Journal of Marine Research, 64, 651–668, 2006

Journal of
MARINE RESEARCH

Volume 64, Number 5

Cluster analysis for physical oceanographic data and oceanographic surveys in Turkish seas

by Derya Birant[1] and Alp Kut[1]

ABSTRACT

Cluster analysis is a useful data mining method to obtain detailed information on the physical state of the ocean. The primary objective of this study is the development of a new spatio-temporal density-based algorithm for clustering physical oceanographic data. This study extends the regular spatial cluster analysis to deal with spatial data at different epochs. It also presents the sensitivity of the new algorithm to different parameter settings. The purpose of the sensitivity analysis presented in this paper is to identify the response of the algorithm to variations in input parameter values and boundary conditions. In order to demonstrate the usage of the new algorithm, this paper presents two oceanographic applications that cluster the sea-surface temperature (SST) and the sea-surface height residual (SSH) data which records the satellite observations of the Turkish Seas. It also evaluates and justifies the clustering results by using a cluster validation technique.

1. Introduction

Clustering is the process of identifying subsets of events in a data collection with similar characteristics. Some distance functions are used for measuring the similarity between objects. As a data mining function, clustering is known as an unsupervised learning process, which means that no human intervention is needed during the clustering and that little need to be known about the characteristics of the input data. Cluster analysis have been used in many areas, including image processing, pattern analysis, data compression, data segmentation, data reduction, outlier detection and noise filtering. In this study, we focus on pattern analysis on physical oceanographic data about a marine environment in

[1] Department of Computer Engineering, Dokuz Eylul University, 35100 Izmir, Turkey.
[2] Corresponding author. ***email: derya@cs.deu.edu.tr***

order to discover interesting patterns in large amounts of data. The main goal of this pattern analysis is understanding how the physical properties of the water are distributed in a marine environment.

Seawater data generally involve both temporal and spatial dimensions. Spatio-temporal data refers to data that have a spatial component (e.g. latitude and longitude) and a time component (e.g. 12:00pm 23/01/2005). Most studies on the cluster analysis focus on discovering clusters from ordinary data (nonspatial and nontemporal data). This study extends the regular spatial cluster analysis to deal with spatial data at different epochs.

Cluster discovery from marine data is one of the very promising subfields of data mining because increasingly large volumes of marine data are collected from satellites and need to be analyzed. The focus of the marine data mining is to maximize the information that can be derived from data of a marine environment. In the literature, a few works (Robinson and Golnaraghi, 1994; Emery and Thomson, 2001) present the modern techniques for the analysis of temporal and spatial data sets collected in oceanography, geophysics, and other disciplines in earth and ocean sciences. The method described in this study is intended for use in oceanographic and interdisciplinary scientific research. This study presents a new density-based clustering algorithm which is based on the algorithm DBSCAN (*Density Based Spatial Clustering of Applications with Noise*) (Ester *et al.,* 1996). It has been proven that DBSCAN algorithm has the ability to process very large databases. (Ester *et al.,* 1998). It has the ability to discover clusters with arbitrary shape such as linear, concave, oval, etc. Furthermore, it does not require the pre-determination of the number of clusters in contrast to some other clustering algorithms. In addition to DBSCAN, ST-DBSCAN algorithm can cluster spatio-temporal data, can detect the noise objects when clusters of different densities exist and can identify adjacent clusters.

Sensitivity analysis is an important method for checking the quality of a given model, as well as a powerful tool for checking the robustness and reliability of its analysis. It is used to determine how a given model output depends upon the input parameters. This paper also presents the sensitivity of new clustering algorithm to the input parameters. In the sensitivity test, we changed one parameter at a time and ran the algorithm again. We changed each parameter value by ±10% each time until reaching ±50%.

The data warehouse designed for storing and clustering oceanographic physical data of Turkish seas contains sea-surface temperature, sea-surface height residual, significant wave height, ocean wind speed and ocean current values of four seas (the Black Sea, the Marmara Sea, the Aegean Sea, and the East Mediterranean Sea). This data warehouse was constructed by collecting data from different satellites to discover the regions that have similar seawater characteristics. Special functions were developed for data integration, data conversion, query, visualization, analysis and management.

One of the most important issues in cluster analysis is the evaluation of clustering results to validate the outcome of a clustering method. In this paper, the new method is also justified by using a cluster validation technique. We used a RS cluster validity index for evaluating our clustering results.

The remainder of this paper is organized as follows. In Section 2 we first introduce the basics of density-based clustering before discussing the flat density-based clustering algorithm DBSCAN. Then, we explain in which situations the existing clustering algorithms are not suitable for efficient distance computations and how we solved these problems. In Section 3, we present our new approach which has the ability of clustering spatio-temporal data, the ability of detecting the noise objects when clusters of different densities exist and the ability of identifying adjacent clusters. Section 4 explains the concept of sensitivity analysis and then presents the sensitivity analysis results of the input parameters of our algorithm. The results presented in this section show the sensitivity of the algorithm to parameter settings. Section 5 presents two applications which are implemented to show the spatio-temporal distributions of physical parameter values in Turkish seas. It shows and discusses the cluster analysis results. Section 6 shows the validation and evaluation of the clustering results. It presents the mathematical quality and reliability of the clustering results by using a cluster validation technique. We close this paper, in Section 7, with a short summary and a few remarks on future work.

2. Basic concepts

a. Density-based clustering

Clustering algorithms can be categorized into Partitioning Methods, Hierarchical Methods, Density-Based Methods, Grid-Based Methods, and Model-Based Methods. (Han and Kamber, 2001) Density-based clustering algorithms have recently gained popularity in the data mining field due to their ability to discover arbitrarily shaped clusters while preserving spatial proximity of data objects. Density-based clustering algorithms consider clusters as dense regions of objects in the data space and clusters are separated by regions of low density (noise). As a general definition, density is an entity that is determined by using a threshold value. For at least one object in a cluster, the neighborhood of a given radius Eps has to contain at least a minimum number of $MinPts$ objects. In our study, density is determined by measuring the distances in kilometers between stations observed by the satellite and by comparing the similarity of the oceanographic values (e.g. water temperature) of neighbor stations.

Density-based clustering algorithms have two input parameters: Eps (maximum radius of the neighborhood) and $MinPts$ (minimum number of objects in an Eps-neighborhood of current object). The objects are categorized into three basic types: *core object, border object* and *noise object.* (Fig. 1) An object is a *core object* if it has more than a specified number of points ($MinPts$) within Eps radius. These are objects that are at the interior of a cluster. An object is a *border object* if it has fewer than $MinPts$ within Eps radius, but it is in the neighborhood of a core object. A *noise object* is any object that it is not neither a core object nor a border object. These objects are not located in any cluster. In the following, we present the basic definitions of density-based clustering.

