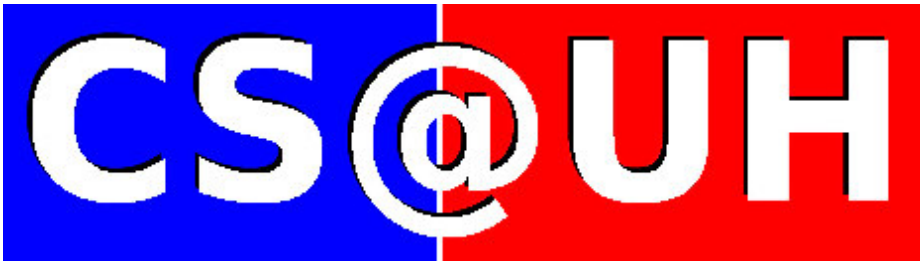


ON SUPERVISED DENSITY ESTIMATION TECHNIQUES AND THEIR APPLICATION TO CLUSTERING

Dan Jiang, Christoph F. Eick, and Chun-sheng Chen

Department of Computer Science
University of Houston
Houston, TX, 77204, USA
http://www.cs.uh.edu

Technical Report Number UH-CS-07-09

August 13, 2007

Keywords: Density estimation, spatial data mining, hot spot discovery, density-based clustering.

Abstract

The basic idea of traditional density estimation is to model the overall point density analytically as the sum of influence functions of the data points. However, traditional density estimation techniques only consider the location of a point. Supervised density estimation techniques, on the other hand, additionally consider a variable of interest that is associated with a point. Density in supervised density estimation is measured as the product of an influence function with the variable of interest. Based on this novel idea, a supervised density-based clustering named SCDE is introduced and discussed in detail. The SCDE algorithm forms clusters by associating data points with supervised density attractors which represent maxima and minima of a supervised density function. Results of experiments are presented that evaluate SCDE for hot spot discovery and co-location discovery in spatial datasets. Moreover, the benefits of the presented approach for generating thematic maps are briefly discussed.

ON SUPERVISED DENSITY ESTIMATION TECHNIQUES AND THEIR APPLICATION TO CLUSTERING

Dan Jiang, Christoph F. Eick, and Chun-sheng Chen

Abstract

The basic idea of traditional density estimation is to model the overall point density analytically as the sum of influence functions of the data points. However, traditional density estimation techniques only consider the location of a point. Supervised density estimation techniques, on the other hand, additionally consider a variable of interest that is associated with a point. Density in supervised density estimation is measured as the product of an influence function with the variable of interest. Based on this novel idea, a supervised density-based clustering named SCDE is introduced and discussed in detail. The SCDE algorithm forms clusters by associating data points with supervised density attractors which represent maxima and minima of a supervised density function. Results of experiments are presented that evaluate SCDE for hot spot discovery and co-location discovery in spatial datasets. Moreover, the benefits of the presented approach for generating thematic maps are briefly discussed.

I. INTRODUCTION

The goal of density estimation techniques [1] is to model the distribution of the underlying population from the sample data collected. This technique measures the density at a point according to the impact of other points observed within its neighborhood using influence functions—the influence of a point on another point decreases as their distance increases. Density at a point is measured as the sum of the influences of data points in its neighborhood.

However, traditional density estimation techniques only consider the spatial dimension of data points ignoring non-spatial information. In this paper, we propose a novel density estimation technique called *supervised density estimation*. It differs from the traditional density estimation by additionally considering a non-spatial variable of interest in its influence function. The new influence function is defined as the product of a traditional influence function and the variable of interest.

In the remainder of this paper we will try to convince the reader that this generalization leads to novel applications for spatial data mining. One direct application of supervised density estimation techniques is the generation of thematic maps for a selected variable [11]. Thematic maps can serve as a visual aid to the domain experts for quickly identifying the interesting regions for further investigations or for finding relations between different features. Another application for supervised density estimation techniques is hot spot discovery in spatial datasets. For example, we might be interested in identifying regions of high risk from earthquakes, based on the past earthquakes that are characterized by longitude, latitude, and severity of the earthquake measured using the Richter scale. Algorithms to find such hot spots will be proposed in this paper.

The main contributions of the paper include: the introduction of a novel supervised density estimation technique (discussed in Section 3), the proposal of a new supervised density-based clustering algorithm called SCDE (discussed in Section 4) that operates on the top of supervised density functions, and the evaluation of the proposed

techniques for applications that originate from geology, planetary, and environmental sciences (discussed in Section 5).

II. RELATED WORK

Many clustering algorithms exist for spatial data mining [11]; among them, density-based algorithms [4, 6, 7, and 16] have been found to be most promising for discovering arbitrary shaped clusters. DBSCAN [4] uses a straightforward definition for a density function which counts the number of points within a predefined radius. Its main drawbacks are the need for parameter tuning and poor performance for datasets having varying density. DENCLUE [6] uses kernel density estimation techniques and its clusters are formed by associating objects with maxima of the so-defined density function, relying on a steepest decent hill climbing procedure. DENCLUE 2.0 [21] improves the original algorithm by using a different hill climbing technique.

Methods of finding hot spots in spatial datasets have been investigated in the past both explicitly and implicitly. Because hot spots represent clusters with respect to spatial coordinates, their detection lies at the heart of spatial data mining and has been investigated in [5, 8, 9, 12, 21]. More explicitly, detection of hot spots using variable resolution approach [17] was investigated in order to minimize the effects of spatial super imposition. In [13] a region growing method for hot spots discovery was described, which selects seed points first and then grows clusters from these seed points by adding neighbor points as long as a density threshold condition is satisfied. The definition of hot spots was extended in [10] to cover a set of entities that are of some particular, but crucial, importance to the domain of experts. This is a feature-based definition, somewhat similar, but less specific, to what we are using in the presented paper. This definition was applied to relational databases to find important nuggets of information. Finally, in [18] feature-based hot spots are defined in a similar sense as in this paper, but their discovery is limited to datasets with a single categorical variable.

