

A Unifying Framework for Clustering with Plug-In Fitness Functions and Region Discovery

Last updated: Dec. 24, 2007, noon

CHRISTOPH F. FICK

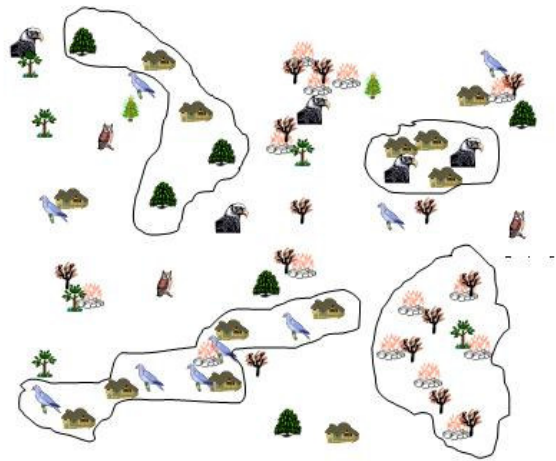
Department of Computer Science, University of Houston,
Houston, TX 77204-3010

1 Motivation

The goal of spatial data mining [SPH05] is to automate the extraction of interesting and useful patterns that are not explicitly represented in spatial datasets. Of particular interests to scientists are techniques capable of finding scientifically meaningful regions in spatial datasets as they have many immediate applications in medicine, geosciences, and environmental sciences, e.g., identification of earthquake hotspots, association of particular cancers with environmental pollution, and detection of crime zones with unusual activities. The ultimate goal of region discovery is to provide search-engine-style capabilities that enable scientists to find interesting places in geo-referenced data automatically and efficiently.

Many applications of region discovery in science exist. First, scientists are frequently interested in identifying disjoint, contiguous regions that are unusual with respect to the distribution of a given class; for example, a region that contains an unexpected low or high number of instances of a particular class. Examples of applications that belong to this task include identifying crime hotspots, cancer clusters, and wild fires from satellite photos. A second region discovery task is finding regions that satisfy particular characteristics of a continuous variable. For example, someone might be interested in finding regions in the state of Wyoming (based on census 2000 data) with a high variance of income—poor people and rich people are living in close proximity of each other. The third application of region discovery is co-location mining in which we are interested in finding regions that have an elevated density of instances belonging to two or more classes. For example, a region discovery algorithm might find a region where there is high density of polluted wells and farms. This discovery might lead to further field study that explores the relationship between farm use and well pollution in a particular region. Figure 1 gives a generic example of a co-location mining problem. Global co-location mining techniques might infer that fires and trees and birds and houses are co-located. Regional co-location mining proposed here, on the other hand, tries to find

regions in which the density of two or more classes is elevated. For example, a regional co-location mining algorithm would identify a region on the upper right in which eagles and houses are co-located. Fourth region discovery algorithms have been found useful [DEWY06] for mining regional association rules. Regional association rules are only valid in a sub-space of a spatial dataset and not for the complete dataset. Finally, region discovery algorithms are also useful for data reduction and sampling. For example, let us assume a European company wants to test the suitability of a particular product for the US market. In this case, the company would be very interested in finding small sub-regions in US that have the same or quite similar characteristics as US as a whole. The company would then try to market and sell their product in a few of those sub-regions, and if this works out well, would extend its operations to the complete US market.




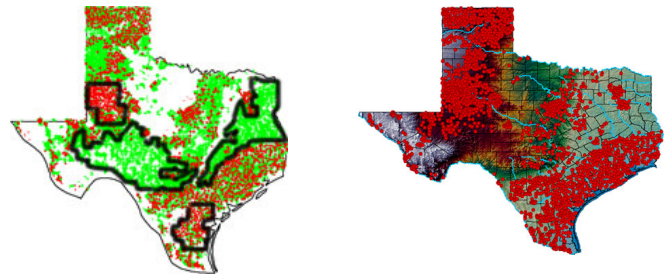
Global Answer:  and 

Figure 1: Finding regions with interesting co-location characteristics.



(a)

(b)

Figure 2: Finding regions with a very high and low density of “good” (in green) and “bad” (in red) wells.

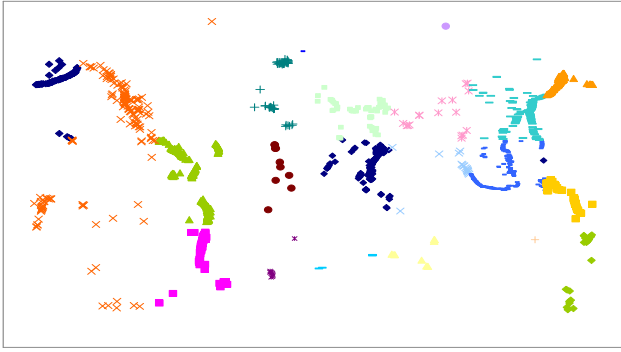


Figure 3: Finding groups of violent volcanoes and groups of non-violent volcanoes.

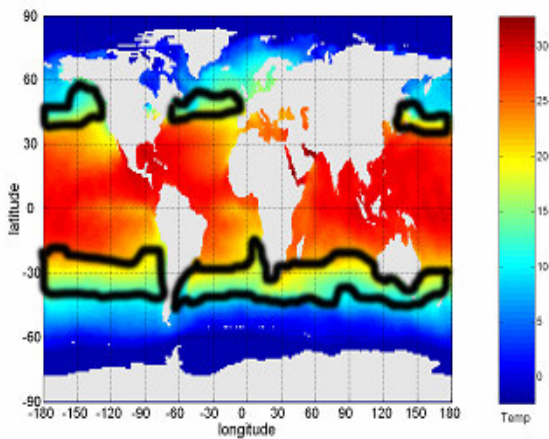


Figure 4: Finding the regions where hot water meets cold water.

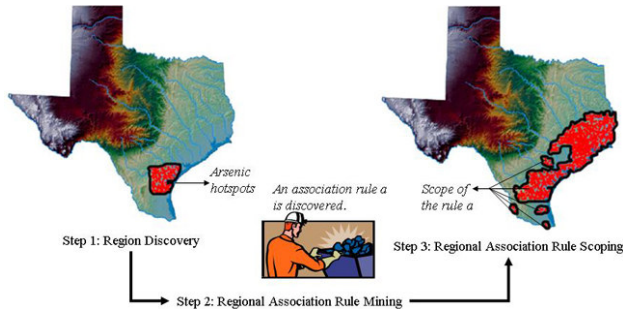


Figure 5: Regional association rule mining and scoping.

Figures 1-5 depict results of using our region discovery algorithms for some examples applications. Figure 2 depicts water wells in Texas. Wells that have high levels of arsenic are in red, and wells that have low levels are in green. As a result of the application of a region discovery algorithm [DEWY06] 4 majors regions in Texas were identified, two of which have a high density of good wells and two of which have a high density of bad wells. Figure 1 gives an example of co-location region discovery result in which regions are identified in which the density of two or more classes is elevated. Figure 3 [EVDW06] depicts the results of identifying groups/regions of violent and non-

violent volcanoes. Figure 4 shows the regions where hot water meets cold water (characterized by high variance in water temperature). Figure 5 illustrates how a regional association rule is discovered from an arsenic hotspot, and then the scope of the discovered association rule is computed that identifies the regions in which a particular rule is valid.