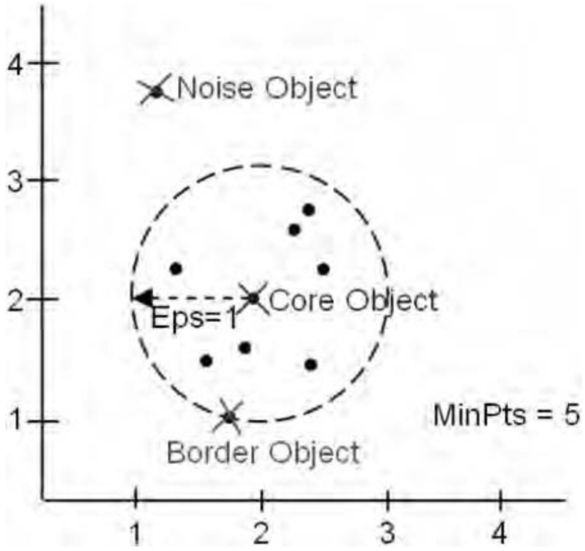


Figure 1. An example core object, border object and noise object.

Definition 1 (*directly density-reachable*). An object p is *directly density-reachable* from an object q with respect to Eps and $MinPts$ in a dataset D, if $p \in N_{Eps}(q)$ and $|N_{Eps}(q)| \geq MinPts$, where $N_{Eps}(q)$ denotes the subset of D contained in the Eps-neighborhood of q.

Eps-Neighborhood ($N_{Eps}(q)$) is determined by a distance measure such as Manhattan Distance, Euclidean Distance, and Minkowski Distance. It can be defined as $\{q \; belongs \; to \; D | dist(p, q) \leq Eps\}$. $|N_{Eps}(q)|$ denotes the number of Eps-neighborhoods of q.

Definition 2 (*density-reachable*). An object p is *density-reachable* from an object q with respect to Eps and $MinPts$ in a dataset D, if there is a chain of objects $p_1, \ldots p_n$, $p_1 = q$ and $p_n = p$ such that $p_i \in D$ and p_{i+1} is directly density-reachable from p_i with respect to Eps and $MinPts$.

Definition 3 (*density-connected*). An object p is *density-connected* to an object q with respect to Eps and $MinPts$ in a dataset D, if there is an object $o \in D$ such that both p and q are density-reachable from o with respect to Eps and $MinPts$ in D.

The most popular density-based clustering algorithm, DBSCAN (Ester *et al.,* 1996), is designed to discover clusters of arbitrary shape as well as to distinguish noise. The basic idea of DBSCAN is that the density of points in a radius around each point in a cluster has to be above a certain threshold. It grows a cluster as long as, for each data point within this cluster, a neighborhood of a given radius contains at least a minimum number of points. Another density-based clustering algorithm, OPTICS (Ankerst *et al.,* 1999), was proposed as an improvement on DBSCAN. It computes an augmented cluster ordering for iterative clustering analysis. DENCLUE (Hinneburg and Keim, 1998) is also one of the density-based clustering algorithms. It uses an influence function to model the impact of an object within that object's neighborhood. The density of the data space is then calculated as the sum of the influence functions over all objects. Clusters (called density-attractors) are then defined as the local maxima of the overall density function. Another density-based clustering algorithm CURD (Ma *et al.,* 2003) firstly captures the shape and extent of a cluster with references, and then it analyzes the data based on the references.

b. Basic extensions

Current density-based algorithms are to some extent capable of clustering databases. (Murray and Estivill-Castro, 1998; Qian and Zhou, 2002) However, since the main objective of clustering algorithms is to find clusters, they are developed to discover clusters on ordinary data (nonspatial and nontemporal data) or spatial data, not to discover clusters on spatio-temporal data. In this paper, we extend these regular clustering algorithms to deal with spatio-temporal data because seawater data generally involve both spatial and temporal dimensions. We use two threshold parameters, $Eps1$ and $Eps2$, to determine whether a set of points is *similar* enough to be considered as a cluster or not. While $Eps1$ is used for spatial values to measure the geographical closeness of two points (latitude and longitude), $Eps2$ is used to measure the similarity of the values of oceanographic parameters measured in a marine environment such as seawater temperature, ocean current vector and others. In this study, $Eps1$ and $Eps2$ threshold parameters are compared with the distance between two objects which is calculated by Euclidean distance metric. For example, the geographical distance ($Dist1$) between two objects A and B must be smaller than the $Eps1$ threshold parameter and the closeness of the oceanographic values ($Dist2$) of these points must also be smaller than $Eps2$ threshold ($Dist1(A, B) \Leftarrow Eps1$ AND $Dist2(A, B) \Leftarrow Eps2$). It is also possible to use other distance metrics such as Manhattan distance and Minkowski distance.

A major problem of clustering algorithms, particularly in subspace and correlation clustering, is their sensitivity to noise objects. Clusters cannot be discovered with satisfactory accuracy if noise objects are present. Some clustering algorithms may detect noise objects in normal conditions, but they capture only certain kinds of noise objects, not all of them, when clusters of different densities exist. In order to produce meaningful and adequate results when clusters of different densities exist, we propose two concepts: *density distance* and *density factor.* The *density distance* of an object p is defined as $density\_distance\_max\ (p)/density\_distance\_min\ (p)$, where $density\_distance\_max$ of an object p denotes the maximum distance between the object p and its neighbor objects within the radius $Eps(\max\{\text{dist}(p, q)|q \in D\ \Lambda\ \text{dist1}(p, q) \leq Eps1\ \Lambda\ \text{dist2}(p, q) \leq Eps2\})$ and $density\_distance\_min$ of an object p denotes the minimum distance between the object p and its neighbor objects within the radius $Eps(\min\{\text{dist}(p, q)|q \in D$

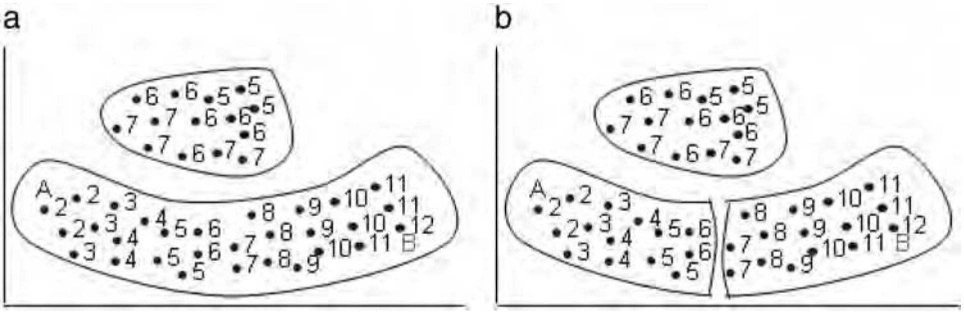


Figure 2. An example adjacent clusters which should be divided into more than one sub-clusters.

