

# Technology for Discovering Interesting Regions in Spatial Data Sets

*Christoph F. Eick, Wei Ding, Jing Wang,  
Rachsuda Jiamthapthaksin, and Dan Jiang*

Department of Computer Science, University of Houston  
Houston, Texas 77204-3010, U.S.A  
{ceick, wding, jwang, rachsuda, djiang}@cs.uh.edu

## 1 Motivation and Problem Definition

Finding interesting documents on the Internet has been major focus of recent research and commercial projects. For example, GOOGLE became famous for building search engines that can efficiently navigate through millions of documents and return a ranked set of documents based on user interests and user feedback. Earth scientists are interested to have similar capabilities to search for interesting regions on the planet earth based on knowledge stored in multiple databases. The Data Mining and Machine Learning Group of the University of Houston (UH-DMML) is dedicated to developing technologies that can satisfy this very need. The focus of this report is to describe in details what techniques, tools, and methodologies we have to offer to scientists who face challenging region discovery problems and who are looking for search-engine style capabilities to find interesting subregions on the planet earth.

Data mining has been identified as a key technology to automate the extraction of interesting, useful, but implicit patterns in large spatial datasets. In particular, we are interested in assisting scientists in finding interesting regions in spatial data sets. 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 next two 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. Figure 1 shows the results of identifying arsenic concentration in Texas wells, with green color represents good wells, while red color represents wells with high arsenic concentration. This discovery might lead to further field study that explores the relationship between farm use and well pollution in a particular region. Figure 2 gives an 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 subregion 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.

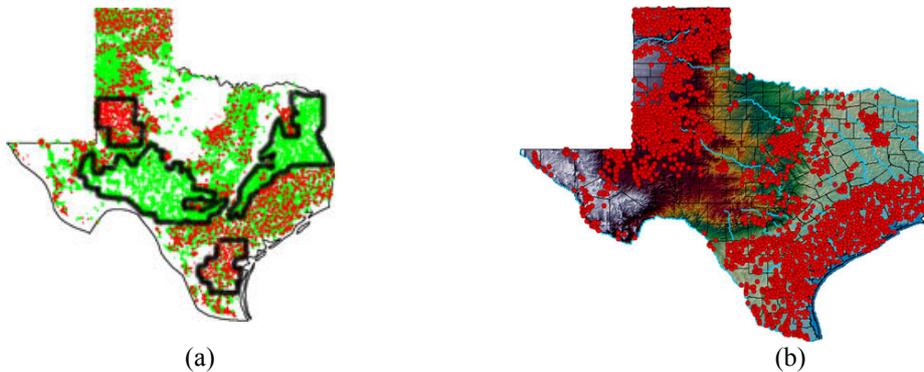
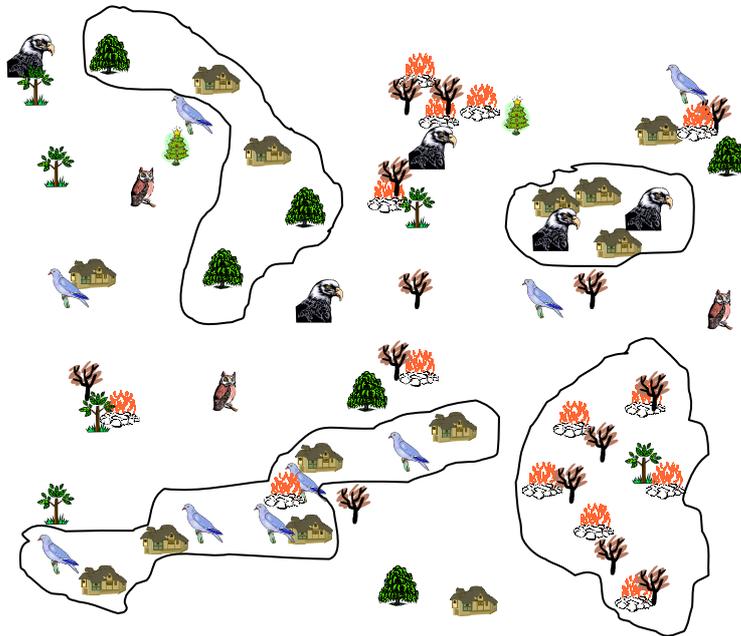


Figure 1: Finding regions with a very high and low density of “good” (in green) and “bad” (in red) wells.