Developing a region discovery system faces the following challenges. First, the system must be able to find regions of arbitrary shape at different levels of resolution. Second, the system needs to provide suitable, plug-in measures of interestingness to instruct discovery algorithms what they should seek for. Third, the identified regions should be properly ranked by relevance. Fourth, due to the high complexity of region discovery, it is desirable for the framework to provide pruning and other sophisticated search strategies as the goal is to seek for interesting, highly ranked regions in an efficient manner.

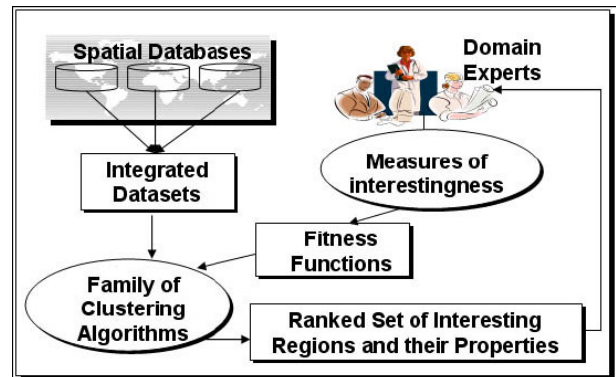


Figure 6: Architecture of the Proposed Region Discovery Framework

An initial region discovery framework (see Fig. 6) that is geared towards finding scientifically interesting places in spatial datasets has been introduced in our past work [EVJW06, DKPJSE08]. The framework views region discovery as a clustering problem in which an externally given fitness function has to be maximized. The framework allows for plug-in fitness functions to support a variety of region discovery applications correspondent to different spatial data mining tasks. The fitness function combines contributions of interestingness from constituent clusters and can be customized by domain experts. Moreover, nine region discovery algorithms (four representative-based, three agglomerative, one divisive, and one density-based region discovery algorithm) have already been designed and implemented by our past work [CJCCGE07, DEWY06, EPDSN07, EVJW06, EZZ04].

It is worth mentioning that the use of fitness functions is not very common in traditional clustering; the only exception to this point is the CHAMELEON [KHK99] clustering algorithm. However, fitness functions play a

more important role in semi-supervised and supervised clustering [EZZ04] and in adaptive clustering [BECV05].

The goal of this paper is to define our region discovery framework with much more rigor and in more depth and to extend it to foster novel applications. Moreover, region discovery algorithms will be categorized, and a theoretical framework to analyze and compare region discovery results will be introduced. In general, the paper provides “theoretical glue” that intended to lead to a clearer understanding concerning the objectives of regions discovery, parameters, inputs and outputs of region discovery algorithms, classes of algorithms for region discovery, frameworks for regions discovery, a foundation for clustering with plug-in fitness functions, and their relationships. Another critical problem in region discovery is finding suitable measures of interestingness for different region discovery tasks; this topic will not be addressed in this paper. Although the paper centers on giving a foundation for region discovery and thoroughly analyzes the contributions and relationships of our and other past work in the area, the paper makes a few contributions of its own:

- It introduces *intensional clustering algorithms* that generate models from a clustering, and explains how they differ from the traditional extensional clustering algorithms; moreover, procedures to create *region discovery models* are introduced. We also describe how region discovery models can be used to analyze relationships between different clusterings.
- It introduces highly generic representative-based, divisive, and agglomerative clustering frameworks; popular clustering algorithms are presents as a special case of the proposed generic frameworks.
- It presents a foundation for clustering with plug-in fitness functions.
- It outlines how density-based clustering algorithms can be adapted to become region discovery algorithms. Moreover, a novel density-based clustering approach called *contour clustering* is introduced.

2 A Framework for Region Discovery in Spatial and Spatio-Temporal Datasets

As mentioned in the previous section, we are interested in the development of frameworks and algorithms that find interesting regions in spatial and spatio-temporal datasets. Our work assumes that region discovery algorithms we develop operate on datasets containing objects o_1, \dots, o_n : $O = \{o_1, \dots, o_n\} \in F$ where F is relational database schema and the objects belonging to O are tuples that are characterized by attributes $S \cup N$, where:

$S = \{s_1, \dots, s_q\}$ is a set of spatial and temporal attributes.

$N = \{n_1, \dots, n_p\}$ is a set of other, non-geo-referenced attributes.

$\text{Dom}(S)$ and $\text{Dom}(N)$ describe the possible values the attributes in S and N can take; that is, each object $o \in O$ is characterized by a single tuple that takes values in $\text{Dom}(S) \times \text{Dom}(N)$. Datasets that have the structure we just introduced, are called *geo-referenced datasets* in the following, and O is assumed to be a geo-referenced dataset throughout this paper.

In general, clustering algorithms can be subdivided into *intensional clustering* and *extensional* algorithms: extensional clustering algorithms just create clusters for the input data set O , partitioning O into subsets, but do nothing else. Intensional clustering algorithms, on the other hand, create a clustering model based on O and other inputs. Most popular clustering algorithms have been introduced as extensional clustering algorithms, but—as we will see in the remainder of the paper—it is not too difficult to generalize most extensional clustering algorithms so that they become intensional clustering algorithms: in sections 4 and 7 intensional versions of the popular clustering algorithms K-means and DENCLUE will be proposed.

Extensional clustering algorithms create clusters X with respect to O that are sets of disjoint subsets of O :

$$X = \{c_1, \dots, c_k\} \text{ with } c_i \subseteq O (i=1, \dots, k) \text{ and } c_i \cap c_j = \emptyset (i \neq j)$$

Intensional clustering algorithms create a set of disjoint regions Y in F :

$$Y = \{r_1, \dots, r_k\} \text{ with } r_i \subseteq F (i=1, \dots, k) \text{ and } r_i \cap r_j = \emptyset (i \neq j)$$

In the case of region discovery, cluster models have a peculiar structure in that regions belong to the subspace $\text{Dom}(S)$ and not to F itself: a *region discovery model* is a function¹ $\rho: \text{Dom}(S) \rightarrow \{1, \dots, k\} \cup \{\perp\}$ that assigns a region $\rho(p)$ to a point p in $\text{Dom}(S)$ assuming that there are k regions in the spatial dataset—the number of regions k is chosen by the region discovery algorithm that creates the model. Models support the notion of outliers; that is, a point p' can be an outlier that does not belong to any region: in this case: $\rho(p') = \perp$.

Intensional region discovery algorithms obtain a clustering Y in $\text{Dom}(S)$ that is defined as a set of disjoint regions in $\text{Dom}(S)$ ²:

$$Y = \{r_1, \dots, r_k\} \text{ with } r_i \subseteq F[S] (i=1, \dots, k) \text{ and } r_i \cap r_j = \emptyset (i \neq j)$$

Moreover, regions r belonging to Y are described as functions over tuples in $\text{Dom}(S)$ — $\phi_r: \text{Dom}(S) \rightarrow \{t, f\}$ indicating if a point $p \in \text{Dom}(S)$ belongs to r : $\phi_r(p) = t$. ϕ_r is called *the intension of r* . ϕ_r can easily be constructed from a the model ρ that has been generated from a clustering Y .

