty$, where, for instance, $H_i(\boldsymbol{\xi}) = e^{-\|\boldsymbol{\xi}-\boldsymbol{\xi}_i\|_2^2}$. Specifically, in the following H is defined as the sum, namely the 1-norm, of the H_is, while the single Hamiltonian function is defined as a Gaussian function, that is,

$$H(\boldsymbol{\xi}) = \sum_{i=1}^{N} H_i(\boldsymbol{\xi}) = \sum_{i=1}^{N} e^{-\|\boldsymbol{\xi}-\boldsymbol{\xi}_i\|_2^2}. \quad (2)$$

The Gaussian function is chosen because it is positive and smooth and can be interpreted as a probability density having its maximum in $\boldsymbol{\xi}_i$, which is an effective way of representing the information associated with the data points.

The level function can be interpreted as a clustering function by choosing a reference value H_r and considering a cluster as the set of data points lying within the same MCCs of the level set corresponding to H_r. In Figure 4, for instance, the level function and the value H_r are such that the level set corresponding to H_r is made of three MCCs, namely, $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$, each of which identifies a cluster, namely, $C_1 = \{\boldsymbol{\xi}_1\}$, $C_2 = \{\boldsymbol{\xi}_2, \boldsymbol{\xi}_4, \boldsymbol{\xi}_5, \boldsymbol{\xi}_7, \boldsymbol{\xi}_9\}$, and $C_3 = \{\boldsymbol{\xi}_3, \boldsymbol{\xi}_6, \boldsymbol{\xi}_8, \boldsymbol{\xi}_{10}\}$.

To explicitly find the level line $\mathcal{S}_j$ associated to $\mathcal{R}_j$, the clustering function H is regarded as a Hamiltonian function and the corresponding Hamiltonian system is defined. The simplest form of a planar Hamiltonian system is described by the equations

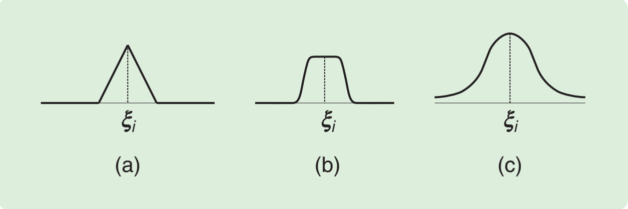


FIGURE 3 Definition of level functions. The figure shows some examples of possible choices for the function H_i, which is the basis for the construction of the overall Hamiltonian function H: (a) a cone function, (b) a hat function, and (c) a Gaussian function.

$$\dot{x} = \frac{\partial H}{\partial y}, \qquad x(0) = x_0,$$

$$\dot{y} = -\frac{\partial H}{\partial x}, \qquad y(0) = y_0,$$

where $(x, y)^\top$ is the state. The time derivative of H is

$$\dot{H} = \frac{\partial H}{\partial x}\dot{x} + \frac{\partial H}{\partial y}\dot{y} = 0, \quad (3)$$

which means that $H(x(t), y(t))$ is constant for all $t \geq 0$. Therefore the trajectory $(x(t), y(t))$ lies on a level line of H. The property $\dot{H} = 0$ holds also for the system

$$\dot{x} = \frac{\partial H}{\partial y} f(x, y), \qquad x(0) = \bar{x}, \quad (4)$$

$$\dot{y} = -\frac{\partial H}{\partial x} f(x, y), \qquad y(0) = \bar{y}, \quad (5)$$

where $f(x, y)$ is any continuous positive function. Hence, denote with x and y the coordinates of $\boldsymbol{\xi}$, that is, $(x, y) = \boldsymbol{\xi}$, and assume that the initial condition $(\bar{x}, \bar{y})$ is such that $(\bar{x}, \bar{y}) \in \mathcal{S}_j$, namely, $H(\bar{x}, \bar{y}) = H_r$. Then the trajectory of system (4)–(5) lies on $\mathcal{S}_j$ for all t. As a matter of fact, the

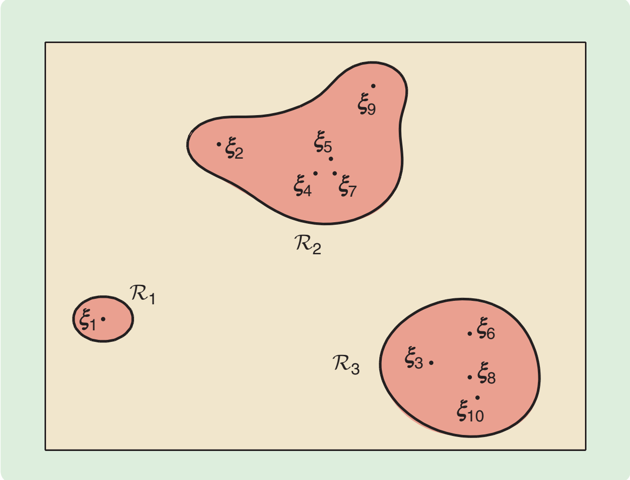


FIGURE 4 Example of the data set. A possible configuration for $\mathcal{P} = \{\boldsymbol{\xi}_1, \dots, \boldsymbol{\xi}_{10}\}$. The level reference H_r is such that the level set is made of three MCCs, namely, $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$. The MCCs identify the clusters $C_1 = \{\boldsymbol{\xi}_1\}$, $C_2 = \{\boldsymbol{\xi}_2, \boldsymbol{\xi}_4, \boldsymbol{\xi}_5, \boldsymbol{\xi}_7, \boldsymbol{\xi}_9\}$, and $C_3 = \{\boldsymbol{\xi}_3, \boldsymbol{\xi}_6, \boldsymbol{\xi}_8, \boldsymbol{\xi}_{10}\}$.

Each cluster is obtained by grouping together points belonging to the same connected region of the level set.

knowledge of an initial condition $\bar{\boldsymbol{\xi}}$ such that $H(\bar{\boldsymbol{\xi}}) = H_r$ is not necessary. In fact, a modified version of system (4)–(5) may be considered. To state the main result associated with the modified system, a preliminary definition and a lemma are needed.

Definition 2

The *diameter* of a set C is $\operatorname{diam}(C) \triangleq \sup\{d(x,y) : x \in C, y \in C\}$, where $d(x,y)$ is the Euclidean distance between x and y.

Lemma 1

Let $\gamma : [0,T] \to \mathbb{R}^2$ be a continuous curve with continuous first-order derivatives, let $L(\gamma)$ be its length and let $(x(t), y(t))$ denote the parametric description of γ, namely, the image in $\mathbb{R}^2$ of $t \in [0,T]$. Moreover, suppose that γ is contained in a compact set $\mathcal{A}$, whose diameter is $\operatorname{diam}(\mathcal{A}) = M$, the equation $\dot{x}(t) = 0$ has at most z_x solutions in $[0,T)$ and the equation $\dot{y}(t) = 0$ has at most z_y solutions in $[0,T)$. Then $L(\gamma) \le M(z_x + z_y + 2)$ (for the proof see "Proof of Lemma 1").

As mentioned previously, the property of the Hamiltonian systems, namely, that $H(t)$ is constant, can be exploited for a Hamiltonian clustering even though the initial condition $\bar{\boldsymbol{\xi}}$ is such that $H(\bar{\boldsymbol{\xi}}) \ne H_r$, as stated by the following result.

Theorem 1

Let H be defined by (2). For each initial condition $(x(0), y(0))$, the dynamical system

$$\dot{x} = f(x,y)\left(\frac{\partial H}{\partial y} - \frac{\partial H}{\partial x}(H - H_r)^{1/3}\right), \tag{6}$$

$$\dot{y} = -f(x,y)\left(\frac{\partial H}{\partial x} + \frac{\partial H}{\partial y}(H - H_r)^{1/3}\right), \tag{7}$$

with the positive-definite function $f(x,y)$ defined as

$$f(x,y) = \begin{cases} \left(\left(\dfrac{\partial H}{\partial x}\right)^2 + \left(\dfrac{\partial H}{\partial y}\right)^2\right)^{-1}, & \text{if } H(x,y) \ne H_r, \\ \left(\left(\dfrac{\partial H}{\partial x}\right)^2 + \left(\dfrac{\partial H}{\partial y}\right)^2\right)^{-1/2}, & \text{if } H(x,y) = H_r \end{cases}$$

provided the trajectory satisfies $(\partial H/\partial x)^2 + (\partial H/\partial y)^2 \ne 0$ for all t, is such that the state of the system is driven to a level line $\mathcal{S}_j$ corresponding to the level set $H(x,y) = H_r$ in finite time and the level line $\mathcal{S}_j$ is spanned in finite time (for the proof of the theorem see "Proof of Theorem 1").

The time required for the state of the system to converge to $\mathcal{S}_j$ and the time needed to span the level line can be rendered arbitrarily small by multiplying f by a positive constant $k > 1$.

A naive methodology to compute all the level lines consists of integrating the Hamiltonian system (6)–(7) N times, the ith time picking as initial condition a point close to the datum point $\boldsymbol{\xi}_i \in \mathcal{P}$. Each $\boldsymbol{\xi}_i$, in fact, belongs to some connected subset $\mathcal{R}_j$ of the level set and, therefore, is internal to the level line $\mathcal{S}_j$. However, this exhaustive approach is not very efficient as it requires to compute the same level line $\mathcal{S}_j$ as many times as there are data points in $\mathcal{R}_j$. A more efficient algorithm makes use of the concept of winding number [9] of a point $\boldsymbol{\xi}_i = (x_i, y_i)$ with respect to the closed curve $\mathcal{S}_j$, defined as

$$w(\boldsymbol{\xi}_i, \mathcal{S}_j) \triangleq \frac{1}{2\pi}\oint_{\mathcal{S}_j} \frac{-(y - y_i)dx + (x - x_i)dy}{(x - x_i)^2 + (y - y_i)^2}.$$

The above ideas lead to the following clustering algorithm, which terminates after a finite number of iterations.

- Step 1: Let $i = 1$ and $j = 1$.
- Step 2: Integrate system (6)–(7) with $\boldsymbol{\xi}(0) = \boldsymbol{\xi}_i$. For all $k = i+1, \ldots, N$, compute $w(\boldsymbol{\xi}_i, \mathcal{S}_j)$.
- Step 3: For all $k = i+1, \ldots, N$, if $w(\boldsymbol{\xi}_i, \mathcal{S}_j) = 1$ then $\mathcal{P} \leftarrow \mathcal{P} \backslash \{\boldsymbol{\xi}_i\}$.
- Step 4: If $\mathcal{P} \ne \emptyset$, then go to Step 2 with $i \leftarrow i+1$ and $j \leftarrow j+1$. Otherwise STOP.