**Global Answer:**  and 

Figure 2: Finding regions with interesting co-location characteristics.

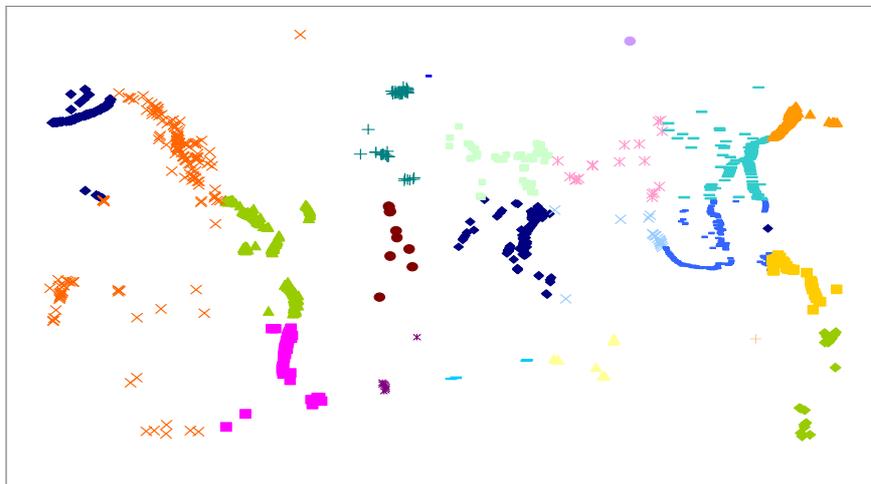


Figure 3: Finding groups of violent volcanoes and groups of non-violent volcanoes.

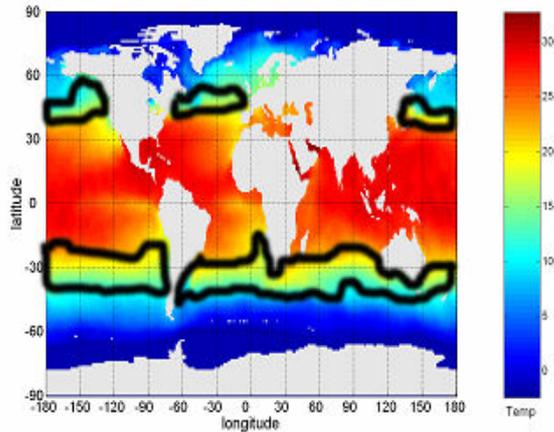


Figure 4: Finding the regions where hot water meets cold water.

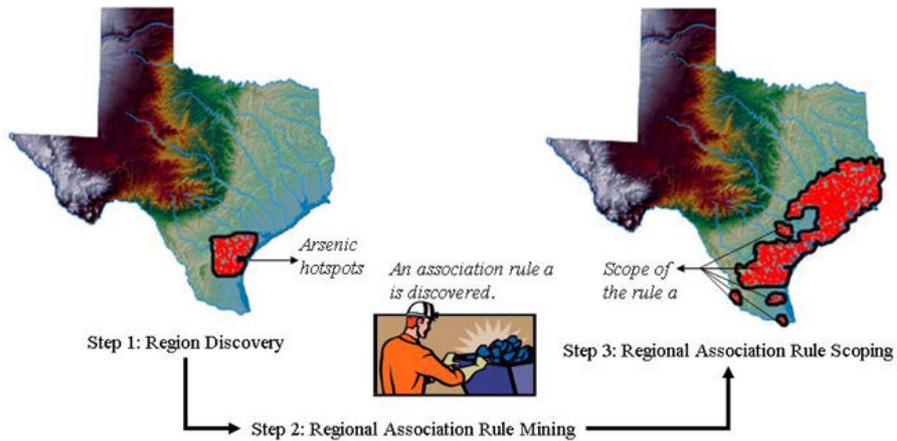


Figure 5: Regional association rule mining and scoping.