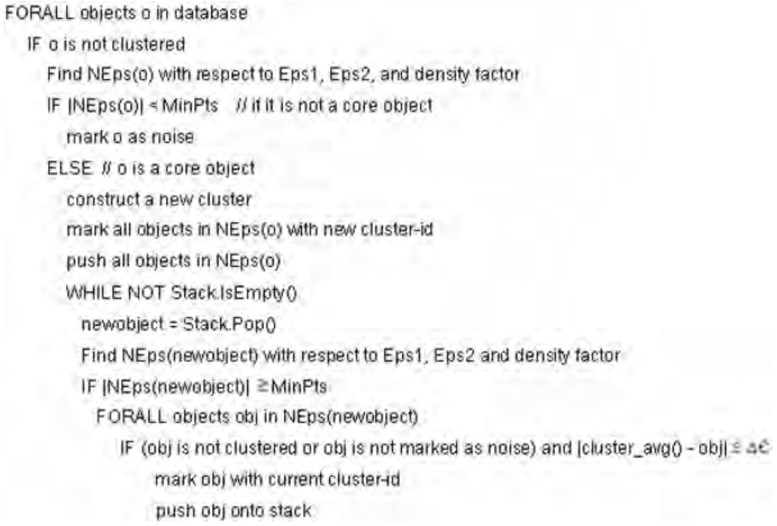
Λ dist1$(p, q) \leq Eps1$ Λ distribute$(p, q) \leq Eps2\})$. We assign to each cluster a *density factor,* which is the degree of the density of the cluster. The *density factor* of a cluster C is defined $1/(\Sigma_{p \in C}$ density distance $(p)/|C|)$, where $|C|$ is the number of objects in the cluster C.

Some clustering algorithms are not suitable for efficient distance computations when clusters are adjacent to each other. If the values of neighbor objects have little differences, the values of two border objects, like A and B objects, in a cluster may have very different values (Fig. 2a) However, cluster objects should be within a certain distance from the cluster means. We solve this problem by comparing the cluster mean (the average of the objects in the cluster) with the value of the new object (e.g. temperature, surface height) which tries to join this cluster. If the absolute difference between the cluster mean and the value of the new object is bigger than a threshold value, $\Delta\varepsilon$, then the new object is not appended to this cluster. In this case, the cluster should be divided into more than one sub-cluster. (Fig. 2b)

3. Description of the method

DBSCAN accepts a radius value *Eps* based on a user defined distance measure and a value *MinPts* for the number of minimal objects that should occur within *Eps* radius. The algorithm starts with an arbitrary object p. If the object p is a core object with respect to *Eps* and *MinPts*, a new cluster with p as the core object is created. Then, it continues to retrieve all density-reachable objects from the core object and add them to the cluster. If p is a border object or noise object, DBSCAN visits the next object of the database. The process terminates when no new object can be added to any cluster.

ST-DBSCAN requires four parameters: *Eps*1, *Eps*2, *MinPts*, and $\Delta\varepsilon$. *Eps*1 is the threshold parameter for spatial attributes (latitude and longitude). *Eps*2 is the threshold parameter for oceanographic values of seawater. *MinPts* is the minimum number of objects within *Eps*1 and *Eps*2 distance of an object. If a region is dense, then it should contain more points than *MinPts* value. In (Ester *et al.,* 1996), a simple heuristic is



```
FORALL objects o in database
  IF o is not clustered
    Find NEps(o) with respect to Eps1, Eps2, and density factor
    IF |NEps(o)| < MinPts   // if it is not a core object
      mark o as noise
    ELSE // o is a core object
      construct a new cluster
      mark all objects in NEps(o) with new cluster-id
      push all objects in NEps(o)
      WHILE NOT Stack.IsEmpty()
        newobject = Stack.Pop()
        Find NEps(newobject) with respect to Eps1, Eps2 and density factor
        IF |NEps(newobject)| ≥MinPts
          FORALL objects obj in NEps(newobject)
            IF (obj is not clustered or obj is not marked as noise) and |cluster_avg() - obj| ≤ Δε
                mark obj with current cluster-id
                push obj onto stack
```

Figure 3. Pseudo code of the algorithm.

presented which in many cases is effective in determining the parameters Eps and $MinPts$. The last parameter $\Delta\varepsilon$ is used to prevent the discovering of adjacent clusters.

The ST-DBSCAN algorithm starts with the first object in the database. (Fig. 3) If the current object does not belong to a cluster, the neighbor objects $N_{Eps}(o)$ of the current object o within $Eps1$ and $Eps2$ radius are retrieved efficiently by spatial access methods such as Quadtrees (Samet, 1990), R-Trees (Guttman, 1984), or others, see (Guting, 1994). If the number of returned points is smaller than $MinPts$ input (if the object o is not a core object), the object is assigned as noise. The points that have been marked to be noise may be changed later, if they are density-reachable from some other point of the database. This happens for border points of a cluster. If the neighborhood $N_{Eps}(o)$ of an object o has more than $MinPts$ elements, o is a so-called core object, and a new cluster containing the objects in $N_{Eps}(o)$ is created. In the next step, the neighbors of o which are not already contained in any cluster are added to the new cluster and their neighborhood is checked. The retrieval of density-reachable objects is performed by iteratively collecting directly density-reachable objects. This procedure is repeated until no new point can be added to the current cluster C. Then the algorithm continues with a point which has not yet been processed trying to expand a new cluster.

The computational complexity of DBSCAN comes up to $O(n{*}\text{logn})$ under the assumption that the data are organized in a spatial index (R*-tree). Our modifications don't change the runtime complexity of the algorithm. DBSCAN has been proven to have the ability to process very large databases (Ester *et al.,* 1998). The papers (Ester *et al.,* 1996, 1998) show that the runtime of other clustering algorithms such as DBCLASD (Xu *et al.,* 1998), CLARANS (Ng and Han, 1994) is between 1.5 and 3 times the runtime of DBSCAN. This factor increases with increasing size of the database.

Table 1. The sensitivity analysis results of $Eps1$, $Eps2$, and $MinPts$ parameters with respect to number of noise objects and number of clusters.