Note that Step 2 allows to compute the boundary $\mathcal{S}_j$ of the connected set to which $\boldsymbol{\xi}_i$ belongs. Note also that

Proof of Lemma 1

The length of the curve γ is given by

$$L(\gamma) = \int_0^T \sqrt{\dot{x}^2 + \dot{y}^2}\,dt \le \int_0^T (|\dot{x}(t)| + |\dot{y}(t)|)dt.$$

If $\gamma \subset \mathcal{A}$, then the length spanned in the x direction in a time interval in which $\dot{x}$ does not change sign is bounded by M. Thus, if $\dot{x}$ changes sign at most z_x times, then

$$\int_0^T |\dot{x}(t)|dt \le M(z_x + 1).$$

An analogous result holds for y and the claim follows immediately. ■

Proof of Theorem 1

The first claim holds thanks to the fact that the term $-(H-H_r)^{1/3}$ has a sign that steers the trajectory towards a line corresponding to H_r. Moreover, the steering term vanishes when $H = H_r$. In particular, the time derivative of the Hamiltonian function yields the dynamical system

$$\begin{aligned}\dot{H} &= \frac{\partial H}{\partial x}\dot{x} + \frac{\partial H}{\partial y}\dot{y} = -\left(\left(\frac{\partial H}{\partial x}\right)^2 + \left(\frac{\partial H}{\partial y}\right)^2\right)(H-H_r)^{1/3} f(x,y) \\ &= -(H-H_r)^{1/3}, \end{aligned} \tag{S1}$$

the solution of which is

$$H(t) = \begin{cases} H_r + \operatorname{sign}(H(0)-H_r)\sqrt{\left(|H(0)-H_r|^{2/3} - \frac{2}{3}t\right)^3}, & \text{for } 0 \le t \le \frac{3}{2}|H(0)-H_r|^{2/3}, \\ H_r, & \text{for } t > \frac{3}{2}|H(0)-H_r|^{2/3}, \end{cases}$$

hence $H(t) = H_r$ for $t \ge 3/2|H(0)-H_r|^{2/3}$ and the system (6)–(7) reduces to (4)–(5). The computation of the linear velocity of the point $\boldsymbol{\xi}(t)$ along the trajectory of system (4)–(5) yields

$$\dot{s} = \sqrt{\dot{x}^2 + \dot{y}^2} = \sqrt{\left(\frac{\partial H}{\partial x}\right)^2 f(x,y)^2 + \left(\frac{\partial H}{\partial y}\right)^2 f(x,y)^2} = 1,$$

hence the level line is spanned with constant unitary velocity. If H is as in (2), if the data points belong to some compact set $\mathbb{K}$ and if $\mathbb{K}$ is a subset of the set $\{\boldsymbol{\xi} \in \mathbb{R}^2 : d(\boldsymbol{\xi}, 0) \le R\}$, for some $R \in \mathbb{R}, R > 0$, then $\mathcal{S}_j$ is a subset of $\{\boldsymbol{\xi} \in \mathbb{R}^2 : d(\xi, 0) \le R'\}$ with $R' = R + \sqrt{\ln(N/H_r)}$. In fact, consider a point $\boldsymbol{\zeta}$ such, that $d(\boldsymbol{\zeta}, 0) > R'$; since $d(\boldsymbol{\xi}_i, 0) \le R$, for all $i = 1, \ldots, N$, then $\rho_i(\boldsymbol{\zeta}) > \sqrt{\ln(N/H_r)}$. Hence $H_i(\boldsymbol{\zeta}) < e^{-(\ln(N/H_r))} = H_r/N$ and $H(\boldsymbol{\zeta}) = \sum_{i-1}^{N} H_i(\boldsymbol{\zeta}) < N(H_r/N) = H_r$. Therefore $\boldsymbol{\zeta} \notin \mathcal{S}_j$. Thus the first hypothesis in Lemma 1 holds. Moreover, due to the choice of the Hamiltonian function, the second hypothesis in Lemma 1 also holds and it can be concluded that the length of $\mathcal{S}_j$ is finite. Hence, since $\mathcal{S}_j$ is spanned with a constant velocity, the level line is spanned in finite time. ■

Step 3 makes use of the notion of the winding number to delete, form the set of data, the points that have already been classified as internal to a level line.

DISCRETE-TIME MODEL OF THE CLUSTER DYNAMICS

The algorithm described in the previous section can be exploited to model the dynamics of the clusters in two different ways, according to the time scale considered. The discrete-time approach, explained in this section, and the continuous-time approach, to which the next section is dedicated, lead to paradigms that are different not only in the time scale but also in the way the static algorithm is used.

In this section the discrete-time motion of a cluster is reconstructed by iterative applications of the static algorithm. In particular, various time instants may be associated with data sets that differ for the number and the position of the data points. More precisely, consider a sequence of time instants $\mathcal{T} = \{\tau_k\}_{k \in \mathbb{N}}$, and let $\mathcal{P}(\tau_{k_1})$ and $\mathcal{P}(\tau_{k_1+1})$ denote the sets of data points at the time instants τ_{k_1} and τ_{k_1+1}, respectively. Since the position of the data points at time τ_{k_1+1} is, in general, different from their position at time τ_{k_1}, the number of MCCs of the level set may also be different. Let $\mathcal{R}_1(\tau_k), \ldots, \mathcal{R}_{N(\tau_k)}(\tau_k)$ denote the MCCs of the level set at time τ_k. Suppose that $\mathcal{G} \in \{\mathcal{R}_1(\tau_k), \ldots, \mathcal{R}_{N(\tau_k)}(\tau_k)\}$ is an MCCs and consider the problem of associating $\mathcal{G}$ with an MCC $\mathcal{F} \in \{\mathcal{R}_1(\tau_{k-1}), \ldots, \mathcal{R}_{N(\tau_{k-1})}(\tau_{k-1})\}$ in such a way that $\mathcal{G}$ is the time-evolution of $\mathcal{F}$.

Obviously, to begin with the concept of time evolution needs to be specified. A possible solution exploits the notion of geometric moments of a two-dimensional region (see "Geometric Moments").

Computation of the Moments

The moments of any order of $\mathcal{F}_{\mathcal{R}_j}$ can be computed by integrating (6)–(7). In fact, by using Green's theorem a double integral over a region is equal to a line integral over its boundary.

Theorem 2 (Green's)

Let $\mathcal{S}$ be a positively oriented, piecewise smooth, simple closed curve in $\mathbb{R}^2$, and let $\mathcal{R}$ be the region bounded by $\mathcal{S}$. If L and M are functions of x and y defined on an open region containing $\mathcal{R}$, and with continuous partial derivatives in $\mathcal{R}$, then

$$\iint_{\mathcal{R}} \left(\frac{\partial M}{\partial x} - \frac{\partial L}{\partial y}\right) \mathrm{d}x\mathrm{d}y = \int_{\mathcal{S}} (L\mathrm{d}x + M\mathrm{d}y).$$

By applying Theorem 2 and selecting, for instance, $(\partial M/\partial x) = x^p y^q$ and $L = 0$ one obtains

$$\iint_{\mathcal{R}_j} x^p y^q \mathrm{d}x\mathrm{d}y = \int_{\mathcal{S}_j} \frac{1}{p+1} x^{p+1} y^q \mathrm{d}y, \tag{8}$$

where the term on the right-hand side is an integral along a (closed) trajectory $\mathcal{S}_j$ corresponding to one of the level lines of the clustering function H. From (8) the time

Geometric Moments

DEFINITION 4

For a function $F(x, y): \mathbb{R}^2 \to \mathbb{R}$ the *moment* of order $p+q$, with p and q integers and such that $p+q \geq 0$, is defined as

$$m_{pq}(F) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x^p y^q F(x, y)\, \mathrm{d}x\mathrm{d}y. \qquad \text{(S1)}$$

Considering the MCC $\mathcal{R}_j$ associated with the cluster C_j, the *index* function of $\mathcal{R}_j$ is

$$F_{\mathcal{R}_j}(x, y) \triangleq \begin{cases} 1 & (x, y) \in \mathcal{R}_j, \\ 0 & \text{otherwise.} \end{cases} \qquad \text{(S2)}$$

From(S1) and (S2), it is possible to relate the moments of order zero and one of $F_{\mathcal{R}_j}$ to the position and the size of $F_{\mathcal{R}_j}$. In fact the moment of order 0 of $F_{\mathcal{R}_j}$ is the area of $\mathcal{R}_j$:

$$m_{00}(F_{\mathcal{R}_j}) = \iint_{\mathcal{R}_j} \mathrm{d}x\mathrm{d}y,$$

whereas the position of the center of mass of $\mathcal{R}_j$ can be obtained from the moments of order 0 and 1. Denoting by $x_{\mathcal{R}_j}$ and $y_{\mathcal{R}_j}$ its coordinates, it follows that

$$x_{\mathcal{R}_j} = \frac{m_{10}(F_{\mathcal{R}_j})}{m_{00}(F_{\mathcal{R}_j})}, \qquad y_{\mathcal{R}_j} = \frac{m_{01}(F_{\mathcal{R}_j})}{m_{00}(F_{\mathcal{R}_j})}.$$

A further step is the quantitative description of the shape of the cluster, which can be done with the help of the moments of higher order of $\mathcal{R}_j$ computed with respect to its center of mass, which are called the central moments.

DEFINITION 5

For a function $F(x,y): \mathbb{R}^2 \to \mathbb{R}$, the *central moment* of order $p+q$, with $p+q > 2$, is defined as

$$\mu_{pq}(F) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} x_T^p y_T^q F(x_T, y_T)\, \mathrm{d}x\mathrm{d}y,$$

where $x_T = x - x_{\mathcal{R}_j}$ and $y_T = y - y_{\mathcal{R}_j}$.

The central moments of order two are associated with geometric quantities such as the principal axes of the image ellipse of $\mathcal{R}_j$ (for further details, see [S20]).

REFERENCE

[S20] R. J. Prokop and A. P. Reeves, "A survey of moment-based techniques for unocculted object representation and recognition," *Graph. Models Image Process.*, vol. 54, no. 5, pp. 438–460, 1992.

variation of the value of the moments on the time scale of the Hamiltonian system (4)–(5) can be obtained. In particular, the differential equations

$$\dot{m}_{00} = x\dot{y}, \qquad (9)$$

$$\dot{m}_{10} = \frac{1}{2}x^2\dot{y}, \quad \dot{m}_{01} = xy\dot{y}, \qquad (10)$$

$$\dot{m}_{20} = \frac{1}{3}x^3\dot{y}, \quad \dot{m}_{11} = \frac{1}{2}x^2y\dot{y}, \quad \dot{m}_{02} = xy^2\dot{y}, \qquad (11)$$

with $m_{pq}(0) = 0$, are such that $m_{pq}(\hat{t}) = m_{pq}(F_{\mathcal{R}_j})$, where $\hat{t}$ is the period of the orbit of system (4)–(5) along $\mathcal{S}_j$.

It is worth noting that the differential equations (9)–(11) do not describe the dynamics of the moments on the time scale of the motion of the points $\boldsymbol{\xi}$, but only the variation of their values on the time-scale of the Hamiltonian system (4)–(5), the trajectory of which is the level line $\mathcal{S}_j$.