III. SUPERVISED DENSITY ESTIMATION

Throughout the paper, we assume that datasets have the form (<location>,<variable_of_interest>); for example, a dataset describing earthquakes whose location is described using longitude and latitude and whose variable of interest is the earth quake's severity measured using the Richter scale.

More formally, a dataset O is a set of data objects, where n is the number of objects in O belonging to a feature space F.

$$O = \{o_1, \ldots, o_n\} \subseteq F \qquad (3\text{-}1)$$

For the reminder of this section we assume that objects $o \in O$ have the form $((x, y), z)$ where (x, y) is the location of object o, and z—denoted as $z(o)$ in the following—is the value of the variable of interest of object o. The variable of interest can be continuous or categorical. Moreover, the distance between two objects in O $o1=((x_1, y_1), z_1)$ and $o_2=((x_2, y_2), z_2)$ is measured as $d((x_1, y_1), (x_2, y_2))$ where d denotes a distance measure. Throughout this paper d is assumed to be Euclidian distance.

In the following, we will introduce *supervised density estimation* techniques. Density estimation is called supervised, because in addition to distance information the feedback $z(o)$ is used in density functions. Supervised density estimation is further subdivided into *categorical density estimation*, and *continuous density estimation*, depending whether the variable z is categorical or continuous.

In general, density estimation techniques employ *influence functions* that measure the influence of a point $o \in O$ with respect to another point $v \in F$; in general, a point o's influence on another point v's density decreases as the distance between o and v increases. In contrast to past work in density estimation, our approach employs *weighted influence functions* to measure the density in datasets O: the influence of o on v is measured as a product of $z(o)$ and a Gaussian kernel function. In particular, the influence of object $o \in O$ on a point $v \in F$ is defined as:

$$f_{Influence}(v, o) = z(o) * e^{-\frac{d(v,o)^2}{2\sigma^2}} \qquad (3\text{-}2)$$

If $\forall o \in O$ $z(o)=1$ holds, the above influence function becomes a Gaussian kernel function, commonly used for density estimation and by the density-based clustering algorithm DENCLUE[6]. The parameter σ determines how quickly the influence of o on v decreases as the distance between o and v increases.

The influence function measures a point's influence relying on a distance measure. For the points in Figure 3-1, the influence of o_1 with respect to point v larger than the influence of o_2 on point v, that is $f_{Influence}(v,o_1) > f_{Influence}(v,o_2)$ because $d(v,o_1) < d(v,o_2)$

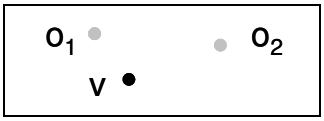


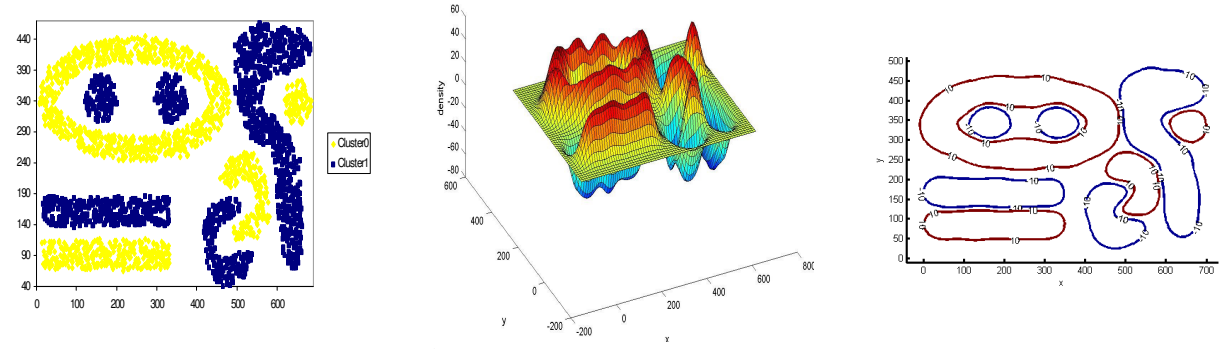
Figure 3-1. Example of Influence Computations.

The overall influence of all data objects $o_i \in O$ for $1 \le i \le n$ on a point $v \in F$ is measured by the density function $\psi^O(v)$, which is defined as follows:

$$\psi^o(v) = \sum_{i=1}^{n} f_{Influence}(v, o_i) \qquad (3\text{-}3)$$

In *categorical density estimation*, we assume that the variable of interest is categorical and takes just two values that are determined by the membership in a class of interest. In this case, $z(o)$ is defined as follows:

$$z(o) = \begin{cases} 1 & \text{if } o \text{ belong to the class of interest} \\ -1 & \text{otherwise} \end{cases} \qquad (3\text{-}4)$$



a) b) c)

Figure 3-2. B-Complex9 dataset with its density function and a density contour map.

Figure 3-2 depicts a dataset in which objects belong to two classes, depicted in blue and yellow color. O contains the points of Figure 3-2 and is assumed to have the form $((x, y), z)$ where z takes the value +1 if the object belongs to class yellow and -1 if the object belongs to class blue. Figure 3-2.b visualizes the density function ψ^O for dataset O. In the display, maxima identify areas in which the class yellow dominates, minima indicate areas in which the class blue dominates, and the density value of 0 represents decision boundaries—areas in which we have a complete balance with respect to instances of the two classes blue and yellow. Figure 3-2.c shows the density contour map for density values 10 as the red lines and -10 as the blue lines for ψ^O.