			Num. of			Num. of			Num. of
Parameter		Num. of	noise		Num. of	noise		Num. of	noise
change	Eps1	clusters	objects	Eps2	clusters	objects	MinPts	clusters	objects
0%	3.0	7	100	0.50	7	100	15.0	7	100
−10%	2.7	7	101	0.45	8	128	13.5	7	100
−20%	2.4	7	103	0.40	9	177	12.0	7	93
−30%	2.1	7	108	0.35	9	182	10.5	7	71
−40%	1.8	7	114	0.30	11	377	9.0	7	70
−50%	1.5	7	134	0.25	13	424	7.5	7	68
+10%	3.3	7	95	0.55	6	82	16.5	7	105
+20%	3.6	7	95	0.60	5	79	18.0	7	108
+30%	3.9	7	95	0.65	4	72	19.5	6	128
+40%	4.2	7	95	0.70	4	69	21.0	6	136
+50%	4.5	6	95	0.75	4	65	22.5	6	148

4. Sensitivity analysis of the algorithm

A sensitivity analysis is the process of varying model input parameters over a reasonable range and observing the relative change in model response. The purpose of the sensitivity analysis is to demonstrate the sensitivity of the model to uncertainty in values of input data. The sensitivity analysis helps to identify parameters that strongly affect model output. (Saltelli *et al.,* 2000).

Most sensitivity analyses involve changing one parameter at a time and changing each parameter value by ±10% in each time until reaching ±50%. The sensitivity analysis indicates a *robust* model when small changes in input parameters result in small changes in model output. If a small change in a parameter results in relatively large changes in the outcomes, the outcomes are said to be sensitive to that parameter. This may mean that the parameter has to be determined very accurately or that the model has to be redesigned for low sensitivity.

Our algorithm ST-DBSCAN requires three basic parameters: $Eps1$ (to measure the closeness of two objects geographically), $Eps2$ (to measure the similarity of oceanographic parameters of a marine environment), and *MinPts* (the minimum number of objects within $Eps1$ and $Eps2$ distance of an object). In the analysis, we used sea surface temperature values of Turkish Seas measured in years between 2001 and 2004. According to the heuristic defined in Ester *et al.* (1996), the input parameters should be assigned as $Eps1 = 3$, $Eps2 = 0.5$, and $MinPts = 15$. Table 1 shows the sensitivity analysis results of $Eps1$, $Eps2$, and $MinPts$ parameters with respect to number of noise objects and number of clusters. For example, when we increase the value of the $Eps1$ parameter by +10% and we stay fixed on other parameters, the number of clusters doesn't change and number of noise objects decrease by 5 units.

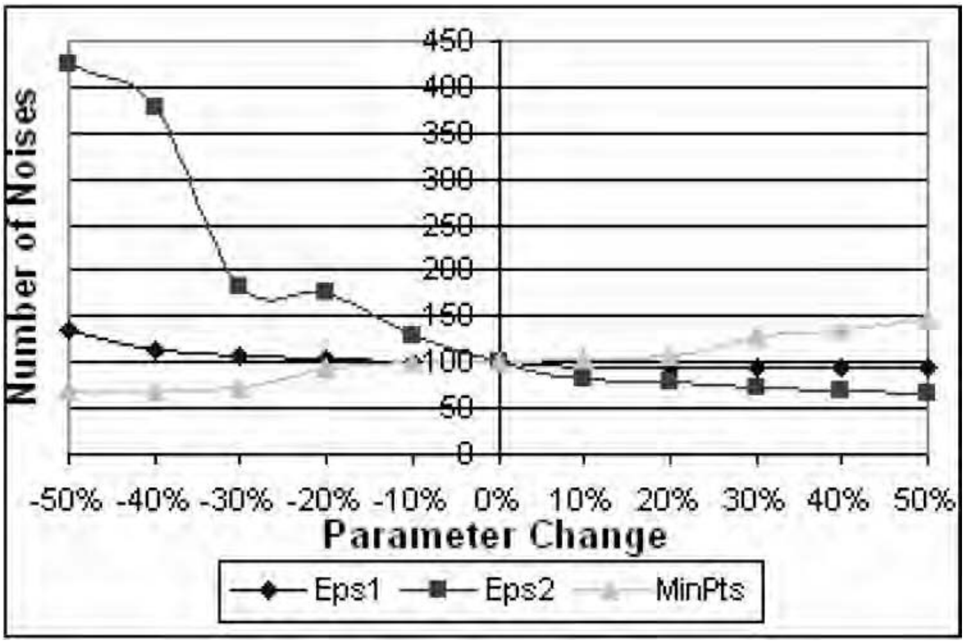
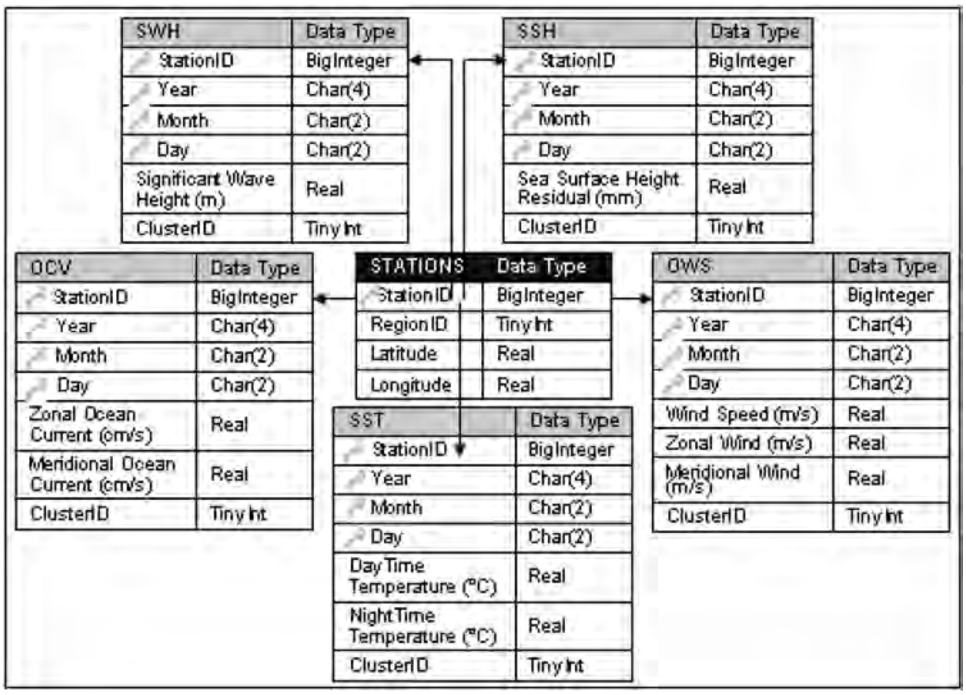


Figure 4. The sensitivity test results for *Eps*1, *Eps*2, and *MinPts* parameters.