The Sequence of Clusters

Within the theoretical framework described above, it is now possible to solve the problem posed at the beginning of the section. In fact, moments and central moments provide a quantitative description of the dimension, the position, the orientation, and the shape of the regions containing the clusters. More precisely, the computation of the moments of order zero and order one and the computation of the central moments of order two originates a map $\mathcal{M}$ from the set of all possible shapes in $\mathbb{R}^2$ to a point $\mathcal{M}(\mathcal{R}) = (m_{00}, m_{10}, m_{01}, \mu_{20}, \mu_{11}, \mu_{02}) \in \mathbb{R}^6$. This map may be used to construct a sequence of clusters each of which is the evolution of the previous. Suppose $\tau_{k_1-1}-\tau_{k_1}$ is small with respect to the time variation of the data points. Then a cluster C_j and the region $\mathcal{R}_j$ surrounding it experience small changes between τ_{k_1-1} and τ_{k_1}. As a consequence, the images of the two regions, through $\mathcal{M}$, are not too far one from the other. Now, consider a region $\mathcal{F} \in \{\mathcal{R}_1(\tau_k), \ldots, \mathcal{R}_{N(\tau_k)}(\tau_k)\}$ and an ellipsoid in $\mathbb{R}^6$ having the center of mass in $\mathcal{M}(\mathcal{F})$ and the semiaxis of which are the coordinates of a vector $\mathbf{d} = (d_1, \ldots, d_6) \in \mathbb{R}^6$, namely, the set

$$E_{\mathbf{d}}(\mathcal{F}) = \left\{ (z_1, \ldots, z_6) \in \mathbb{R}^6 : \sum_{i=1}^{6} \frac{(z_i - m_{\mathcal{F},i})^2}{d_i^2} \leqslant 1 \right\},$$

where $m_{\mathcal{F},i}$ denotes the ith component of $\mathcal{M}(\mathcal{F})$. The definition of the ellipsoid $E_{\mathbf{d}}$ allows to formalize the notion of time-evolution of a cluster.

Definition 3

A maximal connected component $\mathcal{G} \in \{\mathcal{R}_1(\tau_{k+1}), \ldots, \mathcal{R}_{N(\tau_{k+1})}(\tau_{k+1})\}$ is the one-step E_d *evolution* of $\mathcal{F} \in \{\mathcal{R}_1(\tau_k), \ldots, \mathcal{R}_{N(\tau_k)}(\tau_k)\}$, which is denoted by $\mathcal{G} \sim_{E_{\mathbf{d}}} \mathcal{F}$, if $\mathcal{M}(\mathcal{G}) \in E_{\mathbf{d}}(\mathcal{F})$.

With the help of Definition 3 the generic K-step evolution from $\mathcal{G}_0 \in \{\mathcal{R}_1(\tau_0), \ldots, \mathcal{R}_{N(\tau_0)}(\tau_0)\}$ can be defined as the sequence $\{\mathcal{G}_k\}_{k \in \{1,\ldots,K\}}$ such that, for $i = 1, \ldots, K$, $\mathcal{G}_i \in \{\mathcal{R}_1(\tau_i), \ldots, \mathcal{R}_{N_{\tau_i}}(\tau_i)\}$ and $\mathcal{G}_i \sim_{E_{\mathbf{d}}} \mathcal{G}_{i-1}$.

CONTINUOUS-TIME MODEL OF THE CLUSTER DYNAMICS

The results presented in the previous section refer to a discrete-time scenario where the sequence of MCCs and of clusters contained in them may be built by repeated application of the static algorithm described above. In this section a continuous-time scenario is considered leading to a different description of the dynamics of the clusters and to an extension of the static algorithm.

Suppose that the data points move in time, which happens, for instance, when the points represent the position on the ground of moving objects. In this situation, not only the clusters change in time and assume different shapes but they may also merge or split, hence their number also varies. Moreover, data points may appear or disappear, meaning that the data set may have different cardinality in different time instants. Thus N is indeed a discrete-valued function $N(\tau)$ of the time variable τ. However, to simplify the reasoning, a constant value of N is initially assumed.

Due to the motion of points, each Hamiltonian function H_i associated to $\boldsymbol{\xi}_i$ depends on time and so does the Hamiltonian function H

$$H(\boldsymbol{\xi},\tau)=\sum_{i=1}^{N}H_i(\boldsymbol{\xi},\tau). \tag{12}$$

From (12) it is evident that while in the static version of the algorithm the level function H is defined on $\mathbb{R}^2$ and the manifolds corresponding to a constant value of H are curves in $\mathbb{R}^2$, in the dynamic version developed herein the time τ is added as a variable for the Hamiltonian function. As a result the manifolds corresponding to a constant value of H, described by the equation $H(\boldsymbol{\xi},\tau)=H_r$, are surfaces in $\mathbb{R}^3$.

To illustrate the surfaces, consider two points that are close to each other at the initial time-instant and are moving in opposite directions. Up to some instant the intersection between the level surface and a plane corresponding to a constant value of τ is a closed curve surrounding the two points [Figure 5(a)]. When the distance between the two points increases, the intersection becomes the union of two distinct closed lines, each of which encloses one of the points. The resulting three-dimensional surface is a cobordism [10] between the initial closed curve surrounding the two points and the final pair of closed lines [Figure 5(b)].

The introduction of time as a new variable generates a port-Hamiltonian system without inputs, namely,

$$\begin{pmatrix}\frac{\mathrm{d}x}{\mathrm{d}t}\\ \frac{\mathrm{d}y}{\mathrm{d}t}\\ \frac{\mathrm{d}\tau}{\mathrm{d}t}\end{pmatrix}=\begin{pmatrix}0 & J_{12} & J_{13}\\ -J_{12} & 0 & J_{23}\\ -J_{13} & -J_{23} & 0\end{pmatrix}\begin{pmatrix}\frac{\partial H}{\partial x}\\ \frac{\partial H}{\partial y}\\ \frac{\partial H}{\partial \tau}\end{pmatrix}=\mathbf{J}\begin{pmatrix}\frac{\partial H}{\partial x}\\ \frac{\partial H}{\partial y}\\ \frac{\partial H}{\partial \tau}\end{pmatrix}. \tag{13}$$

It can be seen that, due to the skew-symmetry of $\mathbf{J}$, $\dot{H}=0$ for all $t\geq 0$, that is, the trajectory of system (13) lies on a surface corresponding to a constant value of H. In the remainder of the section $\mathcal{H}_r$ denotes the surface corresponding to the value H_r, that is, $\mathcal{H}_r \triangleq \{(x,y,\tau)\in\mathbb{R}^3 : H(x,y,\tau)=H_r\}$. As in the static version, since $\dot{H}=0$, if $x(0)$, $y(0)$, and $\tau(0)$ are such that $H(x(0),y(0),\tau(0))\in\mathcal{H}_r$, then $H(x(t),y(t),\tau(t))=H_r$ for all $t>0$.

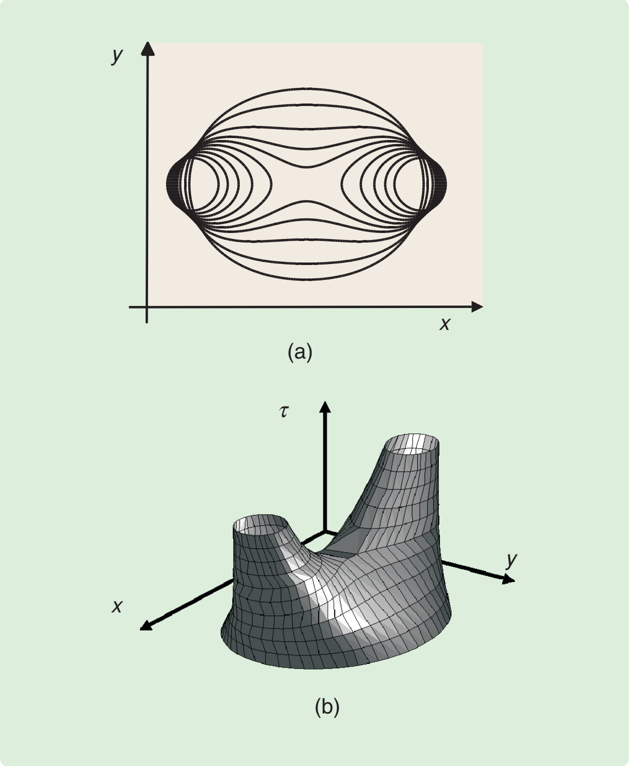


FIGURE 5 Cobordism representing two moving objects. The surface $H(x,y,\tau)=H_r$ corresponds to a pair of points moving in opposite directions. In (a), the projection on the (x,y) plane of the level surface for different values of τ; in (b), the surface in the space (x,y,τ).

The time scale of (13), identified by the variable t, is different from the time scale of the motion of data points. The time-variable τ corresponding to the latter is interpreted as a third variable of the Hamiltonian function, thus originating a vector field in $\mathbb{R}^3$. On the other hand, the time-variable t is the time scale of the trajectory of the state of the system which lies on $\mathcal{H}_r$. The two time scales coincide if and only if J_{13} and J_{23} are such that $\dot{\tau}=1$.

The parameters J_{ij} can be arbitrarily chosen and determine the directions, on $\mathcal{H}_r$, of the trajectories of the system. Therefore, various cases can be considered.

Real-Time Trajectory

If J_{13} and J_{23} are such that $\dot{\tau}=1$, then the two time scales t and τ coincide, which implies that the state of system (13) runs along $\mathcal{H}_r$ in real time, namely, with no delay with respect to the dynamics of the measurements. This situation can be achieved, for instance, by setting

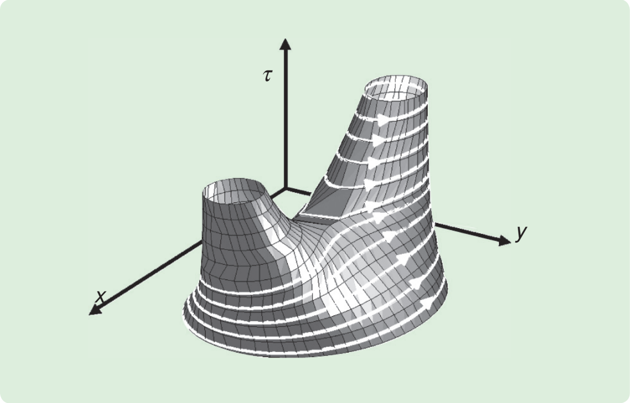


FIGURE 6 Real-time evolution. The trajectory (white line) of a time-varying port-Hamiltonian system twists around the level surface $\mathcal{H}_r$ in real time.

$$J_{13} = \frac{-\dfrac{\partial H}{\partial x}}{\left(\dfrac{\partial H}{\partial x}\right)^2 + \left(\dfrac{\partial H}{\partial y}\right)^2}, \quad J_{23} = \frac{-\dfrac{\partial H}{\partial y}}{\left(\dfrac{\partial H}{\partial x}\right)^2 + \left(\dfrac{\partial H}{\partial y}\right)^2},$$

provided that the trajectory is such that $(\partial H/\partial x)^2 + (\partial H/\partial y)^2 \neq 0$ for all $t \geq 0$. In this case, the remaining parameter J_{12} can be used to determine how rapidly the trajectory wraps around $\mathcal{H}_r$. Figure 6 refers to the same setting considered in Figure 5(a). The surface $\mathcal{H}_r$ "grows" in time along the vertical axis, while the trajectory of (13) wraps around $\mathcal{H}_r$.