Moreover, the variable of interest can be continuous. Let us consider we are conducting a survey for an insurance company and we are interested in measuring the risk of potential earthquake damage at different locations. Considering the earthquake dataset depicted below in Figure 3-3, the variable of interest is the severity of a past earthquake, which is continuous. In region A there have been many low severity earthquakes; in region B, there have been few slightly more severe earthquakes. According to our previous discussion, the influence of data objects far away is much less than the influence of data objects nearby and the influence of nearby data objects is more significant than the influence of data objects far away. Using formula (3-3) for this example, we will reach the conclusion that Region A is more risky than Region B, because of its higher frequency of earthquakes, whereas

region B that is characterized by rare, although slightly more severe earthquakes. It should be noted that traditional interpolation functions will fail to reach the proper conclusion for this example: the average severity in region B is 4, whereas the average severity in region A which is 3. In summary, continuous density estimation does not only consider the value of the variable of interest but also takes the frequency with which spatial events occur into consideration—in general, density increases as frequency increases.

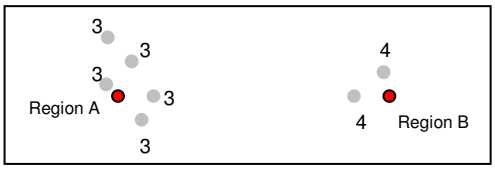


Figure 3-3. Example of Continuous Density Estimation

Finally, it should be noted that the proposed supervised density estimation approach uses kernels that can take negative values, which might look unusual to some readers. However, in a well-known book on density estimation [1] *Silverman observes "there are some arguments in favor of using kernels that take negative as well as positive values...these arguments have first put forward by Parzen in 1962 and Bartlett in 1963"*.

IV. SUPERVISED HOT SPOTS DISCOVEY USING DENSITY ESTIMATION TECHNIQUES

Discovering hot spots and cool spots in spatial datasets with respect to a variable of interest is a challenging ongoing topic. The variable of interest can be a categorical or continuous. There are many possible algorithms to compute hot and cool spots; one such algorithm called SCDE (Supervised Clustering Using Density Estimation) will be introduced in the remainder of this section. It is a density based clustering algorithm that operates on supervised density functions that have been introduced earlier. The way SCDE constructs clusters has some similarity with the way the popular, density-based clustering DENCLUE [6] forms clusters. However, it is important to stress that SCDE—as we pointed out in Section2—operates on a much more general density function allowing to address a much broader class of problems, such as supervised clustering and co-location mining involving continuous variables.

A. The SCDE Algorithm

The SCDE algorithm operates on the top of the supervised influence and density functions that were introduced in formulas (3-2) and (3-3). Its clustering process is a hill-climbing process that computes density attractors. During the density attractor calculation process, data objects are associated with density attractors forming clusters. The final clusters represent hot and cool spots with respect to the variable of interest z.

Density Attractor

A point, a, is called a density attractor of a dataset O if and only if it is a local maximum or minimum of the supervised density function ψ^{ρ} and $|\psi^{\rho}(a)| > \xi$, where ξ is a density threshold parameter. For any continuous and differentiable influence function, the density attractors can be calculated by using a steepest decent hill-climbing procedure.

Pseudo-Code of the SCDE Algorithm

SCDE computes[1] clusters by associating the objects in a dataset O with density attractors with respect to the density function ψ^{ρ}. This is similar with the way the popular, density-based clustering algorithm, DENCLUE [6], creates clusters. However, it is important to stress that SCDE has to compute maxima and minima of the supervised density function; therefore, the sign of $\psi^{\rho}(o)$ plays an important role for determining whether to ascent or to decent in the employed hill climbing procedure. Moreover, this paper describes in much more detail how the clusters are

[1]To increase the calculation speed, we use a local density function to replace the global density function. It only considers the influence of the data point in a predefined neighborhood. Only the data objects $o \in O$ within $o \in$ neighborhood(v) will therefore be used to estimate the density at point v. Neighborhoods can be defined in many different ways. In SCDE, we partition the dataset into grid-cells and only objects in the grid-cell that v belongs to and its neighboring grid cells are used to compute the density of point v.

formed during the hill climbing procedure, whereas the original DENCLUE paper only gives an implementation sketch still leaving many implementation details un-discussed. Figure 4-1 gives the pseudo code of SCDE. It employs a steepest decent/ascent hill climbing procedure which moves in the direction of the largest/smallest gradient of $\psi^{O}(v)$ based on the sign of the density value of v; the length of the movement in the chosen direction is determined by the parameter step. The hill climbing procedure stops when the absolute value of difference of supervised density value between two iterations is less than the density value threshold ϖ.

Inputs

- Dataset $O=\{o_1,...,o_n\}$
- Step size during the iteration, *step*
- Standard deviation of supervised influence function, σ.
- Density value threshold, ξ.
- Density difference threshold, ω.

Output

- Final clusters $X=\{c_1, \ldots, c_p\}$

Algorithm

1. Partition the feature space F to M cells with cell edge size, 4σ.

2. For any un-clustered data objects, calculate the density attractors iteratively

a) $o_i^0 := o_i$ and $c_i := \{ o_i \}$;

b) Repeat until ($sign(\psi(o_i^{k+1})) * (\psi(o_i^{k+1}) - \psi(o_i^k)) < \varpi)$)

$$o_i^{j+1} = o_i^j + sign\ (\psi(o_i^j)) * step\ * \frac{\nabla \psi(o_i^j)}{\| \nabla \psi(o_i^j) \|}$$

where

$$\nabla \psi(x,y) = (dx/(\|dx,dy\|), dy/\|(dx,dy)\|)$$

where

$$dx = \psi(x+step, y) - \psi(x,y)\ and\ dy = \psi(x, y+step) - \psi(x,y)$$

c) If $|\psi(o^k)| > \xi$ then

For any data object u close to o_i^{j+1}, with $distance(u, o_i^{j+1}) < \sigma/2$ and density values $\psi(u)$ and $\psi(o_i^{j+1})$ are same sign

i) If u is an un-clustered data object

Add the data object u to the same cluster c_i which o_i belongs to, cluster c_i is a new cluster

ii) If u belongs another cluster c_j

Merge cluster c_i and c_j

Figure 4-1 SCDE pseudo code

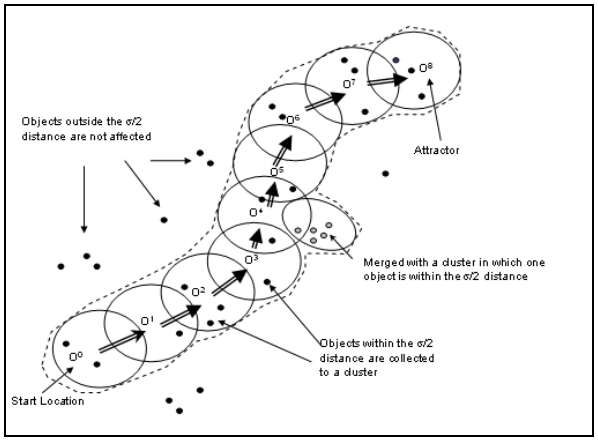


Figure 4-2 Schematic of the SCDE clustering process in one attractor calculation step