We analyzed the sensitivity of our algorithm to the *Eps*1, *Eps*2 and *MinPts* parameters. (Fig. 4) According to the test results, *Eps*1 parameter has little influence on the algorithm output, because even a huge variation of *Eps*1 parameter, there is nearly no change in the number of clusters. The number of noise objects continuously increases when the value of *Eps*1 parameter decreases. But there is no change in the number of noise objects when the value of *Eps*1 parameter increases. The results show that *Eps*2 parameter is the most sensitive parameter. Decreasing in the value of the *Eps*2 parameter greatly increases the number of noise objects and the number of clusters. The output of the algorithm is very much dependent on the reliability of this parameter. The results also show that there is a low sensitivity between the output of the algorithm and *MinPts* parameter values. The output is lowly sensitive to big changes in the *MinPts* parameter values.

5. Oceanographic surveys in Turkish seas

We used the method described in the previous section for two physical oceanographic datasets which were collected from satellites between 1992 and 2004. The first run tries to discover the regions that have similar Sea-Surface Temperature (SST) values. The goal of the second example study is to identify spatially based partitions which have the similar Sea Surface Height (SSH) values.



SWH	Data Type
StationID	BigInteger
Year	Char(4)
Month	Char(2)
Day	Char(2)
Significant Wave Height (m)	Real
ClusterID	Tiny Int

SSH	Data Type
StationID	BigInteger
Year	Char(4)
Month	Char(2)
Day	Char(2)
Sea Surface Height Residual (mm)	Real
ClusterID	Tiny Int

OCV	Data Type
StationID	BigInteger
Year	Char(4)
Month	Char(2)
Day	Char(2)
Zonal Ocean Current (cm/s)	Real
Meridional Ocean Current (cm/s)	Real
ClusterID	Tiny Int

STATIONS	Data Type
StationID	BigInteger
RegionID	Tiny Int
Latitude	Real
Longitude	Real

OWS	Data Type
StationID	BigInteger
Year	Char(4)
Month	Char(2)
Day	Char(2)
Wind Speed (m/s)	Real
Zonal Wind (m/s)	Real
Meridional Wind (m/s)	Real
ClusterID	Tiny Int

SST	Data Type
StationID	BigInteger
Year	Char(4)
Month	Char(2)
Day	Char(2)
Day Time Temperature (°C)	Real
Night Time Temperature (°C)	Real
ClusterID	Tiny Int

Figure 5. The logical schema of the data warehouse.

a. Data warehouse description

We designed a data warehouse system which contains information about four seas: the Black Sea, the Marmara Sea, the Aegean Sea, and the East Mediterranean Sea. These seas surround the countries Turkey to the north, west, and south. The geographical coordinates of our work area are 30° to 47.5° north latitude and 17.0° to 42.5° east longitude. The data warehouse contains a central table, STATIONS, which interconnects the tables SST, SSH, (Significant Wave Height) SWH, Ocean Wind Speed (OWS), and Ocean Current Vector (OCV) by using StationID column. The data size is approximately 2.5 GB.

The logical schema of a data warehouse is a multi-dimensional data model. (Fig. 5) It contains space dimension, time dimension and fact dimension. The columns related with space dimension are StationID, RegionID, Latitude and Longitude. The first column, StationID, identifies the geographic location of an observed station. The column, RegionID, identifies the name of the sea (Black Sea, Marmara Sea, Aegean Sea, or Mediterranean Sea) which includes related station identifier. The time dimension can be grouped into Year, Month, and Day. The third dimension, fact dimension, represents the observations of the physical oceanographic parameters at different instances in time and spatial locations. Initially, the ClusterID column in all tables is set to zero as the default value, which indicates that all objects are not clustered. After clustering, it identifies a particular cluster that the current object in the same row is in it. If ClusterID column in a row is equal to -1 after clustering, this means the object in this row is a noise object.

Data in the data warehouse were obtained from The Physical Oceanography Distributed Active Archive Center (PO.DAAC) which is responsible for archiving and distributing data relevant to the physical state of the ocean. This data center distributes physical oceanographic data acquired by NASA instruments at different spatial and temporal resolutions. Data available at the PO.DAAC web site (http://podaac.jpl.nasa.gov/poet) were obtained from satellites and are intended for use in oceanographic and interdisciplinary scientific research.

SST data measured in the years between 2001 and 2004 were recorded at a spatial resolution of 18 km at weekly time periods. It was provided by NOAA-Series Satellites (National Oceanic and Atmospheric Administration) (NOAA/AVHRR, 2005). SSH dataset are currently available from 1992 to 2002 at 1 degree resolution with daily interval. It is provided by Topex/Poseidon Satellite (Topex/Poseidon, 2005). SWH data measured in years between 1992 and 2002 has also been provided by Topex/Poseidon Satellite. This satellite provides data as a uniform along-track grid with 6.2 km spacing. OWV data measured in between 1999 and 2004 is provided by QuikSCAT Satellite (QuikSCAT, 2005). The dataset is gridded at a resolution of 0.25 degree and daily time periods. OCV data can be provided by OSCAR Satellite (OSCAR, 2005).

b. Data transformation

Data transformation is an important aspect of data preprocessing to enable the dataset to assure the accuracy of the model. Data transformation, in terms of data mining, is the process of changing the form or structure of existing attributes. Data transformation involves converting data into a single common format acceptable to the data mining methodology. One of the most common forms of data transformation used in data mining is the normalization of the attributes, especially min-max normalization. In min-max normalization, attribute values are normalized to lie in a fixed interval given by the minimum and maximum values. It maps a value v of attribute A to v' in the range $[new\_min_A, new\_max_A]$ by the following formula:

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A.$$

In our case studies, we have to normalize temperature values because the temperature value of the same location changes in winter season (e.g. 9°C) and summer season (e.g. 20°C). We normalized the daytime and nighttime temperature attribute values of the SST table in the range [1, 24] and sea-surface height values of the SSH table in the range $[-14, +14]$. We applied min-max normalization on the oceanographic data in the data warehouse by using the following function:

```
Double min max_norm(Double a, Double MinA, Double MaxA,
                                        Double NewMinA, Double NewMaxA)

    Return(a − MinA)/(MaxA − MinA)∗(NewMaxA − NewMinA) + NewMinA;
```

SST satellite data are computed by applying the MultiChannel Sea-Surface Temperature (MCSST) algorithm on the maps derived from the AVHRR (Advanced Very High Resolution Radiometer). This algorithm uses the formula SST $= a{*}T4 + b{*}(T4 - T5){*}Tf + c{*}(\sec(q)\ 1){*}(T4 - T5) - d$, where $T4$ and $T5$ are the brightness radiances in channels 4 and 5, Tf is a analyzed SST or MCSST fields, and a, b, c, d are empirically derived coefficients.