Figures 1-5 depict results of using our region discovery algorithms for some examples applications. Figure 1 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 2 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 volcanos. 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 association rule is computed that indicates the region in which a particular rule holds.

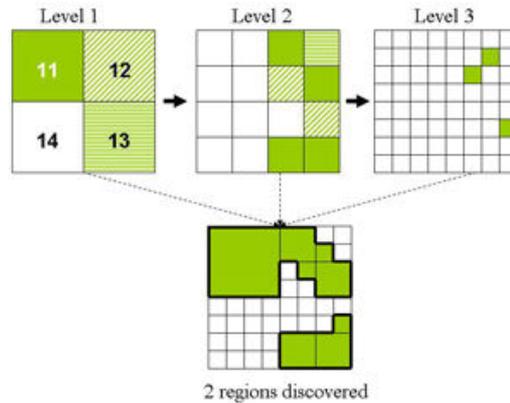


Figure 6: The Supervised Clustering using Multi-Resolution Grids algorithm (SCMRG)

In general, our particular approach views region discovery as a clustering problem. The goal of region discovery is to find a set of clusters (regions) that maximizes an externally given fitness function. Figure 6 depicts the running of divisive region discovery algorithm named Supervised Clustering using Multi-Resolution Grids (SCMRG) searches for interesting regions (depicted in dark green) at a particular level of granularity, and drills down, looking at subregions by splitting regions using rectangular grids, if such an exploration shows some “promise”. During the process the algorithm collects interesting regions at different levels of resolution, and merges the regions to obtain the final region discovery result that consists of 2 regions (more details how the algorithm works are given in section 3.3)

As of now 6 different clustering algorithms for region discovery already exist and 2 more are currently under development. In general, the different clustering algorithms have different characteristics, some are very good in detecting clusters at different levels of granularity, some are good in detecting regions with unusual shapes, some are fast and therefore can cope with very large datasets, and some are good in identifying relatively small clusters. More details about the employed clustering framework are given in Section 2, and more about clustering algorithms is discussed in Section 3.