Obviously, when the level surface splits into two branches, the trajectory of the port-Hamiltonian system can follow only one of them. Therefore, to have a complete approximation of $\mathcal{H}_r$ another port-Hamiltonian system that follows the other branch needs to be considered.

The Static Case

If J_{13} and J_{23} are such that $\dot{\tau} = 0$, then $\tau(t) = \tau(0)$ for all $t \geq 0$ and the trajectory of system (13) lies on a plane corresponding to a constant value of time, as in the static case. Set, for instance,

$$J_{13} = \frac{\partial H}{\partial y}, \quad J_{23} = -\frac{\partial H}{\partial x}.$$

As in the previous case, the remaining parameter J_{12} can be used to determine how rapidly the trajectory "wraps around" the surface corresponding to $\mathcal{H}_r$.

Time-Dependent Cardinality of the Data Set

Suppose now that the number of points in the data set is a function of time, $N(\tau)$. This may happen, for instance, when the data points represent objects moving in the two-dimensional plane and not all objects are detected at all time instants. Specifically, an object may appear at a particular time-instant $\overline{\tau}_a$, while it is not detected for $\tau < \overline{\tau}_a$. On the other hand, a point may disappear at the time-instant $\overline{\tau}_d$, while it is detected for $\tau < \overline{\tau}_d$. This assumption has two main consequences on the structure of the Hamiltonian function. To begin with the total number of objects is indeed a function of time $N(\tau)$. This occurrence has to be taken into careful account since, due to the definition of H given by (12), it results in a discontinuity of its time derivative. In addition, in the case of a disappearing point, a method to propagate in the future the information provided up to the time $\overline{\tau}_a$must be designed.

Suppose that a moving object, identified by the point $\boldsymbol{\xi}_{i^*}(\tau)$, is tracked up to the time-instant τ_{d,i^*} and suddenly disappears. If the velocity of the object is known to be bounded from above by a value V_u, then at time $\tau > \tau_{d,i^*}$ the actual position of the object can be any point in a ball centered in $\boldsymbol{\xi}_{i^*}(\tau_k)$ and of radius $V_u(\tau - \tau_{d,i^*})$. To take this occurrence into account, consider the Heaviside function $\eta : \mathbb{R} \rightarrow \{0,1\}$, defined by $\eta(t) = 0$ if $t < 0$ and $\eta(t) = 1$ if $t \geq 0$, and add to the summation (12) the function $L_{i^*}(\boldsymbol{\xi}, \tau) = H_{i^*}(\boldsymbol{\xi}, \tau) e^{V_u(\tau - \tau_{d,i^*})} \eta(\tau - \tau_{d,i^*})$, that behaves as a storing function. Note that $L_{i^*}(\boldsymbol{\xi}, \tau_{d,i^*}) = H_{i^*}(\boldsymbol{\xi}, \tau_{d,i^*})$, hence the continuity of the total clustering function H is guaranteed at the disappearing time instants.

Furthermore, it is reasonable to assume that the information corresponding to a disappeared point does not need to be carried on forever. The natural way to deal with this requirement is to forget the point $\boldsymbol{\xi}_{i^*}$ after a finite time interval T_R. To this purpose, the storing function can be modified as $L_{i^*}(\boldsymbol{\xi}, \tau) = H_{i^*}(\boldsymbol{\xi}, \tau) e^{V_u(\tau - \tau_{d,i^*})} \mu_{i^*}(\tau)$, where $\mu_{i^*}(\tau) = \eta(\tau - \tau_{d,i^*}) - \eta(\tau - \tau_{d,i^*} - T_R)$.

To cope with the fact that the number of points changes in time, and so does the maximum value of the level function, a normalizing factor has to be defined. This factor can be the inverse of the number of points considered. Denoting by $M(\tau)$ the number of points that disappear but are stored, then the final expression of the clustering function is

$$H(\boldsymbol{\xi}, \tau) = \frac{1}{N(\tau) + M(\tau)} \left(\sum_{i=1}^{N(\tau)} H_i(\boldsymbol{\xi}, \tau) + \sum_{j=1}^{M(\tau)} L_j(\boldsymbol{\xi}, \tau) \right).$$

From the previous considerations it can be deduced that the total Hamiltonian function is discontinuous in the time instants that coincide with the appearance and with the discarding of a data point. In fact, if a new point is detected at time τ_a, then $N(\tau_a^+) = N(\tau_a^-) + 1$. Analogously, if a point is discarded after being stored for T_R time units, at time $\tau_d + T_R$, then $M(\tau_d + T_R^+) = M(\tau_d + T_R^-) - 1$. Nevertheless the discontinuity generated by the appearance of the ith point can be overcome. Consider the function

$$f_s(\tau) \triangleq \begin{cases} 0 & \text{for } \tau < 0, \\ 4\tau^3 & \text{for } \tau \in \left(0, \dfrac{1}{2}\right), \\ 4(\tau - 1)^3 + 1 & \text{for } \tau \in \left(\dfrac{1}{2}, 1\right), \\ 1 & \text{for } \tau \geq 1. \end{cases} \tag{14}$$

where $f_s(\tau)$ is continuous and with continuous first-order derivatives for all $\tau \in \mathbb{R}$. Therefore, denoting by $\tau_{a,i}$ the time instant in which the ith point appears, the function

The level function is interpreted as the Hamiltonian function of a Hamiltonian dynamical system.

$H_i(\boldsymbol{\xi}, \tau) = e^{-\rho_i(\boldsymbol{\xi})} f_s(\tau - \tau_{a,i})$ is a continuous function with continuous first-order derivatives in all its variables. The introduction of the factor f_s has the effect of transforming the appearance of the ith point into a fading appearance. Moreover, the time to completely appear, which in (14) is one time-unit, can be rendered arbitrarily small by a proper scaling.

As a consequence of the introduction of the function (14) the level line defined by $H_i(\boldsymbol{\xi}, \tau) = H_r$ coincides with $\boldsymbol{\xi}_i$ at the initial time-instant and for the following time-instants is a circle the radius of which grows continuously. This growing process is coherent with the fact that each measurement can be interpreted as exact at the appearing time-instant while its position in the following instants may be affected by some tracking error, thus originating a growing region of interest surrounding the point. However, function (14) cannot be used to discard the measurement in a faded way. In fact, a reasoning identical to the appearance situation implies that at the very last time instant the level line degenerates into a point, which is not coherent with a loss of information about the position of the point.

Initial Conditions Not on the Level Set

Analogously to the static case described above, when the initial condition does not belong to the level surface $\mathcal{H}_r$, modified dynamics need to be considered to steer the state of the Hamiltonian system to $\mathcal{H}_r$. In particular, the solution of the system described by

$$\dot{x} = -J_0(H(x, y, \tau) - H_r)^{1/3}\frac{\partial H}{\partial x}, \tag{15}$$

$$\dot{y} = -J_0(H(x, y, \tau) - H_r)^{1/3}\frac{\partial H}{\partial y}, \tag{16}$$

$$\dot{\tau} = -J_0(H(x, y, \tau) - H_r)^{1/3}\frac{\partial H}{\partial \tau}, \tag{17}$$

with $J_0 = ((\partial H/\partial x)^2 + (\partial H/\partial y)^2 + (\partial H/\partial \tau)^2)^{-1}$ converges in finite time to the value H_r, provided that the trajectory does not go through singular points of H.

On the basis of these considerations, the following two-step dynamics can be designed to reconstruct the surface $\mathcal{H}_r = \{(x, y, \tau) : H(x, y, \tau) = H_r\}$.

1) If $H(x, y, \tau) \neq H_r$ (that is, the state is not on the level surface, for instance in time instants corresponding to appearing and disappearing points), the trajectory of the port-Hamiltonian system is determined by (15)–(17).
2) If $H(x, y, \tau) = H_r$, the trajectory is determined by (13).

HIGHER DIMENSIONAL CLUSTERING

The extension of the two-dimensional algorithm to the clustering of data belonging to $\mathbb{R}^n$ can be designed in several ways. A possible approach is to consider the coordinates of the data points pair by pair and to apply to each pair the static two-dimensional algorithm. More precisely, suppose that $\mathcal{P} \subset \mathbb{R}^n$ and suppose, without loss of generality, that n is even. In fact, if n is odd, a new coordinate may be added without affecting the reasoning. If two points $\boldsymbol{\xi}_1 \in \mathcal{P}$ and $\boldsymbol{\xi}_2 \in \mathcal{P}$ belong to different clusters, then they must differ at least in one of the n coordinates, say the jth. As a consequence, the application of the planar algorithm to the projection of the data points onto the plane (ξ_j, ξ_l), for $l \neq j$, classifies $\boldsymbol{\xi}_1 \in \mathcal{P}$ and $\boldsymbol{\xi}_2 \in \mathcal{P}$ in different clusters. Note that, since the original Hamiltonian surface defined in $\mathbb{R}^n$ is projected in the plane (ξ_j, ξ_l), namely, in $\mathbb{R}^2$, the reference value with respect to which the level set are computed also changes, becoming smaller. Summarizing these considerations, an algorithm for the n-dimensional clustering method can be described as follows.

- Step 1: Construct data sets

$$\begin{aligned} \mathcal{Q}_1 &= \{(\xi_{1,1}, \xi_{1,2}), (\xi_{2,1}, \xi_{2,2}), \ldots\}, \\ \mathcal{Q}_2 &= \{(\xi_{1,3}, \xi_{1,4}), (\xi_{2,3}, \xi_{2,4}), \ldots\}, \\ &\vdots \\ \mathcal{Q}_{n/2} &= \{(\xi_{1,n/2-1}, \xi_{1,n/2}), (\xi_{2,n/2-1}, \xi_{2,n/2}), \ldots\}, \end{aligned} \tag{18}$$

  where $\xi_{i,j}$ denotes the jth coordinate of the ith data point. These sets consist of the projections of the n-dimensional data points, and, in particular, they are made of two-dimensional data points.
- Step 2: Apply the static two-dimensional algorithm iteratively. The application to $\mathcal{Q}_1$ yields the clusters $C_{1,1}, C_{1,2}, \ldots, C_{1,K_1}$, the application to $\mathcal{Q}_2$ yields the clusters $C_{2,1}, C_{2,2}, \ldots, C_{2,K2}$, and so on, up to the application to $\mathcal{Q}_{n/2}$, which yields the clusters $C_{n/2,1}, C_{n/2,2}, \ldots, C_{n/2,K_{n/2}}$. In this way each data point $\boldsymbol{\xi}_j$ belongs to the cluster C_{1,j_1} as far as the first pair of coordinates is concerned, with $j_1 \in \{1, \ldots, K_1\}$, to the cluster C_{2,j_2}as far as the second pair of coordinates is concerned, with $j_2 \in \{1, \ldots, K_2\}$, and so on.
- Step 3: Let $N_K = K_1 K_2 \cdots K_{n/2}$. For each $k \in \{0, \ldots, N_K\}$ consider the set of indexes $\mathcal{I}_k = \{j_1, \ldots, j_{n/2}\}$, such that $j_1 \in \{1, \ldots K_1\}$, $j_2 \in \{1, \ldots K_2\}, \ldots j_{n/2} \in \{1, \ldots K_{n/2}\}$. This set of indexes is associated with the cluster

$$C_k = \bigcap_{i=1}^{n/2} C_{i,j_i}.$$

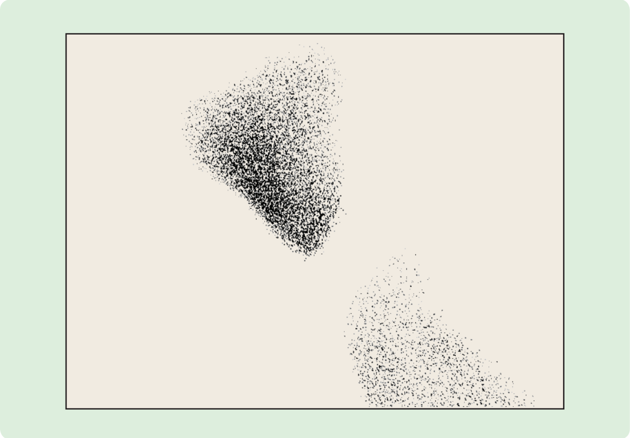


FIGURE 7 Flock of birds. A black and white picture of two flocks of birds can be used to test a clustering algorithm. In fact, a reasonable clustering method should be able to distinguish the two flocks.