SSH values are calculated by the formula SSHR = SSH − MSS − Tide Effects − Inverse Barometer, where SSHR is the Sea Surface Height Residual value, SSH is the Sea Surface Height value, and MSS is Mean Sea Surface height value. The residual sea surface is defined as the sea-surface height minus the mean sea-surface and minus known effects; i.e., tides and inverse barometer.

SWH data are calculated from altimeter data based on the shape of a radar pulse after it bounces off the sea surface. A calm sea with low waves returns a sharply defined pulse whereas a rough sea with high waves returns a stretched pulse. The significant wave height is the average height of the highest one-third of all waves in a particular time period.

c. *Oceanographic applications and results*

We used our clustering algorithm two times on two different datasets. The first application clusters SST data by assigning input parameters as $Eps1 = 3$, $Eps2 = 0.5$, and $MinPts = 15$ to find the regions that have similar sea-surface temperature characteristics. These values for the input parameters are determined by using the heuristics given in (Ester *et al.,* 1996). We first worked on the spatial distribution of temperature in surface water (30–47.5N and 17–42.5E) (Fig. 6). In the work area, there are 5340 stations and each station is shown as a dot. Seven clusters were discovered at the end of the data mining and each cluster has data points that have similar sea surface temperature characteristics. Cluster number 1 is bordered by Ukraine and Russia. This region is the coldest area. Cluster number 2 at the north of the Ukraine is the second coldest area. The seawater temperatures of other parts of the Black Sea are similar with the Marmara Sea. Cluster number 4 covers the north of the Aegean Sea. Cluster number 5 forms a great single cluster. The temperature values of the stations in Cluster 6 have also similar characteristics. Cluster number 7 is the hottest region because it is the closest area to equator. In winter seasons, C5 and C7 clusters can be marked as one cluster because they are not well distinguished. In summer seasons, C6 cluster decreases in size. Many factors can effect this distribution of seawater temperature. The temperature varies both latitudinally and depthwise in response to changes in air-sea interactions. Heat fluxes, evaporation, river inflow, the movement of water and rain all influence the distribution of seawater temperature.

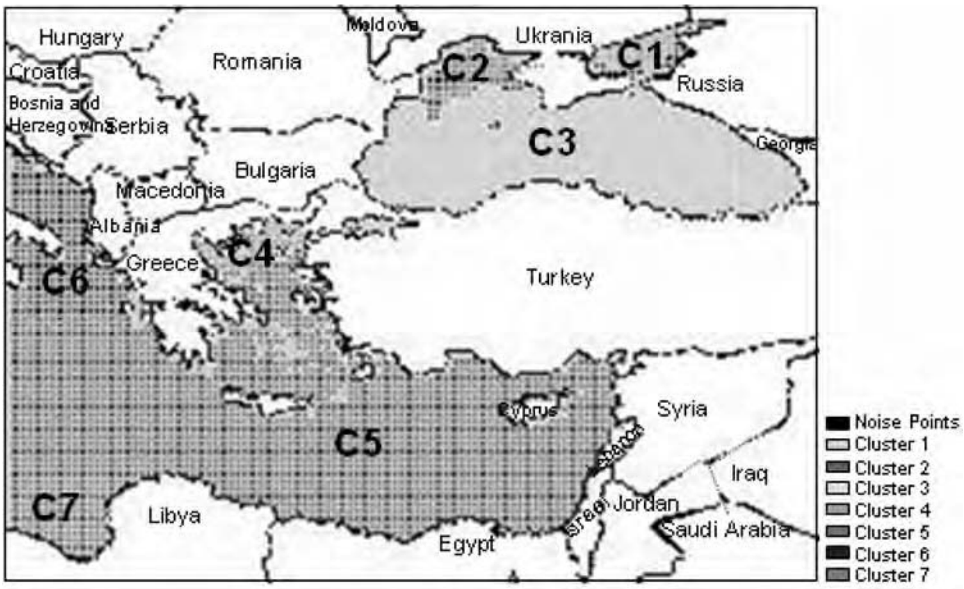


Figure 6. The locations of 5340 stations and cluster analysis results for SST data.

Table 2 also shows the clustering results. In the table, the first column identifies the table, the second column presents the number of objects in each cluster, the third and sixth columns display the average values of objects in each cluster. Other columns show the minimum and maximum values in each cluster.

The second application discovers the regions that have similar sea-surface height residual values by using the SSH table and by assigning input parameters as $Eps1 = 3$, $Eps2 = 1$, and $MinPts = 4$. Table 3 shows the clustering results as a list. In the work area, there are 134 stations and ten clusters are obtained by the usage of the SSH table (Fig. 7). Each cluster has data points that have similar sea-surface height residual values. The clusters named by C1, C2, C3 and C4 are located in the Black Sea. The cluster named by C7 is located in the Aegean Sea. The rest of the clusters are located in the Mediterranean

Table 2. Clustering results for SST data.

ClusterID	Num. of obj.	AVG (day temp.)	MIN (day temp.)	MAX (day temp.)	AVG (night temp.)	MIN (night temp.)	MAX (night temp.)
1	63	2.74	2.09	4.59	2.71	1.00	4.55
2	92	10.05	9.60	11.10	9.67	8.39	10.75
3	1306	13.70	11.40	16.20	13.41	9.44	15.75
4	90	16.35	15.75	16.65	16.25	15.30	16.54
5	3061	20.14	16.35	22.5	20.04	16.20	22.35
6	349	14.35	13.25	15.75	14.01	13.10	15.65
7	279	22.73	21.50	24.00	22.67	21.45	23.60

Table 3. Clustering results for SSH data.

ClusterID	Num. of obj.	AVG (height)	MIN (height)	MAX (height)
1	9	4.53	3.80	5.40
2	8	12.60	10.90	14.00
3	6	7.58	7.00	8.50
4	11	11.41	9.69	13.00
5	4	1.30	1.20	1.50
6	17	−11.21	−14.00	−9.40
7	6	−0.91	−1.40	−0.40
8	7	2.51	1.80	3.80
9	29	−0.86	−6.00	3.50
10	32	−4.31	−5.90	−2.90

Sea. Many factors contribute to changes in sea-surface height, including sea eddies, temperature of the upper seawater, tides, sea currents, and gravity.