¹ \perp denotes “undefined”.

² $F[S]$ denotes the projection of F on the attributes in S .

Moreover, *the extension of a region* $r \in \mathcal{E}_r$ is defined as follows:

$$\mathcal{E}_r = \{o \in O \mid \phi_r(o[S]) = t\}$$

In the above definition $o[S]$ denotes the projection of o on its spatial and temporal attributes.

In the subsequently sections 4-7 of this paper procedures will be described that generate models from extensional clusterings X that have been obtained for a dataset O .

Our approach requires that discovered regions are contiguous. To cope with this constraint, we assume that we have neighbor relationships *no* between the objects in O and cluster neighbor relationship *nc* between regions in X defined with respect to O : if *no*(o, o') holds objects o and o' are neighboring; if *nc*(r, r') holds regions r and r' are neighboring.

$$no \subseteq O \times O$$

$$nc \subseteq 2^O \times 2^O$$

Moreover, neighbor relationships are solely determined by the attributes in S ; that is, the temporal and spatial attributes in S are used to determine which objects and clusters are neighboring. A region r is contiguous if for each pair of points u and v in r there is a path between u and v that solely traverses r and no other regions. More formally, contiguity is defined as a predicate over subsets c of O :

$$\text{contiguous}(c) \Leftrightarrow \forall w \in c \forall v \in c \exists m \geq 2 \exists x_1, \dots, x_m \in c: w = x_1 \wedge v = x_m \wedge no(x_i, x_{i+1}) \quad (i=1, \dots, m).$$

$$\text{contiguous}(X) \Leftrightarrow \forall c \in X: \text{contiguous}(c)$$

Our approach employs arbitrary plug-in, reward-based fitness functions to evaluate the quality of a given set of regions. The goal of region discovery is to find a set of regions X that maximize an externally given fitness function $q(X)$; moreover, q is assumed to have the following structure:

$$q(X) = \sum_{c \in X} \text{reward}(c) = \sum_{c \in X} i(c) * |c|^\beta$$

where $i(c)$ is the interestingness of a region c —a quantity designed by a domain expert to reflect a degree to which regions are “newsworthy”. The number of objects in O belonging to a region is denoted by $|c|$, and the quantity $i(c) * |c|^\beta$ can be considered as a “reward” given to a region c ; we seek X such that the sum of rewards over all of its constituent regions is maximized. The amount of premium put on the size of the region is controlled by the value of parameter β ($\beta > 1$). A region reward is proportional to its interestingness, but bigger regions receive a higher reward than smaller regions having the same value of interestingness to reflect a preference given to larger regions. Furthermore, it is assumed that the fitness function q is additive; the reward associated with X is the sum of the

rewards of its constituent regions. One reason to reward size non-linearly is to encourage region discovery algorithms to merge neighboring regions as long as their interestingness does not increase through the merging: Let $r = r_1 \cup r_2$, $|r_1| + |r_2| = |r|$ and $i(r) = i(r_1) = i(r_2)$; assuming $\beta > 1$, $|r_1|, |r_2| \geq 1$ we yield: $q(r) / (q(r_1) + q(r_2)) = (|r_1| + |r_2|)^\beta / (|r_1|^\beta + |r_2|^\beta) > 1$ —that is, region discovery algorithms should prefer $\{r\}$ over $\{r_1, r_2\}$ in this case. Moreover, if region size would be rewarded linearly ($\beta = 1$) for many measures of interestingness, single object regions represent a maximum for q ; this is for example the case for supervised clustering whose objective is to in obtaining pure clusters that only contain objects belonging to the same class—a fitness function for supervised clustering will be introduced later in Section 3 of this paper.

Given a geo-referenced dataset O , there are many possible approaches to seek for interesting regions in O with respect to a plug in fitness function q . In general, the *objective of region discovery with plug-in fitness functions* is:

Given: O , q , and possibly other input parameters

Find: regions r_1, \dots, r_k that maximize $q(\{r_1, \dots, r_k\})$ subject to the following constraints:

- (1) $r_i \subseteq O$ ($i=1, \dots, k$)
- (2) $\text{contiguous}(r_i)$ ($i=1, \dots, k$)
- (3) $r_i \cap r_j = \emptyset$ ($i \neq j$)

It should be emphasized that the number of regions k is not an input parameter in the proposed framework; that is, region discovery algorithms are assumed to seek for the optimal number of regions k . In the upcoming sections of this paper, several algorithms for region discovery with plug-in fitness functions will be introduced.

3 An Example Application for Region Discovery

As explained earlier, our approach identifies regions as clusters and employs reward-based fitness functions to measure the interestingness of a given set of regions. As we described earlier, 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 depend on the interestingness $i(c)$ and are weighted by the number of objects that belong to a cluster c .

$$q(X) = \sum_{c \in X} (i(c) * |c|^\beta)$$

The remainder of this section illustrates the use of the proposed framework for a region discovery task that centers on discovering *hot spots* and *cold spots* in a categorical datasets. In the example, datasets are assumed

to take the form $P=(\langle \text{longitude} \rangle, \langle \text{latitude} \rangle, \langle \text{class-variable} \rangle)$ and describe class memberships of object residing at a particular geographic location: P describes people at geographic locations that are either belong to the class “Poor” or to a class “Not_Poor”. Our goal it is to identify regions in the dataset P where an unusually high number of poor people live and regions in which an unusually low number of poor people live.

A categorical measure of interestingness $i(c)=\tau_c(c)$ that relies on a class of interest C is introduced for this purpose, which rewards the regions whose distribution of class C significantly deviates from the prior probability of class C for the whole dataset. τ_c that assigns rewards to each cluster c that is depicted in Fig. 1. τ_c itself is computed based on $p(c,C)$, $\text{prior}(C)$, and based on the following parameters: $\gamma_1, \gamma_2, R_+, R_-$ with $\gamma_1 \leq 1 \leq \gamma_2; 1 \geq R_+, R_- \geq 0, \eta > 0$ (see Fig. 7).