SCDE does not calculate the density attractor for every data object. During the density attractor calculation, it examines data objects close to the current point when moving towards the supervised density attractor. If their density values have the same sign as the current density value at the calculating point, these data objects will be associated with the same final density attractors after it has been computed.

Figure 4-2 illustrates the clustering process during the calculation of a particular attractor. All the solid objects in the dashed curve belong to one cluster. The final cluster includes all the data objects inside the curve that is enclosed by a dashed line. In each iteration step, we locate the data objects close to o_i^{j+1}, i.e. data objects whose distance to o_i^{j+1} is less than $\sigma/2$. Each data object close to o_i^{j+1} is processed as follows:

If the data object does not belong to any cluster yet—such as the solid black objects inside the dashed curve in Figure 4-2—the object is marked with the same cluster ID as those attracted by the current attractor computation.

If the data object already belongs to a cluster c_i—such as the solid grey objects in Figure 4-2—all the data points in the cluster c_i are marked with the same cluster ID as those objects that have been collected by the current attractor computation.

The hill climbing procedure stops returning a density attractor a; if $|\psi^{\rho}(a)| > \xi$, a is considered a supervised density attractor, and a cluster is formed.

Our currently implemented hill climbing procedure just considers the density of two points—that are step apart from the current point—when determining the direction using formula 4-1.

$$\begin{aligned} &Let\ dx=\psi(x+step,y)-\psi(x,y)\ and\ dy=\psi(x,y+step)-\psi(x,y) \\ &\nabla\psi(x,y)=\nabla\psi(u)=(dx/(||(dx,dy)||),dy/(|dx|+|dy|) \end{aligned} \tag{4-1}$$

Alternatively, instead of using just two points to determine directions, all points in the neighborhood of the current hill climbing point could be used.

$$\nabla\psi(u)=\sum_{v\in neighbor(u)}(v-u)*(\psi(v)-\psi(u))/\|v-u\| \tag{4-2}$$

It is important to stress that DENCLUE's hill climbing procedure [6] uses the influence function and not the density function to determine in which direction to proceed.

$$\nabla\psi(u)=\sum_{v\in neighbor(u)}(v-u)*f_{Influence}(u,v) \tag{4-3}$$

However, we strongly believe this formula should be normalized—otherwise, far away points would receive a higher weight in direction computations—obtaining formula 4-4. This approach considers all the points in the neighborhood of an iteration point, and the influence of a point in the neighborhood decreases as the distance to the iteration point increases.

$$\nabla\psi(u)=\sum_{v\in neighbor(u)}(v-u)*f_{Influence}(u,v)/\|v-u\| \tag{4-4}$$

Finally, DENCLUE 2.0 [21] uses a novel iteration procedure that directly solves $\nabla\psi=0$ for the influence function, formulation 4-3, of the optimization problem; if we adapt this approach for our modified influence function we obtain:

$$o_i^{j+1}=\frac{\sum_{t=1}^{n}f_{Influence}(o_i^j,x_t)*x_t}{\sum_{t=1}^{n}f_{Influence}(o_i^j,x_t)} \tag{4-5}$$

$$o_i^{j+1}=\frac{\sum_{x\in near(o_i^j)}f_{Influence}(o_i^j,x)*x}{\sum_{x\in near(o_i^j)}f_{Influence}(o_i^j,x)} \tag{4-6}$$

The most critical parameter when using SCDE is σ, which determines the size of the influence region of an object. Larger values for σ increases the influence regions and more neighboring objects are affected. From a global point of view, larger values for σ results more objects are connected and joined together by density paths between them. Therefore when σ decreases, there are more local maxima and minima and the number of clusters increases.

To select proper parameters for SCDE, we can visualize the density function and/or create a density contour map of the dataset for some initial values for σ and ξ first, and then use the displays to guide the search for good parameter values. Sometimes, the domain experts will give us the approximate number of clusters or number of outliers; this information can be used to determine good values for σ and ξ automatically using a simple binary search algorithm. Based on our past experience, choosing good values for the other parameters of SCDE is straight forward.

B. SCDE Run Time Complexity

The run time of SCDE depends on algorithm parameters, such as σ, ξ, ω and step. The grid partitioning part takes *O(n)* in the worst case, where n is the total number of data objects in the dataset. The run time for calculating the density attractors and clustering varies for different values of parameters σ and ξ. It will not go beyond *O(h*n)* where *h* is the average hill-climbing time for a data object. The total run time is approximately *O(n + h*n)*.