6. Cluster validation

Since clustering algorithms define clusters that are not known *a priori* information, the clustering results require some kind of evaluation in most applications. The correctness of clustering algorithm results is verified using appropriate criteria and techniques. (Halkidi *et al.,* 2001) Visualization of the data set is a crucial verification of the clustering results. (Fig. 8) While the left side of the figure shows the Topex/Poseidon satellite map, the right side of

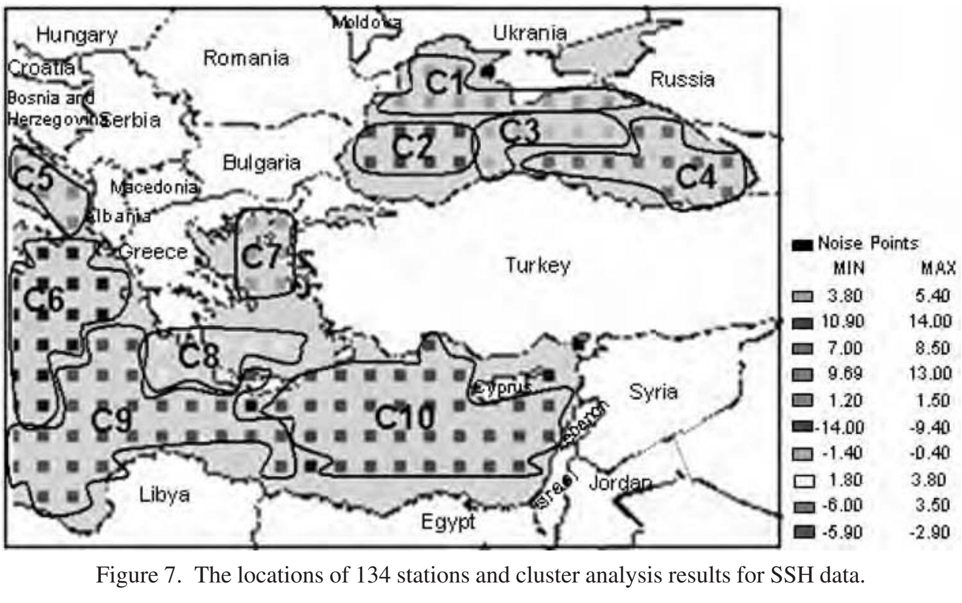


Figure 7. The locations of 134 stations and cluster analysis results for SSH data.

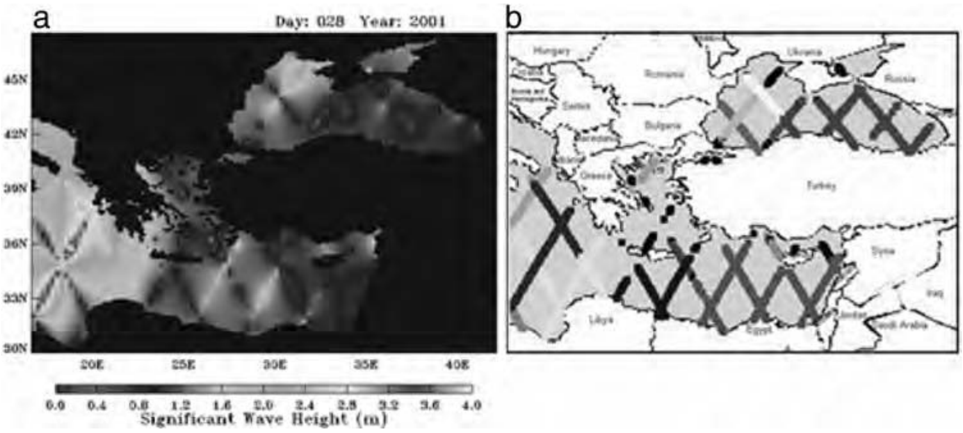


Figure 8. Visual cluster validation.

it shows our clustering result obtained by the usage of the SWH dataset measured on January 28, 2001. The goal of this application is to discover the regions that have similar SWH values. In this application, we assigned input parameters as $Eps1 = 1$, $Eps2 = 0.25$, and $MinPts = 15$.

In order to validate the clustering results, some validity indexes can be used such as RS index, CD index, and SD validity index. (Halkidi *et al.,* 2001) In this study, we used the RS index which may be considered as a measure of the degree of difference between clusters. Furthermore, it measures the degree of homogeneity between groups. RS is the ratio of SS_b to SS_t, where SS means sum of squares, SS_b is a measure of difference between clusters, SS_w is a measure of difference within clusters and SS_t is equal to $\mathrm{SS}_b + \mathrm{SS}_w$.

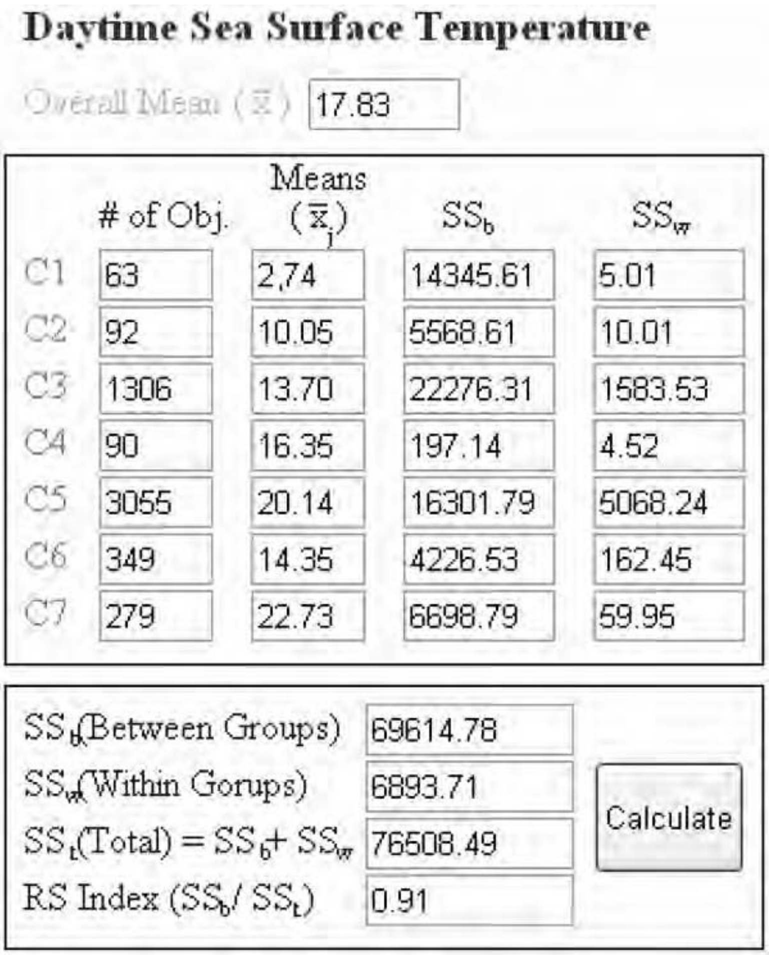
$$\mathrm{SS}_b = \sum_{j=1}^{p} n_j(\bar{x}_j - \bar{x})^2$$

$$\mathrm{SS}_w = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (x_{ij} - \bar{x}_j)^2$$

$$\mathrm{SS}_t = \sum_{j=1}^{p} \sum_{i=1}^{n_j} (x_{ij} - \bar{x})^2$$

where p is number of clusters, n_j is the number of observations in the cluster j, $\bar{x}$ is the overall mean (the sum of all the observations divided by the total number of observations), $\bar{x}_j$ is the mean of the cluster j, and x_{ij} is the i^{th} observation in the cluster j.