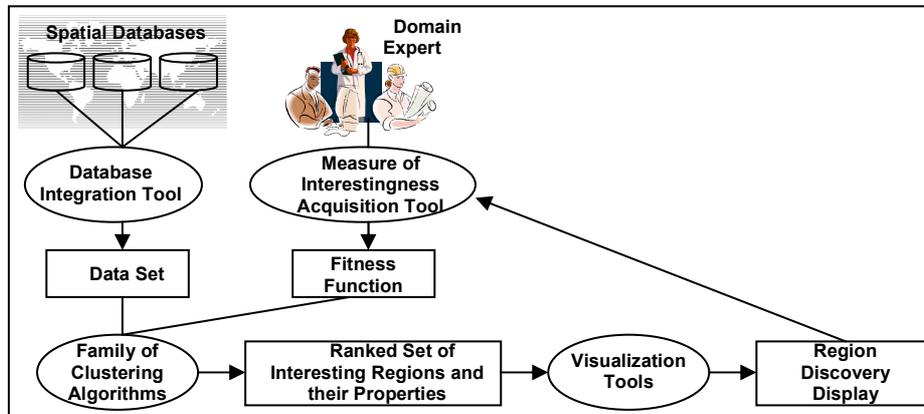


Figure 7: Region Discovery System Architecture

The ultimate vision of this research is the development of region discovery engines that assist scientist in finding interesting regions in spatial datasets in a highly automated fashion. Fig. 7 depicts the architecture of a region discovery system. The key-component of the system is a family of clustering algorithms that detect interesting regions with respect to a given reward-based fitness function. Additionally, the Measure of Interestingness Acquisition Tool assists domain experts in selecting the appropriate fitness functions and in selecting in tuning parameters of the employed fitness functions. Region discovery usually involves data that are stored in multiple spatial databases. The task of the Database Integration Tool is to integrate data from different databases before the region discovery algorithms can be applied. Finally, a visualization tool displays the discovered regions and their properties in an understandable form and interactively allows the domain expert to change what is visualized and how it is visualized.

In summary, key contribution of our work is the proposal of a generic region that are capable of capturing what domain scientist are interested in, and families of clustering algorithms that find regions the user is looking for. Region discovery faces several challenges that do not exist in information retrieval, such as the need to find regions of arbitrary shape and at arbitrary levels of resolution, the definition of suitable parameterized measures of interestingness to instruct discovery algorithms what they are supposed to be looking for, and the need to reduce computational complexity due to the large size of most spatial datasets. The next session will discuss the proposed region discovery framework in more detail. Table 1 summarizes the notations used in this paper.

Table 1: Notations used in the paper.

Notation	Description
$O=\{o_1, \dots, o_n\}$	Objects in a dataset (or training set)
$N$	Number of objects in the dataset
$c_i \subset O$	The $i$ -th cluster
$X=\{c_1, \dots, c_k\}$	A clustering solution consisting of clusters $c_1$ to $c_k$
$q(X)$	Fitness function that evaluates a clustering $X$
$C$	A class label

## 2 A Region Discovery Framework

As we explained earlier, our approach uses supervised clustering algorithms to identify interesting regions in a dataset. A region, in our approach, is defined as a surface containing a set of spatial objects; for instance, the convex hull of the objects belonging to a cluster. Moreover we require regions to be disjoint and contiguous; for each pair of objects belonging to a region, there must always be a path within this region that connects the pair of objects. Furthermore, we assume that the number of regions is not known in advance, and therefore finding the best number of regions is one of the objectives of the clustering process. Consequently, our evaluation scheme must be capable of comparing clusterings that use a different number of clusters.

Our approach employs a reward-based evaluation framework. The fitness function  $q(X)$  of a clustering  $X$  is computed as the sum of the rewards obtained for each cluster  $c_i \in X$ . Cluster rewards are weighted by the number of objects that belong to a cluster  $c_i$ . In general, we are interested in finding larger clusters if larger clusters are equally interesting as smaller clusters. Consequently, our evaluation scheme uses a parameter  $\beta$  with  $\beta > 1$  and fitness increases nonlinearly with cluster-size dependent on the value of  $\beta$ , favoring clusters  $c_i$  with more objects.

$$q(X) = \sum_{c_i \in X} (\text{reward}(c_i) * |c_i|^\beta) \quad (1)$$

Selecting larger values for the parameter  $\beta$  usually results in a smaller number of clusters in the best clustering  $X$ . The proposed evaluation scheme is very general; different reward schemes that correspond to different measures of interestingness can easily be supported in this framework. Accordingly, the clustering algorithms that will be introduced in the second half of the paper

can be run with any fitness functions without modifying the clustering algorithm itself.

In this section, we introduce a single measure of interestingness that centers on discovering *hotspots* and *coldspots* in a dataset. Basically, the measure is based on a class of interest  $C$ , and rewards regions in which the distribution of class  $C$  significantly deviates from its prior probability, relying on a reward function  $\tau$ .  $\tau$  itself is computed based on  $p(c_i, C)$ ,  $prior(C)$ , and the following parameters:  $\gamma_1, \gamma_2, R+, R-$  with  $\gamma_1 \leq 1 \leq \gamma_2, 1 \geq R+, R- \geq 0, \eta > 0$  as shown in figure 8.