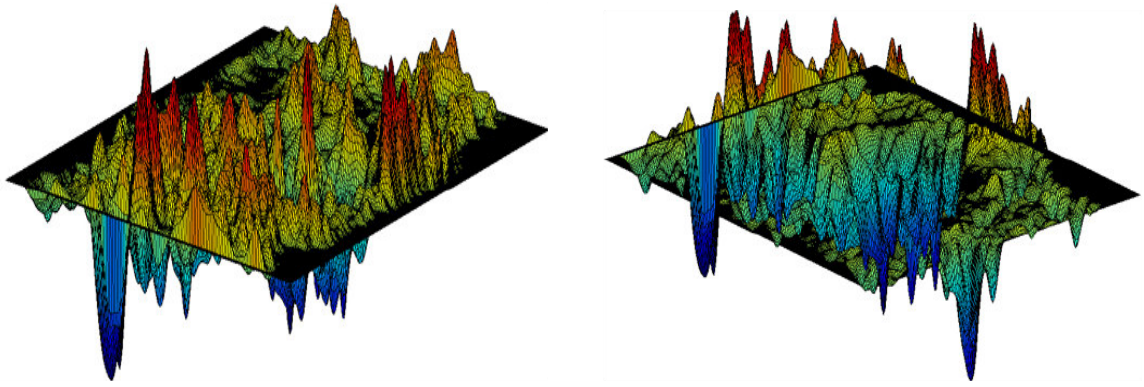
V. SCDE EXPERIMENTAL RESULTS

In Section *A* SCDE will be evaluated for a hot spot discovery problem involving a continuous variable, and in Section *B* SCDE will be evaluated for a benchmark of categorical hot spot discovery problems.

A. Experiments Involving Continuous Density Estimation

It is widely believed [2] that a significant quantity of water resides in the Martian subsurface in form of ground ice. In this section, we will evaluate SCDE for a binary co-location mining problem that centers on identifying places on Ice in which we observe a strong correlation or anti-correlation between shallow and deep subsurface ice. The approximate 35,000 objects in the original dataset are characterized by longitude, latitude, shallow ice density and deep ice density, and the original dataset was transformed into a dataset Ice'=(longitude, latitude, *z*), where *z* is a continuous number which is the product of the z-scores of shallow ice and deep ice density. SCDE was then used to find hot spots and cool spots in this dataset. Knowing regions of Mars where deep and shallow ice abundances coincide provides insight into the history of water on Mars.

Figure 5-1 visualizes the supervised density function for dataset Ice' that was constructed using formula 3-2 with σ=0.75. In addition to visualizing the dataset we can identify hot spots and cool spots in the dataset algorithmically which will be the subject of the remainder of this sub-section.



a) b)

Figure 5-1 Density Map of Ice Dataset

The results of using SCDE for the dataset have been compared with two other algorithms: SPAM and SCMRG. SPAM (Supervised PAM) is a variation of PAM [15]. SPAM uses the fitness function q(x)—and not the mean square error— to determine the best cluster representatives as PAM does. SPAM starts its search with a randomly

created set of representatives, and then greedily replaces representatives with non-representatives as long as q(x) improves.

Clustering algorithms were evaluated by measuring the absolute average z-value of clusters—obviously, we prefer clustering solutions which contain clusters whose average z values are highly positive or negative. However, the measure based on absolute average z value does not take the number of clusters used into consideration. For example, one clustering solution might have five clusters whereas the other clustering solution might have two clusters all with the same average z-value; obviously, we would prefer the two cluster solution. To measure this aspect, we use the following fitness function as a second evaluation measure when comparing clustering results

$$q(X) = \sum_{c \in X} Reward(c) = \sum_{c \in X} i(c) \times (|c|)^{\beta} \qquad (5\text{-}1)$$

The interestingness function *i(c)* is defined as:

$$i(c) = \frac{|(\sum_{o \in c} z(o))|}{|c|} \qquad (5\text{-}2)$$

where $\beta > 1$, o is an object in cluster c and $z(o)$ is the value of attribute z for object o; $|c|$ is the number of objects in cluster c. The function i assesses interestingness based on the absolute mean value for attribute z in cluster c. Parameter β determines how much penalty we associate with the number of clusters obtained. If we are interested in finding very local hot spots, 1.01 is a good choice for β; if we are interested in finding more regional hot spots, 1.2 turned out to be a good value for β.

Figure 5-2 shows fractions of clusters having absolute average z values in a given range for each clustering algorithm. Observing formula 5-1, the larger fractions of clusters having high absolute average z values the larger q(x) will be. From this perspective the SCDE results are superior, because they contain the largest fraction of clusters characterized by the high values of |average(z)|.

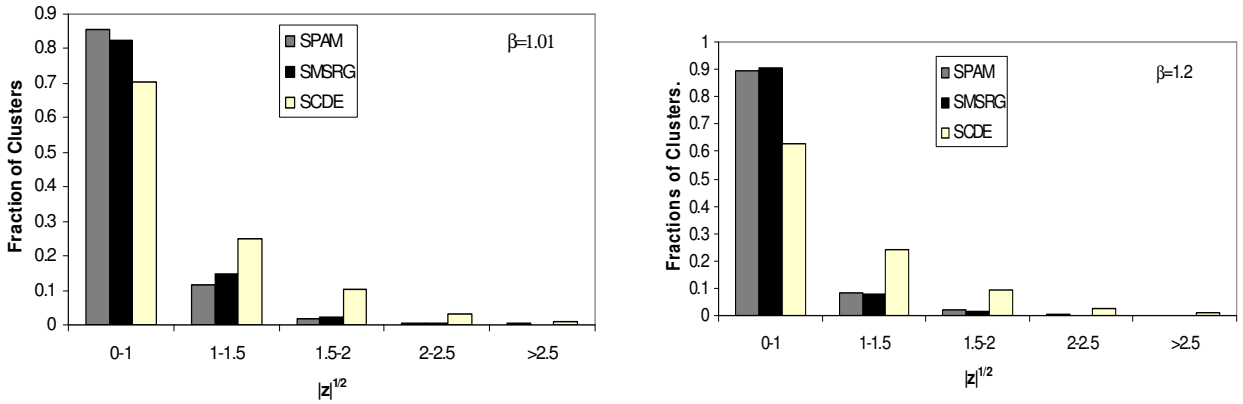


Figure 5-2 Statistics Comparison of SPAM, SCMRG and SCDE on Ice' Dataset

Table 5-1 compares solutions obtained from SCMRG and SPAM with the one produced by SCDE. As we can see SCDE get slightly higher value of $q(x)$ than the other two clustering algorithms, but more importantly its maximum rewards are much higher than those of the other clustering algorithms, and this is accomplished by using fewer clusters than the other two algorithms.