The values of RS range between 0 and 1. In case that the value of RS is zero (0) indicates that no difference exists among groups. On the other hand, when RS equals 1



Daytime Sea Surface Temperature

Overall Mean ($\bar{x}$) 17.83

	# of Obj.	Means ($\bar{x}_j$)	SS_b	SS_w
C1	63	2,74	14345.61	5.01
C2	92	10.05	5568.61	10.01
C3	1306	13.70	22276.31	1583.53
C4	90	16.35	197.14	4.52
C5	3055	20.14	16301.79	5068.24
C6	349	14.35	4226.53	162.45
C7	279	22.73	6698.79	59.95

SS_b(Between Groups)	69614.78
SS_w(Within Gorups)	6893.71
SS_t(Total) = SS_b+ SS_w	76508.49
RS Index (SS_b/ SS_t)	0.91

Figure 9. The evaluation and validation of the clustering results.

there is an indication of significant difference among groups. We developed a small program to present the evaluation and validation of our clustering results. (Fig. 9) As shown in this figure, RS validity index of our clustering results is 0.91. So the difference between clusters discovered by new method is quite high and the homogeneity within the clusters is quite low. Thus the mathematical quality and reliability of our clustering results are quite good.

7. Conclusions and future works

This study focuses on clustering to obtain a meaningful water map according to the different physical characteristics. It presents a new density-based clustering algorithm. The main objective of this algorithm is to obtain the regions (clusters) that have similar physical parameter values in a marine environment. As an example, we used the algorithm to show the spatio-temporal distributions of sea-surface temperature and sea-surface height residual values in Turkish seas. Experimental results show that the new algorithm is useful for describing physical characteristics of seas. According to the sensitivity analysis results of the algorithm, $Eps2$ parameter is the most sensitive parameter for the output of the method. In order to evaluate and validate our clustering results, we used RS validity index. RS index values calculated from our clustering results show that the mathematical quality and reliability of the clustering results is very high.

Extensive analysis in large databases requires too much time and needs extreme computing power. In future studies, we will try to implement the same algorithm by using parallel processing techniques in order to improve the performance.

Acknowledgments. This study was supported by the Scientific Research Projects Directorate of the Dokuz Eylul University. We would like to thank the staff of the Institute of Marine Sciences and Technology at Dokuz Eylul University for sharing their knowledge with us and for their helps in obtaining data.

REFERENCES

Ankerst, M., M. M. Breunig, H.-P. Kriegel and J. Sander. 1999. OPTICS: Ordering points to identify the clustering structure, *in* Proceedings of ACM SIGMOD International Conference on Management of Data, Philadelphia, PA, 49–60.

Emery, W. J. and R. E. Thomson. 2001. Data Analysis Methods in Physical Oceanography, 2nd ed., Elsevier Science, 658 pp.

Ester, M., H.-P. Kriegel, J. Sander, M. Wimmer and X. Xu. 1998. Incremental clustering for mining in a data warehousing environment, *in* Proceedings of International Conference on Very Large Databases (VLDB '98), New York, 323–333.

Ester, M., H.-P. Kriegel, J. Sander and X. Xu. 1996. A density-based algorithm for discovering clusters in large spatial databases with noise, *in* Proceedings of Second International Conference on Knowledge Discovery and Data Mining, Portland, OR, 226–231.

Guting, R. H. 1994. An introduction to spatial database system. VLDB Journal, *3,* 357–399.

Guttman, A. 1984. R-trees: A dynamic index structure for spatial searching, *in* Proceedings of ACM SIGMOD Int. Conference on Management of Data, Boston, MA, 47–57.

Halkidi, M., Y. Batistakis and M. Vazirgiannis. 2001. On clustering validation techniques. J. Intelligent Infor. Syst., *17,* 107–145.

Han, J. and M. Kamber. 2001. Data Mining Concepts and Techniques, Morgan Kaufmann Publishers, San Francisco, CA, 533 pp.

Hinneburg, A. and D. A. Keim. 1998. An efficient approach to clustering in large multimedia databases with noise, *in* Proceedings of 4th International Conference on Knowledge Discovery and Data Mining, NY, 58–65.

Ma, S., T. J. Wang, S. W. Tang, D. Q. Yang and J. Gao. 2003. A new fast clustering algorithm based on reference and density. Lecture Notes in Computer Science, *2762,* 214–225.

Murray, A. T. and V. Estivill-Castro. 1998. Cluster discovery techniques for exploratory spatial data analysis. Inter. J. Geograph. Infor. Sci., *12,* 431–443.

NAVOCEANO Satellite—sea surface temperature data web site. Retrieved January 2005, from http://podaac.jpl.nasa.gov/navoceano_mcsst/

Ng, R. T. and J. Han. 1994. Efficient and effective clustering methods for spatial data mining, *in* Proceedings of 20th Inter. Conf. on Very Large Data Bases, Santiago, Chile, 144–155.

OSCAR Satellite—ocean current vector data web site. Retrieved January 2005, from http://www.oscar.noaa.gov/

Qian, W. N. and A. Y. Zhou. 2002. Analyzing popular clustering algorithms from different viewpoints. J. Software, *13,* 1382–1394.

QuikSCAT Satellite—sea winds data web site. Retrieved January 2005, from http://podaac.jpl.nasa.gov/quikscat/

Robinson, A. R. and M. Golnaraghi. 1994. The physical and dynamical oceanography of the Mediterranean, *in* Ocean Processes in Climate Dynamics: Global and Mediterranean Examples, P. Malanotte-Rizzoli and A. R. Robinson eds., Kluwer Academic Publishers, The Netherlands, 255–306.

Saltelli, A., K. Chan and E. M. Scott. 2000. Sensitivity Analysis, John Wiley & Sons, Chichester, England, 10.

Samet, H. 1990. The Design and Analysis of Spatial Data Structures, Addison-Wesley, 493 pp.

Topex/Poseidon Satellite—Sea Surface Height and Sea Wave Height Data Web Site. Retrieved January 2005, from http://podaac.jpl.nasa.gov/woce/

Xu, X., M. Ester, H.-P. Kriegel and J. Sander. 1998. A nonparametric clustering algorithm for knowledge discovery in large spatial databases, *in* Proceedings of IEEE International Conference on Data Engineering, Orlando, FL, 324–331.

Received: *19 September, 2005;* revised: *6 June, 2006.*