In general, different choices of the set of indexes may originate the same cluster, that is, $\mathcal{I}_{k_1}$ and $\mathcal{I}_{k_2}$, with $k_2 \neq k_1$ may be such that $C_{k_1} = C_{k_2}$. However, the total number of n-dimensional clusters yielded by this process is at most $K = k_1 \cdots k_{n/2}$.

APPLICATIONS

In this section several applications are presented to show the effectiveness of the clustering methods described in this article.

Detecting Groups of Objects from Images

Consider the static two-dimensional algorithm and suppose that the set of data points to be clustered represent the black pixels in a black-and-white picture. This setting is the case, for instance, of an image representing moving objects that have to be grouped to deduct some common behavior of the members within the same group. Consider, for instance, two flocks of birds, as in Figure 7, obtained from an original colored image, where each bird is mapped into a set of black pixels, possibly a singleton. The image has 700×961 pixels, 27,016 of which are black and each of which potentially corresponds to a bird.

To simplify the computation, the image can be sampled before applying the algorithm. For instance, with a decimation rate of ten, a new image is obtained that has 70×97 pixels, 272 of which are black. Figure 8 shows the result of the clustering algorithm on the sampled image. Four clusters are identified. Two of them correspond to the original flocks while two contain only one point, namely, one bird. The two singletons may correspond to birds that are moving out from the flock or, on the contrary, joining the flock, or may be due to the fact that the sampling cut out some pixels between them and the nearest flock. In particular, in each cluster the piece of trajectory from the initial point $\boldsymbol{\xi}_i$ to the level line corresponding to H_r can be identified.

Finally, in Figure 9 the level line provided by the algorithm is compared with the original picture.

Tuning the Value H_r Corresponding to the Level Set

As the example considered in the previous section points out, a level line can be associated either with a group of objects, as in the case of the clusters associated to the main flocks, or with a single one, as in the case of the two birds clustered as singletons in Figure 8. While in that example this effect is due mainly to a sampling process, it can be obtained by properly selecting the reference value H_r. In Figure 10, for instance, different level lines obtained for the same set of data points, namely black pixels corresponding to geese, but with different values of H_r are reported.

On the left, the result obtained with a low value of the reference level yields the identification of a "V" shape, typical of migratory birds. On the right, a high value allows to

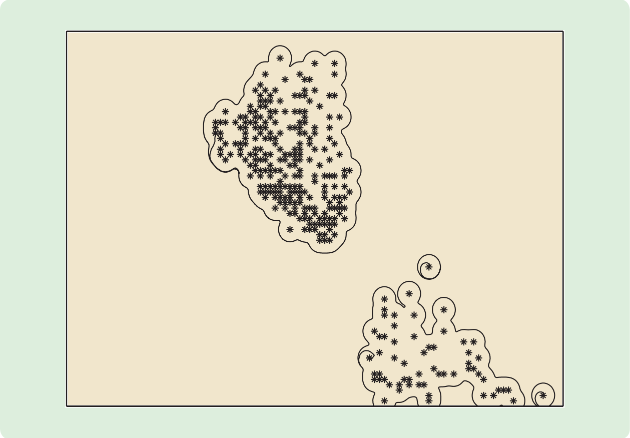


FIGURE 8 Identification of flocks of birds with sampled data. By sampling the black and white image of Figure 7, a set of points in $\mathbb{R}^2$ is obtained. The lines represent the result of the clustering algorithm.

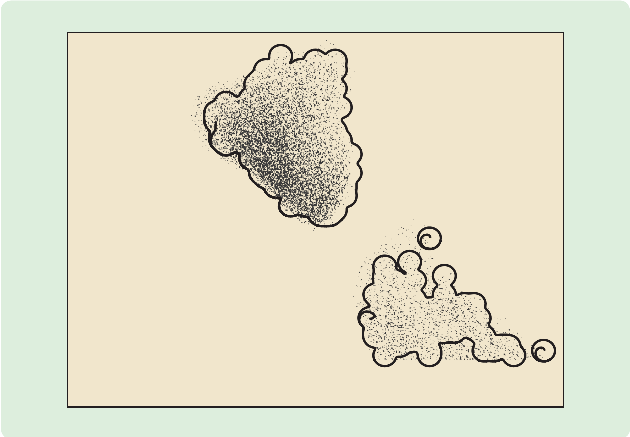


FIGURE 9 Validation of the resulting clustering with the original image. The result of the algorithm overlapped with the original image of Figure 7.

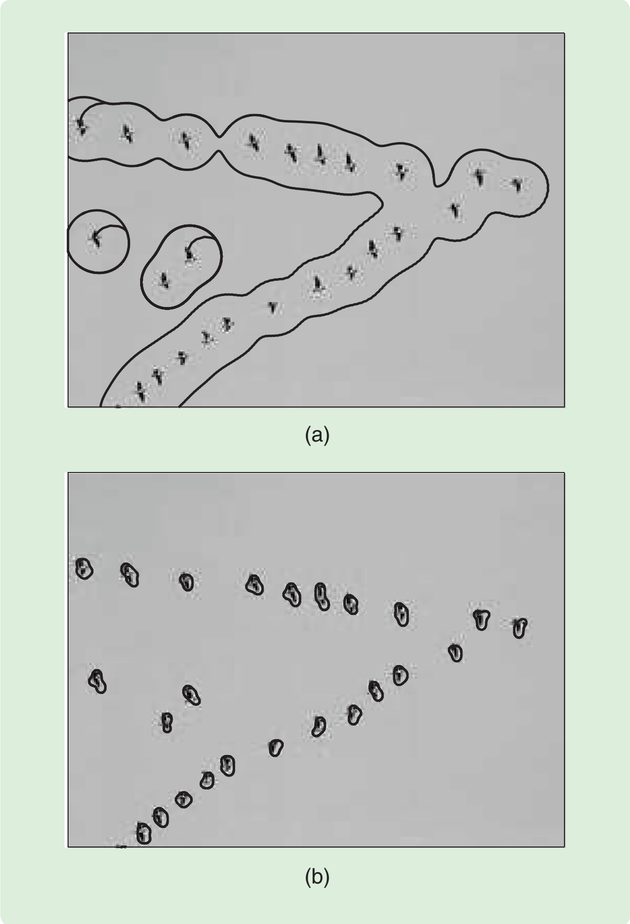


FIGURE 10 Effects of the choice of the reference value on the clustering. A low value of (a) the reference level H_r allows to deduct common behavior of a flock of geese flying in formation while a (b) high value allows to select the single birds.

select each single bird, an operation that can be used to automatically count the number of birds in the image.

Modeling the Motion of a Hurricane

As explained above, by applying iteratively the static algorithm to a sequence of images, the dynamics of the clusters can be identified. Consider, for instance, the sequence of infrared images taken from a geostationary satellite over a region of the Earth. Each image is a black-and-white image the white pixels of which are the data points corresponding to the presence of water. The position of the center of mass $C_{\mathcal{R}}$ of a region $\mathcal{R} \subset \mathbb{R}^2$ can be obtained by computing the moment of order zero of $\mathcal{R}$ together with its two moments of order one. Now, suppose that the motion of $C_{\mathcal{R}}$ in a discrete-time scale is described by an autoregressive linear difference equation of order n with time-invariant coefficients, namely, by $\nu(t) + b_1\nu(t-1) + \cdots + b_n\nu(t-n) = 0$. Standard techniques from system identification [11] can be used to estimate both the order of the difference equation n and the vector of parameters $\mathbf{b} = (b_1, \ldots, b_n)^\top$ giving the best fit for the sequence $\{\nu(k)\}_{k\in\mathbb{N}}$. To test the performance of the method, a real scenario is considered. The sequence of time instants is $\{\tau_k\}_{k\in\mathbb{N}}$, namely, $\tau_k = kT$, being the sampling time T half an hour, in comparison with which the computational time is negligible. Each image is a grayscale image and has 720×480 pixels. To ease the computational load, a spatial sampling of each image is performed, reducing the size to 72×48 pixels. Finally, the level of white associated to each pixel is compared with a threshold and the measurements set ρ is composed of all the pixels having a white level larger than the threshold. Figure 11 shows the first image of a stream where the level lines corresponding to each cluster are detected.

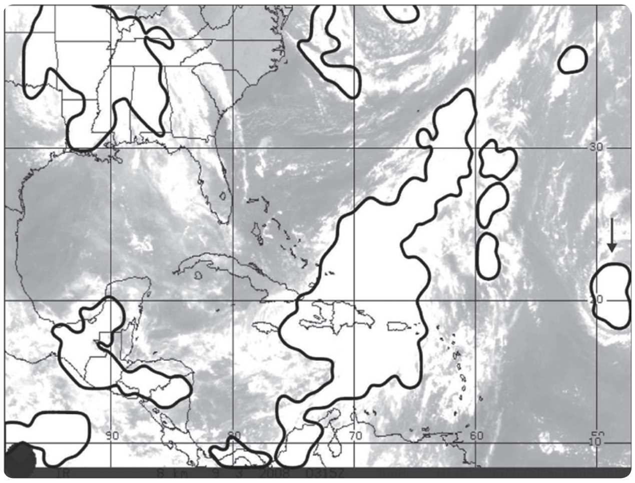


FIGURE 11 Infrared image of a hurricane. An infrared image taken from the satellite and representing the presence of water (white clouds). The closed lines represent the regions, namely, the result of the clustering method on the sampled measurements. The cloud to be tracked is the first region on the right, indicated by an arrow. The background image is taken from [20].

The clustering technique is applied to a sequence of 288 images, corresponding to six days, using the first 48, namely one day, to estimate the model. The obtained values are used to predict the motion of the cloud in the subsequent 240 images, corresponding to five days, with four different values of the prediction horizon and precisely one step, six steps, 12 steps, and 24 steps, corresponding to half an hour, three hours, six hours, and 12 hours, respectively. Since results show that in all four cases the best approximation is a difference equation of order two, such a model is used to estimate both the dynamics of $x_{\overline{\mathcal{R}}}$ and those of $y_{\overline{\mathcal{R}}}$, thus estimating the trajectory of the center of mass. Results corresponding to a prediction horizon of length 12 are reported in Figure 12 where the true trajectory is also plotted.