Algorithm	$q(x)$	Clusters Number	Highest Reward
$\beta = 1.01$ / $\beta = 1.2$			
SPAM	13502 / 24265	2000 / 807	204 / 705
SCDE	14709 / 39935	1253 / 653	671 / 9488
SCMRG	14129 / 34614	1597 / 644	743 / 6380

Table 5-1 Comparison of SPAM, SCMRG and SCDE on Ice Dataset

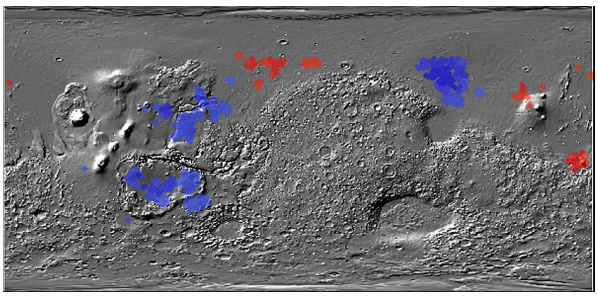


Figure 5-3 SCDE results on Ice dataset; Regional shallow ice/ deep ice co-location patterns in red; Regional Shallow Ice/ Deep Ice anti co-location patterns in blue

Figure 5-3 depicts the hot spots (in red) and cool spots (in blue) that have been identified by running SCDE whose average z value is more than 2.25 standard deviations above or below the mean value of z ($|z|>2.25$). These are places that need to be further studied by domain experts to find what particular set of geological circumstances lead to their existence.

	β=1.01	β=1.2
SPAM	73957.5	42817.5
SCMRG	5.25	1.2
SCDE	618.75	489.375

Table 5-2 Running Time (seconds) Comparison of SPAM, SCMRG and SCDE on Ice Dataset

As shown in Table 5-2, the running time of SCDE is much faster than SPAM. It is slower than SCMARG, but its running time is still acceptable and efficient.

B. Categorical Dataset Experiment Results

The categorical datasets include 3 real-world datasets and 1 artificial dataset, as summarized in Table 5-3. Each object in the data sets has 3 attributes including: longitude, latitude and the class label, which takes values +1 or -1.

	Dataset Name	Number of objects	Class of Interest
1	B-Complex9	3,031	yellow
2	Volcano	1,533	violent
3	Arsenic	2,385	safe
4	Earthquake	3,161	deep

Table 5-3 Categorical Dataset

B-Complex9 is a two dimensional synthetic spatial dataset whose examples are distributed having different, well-separated shapes. The Arsenic dataset has been created from the Texas Ground Water Database (GWDB) [19]. A well is labeled as dangerous if its arsenic concentration level is above 10µg/l, the standard for drinking water by the

Environment Protection Agency [20]. Earthquake and Volcano are spatial datasets obtained from Geosciences Department, University of Houston. The Volcano dataset uses severity of eruptions as the class label, whereas the Earthquake dataset uses depth of the earthquake as a class label.

Figure 5-4 visualizes the Volcano dataset and the SCDE's clustering result for that dataset for σ=10. Figure 5-4.d shows the clusters that were discovered by SCDE, which provides visual evidence that the SCDE algorithms can discover arbitrary hot spots and cool spots in the dataset—the detected hot and cool spots are consistent with the density map 5-4.b

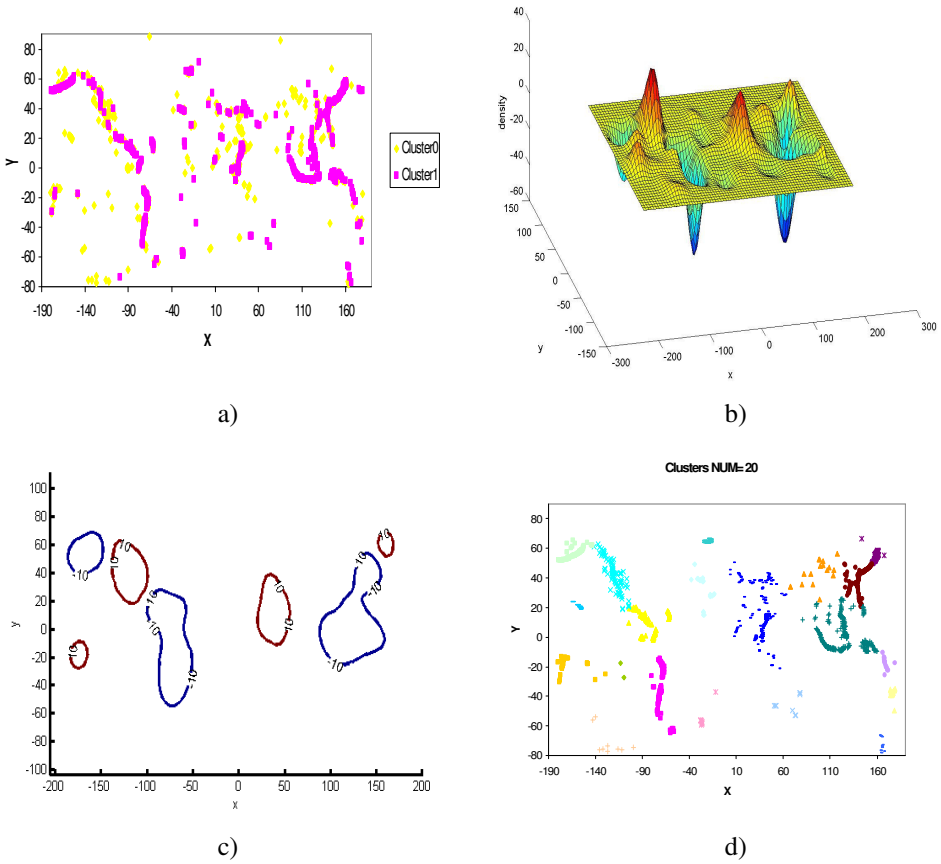


Figure 5-4 Visualization of the Volcano dataset and of clusters computed using SCDE. a) The distribution of the original Volcano dataset; b) Thematic map for ψ^{O}; c) The contour map according to ψ^{O} for density values 10 and -10; d) Clustering results of SCDE