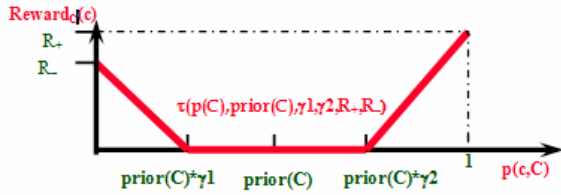


Figure 7: The interpolation function τ_c for $\eta=1$

The evaluation function $q(X)$ is defined as follows:

$$q(X) = \sum_{i=1}^{|X|} \frac{\tau_c(p(c_i, C), \text{prior}(C), \gamma_1, \gamma_2, R_+, R_-, \eta) * (|c_i|)^\beta}{n^\beta}$$

with

$$\tau_c(p(c, C), \text{prior}(C), \gamma_1, \gamma_2, R_+, R_-, \eta) = \begin{cases} \left(\frac{((\text{prior}(C) * \gamma_1) - p(c, C)) * R_-}{(\text{prior}(C) * \gamma_1)} \right)^\eta & \text{if } p(c, C) < \text{prior}(C) * \gamma_1 \\ \left(\frac{(p(c, C) - (\text{prior}(C) * \gamma_2)) * R_+}{(1 - (\text{prior}(C) * \gamma_2))} \right)^\eta & \text{if } p(c, C) > \text{prior}(C) * \gamma_2 \\ 0 & \text{otherwise} \end{cases}$$

In the above formula $\text{prior}(C)$ denotes the probability of objects in dataset belonging to the class of interest C ; $p(C,c)$ denotes the percentage of examples in cluster c that belong to class C . $\gamma_1 * \text{prior}(C)$ is the class density below which clusters receive a reward; $\gamma_2 * \text{Prior}(C)$ is the class density above which clusters receives a reward. R_+ is the reward given to a cluster where 100% of the examples

belong to class C ; R_- is the reward given to a cluster where none of its examples belong to class C . The parameter η determines how quickly the reward function grows to maximum reward (either R_+ or R_-). If η is set to 1 it grows linearly, if it is set to 2 a quadratic function would be used that grows significantly slower initially. In general, if we are interested in giving higher rewards to purer clusters, it is desirable to choose larger values for η : e.g. $\eta=8$.

Let us assume a clustering X has to be evaluated with respect to a class of interest “Poor” with $\text{prior}(\text{Poor}) = 0.2$ for a dataset that contains 1000 examples. Suppose that the generated clustering X subdivides the dataset into five clusters $c1, c2, c3, c4, c5$ with the following characteristics. $|c1| = 50, |c2| = 200, |c3| = 200, |c4| = 350, |c5| = 200$; $p(c1, \text{Poor}) = 20/50, p(c2, \text{Poor}) = 40/200, p(c3, \text{Poor}) = 10/200, p(c4, \text{Poor}) = 30/350, p(c5, \text{Poor}) = 100/200$. Moreover, the following parameters used in the evaluation function are as follows: $\gamma_1 = 0.5, \gamma_2 = 1.5, R_+ = 1, R_- = 1, \beta = 1.1, \eta = 1$.

Due to the settings of $\gamma_1 = 0.5, \gamma_2 = 1.5$, clusters that contain between $0.5 \times 0.2 = 10\%$ and $1.5 \times 0.2 = 30\%$ instances of the class “Poor” do not receive any reward at all; therefore, no reward is given to cluster $c2$. The remaining clusters received rewards because the distribution of class “Poor” in the cluster is significantly higher or lower than its prior distribution. Consequently, the reward for the first cluster $c1$ is $1/7 \times (50)^{1.1}$ since $p(c1, \text{Poor}) = 40\%$ is greater than $\text{prior}(\text{Poor}) * \gamma_2$ which is 30% , $1/7$ is obtained by applying the interpolation function $\tau_c(c)$, thus we get $\tau_{\text{poor}}(c1) = ((0.4 - 0.3) / (1 - 0.3)) * 1 = 1/7$. Rewards of other clusters are computed similarly and the following overall reward for X is obtained:

$$q_{\text{poor}}(X) = \frac{\frac{1}{7} * 50^{1.1} + 0 + \frac{1}{2} * 200^{1.1} + \frac{1}{7} * 350^{1.1} + \frac{2}{7} * 200^{1.1}}{1000^{1.1}} = 0.129$$

4 Representative-Based Region Discovery Algorithms

Representative-based clustering algorithms, sometimes called prototype-based clustering algorithms in the literature, construct clusters by seeking for a set of representatives; clusters are then created by assigning objects in the dataset to their closest representative. Popular representative-based clustering algorithms include K-Medoids/PAM and K-means [KR00].

In a nutshell, *representative-based region discovery algorithms*³ compute the following function:

³ The four representative-based clustering algorithms that developed by our past research additionally require that representatives have to belong to the dataset O ; consequently, they compute mappings:

$\pi: O \times q \times d \times \{\text{other parameters}\} \rightarrow 2^{\text{Dom}(S)}$

π takes O , q , a distance function d over $\text{Dom}(S)$, and possibly other parameters as an input and seeks for an “optimal set”⁴ of representatives in $\text{Dom}(S)$, such that the clustering X obtained by assigning the objects in O to their closest representative in $\pi(O, q, d, \dots)$ maximizes $q(X)$. Moreover, it should be noted that clustering is done in the geo-referenced attribute space S , and not in F ; the attributes in N are only used by fitness function q when evaluating clusters.

Four representative-based clustering algorithms have already been introduced in our previous work: SPAM[ZEZ06], SRIDHCR [ZE04, EZZ06], SCEC[EZZ04, ZEZ06], and CLEVER[EPDSN07, PAR07]. All four algorithms start their search from a randomly generated set of representatives and seek for an optimal set of representatives maximizing $q(X)$. Moreover, all four algorithms limit representatives to objects belonging to the dataset O . SPAM requires that the number of regions k is given as an input, and tries to find the optimal set of k representatives by replacing single representatives with non-representatives as long as $q(X)$ improves. The other three algorithms, on the other hand, seek for the optimal number of representatives k . SRIDHCR greedily searches for a better set of representatives by greedily inserting and deleting representatives from the current set of representatives. CLEVER uses randomized hill climbing and uses neighborhood sizes⁵ larger than 1 when greedily searching for the best set of representatives. SCEC uses evolutionary computing to find the optimal set of representatives. It employs crossover and mutation operators that breed new solutions from a population that is a set of sets of representatives based on the principles of the survival of the fittest: solutions with higher values for $q(X)$ reproduce with a higher frequency.

The region discovery model ρ for the result obtained by running a representative-based clustering algorithm can be constructed as follows:

Let

$\pi': O \times q \times d \times \{\text{other parameters}\} \rightarrow 2^O$

⁴ In general, prototype-based clustering is NP-hard. Therefore, most representative-based clustering algorithm will only be able to find a suboptimal clustering X such that $q(X)$ is below the global maximum of q .

⁵ It modifies the current set of representatives by applying two or more operators to it—battling premature convergence. Operators include: replacing a representative by a non-representative, inserting a new representative, deleting a representative.

$\pi(O, q, d, \dots) = \{\text{rep}_1, \dots, \text{rep}_k\} \subseteq \text{Dom}(S)$ that is; the representative-based clustering algorithm returned $R = \{\text{rep}_1, \dots, \text{rep}_k\}$

Then the model ρ can be derived from R as follows:

$\forall p \in S \rho(p) = m \leftrightarrow d(p, \text{rep}_m) \leq d(p, \text{rep}_j)$ for $j=1, \dots, k$

that is, ρ assigns p to the region associated with the closest representative⁶ in R .

Because representative-based clustering algorithms assign objects to clusters using 1-nearest neighbor queries, the spatial extent of regions $r_i \subseteq \text{Dom}(S)$ can be constructed by computing Voronoi diagrams; this implies that the shape of regions obtained by representative-based clustering algorithms is limited to convex polygons in $\text{Dom}(S)$. Neighboring relationships *no* between objects in O and *nc* between clusters obtained by a representative-based clustering algorithm can be constructed by computing the Delaunay triangulation for $O[S]$ and R , respectively. Moreover, representative-based clustering algorithms do not support the concept of outliers; therefore, representative-based models have to assign a region to every point p in S .

