Published in Proceedings of 3rd International Conference on Knowledge Discovery and Data Mining (KDD-97)

Density-Connected Sets and their Application for Trend Detection in Spatial Databases

Martin Ester, Hans-Peter Kriegel, Jörg Sander, Xiaowei Xu

Institute for Computer Science, University of Munich
Oettingenstr. 67, D-80538 München, Germany
{ester | kriegel | sander | xwxu}@informatik.uni-muenchen.de

Abstract[1]

Several clustering algorithms have been proposed for class identification in spatial databases such as earth observation databases. The effectivity of the well-known algorithms such as DBSCAN, however, is somewhat limited because they do not fully exploit the richness of the different types of data contained in a spatial database. In this paper, we introduce the concept of density-connected sets and present a significantly generalized version of DBSCAN. The major properties of this algorithm are as follows: (1) any symmetric predicate can be used to define the neighborhood of an object allowing a natural definition in the case of spatially extended objects such as polygons, and (2) the cardinality function for a set of neighboring objects may take into account the non-spatial attributes of the objects as a means of assigning application specific weights. Density-connected sets can be used as a basis to discover trends in a spatial database. We define trends in spatial databases and show how to apply the generalized DBSCAN algorithm for the task of discovering such knowledge. To demonstrate the practical impact of our approach, we performed experiments on a geographical information system on Bavaria which is representative for a broad class of spatial databases.

Keywords: Clustering Algorithms, Spatial and non-spatial data, Trend Detection, Application to Geographic Information Systems.

1. Introduction

Increasingly large amounts of data obtained from satellite images, X-ray crystallography or other automatic equipment are stored in databases. Therefore, automated knowledge discovery becomes more and more important in databases. *Knowledge discovery in databases (KDD)* can be defined as the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. *Data mining* is a step in the KDD process consisting of the application of data analysis and discovery algorithms that, under acceptable computational efficiency limitations, produce a particular enumeration of patterns over the data (Fayyad, Piatetsky-Shapiro & Smyth 1996).

Spatial Database Systems (SDBS) (Gueting 1994) are database systems for the management of spatial data. Spatial data are data related to space, e.g. a part of the 2D surface of the earth. While a lot of research has been conducted on knowledge discovery and data mining in relational databases in the last few years, only a few methods for knowledge discovery in spatial databases have been proposed in the literature.

Lu, Han & Ooi (1993) propose a generalization-based method to extract high-level relationships between spatial and non-spatial data stored in a spatial database. Koperski & Han (1995) present an algorithm to discover spatial association rules of the form $X \longrightarrow Y$ ($c\%$), where X and Y are sets of spatial or non-spatial predicates and c is the confidence of the rule. The implementation of the algorithm is based on the spatial join which is one of the basic operations in SDBS .

Recently, several clustering methods have been developed for the application on spatial databases (Ng & Han 1994) (Ester, Kriegel & Xu 1995) (Ester et al. 1996) (Zhang, Ramakrishnan & Linvy 1996). All these methods are designed for point objects, i.e. objects without extension. In a spatial database, however, objects are usually spatially extended with one or more non-spatial attributes. For example, objects in a geographic information system (GIS) may be polygons which represent, e.g., communities or lakes with non-spatial attributes like name, average income, number of houses in the area, etc. One can use all the clustering methods mentioned above to cluster general spatial objects by transforming them to points in some feature space. However, some spatial relationships between them will be lost. If clustering in the original or in the transformed space, it is difficult to find a natural definition for the distance of general spatial objects if their non-spatial attributes should be considered.

The clusters detected by any algorithm can be used as input for other KDD tasks. Knorr & Ng (1996) study the proximity relationships between clusters of points and polygonal objects in a spatial database. For a given cluster of points, they give an algorithm which can efficiently find the “top-k” polygons that are “closest” to the cluster. For n given clusters of points, an algorithm is presented which can find common polygons or classes of polygons that are nearest to most, if not all, of the clusters.

In this paper, we use clustering as a basic operation for KDD in spatial databases. E.g., one may be interested in discovering trends of some non-spatial attribute for all spatial objects in neighboring regions. We present the algorithm GDBSCAN which is a generalized version of DBSCAN (Ester et al. 1996) and can cluster general spatial objects according to both spatial and non-spatial attributes. To demonstrate the applicability as a basic operation for data mining, we use GDBSCN to find interesting regions for trend detection in a geographic database.