On Figure 13 the same trajectory is plotted over the image corresponding to the last position of the trajectory.

Continuous-Time-Varying Data Sets

In many applications the data points change in time. In the following examples, data represent information on a subject collected in a particular instant, and it might be interesting

By applying iteratively the static algorithm to a sequence of images, the dynamics of the clusters can be identified.

to compare how this information evolve in a continuous-time scale. In Figure 14(a), a set of two points is considered. Initially the two points are classified in the same cluster. However, the two points are moving and after five time units, at time $\tau = 5$, they are too far apart and each of them is associated with a different cluster. The trajectory of the Hamiltonian system follows one of the "branch" of the surface $\mathcal{H}_r$; after five more time units, at $\tau = 10$, the datum corresponding to this branch disappears and the storing function term is used. In this case the storing interval lasts three time units; after this three time units, at $\tau = 13$, the disappeared datum is forgot and the trajectory of the Hamiltonian system is driven to the remaining branch.

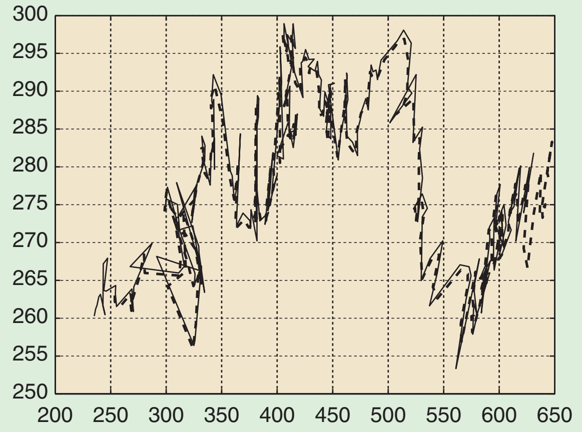


FIGURE 12 Prediction of the motion of the hurricane. The trajectory of the center of mass estimated over a 12-step horizon (dashed bold line) together with the true trajectory (solid line).

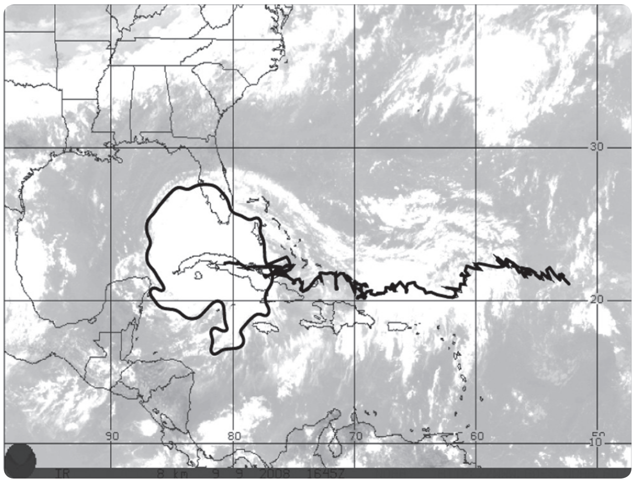


FIGURE 13 Evolution of the hurricane. The estimated trajectory of the center of mass and the boundary of the cloud obtained with the clustering algorithm are plotted over the last image of the test sequence. The background image is taken from [20].

The second scenario, depicted in Figure 14(b), is that of two groups of data points that are classified in the same cluster for the first four time units. Note the trajectory of the Hamiltonian system running around them both. Then, at $\tau = 4$ the two groups, which move in opposite directions, are too far apart and are associated with different clusters; the trajectory of the Hamiltonian system runs around one of the two branches. After eight time units from the initial instant, at $\tau = 8$, one of the two groups inverts its direction while the other stays still. After eight more time units, at $\tau = 16$, the two groups are again associated with the same cluster and all the data points are internal to the trajectory of the Hamiltonian system.

These two examples show that the clustering method valid in the static scenario can be extended to the time-varying case, thus defining a dynamic clustering. To this

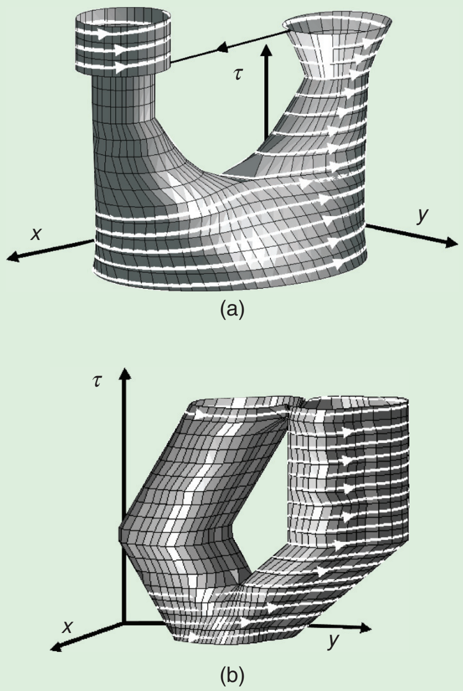


FIGURE 14 Time-varying Hamiltonian functions. (a) shows two points, initially belonging to the same cluster, that split apart; after a while one of the two points disappear. (b) shows two groups of data points that split apart and, after a while, merge again.

The notion of moments can be exploited to solve several classification problems.

aim, time needs to be regarded as an additional variable of the Hamiltonian function. The level surfaces obtained are to be reconstructed in a sort of "wrap around" description by the trajectory of a Hamiltonian system.

Static Optical Character Recognition

The notion of moments can be exploited to solve several classification problems. One of the most challenging is the classification of the letters of the alphabet, which is known as optical character recognition (OCR) [12]–[15]. For this problem, it is reasonable to assume that the size and the position of a letter on the plane need not be relevant to its classification, hence a classification process is expected to be insensitive to these quantities. As a consequence, the moments of order one, providing information on the position of the letter are not needed. On the other hand, the moments of order zero, providing information on the size, can be used to normalize the moments of higher order. Consider, in fact, a magnified version of a letter, being $K > 1$ the magnification factor. Integral (8) on the magnified region $K\mathcal{R}_j$ is equal to the integral on $\mathcal{R}_j$ by scaling x and y of a factor $1/K$; in fact

$$\begin{aligned}\iint_{K\mathcal{R}_j} x^p y^q \mathrm{d}x\mathrm{d}y &= \iint_{\mathcal{R}_j} \left(\frac{x}{K}\right)^p \left(\frac{y}{K}\right)^q \mathrm{d}\left[\frac{x}{K}\right]\mathrm{d}\left[\frac{y}{K}\right]\\ &= \frac{1}{K^{(p+q+2)}} \iint_{\mathcal{R}_j} x^p y^q \mathrm{d}x\mathrm{d}y. \qquad (19)\end{aligned}$$

In the following, ν_{pq} denotes a central moment of order $p+q$ normalized as in (19), being the normalization factor K the square root of the area of the letter

$$\nu_{pq} \triangleq \frac{\mu_{pq}}{\left(\sqrt{m_{00}(\mathcal{R})}\right)^{p+q+2}},$$

that is, all normalized letters have area one. In some applications, an additional invariance property is required, namely, the invariance with respect to rotations, and the moments are redefined according to this requirement. On the contrary, in the particular case of OCR this kind of invariance can be misleading when attempting to recognize characters that are similar but differently oriented, such as "N" and "Z," "6" and "9," or "M" and "W." Therefore herein orientation is considered as an important feature.

In the moments-based OCR, first considered by Hu [16], the features to be extracted are associated to the moments and the minimal distance method is followed. The considered similarity measure is the Euclidean distance between collections of moments. For each symbol σ, a *training set* T_σ is defined, which is a set of different representations of σ. Specifically each representation corresponds to a different font. In Figure 15, for instance, a training set constituted by three different representations for the symbol "C" is shown. Each representation, namely, each shape, is mapped into a point in the space of the normalized central moments. If, for the sake of a simpler computation, only moments of order two and three are considered, then a map M from T_σ to $\mathbb{R}^7$ can be constructed $M(\sigma_i) \triangleq (\nu_{20}(\sigma_i), \nu_{11}(\sigma_i), \nu_{02}(\sigma_i), \nu_{30}(\sigma_i), \nu_{21}(\sigma_i), \nu_{12}(\sigma_i), \nu_{03}(\sigma_i))^\top$ where the subscript i denotes the representation corresponding to the ith font. Finally, a reasonable way to merge the information provided by all the elements of the training set is that of finding the center of mass of their images (in $\mathbb{R}^7$). More precisely, if $F = \{1, \ldots, N_f\}$ is the set of fonts of the training set and σ_k is the shape corresponding to the representation of the symbol σ in the kth font, not necessarily belonging to the training set, then $M(\sigma_k)$ is likely to be close to the center of mass of the images of the representations of σ in the fonts of the training set; hence

$$M(\sigma_k) \simeq B(\sigma) \triangleq \frac{1}{N_f}\sum_{i \in F} M(\sigma_i). \qquad (20)$$

Consider the set of the symbols corresponding to the letters of the English alphabet and the Arabic ciphers, namely, the set $S \triangleq \{A, B, \ldots, Z, 0, 1 \ldots, 9\}$ and let $\sigma_i^1, \sigma_i^2, \ldots \sigma_i^{36}$ denote their representation in the ith font. Within this framework, the classification problem is to find ρ given the representation ρ_k of a symbol $\rho \in S$ associated to the kth font.

As mentioned above, k may, or may not, belong to F. According to the minimum-distance method, the answer to the classification problem is the symbol σ^j providing the $B(\sigma^j)$ nearest to $M(\rho_k)$, that is, $\rho = \mathrm{argmin}_j \| M(\rho_k) - B(\sigma^j) \|$.

To have an idea of how the method works, consider the three sets of symbols represented in Figure 16, each of which corresponds to one of the fonts represented in Figure 15. The 36 centers of mass of all the images $B(\sigma^1), \ldots, B(\sigma^{36})$ can be used for classification. The result of a preliminary test on the training sets of Figure 16 is that all the characters are correctly recognized.

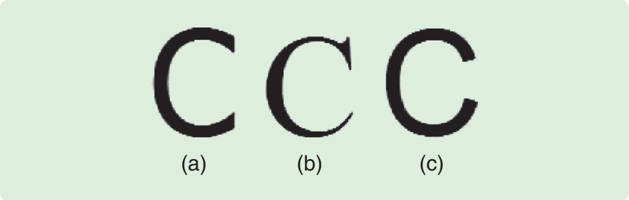


FIGURE 15 A possible training set for the letter "C." The considered fonts are (a) Tahoma, (b) Times New Roman, and (c) Arial.

One of the most challenging problems in bioinformatics is the localization of the position of a protein with respect to the cellular membrane and by exploiting information about the structure of the protein.

To further test the performance of the method, a fourth font, not used to find the centers of mass of each symbol, may be considered, as in Figure 17. In this case all the characters but one are correctly recognized, the only error being that "1" is recognized as "2," yielding a correct recognition rate of 97.22%, comparable with the rate of other moment-based recognition methods [17].