5 Specialization for Agglomerative Clustering

The *agglomerative clustering problem* can be defined as follows:

Given: O , F , S , N , a fitness function q , and an initial clustering X with $\text{contiguous}(X)$

Find: $X' = \{c'_1, \dots, c'_h\}$ that maximizes $q(X')$ and all clusters in X' have been constructed using unions of neighboring clusters in X :

For all $c_i \in X'$: $c_i = c_{i1} \cup \dots \cup c_{ij} \wedge c_{i1}, \dots, c_{ij} \in X \wedge \text{nc}(c_{ik}, c_{ik+1})$ (for $k=1, j-1$)

In the following, we view results that are obtained by agglomerative methods as a meta clustering X' over an initial clustering X of O ; X' over X is defined as an exhaustive set of contiguous, disjoint subsets of X . More formally, the *objectives of agglomerative clustering* can be reformulated—relying on a meta clustering framework—as follows:

Given: O , F , S , N , a fitness function q , and an initial clustering X with $\text{contiguous}(X)$

Find: $X' = \{x_1, \dots, x_r\}$ with $x_i \subseteq X$ ($i=1, \dots, r$) maximizing $q(X')$, subject to the following constraints:

(1) $x_1 \cup \dots \cup x_r = X$

⁶ Our formulation ignores the problem of ties when finding the closest representative; in general, our representative-based clustering algorithms break ties randomly.

- (2) $x_i \cap x_j = \emptyset$ ($i \neq j$)
- (3) contiguous(x_i) (for $i=1, \dots, r$)
- (4) $\forall x \in X' \exists m \geq 1 \exists x'_1 \dots \exists x'_m \in X: x = x'_1 \cup \dots \cup x'_m$

We use the term meta clustering, because it is a clustering of clusters, and not of objects in a dataset—as it is the case in traditional clustering. It should be noted that agglomerative clusters are exhaustive subsets of an initial clustering X ; that is, we assume that outliers are not removed by the agglomerative clustering algorithm itself, but rather by the algorithm that constructs the input X for the agglomerative clustering algorithm. More specifically, an agglomerative clustering framework consists of two algorithms:

1. a preprocessing algorithm that constructs an initial clustering X
2. the agglomerative clustering algorithm itself that derives X' from X .

The preprocessing algorithm is frequently degenerated; for example, its input could consist of single object clusters, or X could be constructed based on a grid-structure; however, as we will point out later, frequently it is beneficiary for many applications to use a full fledged clustering algorithm for the preprocessing step, particularly if q is a complex fitness function to maximize.

An agglomerative clustering algorithm MOSAIC [CECCGE07] has been introduced by our previous work. MOSAIC takes the clustering X obtained by running a representative-based region discovery algorithm as its input, and merges neighboring regions greedily, maximizing $q(X)$. For efficiency reasons, MOSAIC uses Gabriel graphs[GS97]—which are subsets of Delaunay graphs—to compute nc which describes which clusters in X are neighboring; these neighboring cluster identify merge candidates for MOSAIC; nc is updated incrementally as clusters are merged. Fig. 8 gives the pseudo-code for MOSAIC.

```

1. Run a representative-based clustering algorithm to
   create a large number of clusters  $X = \{c_1, \dots, c_k\}$ 
2. Read the representatives of the obtained clusters in  $X$  and
   create a merge-candidate relation  $nc$  using Gabriel graphs.
3. WHILE there are merge-candidates  $(c_i, c_j)$  left
   BEGIN
       Merge the pair of merge-candidates  $(c_i, c_j)$ , that
       enhances  $q(X)$  the most, into a new cluster  $c' = c_i \cup c_j$ 
       Update merge-candidates:
        $\forall C$  Merge-Candidate( $c', c$ )  $\Leftrightarrow$ 
           Merge-Candidate( $c_i, c$ )  $\vee$  Merge-Candidate( $c_j, c$ )
   END
RETURN the best clustering  $X'$  with respect to  $q(X')$  found.

```

Figure 8: Pseudo code for MOSAIC

Moreover, models for the clusters obtained by an agglomerative regions discovery algorithm can be easily constructed from the models of the input clusters in X that have been merged to obtain the region in question. Let us assume r has been obtained as $r = r_1 \cup \dots \cup r_m$; in this case the model for r can be defined as: $\phi_r(p) = \phi_{r_1}(p) \vee \dots \vee \phi_{r_m}(p)$

In the case of MOSAIC, $\phi_r(p)$ is implemented by characterizing MOSAIC clusters by sets of representatives⁷; new points are then assigned to the cluster whose set of representatives contains the representative that is closest to p . Basically, MOSAIC constructs regions as union of Voronoi cells and the above construction takes advantage of this property.

6 Specialization for Divisive Clustering

The *divisive clustering problem* can be defined as follows:

Given: O, F, S, N , a fitness function q , and an initial clustering $X = \{x_1, \dots, x_n\}$ with contiguous(X).

Find: $X' = \{c'_1, \dots, c'_k\}$ that maximizes $q(X')$ and X' has been obtained from X .

Procedure: Initially, X' is set to X . Then X' is modified to increase $q(X')$ by recursively replacing an $x \in X'$ by $x = x'_1 \cup \dots \cup x'_p$ with $p \geq 1$ as long as $q(X)$ improves, and the following conditions are satisfied:

- (1) $x'_j \subseteq x$ ($j=1, \dots, p$)
- (2) $x'_j \cap x'_i = \emptyset$ (for $j \neq i$)
- (3) contiguous(x'_j) ($j=1, \dots, p$)
- (4) $\text{reward}(x) < \text{reward}(x'_1) + \dots + \text{reward}(x'_p)$

Region x is only replaced by regions at a lower level of resolution, if the sum of the rewards of the regions at lower level of resolution is higher than x 's reward. It should be emphasized that the splitting procedure employs a variable number of decompositions; e.g. one region might be split into just two or even one region, whereas another region might be split into 17 sub-regions. Moreover, the splitting procedure is not assumed to be exhaustive; that is, x can be split into y_1, y_2, y_3 with $y_1 \cup y_2 \cup y_3 \subseteq x$; in other words, the above specification allows divisive region discovery algorithms to discard outliers when seeking for interesting regions; basically the objects belonging to the residual region $x / (y_1 \cup y_2 \cup y_3)$ in the above examples are considered to be outliers. When and how larger regions are decomposed into small regions varies between different divisive clustering algorithms.

⁷ If r in X' has been constructed using $r = r_1 \cup \dots \cup r_m$ from X r would be characterized by the representatives of regions r_1, \dots, r_m .