The clustering results of SCDE algorithm were also compared with other supervised clustering algorithms SCEC [14], SCAH and SCMRG [18].[2] All three clustering algorithms seek to find clusters that maximize a reward-based fitness function *q(x)*. SCEC is a K-means-style, representative-based clustering algorithm that uses evolutionary computing to seek for the best set of representatives. Supervised Clustering using Agglomerative Hierarchical Techniques (SCAH) greedily merges clusters as long as *q(x)* improves. Supervised Clustering using Multi-Resolution Grids (SCMRG) is a hierarchical grid-based method that utilizes a divisive grid-based clustering method, which employs a divisive, top-down search: a higher level is partitioned further into a number of smaller cells at the lower level, until the sum of the rewards of the lower level cells is greater than the obtained reward for the cell at the higher level.

All clustering algorithms were applied to the four datasets, and cluster quality was assessed using cluster purity, number of clusters achieved, and the values of the following reward-based fitness function *q(x)*: The quality *q(x)* of a clustering *X* is computed as the sum of the rewards obtained for each cluster $c \in X$. Cluster rewards are computed as the product of interestingness of a cluster and the size of a cluster. More specifically, the evaluation function *q(x)* is defined as follows:

[2] Due to space limitations in this paper we will only center on discussing the most important findings of this evaluation.

$$q(X) = \sum_{c \in X} Reward(c) = \sum_{c \in X} \frac{i(c) \cdot (|c|)^{\beta}}{n^{\beta}} \qquad (5\text{-}3)$$

where

$$i(c) = \begin{cases} \left(\frac{(purity_Y(c) - hst)}{(1 - hst)} \right)^{\eta} & if \quad purity_Y > hst \\ \left(\frac{(cst - purity_Y(c)}{cst} \right)^{\eta} & if \quad purity_Y < cst \\ 0 & otherwise \end{cases} \qquad (5\text{-}4)$$

where *hst* and *cst* are hot spot and cool spot purity thresholds and $purity_Y(c)$ is the percentage of examples in cluster *c* that belong to the class of interest *Y*. In the experimental evaluation, *hst* was set to 1.5 times the prior probability of the class of interest, and *cst* was set to 0.5 times the prior probability of the class of interest.

Algorithms	**SCEC**	**SCAH**	**SCMRG**	**SCDE**
B-Complex9				
Purity	0.830	1	0.863	1
q(x)	0.032	0.008	0.002	0.045
# Clusters	4	17	22	9
Time (sec)	8063	5097	1	440
Volcano				
Purity	0.607	1	0.885	0.723
q(x)	0.0004	1E-5	1E-4	0.001
# Clusters	7	639	221	41
Time	5452	448	446	80
Arsenic				
Purity	0.774	0.857	0.794	0.831
q(x)	0.021	0.0006	0.0009	0.019
# Clusters	6	390	91	15
Time (sec)	14721	3462	1	3348
Earthquake				
Purity	0.846	0.853	0.814	0.843
q(x)	0.016	0.004	0.006	0.019
# Clusters	4	161	93	22
Time (sec)	8178	448	356	264

Table 5-4 SCDE and other algorithms comparisons based on $\beta = 3$, $\eta = 1$

The parameters β and η in formula 5-3 and 5-4 control the preference of how the rewards are given to the clusters. Large value of β causing the evaluation function 5-3 gives more rewards to clusters with more objects whereas the parameter η stresses on the interestingness of the cluster. We have designed 2 sets of experiments that the only different between the two are the parameters β and η. In the first experiment (table-5-4) we assign $\beta = 3$ and $\eta = 1$, so that the size of clusters is considered more important than the interestingness of the cluster. The second experiment (table 5-5) we set $\beta = 1.01$, $\eta = 6$ so that the interestingness of the clusters are more emphasized than the size of the clusters.

Algorithms	**SCEC**	**SCAH**	**SCMRG**	**SCDE**
B-Complex9				
Purity	0.998	1	1	1
q(x)	0.937	0.974	0.957	0.981
# Clusters	98	17	132	9
Time (sec)	23584	5275	1	440
Volcano				
Purity	0.780	1	0.979	0.723
q(x)	0.322	0.940	0.822	0.053
# Clusters	402	639	311	41
Time	6436	445	365	80
Arsenic				
Purity	0.834	1	0.9221	0.831
q(x)	0.391	0.942	0.668	0.019
# Clusters	396	746	184	15
Time (sec)	14062	2142	2	3348
Earthquake				
Purity	0.895	1	0.938	0.84
q(x)	0.575	0.952	0.795	0.158
# Clusters	404	479	380	22
Time (sec)	15440	6769	346	264

Table 5-5 SCDE and other algorithms comparisons based on $\beta = 1.01$, $\eta = 6$

By comparing the value of q(x), we can see that SCDE obtained good clustering results and outperforms the other algorithms, when β is 3 and η is 1. This is because the SCDE generates less number of clusters, larger cluster size, while the purity of its clusters is still comparable with clusters produced by other algorithms. This result is favorable by the parameters setting of the first experiment. On the other hand, for β is 1.01 and η is 6, SCDE obtained the best result for the Complex9 dataset, but did not perform well for the other three datasets. For Complex9, SCDE did a perfect job on identifying all 9 nature clusters with purity equal to 1. For the other three datasets, other three algorithms took advantage of increasing the number of clusters to improve their purity. Since the parameter setting of the second experiment stresses on the interestingness of clusters rather than the size of clusters, the value of q(x) for SCDE are lower than other algorithms, except the Complex9 dataset. In summary, SCDE did a good job in identifying larger size hot spots. We attribute the good performance to SCDE's capability to find arbitrary shape clusters

VI. CONCLUSIONS

This paper proposes a supervised density estimation approach that extends the traditional density estimation techniques by considering a variable of interest that is associated with a spatial object. Density in supervised density estimation is measured as the product of an influence function with the variable of interest. We claim that such a generalization can be used to create thematic maps, and that novel data mining algorithms can be developed with the help of supervised density functions.