A third step of the test may consist of the recognition of some modified letters. In particular, four typical situations that could affect the performance in a practical case are considered and precisely letters that are partially erased, rotated, blurred or stretched (see Figure 18). In all the cases the method performs a correct recognition.

Finally, the method is tested on a handwritten word, the word in Figure 19. In this case only the second "A" is erroneously recognized as an "S"; note that only three sets of fonts are used for the training and that the handwritten font, obviously, does not belong to the training set.

Flower Classification

Consider the Anderson iris flower data set [18], that is, a set of points of $\mathbb{R}^4$ representing the sepal lengths and widths and the petal lengths and widths of 150 samples of three different species of iris, the *virginica*, the *setosa*, and the *versicolor* species. Each species is represented by 50 samples. This data set is used in [19] as a benchmark set for

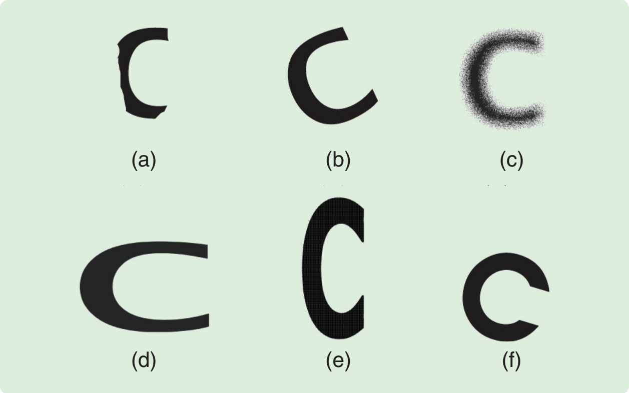
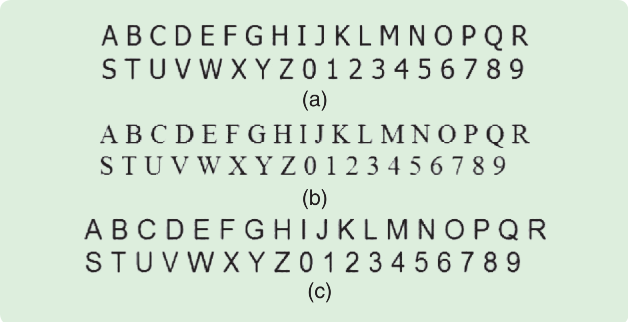


FIGURE 18 Examples of modified letter "C" used to test the performance of the classifying method. In (a) the letter is partially erased, in (b) and (f) it is rotated, in (c) it is blurred, and in (e) and (d) it is stretched. Different fonts are used.


ABCDEFGHIJKLMNOPQR
STUVWXYZ0123456789
(a)

ABCDEFGHIJKLMNOPQR
STUVWXYZ0123456789
(b)

ABCDEFGHIJKLMNOPQR
STUVWXYZ0123456789
(c)

FIGURE 16 The training sets for the letters and the numerals. Three sets of letters and numerals, corresponding to the font (a) Tahoma, (b) Times New Roman, and (c) Arial, are used to train the classification method by computing the moments of each character.

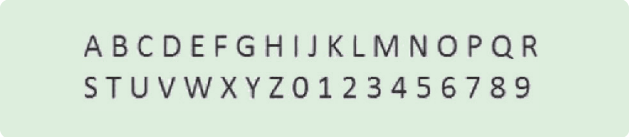

ABCDEFGHIJKLMNOPQR
STUVWXYZ0123456789

FIGURE 17 The Calibri font used to test the classification method. The moments of each letter and each numeral are computed; the classification is performed by comparing them with the average moments of the letters and numerals of Figure 16.

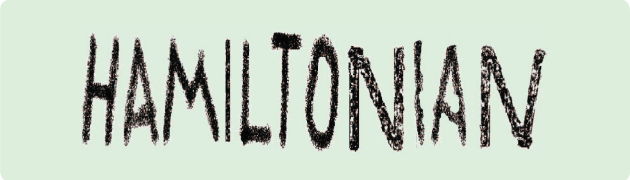
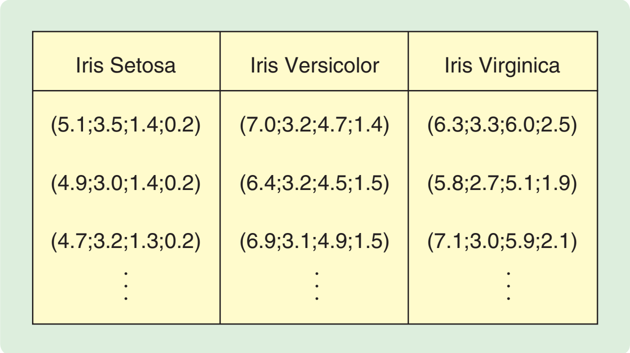


FIGURE 19 A handwritten word. The classification method can be used also to recognize a handwritten word, as in this case where the hand written representation of the word *Hamiltonian* is used.



Iris Setosa	Iris Versicolor	Iris Virginica
(5.1;3.5;1.4;0.2)	(7.0;3.2;4.7;1.4)	(6.3;3.3;6.0;2.5)
(4.9;3.0;1.4;0.2)	(6.4;3.2;4.5;1.5)	(5.8;2.7;5.1;1.9)
(4.7;3.2;1.3;0.2)	(6.9;3.1;4.9;1.5)	(7.1;3.0;5.9;2.1)
⋮	⋮	⋮

FIGURE 20 The Anderson's Iris flower data set. This set is used as a benchmark set to test classification and clustering algorithms.

The notion of level function can be used to cluster data points internal to a level line.

discriminant analysis. The set of measurements is partly reported in Figure 20, where to avoid cluttering only three points per species are reported.

According to the algorithm explained above, the sets

$$Q_1 = \{(5.1; 3.5); (7.0; 3.2), (6.3; 3.3), (4.9; 3.0), (6.4; 3.2),\\ (5.8; 2.7), (4.7; 3.2), (6.9; 3.1), (7.1; 3.0), \ldots\}$$

and

$$Q_2 = \{(1.4; 0.2), (4.7; 1.4), (6.0; 2.5), (1.4; 0.2), (4.5; 1.5),\\ (5.1; 1.9), (1.3; 0.2), (4.9; 1.5), (5.9; 2.1), \ldots\}$$

can be constructed. The algorithm groups all the points in Q_1 in a single cluster $C_{1,1} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_2, \ldots\}$ [see Figure 21(a)], while the set Q_2 is partitioned into the two clusters

$$\begin{aligned} C_{2,1} &= \{(1.4; 0.2), (1.4; 0.2), (1.3; 0.2), \ldots\}\\ &= \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_4, \boldsymbol{\xi}_7, \ldots\},\\ C_{2,2} &= \{(4.7; 1.4), (6.0; 2.5), (4.5; 1.5), (5.1; 1.9),\\ &\quad (4.9; 1.5), (5.9; 2.1), \ldots\}\\ &= \{\boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \boldsymbol{\xi}_5, \boldsymbol{\xi}_6, \boldsymbol{\xi}_8, \boldsymbol{\xi}_9, \ldots\} \end{aligned}$$

[see Figure 21(b)]. Therefore, the possible choices for the indexes are $i_1 = 1$ and $i_2 \in \{1, 2\}$, yielding the clusters

$$\begin{aligned} C_1 &= C_{1,1} \cap C_{2,1} = \{\boldsymbol{\xi}_1, \boldsymbol{\xi}_4, \boldsymbol{\xi}_7, \ldots\},\\ C_2 &= C_{1,1} \cap C_{2,2} = \{\boldsymbol{\xi}_2, \boldsymbol{\xi}_3, \boldsymbol{\xi}_5, \boldsymbol{\xi}_6, \boldsymbol{\xi}_8, \boldsymbol{\xi}_9 \ldots\}. \end{aligned}$$

Hence, the projection of the data points onto the two-dimensional space corresponding to the sepal length and width does not provide any information on the species, since the result of the clustering process is the whole set of measurement points. On the other hand, when petal length and width are considered, the same algorithm classifies the points into two clusters.

In conclusion, with respect to this particular problem, the level set method recognizes the members of the *setosa* quality, namely the cluster on the bottom-left of Figure 21(b), whereas it cannot distinguish between the *virginica* and the *versicolor* quality, implying, as claimed in [19], that "a certain diagnosis of these two species could not be based solely on these four measurements."

Subcellular Protein Localization

One of the most challenging problems in bioinformatics is the localization of the position of a protein with respect to the cellular membrane and by exploiting information about the structure of the protein. It is known that cells are constituted by compartments and that each of the compartments performs specific functions. Since proteins in each compartment are specialized to fulfill a particular function, it is possible to deduct the behavior of a specific protein knowing its position within the cell. Consider a cell of *Escherichia Coli* and the problem of distinguishing among proteins localized within the cytoplasm, proteins localized in the inner membrane and proteins localized in the outer membrane.

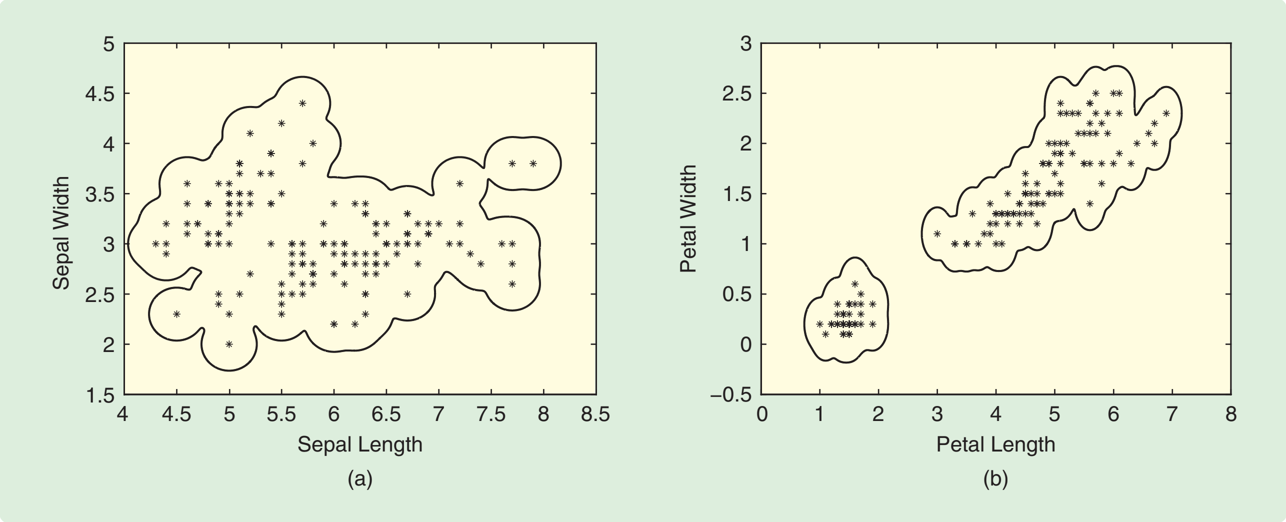


FIGURE 21 Classification of the iris flowers. A set of points belonging to $\mathbb{R}^4$ can be partitioned into (a) a single cluster or into (b) two clusters according to the coordinates considered.