SCMRG⁸ (Supervised Clustering using Multi-Resolution Grids) [EVJW06, DEWY06] is a divisive, grid-based region discovery algorithm that has been developed by our past work. SCMRG partitions the spatial/spatial-temporal space $\text{Dom}(S)$ of the dataset into grid cells. Each grid cell at a higher level is partitioned further into a number of smaller cells at the lower level, and this process continues if the sum of the rewards of the lower level cells is greater than the rewards at the higher level cell. The regions returned by SCMRG usually have different sizes, because they were obtained at different levels of resolution. Moreover, a cell is drilled down only if it is promising (if its fitness improves at a lower level of resolution). SCMRG uses a look-ahead splitting procedure that splits a cell into 4, 16, and 64 cells respectively and analyzes if there is an improvement in fitness in any of these three splits; if this is not the case and the original cell receives a reward, this cell is included in the region discovery result; however, regions who themselves as well as their successors at lower level of resolution do not receive any rewards, will be viewed as outliers, and discarded from the final clustering X' .

7 Specialization for Density-Based Clustering

Given a dataset containing objects o_1, \dots, o_n : $O = \{o_1, \dots, o_n\} \subseteq F$ where F is relational database schema and the objects in O are tuples that are characterized by attributes $S \cup \{z\}$ where: $S = \{s_1, \dots, s_q\}$ is a set of spatial or spatio-temporal attributes, z is a variable of interest

That is, our proposed framework for density-based clustering only allows for a single non-geo-referenced attribute that is called the *variable of interest* and which usually takes values in \mathfrak{R} .

For the remainder of the discussions in this section, we assume that datasets have the form $\langle \text{location}, \text{variable_of_interest} \rangle$; 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.

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 $o_1 = ((x_1, y_1), z_1)$ 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 over objects belonging to S . Throughout this paper d is assumed to be Euclidian distance.

In the following, we will introduce *supervised density estimation* techniques. They are 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:

$$f_{\text{Influence}}(v, o) = z(o) * e^{-\frac{d(v, o)^2}{2\sigma^2}} \quad (7-1)$$

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 [HK98]. 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 9, the influence of o_1 with respect to point v larger than the influence of o_2 on point v , that is $f_{\text{Influence}}(v, o_1) > f_{\text{Influence}}(v, o_2)$ because $d(v, o_1) < d(v, o_2)$



Figure 9: Example of Influence Computations.

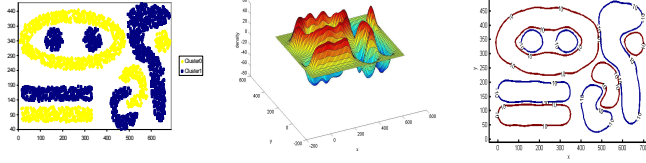
The overall influence of all data objects $o_i \in O$ for $1 \leq i \leq n$ on a point $v \in F$ is measured by the density function $\psi^o(v)$, which is defined as follows:

$$\psi^o(v) = \sum_{i=1}^n f_{\text{Influence}}(v, o_i) \quad (7-2)$$

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:

⁸ This paper introduces a simplified version of the SCMRG algorithm; in general, SCMRG is more complex in that it applies a post processing procedure to a clustering X' . The post-processing procedure greedily merges neighboring grid-cells as long as $q(X')$ improves. In summary, SCMRG is a hybrid grid-based clustering algorithm that both employs divisive and agglomerative clustering.

$$z(o) = \begin{cases} 1 & \text{if } o \text{ belong to the class of interest} \\ -1 & \text{otherwise} \end{cases} \quad (7-3)$$



a) b) c)
Figure 10: B-Complex9 dataset with its density function and a density contour map.

Figure 10 depicts a dataset in which objects belong to two classes, depicted in blue and yellow color. O contains the points depicted in Figure 10.a 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 10.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 10.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 11, 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 (7-2) 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.

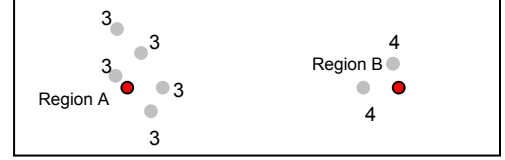


Figure 11: Example of Continuous Density Estimation

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 [SIL86] 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”.

There are many possible approaches to construct regions that represent hotspots and cool spots with respect to ψ^O . One approach—we are currently investigating—centers on finding contour clusters in $\text{Dom}(S)$ based on ψ^O . Contour clusters are defined as contiguous regions in $\text{Dom}(S)$ whose density is above a hotspot density threshold θ_{hot} or below a cool spot density threshold θ_{cool} . A regions r is called a hotspot or a cool spot if and only if:

$$\begin{aligned} \text{Hotspot}(r, \psi, \theta_{\text{hot}}) &\leftrightarrow \forall p \in r \ \psi(p) \geq \theta_{\text{hot}} \\ \text{Coolspot}(r, \psi, \theta_{\text{cool}}) &\leftrightarrow \forall p \in r \ \psi(p) \leq \theta_{\text{cool}} \end{aligned}$$

An implementation of a density-based supervised contour clustering algorithm has to satisfy the following requirements:

Given: O , ψ^O , θ_{hot} , and θ_{cool}

Find: $X = \{r_1, \dots, r_k\}$ such that

- (1) $r_i \subseteq \text{Dom}(S)$ ($i=1, \dots, k$)
- (2) $\text{contiguous}(r_i)$ ($i=1, \dots, k$)
- (3) $\text{Hotspot}(r_i, \psi^O, \theta_{\text{hot}}) \vee \text{Coolspot}(r_i, \psi^O, \theta_{\text{cool}})$ ($i=1, \dots, k$)
- (4) $r_i \cap r_j = \emptyset$ ($i \neq j$)
- (5) $\text{Hotspot}(r_i, \psi^O, \theta_{\text{hot}}) \Rightarrow \neg \exists X'$ that satisfies conditions (1)-(4) $\wedge \exists r' \in X'$ with $r' \supset r_i \wedge \text{Hotspot}(r', \psi^O, \theta_{\text{hot}})$ ($i=1, \dots, k$)
- (6) $\text{Coolspot}(r_i, \psi^O, \theta_{\text{cool}}) \Rightarrow \neg \exists X'$ that satisfies conditions (1)-(4) $r' \in X'$ with $r' \supset r_i \wedge \text{Coolspot}(r', \psi^O, \theta_{\text{cool}})$ ($i=1, \dots, k$)

Conditions (5) and (6) enforce *maximality* for the hot- and cool spots computed by a supervised contour clustering algorithms: basically, for each contiguous region r in X

which is a hotspot (cool spot) there should not be a larger contiguous region $r \supset r$ which is also a hotspot (cool spot) with respect to ψ^o .

The development of a new algorithm *SDF-CONTOUR* (“Supervised Density Function Contour Clusters”) that constructs contour maps that identify hotspot and cool spots for supervised density functions that satisfies the above specification is currently under development.

Moreover, in our past work we already introduced a novel density-based clustering algorithm called SCDE [JEC07] that computes extensional clusters that represent cool spots and hotspots with respect to ψ^o . The SCDE algorithm forms clusters by associating objects in the dataset with supervised density attractors which represent maxima and minima of a supervised density function ψ^o : objects in the dataset that are associated with the same density attractor belong to the same cluster. Hill climbing procedures are employed to associate density attractors with objects in the dataset. Only clusters whose density attractor’s density is above a hotspot threshold ξ_{hot} and below a cool-spot threshold ξ_{cool} are returned by SCDE; these clusters capture hotspots and cool spots with respect to ψ^o ; all other clusters are discarded as outliers. A much more detailed description of SCDE can be found in [JIA06].