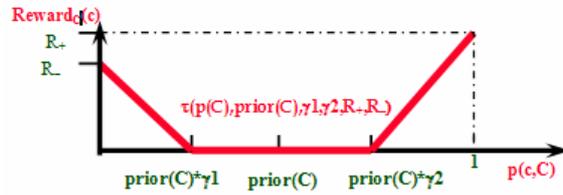


Figure 8: The reward function  $\tau_c$  for  $\eta=1$

Then, the fitness function  $q(X)$  is defined as follows:

$$q(X) = \sum_{i=1}^{|X|} \frac{\tau(p(c_i, C), prior(C), \gamma_1, \gamma_2, R+, R-, \eta) * (|c_i|)^\beta}{n^\beta} \quad (2)$$

with

$$\tau(p(c_i, C), prior(C), \gamma_1, \gamma_2, R+, R-, \eta) = \begin{cases} \left( \frac{((prior(C) * \gamma_1) - P_{c_i, C}) * R-}{(prior(C) * \gamma_1)} \right)^\eta & \text{if } P_{c_i, C} < prior(C) * \gamma_1 \\ \left( \frac{(P_{c_i, C} - (prior(C) * \gamma_2)) * R+}{(1 - (prior(C) * \gamma_2))} \right)^\eta & \text{if } P_{c_i, C} > prior(C) * \gamma_2 \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

In the above formula  $prior(C)$  denotes the probability of objects in dataset belonging to the class of interest  $C$ . The parameter  $\eta$  determines how the reward quickly grows to the maximum reward (either  $R+$  or  $R-$ ). If  $\eta$  is set to 1, it grows linearly. In general, if we are interested in giving higher rewards to purer clusters, it is desirable to choose larger values for  $\eta$ , for instance,  $\eta=8$ .

We provide an example that demonstrates the measure of interestingness for detected “*Poor*” regions. Let us assume a clustering  $X$  has to be evaluated with respect to a class of interest “*Poor*” that contains 1000 examples.

Suppose that the generated clustering  $X$  subdivides the dataset into three clusters  $c_1, c_2, c_3$  with the following characteristics.  $|c_1| = 250, |c_2| = 200, |c_3| = 550; p(c_1) = 130/250, p(c_2) = 20/200, p(c_3) = 50/550$ . The parameters used in the fitness 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, clusters that contain between 10% and 30% instances of the class “*Poor*” do not receive any reward at all; therefore, no reward is given to cluster  $c_2$ . The remaining clusters received rewards because the distribution of class “*Poor*” in the cluster is significantly higher or lower than the corresponding threshold. Consequently, the reward for the first cluster  $c_1$  is  $11/35 \times (250)^{1.1}$  since  $p(c_1) = 52\%$  is greater than 30%,  $11/35$  is obtained by applying the function  $p(c)$ , thus we get  $p(c_1) = ((0.52-0.3)/(1-0.3)) * 1 = 11/35$ . Rewards of other clusters are computed similarly and the following overall reward for  $X$  is obtained:

$$q_{Poor}(X) = \frac{\frac{11}{35} * 250^{1.1} + 0 + \frac{1}{11} * 550^{1.1}}{1000^{1.1}} = 0.115$$

### 3 Clustering Algorithms that Discover Interesting Regions

As part of our research, we have designed and implemented 6 clustering algorithms 3 of which will be described in this section.

#### 3.1 Supervised Clustering Using Agglomerative Hierarchical Techniques (SCAH)

SCAH is an agglomerative, hierarchical supervised clustering algorithm. Initially, it forms single object clusters, and then greedily merges clusters as long as it improves the clustering quality. In particular, a pair of clusters  $(c_i, c_j)$  is considered to be a merge candidate if  $c_i$  is the closest cluster to  $c_j$ , or  $c_j$  is the closest cluster to  $c_i$ . Distances between clusters are measured by using the average distance between the objects belonging to the two clusters. The pseudo code of the SCAH algorithm is given in figure 9.