The basic static algorithm can be exploited to define dynamical clustering, both in the discrete- and continuous-time cases.

More specifically, the problem is twofold; first, part of a data set is clustered, 80 points out of 145, according to the static algorithm. Then the obtained clusters are used to classify the remaining 65 elements of the data set. The nature of the data set is such that each protein $\mathbf{p}_i$ is described by a vector of seven components. In the application of the n-dimensional clustering algorithm, one of these components is discarded since it is a binary digit having the same value for all the elements in the data set. Among the six remaining attributes, the first three represent the score of the protein to different recognition analysis, while the last three attributes are the score discriminant analysis of the amino-acid content and the score of two different versions of the ALOM program. The six coordinates originate three two-dimensional problems, associated with the sets

$$\begin{aligned} Q_1 &= \{(p_{1,1}, p_{1,2}), (p_{2,1}, p_{2,2}), \ldots\}, \\ Q_2 &= \{(p_{1,3}, p_{1,4}), (p_{2,3}, p_{2,4}), \ldots\}, \\ Q_3 &= \{(p_{1,5}, p_{1,6}), (p_{2,5}, p_{2,6}), \ldots\}. \end{aligned}$$

The application of the clustering algorithm to Q_1 yields three clusters $C_{1,1}$, $C_{1,2}$, and $C_{1,3}$ [Figure 22(a)]; its application to Q_2 yields one cluster, $C_{2,1}$, and two singletons, $C_{2,2}$ and $C_{2,3}$, [Figure 22(b)]; its application to Q_3 yields one cluster, $C_{3,1}$, and one singleton, $C_{3,2}$ [Figure 22(c)].

It can be noted that $C_{3,2} \neq C_{2,2}$ and $C_{3,2} \neq C_{2,3}$, hence the intersection between each singleton and the other (proper) clusters is either empty or coincides with the singleton itself. Hence six clusters are obtained

$$\begin{aligned} &C_1 = C_{1,1} \cap C_{2,1} \cap C_{3,1}, \quad C_2 = C_{1,2} \cap C_{2,1} \cap C_{3,1}, \\ &C_3 = C_{1,3} \cap C_{2,1} \cap C_{3,1}, \quad C_4 = C_{2,2}, \quad C_5 = C_{2,3}, \quad C_6 = C_{3,2}. \end{aligned}$$

In conclusion, when neglecting the singletons, which can be considered as measurement errors, three clusters are obtained corresponding to the cytoplasm (C_1), the inner membrane (C_2), and the outer membrane (C_3), respectively. The three corresponding regions $\mathcal{R}_1$, $\mathcal{R}_2$, and $\mathcal{R}_3$ are then used to classify the remaining proteins in the data set in the more natural way, that is, by classifying a protein $\mathbf{p}^* = (p_1^*, p_2^*, \ldots p_6^*)^\top$ as residing in the cytoplasm if $(p_1^*, p_2^*), \in C_1$, as residing in the inner membrane if $(p_1^*, p_2^*), \in C_2$ and as residing in the outer membrane if $(p_1^*, p_2^*), \in C_3$. With this method, a classification rate of 88% is obtained.

CONCLUSIONS

The notion of level function can be used to cluster data points internal to a level line. Level lines can be determined as trajectories of a Hamiltonian system. More precisely, the level function is interpreted as a Hamiltonian function, and the corresponding Hamiltonian system is integrated.

The basic static algorithm can be exploited to define dynamical clustering, both in the discrete- and continuous-time cases. Due to the different nature of the two time scales, different solutions to the problem of dynamic clustering are defined.

The extension of the method to the clustering of n-dimensional data points is straightforward. In fact it basically consists of an iterative application of the two-dimensional version of the algorithm and to the intersection of the results of each iteration. The applications described in the final section of the article show the effectiveness of the method.

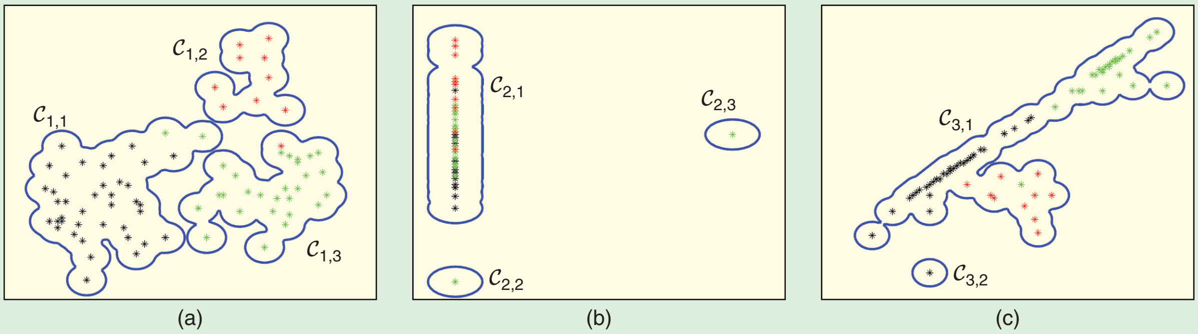


FIGURE 22 Localization of the protein inside the cell. The results of the three two-dimensional clustering processes corresponding to the sub-cellular protein localization problem.

ACKNOWLEDGMENTS

This work has been partially supported by the Austrian Center of Competence in Mechatronics (ACCM), the Engineering and Physical Sciences Research Council (EPSRC), and the Systems Engineering for Autonomous Systems (SEAS). The authors would like to thank Neil Cade from SELEX Sensors and Airborne Systems Limited for many useful discussions that contributed to this work.

AUTHOR INFORMATION

Daniele Casagrande (daniele.casagrande@uniud.it*)* received the "Laurea" in electronic engineering in 1999 and the Ph.D. in information engineering in 2006, both from the University of Trieste, Italy. In 2006 he was a research associate at the Institute of Cybernetics of the Tallinn University of Technology, Estonia. Between 2006 and 2007 he was a research associate in the Department of Electrical, Electronic and Computer Engineering of the University of Trieste. In 2008 he was a research associate in the Department of Electrical and Electronic Engineering of Imperial College London, United Kingdom. Since 2008 he has been with the Department of Electrical, Managerial, and Mechanical Engineering of the University of Udine where he is currently a lecturer in automatic control. In 2009 he was an academic visitor at Imperial College London. His research interests are focused on stability of switched system, passive systems, Hamiltonian systems, LPV stability, and nonlinear control.

Mario Sassano received the B.S. degree in automation system engineering and the M.S. degree in systems and control engineering from the University of Rome ”Sapienza,” Italy, in 2006 and 2008, respectively. He is currently a Ph.D. student with the Electrical and Electronic Engineering Department, Imperial College, London, United Kingdom. His research interests are focused on nonlinear observer design, optimal control, and image processing.

Alessandro Astolfi graduated in electrical engineering from the University of Rome in 1991. In 1992 he joined ETH-Zurich where he obtained an M.Sc. in information theory in 1995 and the Ph.D. degree with a medal of honor in 1995 with a thesis on discontinuous stabilization of nonholonomic systems. In 1996 he was awarded a Ph.D. from the University of Rome "La Sapienza" for his work on nonlinear robust control. Since 1996 he has been with the Electrical and Electronic Engineering Department of Imperial College, London, United Kingdom, where he is currently a professor in nonlinear control theory. From 1998 to 2003 he was also an associate professor with the Department of Electronics and Information of the Politecnico of Milano. Since 2005 he has been a professor in the Dipartimento di Informatica, Sistemi e Produzione, University of Rome Tor Vergata. He has been a visiting lecturer in nonlinear control at several universities. His research interests are focused on mathematical control theory and control applications, with a special emphasis on the problems of discontinuous stabilization, robust stabilization, robust control, and adaptive control. He is the author of more than 100 journal papers, 20 book chapters, and over 200 papers in refereed conference proceedings. He is a coauthor of the monograph *Nonlinear and Adaptive Control with Applications* (Springer Verlag). He has been an associate editor of *IEEE Transactions on Automatic Control*. He has served in the IPC of various international conferences and he is currently the chair of the Conference Editorial Board of the IEEE Control Systems Society.

REFERENCES

[1] European innovation scoreboard 2006, comparative analysis of innovation performance. [Online]. Available: http://www.proinno-europe.eu/doc/EIS2006 final.pdf

[2] G. Punj and D. Stewart, "Cluster analysis in marketing research: Review and suggestions for application," *J. Market. Res.*, vol. 20, no. 2, pp. 134–148, 1983.

[3] P. D'haeseleer, "How does gene expression clustering work?," *Nat. Biotechnol.*, vol. 23, no. 12, pp. 1499–1501, 2005.

[4] J. Han and M. Kamber, *Data Mining: Concepts and Techniques*. San Mateo, CA: Morgan Kaufmann, 2001.

[5] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*. New York: Wiley, 2001.

[6] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*. New York: Wiley, 1973.

[7] S. Theodoridis and K. Koutroumbas, *Pattern Recognition*. New York: Academic Press, 1999.

[8] K. L. Wu and M. S. Yang, "Alternative c-means clustering algorithm," *Pattern Recognit.*, vol. 35, no. 10, pp. 2267–2278, 2002.

[9] I. Stewart and D. Tall, *Complex Analysis*. Cambridge, U.K.: Cambridge Univ. Press, 1983.

[10] Y. Rudyak, *On Thom Spectra, Orientability, and Cobordism*. Berlin, Germany: Springer-Verlag, 1997.

[11] L. Ljung, *System Idenztification. Theory for the User*. Englewood Cliffs, NJ: Prentice-Hall, 1999.

[12] H. I. Avi-Itzhak, T. A. Diep, and H. Garland, "High accuracy optical character recognition using neural networks with centroid dithering," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 17, no. 2, pp. 218–224, 1995.

[13] L. Wang and T. Pavlidis, "Direct gray-scale extraction of features for character recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 10, pp. 1053–1067, 1993.

[14] C.-Y. Liou and H.-C. Yang, "Handprinted character recognition based on spatial topology distance measurement," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 9, pp. 941–945, 1996.

[15] N. Kato, M. Suzuki, S. Omachi, H. Aso, and Y. Nemoto, "A handwritten character recognition system using directional element feature and asymmetric mahalanobis distance," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 3, pp. 258–262, 199.

[16] M.-K. Hu, "Visual pattern recognition by moment invariance," *IRE Trans. Inform. Theory*, vol. 8, no. 2, pp. 179–187, 1962.

[17] G. L. Cash and M. Hatamian, "Optical character recognition by the method of moments," *Comput. Vis., Graph. Image Process.*, vol. 39, no. 3, pp. 291–310, 1987.

[18] E. Anderson, "The irises of the Gaspe Peninsula," *Bull. Amer. Iris Soc.*, vol. 59, pp. 2–5, 1935.

[19] R. Fisher, "The use of multiple measurements in taxonomic problems," *Anna. Eugen.*, vol. 7, no. 2, pp. 179–188, 1936.

[20] Available: http://www.goes-arch.noaa.gov