Moreover, the popular density-based clustering algorithm DENCLUE [HK98, HG07] is a special case of SCDE in that uses a traditional density function and not a supervised density functions; that is: $\forall o \in O: z(o)=1$.

The readers might ask themselves: *what are models for density-based clustering algorithms?* The answer to this question is that ψ^o itself is the model for the dataset O . ψ^o can easily be used to cluster a different dataset O' by hill climbing the objects in O' on ψ^o . Basically, we compute density attractors for each $o' \in O'$ and then clusters are formed by merging objects that share the same density attractor, creating clusters on O' based on ψ^o .

8 Using Region Discovery Models to Characterize Relationships Between Different Clusterings

Let us assume we have two intensional clusterings $Y_{old}=\{r_1, \dots, r_k\}$ with model ρ_{old} and $Y_{new}=\{r'_1, \dots, r'_k\}$ with model ρ_{new} that have been obtained by running the same clustering algorithm on 2 datasets O_{old} and O_{new} having identical attribute sets F . Basically, we can use the model of one dataset to cluster the objects in the other dataset and compare it with original clustering for the dataset. Consequently, we generalize the definition of region extensions, given earlier, to allow for “cross-clustering”:

$$\epsilon_{r,o} = \{o \in O \mid \phi_r(o[S])=t\}$$

To find correspondence we compare

$$\epsilon_{r_1,O_{old}}, \dots, \epsilon_{r_k,O_{old}} \text{ with } \epsilon_{r'_1,O_{old}}, \dots, \epsilon_{r'_k,O_{old}}$$

as well as

$$\epsilon_{r_1,O_{new}}, \dots, \epsilon_{r_k,O_{new}} \text{ with } \epsilon_{r'_1,O_{new}}, \dots, \epsilon_{r'_k,O_{new}}$$

That is we use the model for O_{new} to cluster the examples in O_{old} and compare it with the results the model of O_{old} produces for O_{old} —we call this analysis “going backward analysis”. Similarly we can use the model for O_{old} to cluster the examples in O_{new} and compare it with the results the model of O_{new} produces for O_{new} ; this analysis is called “going forward analysis”. Basically, we compare two clusterings X and X' for the same dataset O and analyze for each pair objects $o, v \in O$ which of the following conditions it satisfies:

1. o and v belong to the same cluster in X and X'
2. o and v belong to a different cluster in X and X'
3. o and v are both outliers in X and X'
4. o and v belong to the same cluster in X and to a different cluster in X' .
5. o and v belong to the same cluster in X' and to a different cluster in X .
6. o is an outlier in X and belongs to a cluster in X'
7. o is an outlier in X' and belongs a cluster in X
8. v is an outlier in X and belongs to a cluster in X'
9. v is an outlier in X' and belongs a cluster in X

Iterating over all pairs of objects in $O \times O$ statistics of agreement between X and X' can be computed: cases 1-3 represent an agreement, and cases 4-9 represent a disagreement between X' and X . One simplified way to perform this analysis is construct *the co-occurrence matrix* M_X for a clustering X on O . The co-occurrence M_X is a $|O| \times |O|$ matrix (there is a column and row for each object $o_i \in O$ for $i=1, \dots, n$) that is computed from a clustering X as follows:

1. If o_j and o_i belong to the same cluster in X , entries (i,j) and (j,i) of M_X are set to 1.
2. If o_i is not an outlier in X , set (i,i) in M_X to 1
3. The remaining entries of M_X are set to 0

Let M_X and $M_{X'}$ be two co-occurrence matrices that have been constructed for two clusterings X and X' over O ; then the *agreement between X and X'* can be computed as follows:

$$Agreement(X, X') := (\text{Number of entries } (i,j) \text{ with } i \leq j \text{ in } M_X \text{ and } M_{X'} \text{ that both are 1 in } M_X \text{ and } M_{X'} \text{) / (Number of entries } (i,j) \text{ with } i \leq j \text{ that contain a 1 in } M_X \text{ or } M_{X'} \text{, or a 1 in both)}$$

If X and X' are identical clusters, then $\text{Agreement}(X, X')=1$.

Given two clusterings X and X' for same dataset O , relationships between the regions that belong to X and X' can be analyzed by comparing their extensions. Let r be a region in X and r' be a region in X' . Then agreement and containment between r and r' can be computed as follows.

- $\text{Agreement}(r, r') = |\epsilon_r \cap \epsilon_{r'}| / |\epsilon_r \cup \epsilon_{r'}|$
- $\text{Containment}(r, r') = |\epsilon_r \cap \epsilon_{r'}| / |\epsilon_{r'}|$

Moreover, the most similar region r' in X' with respect to r in X is the region r' for which $\text{Agreement}(r, r')$ has the highest value. In general, agreement and containment can be used to determine cluster correspondence and containment between different clusterings.

Based on what we defined so far, predicates can be defined that compare $\epsilon_{r_1, \text{Oold}}, \dots, \epsilon_{r_k, \text{Oold}}$ with $\epsilon_{r'_1, \text{Oold}}, \dots, \epsilon_{r'_k, \text{Oold}}$ as well as compare $\epsilon_{r_1, \text{Onew}}, \dots, \epsilon_{r_k, \text{Onew}}$ with $\epsilon_{r'_1, \text{Onew}}, \dots, \epsilon_{r'_k, \text{Onew}}$.

Some examples of such predicates are defined below:

Let c, c_1, c_2 be regions discovered at time t , and c', c_1', c_2' be regions that have been obtained by re-clustering the objects of time t using the model of time $t+1$. Below we list a few predicates that might be useful when analyzing the relationships between regions discovered at time t with regions discovered at time $t+1$.

1. $\text{Permanent_Cluster}(c, c') \leftrightarrow \text{Agreement}(c, c') > 0.95$
2. $\text{Evolving_Cluster}(c, c') \leftrightarrow 0.95 \geq \text{Agreement}(c, c') > 0.85$
3. $\text{New_Cluster}(c') \leftrightarrow$ Not exists c with $\text{Agreement}(c, c') > 0.5$
4. $\text{Growing_Cluster}(c, c') \leftrightarrow \text{Agreement}(c, c') < 0.85$ and $\text{Containment}(c, c') > 0.90$
5. $\text{Merged_Cluster}(c_1, c_2, c') \leftrightarrow \text{Containment}(c_1, c') > 0.90$ and $\text{Containment}(c_2, c') > 0.9$
6. $\text{Splitt_Cluster}(c, c_1', c_2') \leftrightarrow \text{Containment}(c_1', c) > 0.90$ and $\text{Containment}(c_2', c) > 0.9$
7. $\text{Dying_Cluster}(c) \leftrightarrow$ Not exists c' with $\text{Agreement}(c, c') > 0.5$

We claim that the above and similar predicates are useful to identify emergent patterns and other patterns in evolving clusterings. Moreover, these predicates can be extended to take spatial characteristics of clusters into consideration; e.g. we could try to formalize the properties of “moving clusters”.