In general, SCAH differs from traditional hierarchical clustering algorithms which merge the two clusters that are closest to each other in that it considers more alternatives for merging clusters. This is important for supervised clustering because merging two regions that are closest to each other will frequently not lead to a better clustering, especially if the two regions to be merged are dominated by instances belonging to different classes.

**Inputs:**

A dataset  $O = \{o_1, \dots, o_n\}$

A dissimilarity Matrix  $D = \{d(o_i, o_j) \mid o_i, o_j \in O\}$ ,

**Output:**

Clustering  $X = \{c_1, c_2, \dots, c_k\}$

**Algorithm:**

- 1) **Initialize:**
  - Create single object clusters:  $c_i = \{o_i\}, 1 \leq i \leq n$ ;
  - Compute merge candidates
- 2) **DO FOREVER**
  - a) Find the pair  $(c_i, c_j)$  of merge candidates that improves  $q(X)$  the most
  - b) IF no such pair exists THEN
    - Terminate and return  $X = \{c_1, c_2, \dots, c_k\}$
  - c) Delete the two clusters  $c_i$  and  $c_j$  from  $X$  and add the cluster  $c_i \cup c_j$  to  $X$
  - d) Update merge candidates

Figure 9: The SCAH Algorithm

### 3.2 Supervised Clustering Using Hierarchical Grid-based Techniques (SCHG)

Grid-based clustering methods are designed to deal with the large number of data objects in a high dimensional attribute space. A grid structure is used to quantize the space into a set of grid cells on which all clustering operations are performed. The main advantage of this approach is its fast processing time which is typically independent of the number of data objects, and only depends on the number of occupied cells in the quantized space. SCHG is an agglomerative, grid-based clustering method. Initially, each occupied grid cell is considered to be a cluster. Then, SCHG tries to improve the quality of the clustering by greedily merging two clusters that share a common boundary. The algorithm terminates if  $q(X)$  no longer improves by further merging.

### 3.3 Supervised Clustering Using Multi-Resolution Grids (SCMRG)

Supervised Clustering using Multi-Resolution Grids (SCMRG) is a hierarchical grid based method that utilizes a divisive, top-down search. Each cell at a higher level is partitioned further into a number of smaller cells in the next lower level. This iterative process continues only when the sum of the rewards of the lower level cells is higher than the obtained reward for the cell at the higher level. The returned cells usually have different sizes, because they were obtained at different level of resolution. In summary, the algorithm

starts at a user defined level of resolution, and considers three cases when processing a cell:

1. If a cell receives a reward, and its reward is larger than the sum of the rewards associated of its children and larger than the sum of rewards of its grandchildren, this cell is returned as a cluster by the algorithm.
2. If a cell does not receive a reward and its children and grandchildren do not receive a reward, neither the cell nor any of its descendants will be included in the result.
3. Otherwise, all the children cells of the cell are put into a queue for further processing.

The algorithm also uses a user-defined cell size as a depth bound; cells that are smaller than this cell size will not be split any further. The employed framework has some similarity with the framework introduced in the STING algorithm except that our version centers on finding interesting cells instead of cells that contain answers to a given query, and only computes cell statistics when needed and not in advance as STING does.

#### **4. Summary**

In this paper, we introduced a novel framework for region discovery in spatial datasets with the goal to provide search-engine like capabilities to earth scientists. Families of reward-based fitness functions and various clustering algorithms are provided to provide these capabilities for many region discovery tasks. Different measures of interestingness can easily be supported in the proposed framework by designing different reward-based fitness functions; in this case, neither the clustering algorithm itself nor our general evaluation framework has to be modified.

#### **References**

[DEWY06] W. Ding, C. 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.

[EVDW06] C. 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.