One such algorithm, a supervised density-based clustering algorithm SCDE was introduced and discussed in detail in the paper. It uses hill-climbing to calculate the maximum/minimum (density attractors) of a supervised density function and clusters are formed by associating data objects with density attractors during the hill climbing procedure.

We conducted sets of experiments in which SCDE was evaluated for hot spot discovery and co-location discovery in spatial datasets. Compared with other algorithms, SCDE performed quite well for a set of four categorical hot spot discovery problems and was able to compute the hot spots quickly. Moreover, SCDE did particularly well in identifying regions on Mars in which shallow and deep subsurface ice are co-located.

When using categorical density estimation techniques, decision boundaries between the two classes represent areas whose supervised density is equal to 0. In this paper, we successfully used Matlab's contour function to visualize decision boundaries of the Complex9 dataset (also see Figure 3-2). However, in general, computing decision boundaries is a difficult task for most datasets. In our future work, we will focus on following topics:

- Develop novel classification algorithms that rely on decision boundaries that have been extracted from supervised density functions,
- Compare different hill climbing procedures that have been proposed for SCDE in this paper
- Develop automatic preprocessing procedures to determine the influence function parameter σ
- Develop a post-processing procedure to compute hotspots that only contains object whose density is above (below) a given threshold
- Compare SCDE using a benchmark of hotspot discovery problems with CrimeStat [22] and SaTScan[5][3]

REFERENCES

[1] Silverman, B. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.

[2] Clifford, S. A model for the hydrological and climatic behavior of water on mars. *Journal of Geophysical Research*, Vol. 98, No. E6, 1993, 10973-11016.

[3] Koperski, K., Adhikary, J., and Han, J. Spatial data mining: Progress and challenges survey paper. In *Proceedings of ACM SIGMOD Workshop on Research Issues on Data Mining and Knowledge Discovery*, Montreal, Canada, 1996.

[4] Ester, M., Kriegel, H., Sander, J., and Xu, X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, Portland, Oregon, August 1996, 226-231.

[5] Kulldorff M. A spatial scan statistic. *Communications in Statistics: Theory and Methods*, 26:1481-1496, 1997

[6] Hinneburg, A. and Keim, D. A. An Efficient Approach to Clustering in Large Multimedia Databases with Noise, *Proceedings of The Fourth International Conference on Knowledge Discovery and Data Mining,* New York City, August 1998, 58-65.

[7] Sander, J., Ester, M., Kriegel, H.P., and Xu, X., Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications. *Data Mining and Knowledge Discovery*, Kluwer Academic Publishers, Vol. 2, No. 2, 1998, 169-194.

[8] Murray, A. T. and Estivill-Castro, V. Cluster discovery techniques for exploratory spatial data analysis, *International Journal of Geographical Information Science*, Vol. 12, No. 5, 1998, 431-443.

[9] Openshaw, S. *Geocomputation: A primer, Chapter building automated geographical analysis and explanation machines*, Wiley, 1998, 95-115.

[3] SaTScan is developed by Martin Kulldorff of the National Cancer Institute and Farzad Mostashari of the New York Department of Health and Mental Hygiene.

[10] Williams, G.J. Evolutionary hot spots data mining – an architecture for exploring for interesting discoveries. In *Proceedings of the 3rd Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, London, UK, 1999, 184-193.

[11] Kolatch, E. Clustering Algorithms for Spatial Databases: A Survey. Department of Computer Science, University of Maryland, College Park CMSC 725, 2001.

[12] Miller, H. J. and Han, J. *Geographic data Mining and Knowledge Discovery*. Taylor & Francis, London, 2001.

[13] Tay, S.C. , Hsu, W., and Lim, K. H. Spatial data mining: Clustering of hot spots and pattern recognition. In *International Geoscience & Remote Sensing Symposium*, Toulouse France, July 2003.

[14] Eick C., Zeidat N. and Zhao Z., Supervised Clustering - Algorithms and Benefits, In *Proceedings of International Conference on Tools with AI*, Boca Raton, Florida, November 2004, 774-776.

[15] Kaufman, L. and Rousseeuw, P. J. *Finding groups in data: An introduction to cluster analysis*, John Wiley and Sons,New Jersey, USA, 2005.

[16] Kriegel, H.P. and Pfeifle, M. Density-Based Clustering of Uncertain Data. In *Proceedings of 11th ACM SIGKDD International Conference on Knowledge Discovery in Data* Mining, Chicago, Illinois, August 2005, 672-677.

[17] Brimicombe, A. J. Cluster detection in point event data having tendency towards spatially repetitive events. In *Proceedings of 8th International Conference on GeoComputation*, Michigan, August 2005.

[18] Eick C., Vaezian B., Jiang, D. and Wang, J. Discovery of Interesting Regions in Spatial Datasets Using Supervised Clustering, In *Proceedings of The 10th European Conference on Principles and Practice of Knowledge Discovery in Databases*, Berlin, Germany, September 2006.

[19] Texas Water Development Board, http://www.twdb.state.tx.us/home/index.asp, 2006.

[20] U.S. Environmental Protection Agency, http://www.epa.gov/, 2006.

[21] Hinneburg, A. and Gabriel, H.H. Denclue 2.0: Fast Clustering Based on Kernel Density Estimation, In *Proceedings of The 7th International Symposium on Intelligent Data Analysis*, Ljubljana, Slovenia, September, 2007

[22] Ned Levine (2007). CrimeStat: A Spatial Statistics Program for the Analysis of Crime Incident Locations (v 3.1). Ned Levine & Associates, Houston, TX, and the National Institute of Justice, Washington, DC. March