9 Summary and Future Work

Region discovery in spatial datasets is a key technology for high impact scientific research, such as studying global climate change and its effect to regional ecosystems; environmental risk assessment, modeling, and association analysis in planetary sciences. The motivation for mining regional knowledge discovery originates from the common belief in the geosciences that “*whole map statistics are seldom useful*”, and that “*most relationships in spatial data sets are geographically regional, rather than global*” [GOO03, OPE99]. To the best of our knowledge, existing data mining technology is ill-prepared for regional knowledge discovery. To alleviate this problem our work centers on discovering regional knowledge in spatial datasets.

In particular, this paper proposed a novel, integrated framework that assists scientists in discovering interesting regions in spatial datasets in a highly automated fashion. The framework treats region discovery as a clustering problem in which we search for clusters that maximize an externally given measure of interestingness that captures what domain experts are interested in.

Our current and future work centers on analyzing the following problems:

- The current region discovery framework has been mostly developed and tested for spatial datasets. Currently, we start to investigate region discovery for spatio-temporal datasets including tasks such as: spatio-temporal hotspot discovery and emergent patterns discovery.
- The paper introduced clustering models and intensional clustering algorithms, and sketched how models could be used to analyze relationships between clustering. However, what we currently have is quite preliminary, and needs to be extended and refined in our future work.
- Our region discovery algorithms have a lot of parameters, and selecting proper values for these parameters is difficult and critical for the performance of most algorithms. Therefore, preprocessing tools are needed that select good values for these parameters automatically.
- The development of clustering algorithms that operate on the top of density functions derived from the objects in the dataset and not on the objects in the dataset themselves is an intriguing idea that deserves to be explored in much more detail in the future.

References

- [BECV05] A. Bagherjeiran, C. F. Eick, C.-S. Chen, and R. Vilalta, *Adaptive Clustering: Obtaining Better Clusters Using Feedback and Past Experience*, in Proc. Fifth IEEE International Conference on Data Mining (ICDM), Houston, Texas, November 2005.
- [CJCCGE07] J. Choo, R. Jiamthapthaksin, C. Chen, O. Celepcikay, C. Giusti, and C. F. Eick, *MOSAIC: A Proximity Graph Approach to Agglomerative Clustering*, in Proc. 9th International Conference on Data Warehousing and Knowledge Discovery (DaWaK), Regensburg, Germany, September 2007.
- [DEWY06] W. Ding, C. F. Eick, J. Wang, and X. Yuan, *A Framework for Regional Association Rule Mining in Spatial Datasets*, in Proc. IEEE International Conference on Data Mining (ICDM), Hong Kong, China, December 2006.
- [DEYWN07] W. Ding, C. F. Eick, X. Yuan, J. Wang, and J.-P. Nicot, *On Regional Association Rule Scoping*, in Proc. International Workshop on Spatial and Spatio-Temporal Data Mining (SSTD), Omaha, Nebraska, October 2007.
- [DKPJSE08] W. Ding, R. Jiamthapthaksin, R. Parmar, D. Jiang, T. Stepinski, and C. F. Eick, *Towards Region Discovery in Spatial Datasets*, in Proc. Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD), Osaka, Japan, May 2008.
- [EPDSN07] C. F. Eick, R. Parmar, W. Ding, T. Stepinski, and J.-P. Nicot, *Finding Regional Co-location Patterns for Sets of Continuous Variables*, under review.
- [EVJW06] C. F. Eick, B. Vaezian, D. Jiang, and J. Wang, *Discovery of Interesting Regions in Spatial Datasets Using Supervised Clustering*, in Proc. 10th European Conference on Principles and Practice of Knowledge Discovery in Databases (PKDD), Berlin, Germany, September 2006.
- [EZ05] C. F. Eick and N. Zeidat *Using Supervised Clustering to Enhance Classifiers*, in Proc. 15th International Symposium on Methodologies for Intelligent Systems (ISMIS), Saratoga Springs, New York, pp. 248-256, May 2005.
- [EZV04] C. F. Eick, N. Zeidat, and R. Vilalta, *Using Representative-Based Clustering for Nearest Neighbor Dataset Editing*, in Proc. Fourth IEEE International Conference on Data Mining (ICDM), Brighton, England, pp. 375-378, November 2004.
- [EZZ04] C. F. Eick, N. Zeidat, and Z. Zhao, *Supervised Clustering --- Algorithms and Benefits*, short version appeared in Proc. International Conference on Tools with AI (ICTAI), Boca Raton, Florida, pp. 774-776, November 2004.
- [GOO03] Goodchild, M., *The fundamental laws of GIS Science*, invited talk at University Consortium for Geographic Information Science, University of California, Santa Barbara, California, 2003.
- [GS69] Gabriel, K.R., Sokal, R.R., *A new statistical approach to geographic variation analysis*. Systematic Zoology 18, pp.259-278, 1969.
- [HG07] Hinneburg, A. and Gabriel, H.-H, *DENCLUE 2.0: Fast Clustering based on Kernel Density Estimation*, in Proc. 7th International Symposium on Intelligent Data Analysis (IDA), pp. 70-80, Ljubljana, Slovenia, 2007.
- [HK98] Hinneburg, A. and Keim, D. A. *An Efficient Approach to Clustering in Large Multimedia Databases with Noise*, in Proceedings of The Fourth International Conference on Knowledge Discovery and Data Mining, New York City, August 1998, 58-65.
- [JEC07] D. Jiang, C. F. Eick, and C.-S. Chen, *On Supervised Density Estimation Techniques and Their Application to Clustering*, UH Technical Report UH-CS-07-09, short version to appear in Proc. 15th ACM International Symposium on Advances in Geographic Information Systems (ACM-GIS), Seattle, Washington, November 2007.
- [JIA06] Jiang, D., *Design and Implementation of Density-based Supervised Clustering Algorithms*, Master Thesis, University of Houston, December 2006.
- [KHK99] Karypis, G., Han, E.H.S., Kumar, V.: *Chameleon: Hierarchical clustering using dynamic modeling*, IEEE Computer 32(8), pp.68-75, 1999.
- [KR00] Kaufman, L. and Rousseeuw, P. J. *Finding groups in data: An introduction to cluster analysis*, John Wiley and Sons, New Jersey, USA, 2000.
- [OPE99] Stan Openshaw. *Geographical data mining: Key design issues*, GeoComputation, 1999.
- [PAR07] R. Parmar, *Finding Regional Co-location Patterns using Representative-based Clustering Algorithms*, Master's Thesis, University of Houston, December 2007.
- [SIL86] Silverman, B. *Density Estimation for Statistics and Data Analysis*. Chapman & Hall, London, UK, 1986.
- [SPH05] Shekhar S., Zhang P., and Huang Y., *Spatial Data Mining*. The Data Mining and Knowledge Discovery Handbook, pp. 833-851, 2005.
- [ZE04] N. Zeidat and C. F. Eick, *K-medoid-style Clustering Algorithms for Supervised Summary Generation*, in Proc. 2004 International Conference on Machine Learning; Models, Technologies and Applications (MLMTA'04), Las Vegas, Nevada, pp. 932-938, June 2004.
- [ZEZ06] N. Zeidat, C. F. Eick, and Z. Zhao, *Supervised Clustering: Algorithms and Applications*, UH Technical Report UH-CS-06-10, June 2006.