The rest of the paper is organized as follows. We present the notion of density-connected sets and an algorithm to detect them in section 2. Section 3 discusses the task of trend detection in a geographic database and shows how to use GDBSCAN as a basic operation. Section 4 concludes with a summary and some directions for future research.

2. Density-Connected Sets in Spatial Databases

In the following, we introduce the notion of "density-connected sets" which is a significant generalization of the notion of "clusters" as presented in (Ester et al. 1996). We assume a spatial database D to be a finite set of objects characterized by spatial and non-spatial attributes. The spatial attributes may represent, e.g., points or spatially extended objects such as polygons in some d-dimensional space S. The non-spatial attributes of an object in D may represent additional properties of a spatial object, e.g., the unemployment rate for a community represented by a polygon in a geographic information system.

The key idea of a density-based cluster is that for each point of a cluster its ε-neighborhood for some given $\varepsilon > 0$ has to contain at least a minimum number of points, i.e. the "density" in the ε-neighborhood of points has to exceed some threshold. This idea of "density" can be generalized in two important ways. First, we can use any notion of a neighborhood of an object if the definition of the neighborhood is based on a binary predicate which is symmetric and reflexive. Second, instead of simply counting the objects in a neighborhood of an object we can as well use other measures to define the "cardinality" of that neighborhood.

Definition 1. (*neighborhood* of an object) Let *NPred* be a binary predicate on D which is reflexive and symmetric, i.e., for all p, $q \in D$: $NPred(p, p)$ and, if $NPred(p, q)$ then $NPred(q, p)$. Then the *NPred*-neighborhood of an object $o \in D$ is defined as $N_{NPred}(o) = \{o' \in D \mid NPred(o, o')\}$.

The definition of a cluster in (Ester et al. 1996) is restricted to the special case of a distance based neighborhood, i.e., $N_\varepsilon(o) = \{o' \in D \mid$ *the distance between o and o' is less than or equal to* $\varepsilon\}$. A distance based neighborhood is a natural notion of a neighborhood for point objects, but it is not clear how to apply it for the clustering of spatially extended objects such as a set of polygons of largely differing sizes. Neighborhood predicates like *intersects* or *meets* are more appropriate for finding clusters of polygons, i.e. density-connected sets of polygons, in many cases.

Although in many applications the neighborhood predicate will be defined using only spatial properties of the objects, the formalism is in no way restricted to purely spatial neighborhoods. We can as well use non-spatial attributes and combine them with spatial properties of objects to derive a neighborhood predicate. Suppose, we have a database of polygons representing communities in a country with the non-spatial attribute "unemployment rate" taking values "very low", "low", "medium", "high", "very high". Then, we can define that o is a neighbor of o' if and only if polygon(o) intersects polygon(o') *and* the unemployment rate of o is equal to the unemployment rate of o'.

Another way to take into account the non-spatial attributes of objects is as a kind of "weight" when calculating the "cardinality" of the neighborhood of an object. To keep things as simple as possible, we will not introduce a weight function operating on objects, but a *weighted cardinality* function *wCard* for sets of objects. The "weight" of a single object o can then be expressed by the weighted cardinality of the singleton containing o, i.e. $wCard(\{o\})$. This particular generalization of the parameter *MinPts* in the algorithm DBSCAN and some example applications on databases containing point objects can also be found in (Sander et al. 1997).

Definition 2. (*MinWeight* of a set of objects) Let *wCard* be a function from the powerset of the Database D into the non-negative Real Numbers, $wCard: 2^D \longrightarrow \Re^{\geq 0}$ and *MinCard* be a positive real number. Then, the predicate *MinWeight* for sets of objects S is defined to be true iff $wCard(S) \geq MinCard$.

There are numerous possibilities to define $wCard(S)$ for subsets of the database D. A special *wCard* function, called the "*default* weighted cardinality" is the common cardinality from set theory (i.e. the number of objects in subsets of the database). Simply summing up the values of some non-spatial attribute for the objects in S is another example of a *wCard* function. E.g., if we want to cluster objects represented by polygons and if the size of the objects should be considered to influence the "density" in the data space, then the area of the polygons could be used as a weight for objects. A further possibility is to sum up a value derived from several non-spatial attributes, e.g. by specifying ranges for some non-spatial attribute values of the objects (i.e. a selection condition), we can realize the clustering of only a subset of the database D by attaching a weight of 1 to objects that satisfy the selection condition and a weight of 0 to all other objects. Note that using non-spatial attributes as a weight for objects one can "induce" different densities, even if the objects are equally distributed in the space of the spatial attributes. Note also that by means of the *wCard* function the combination of a clustering with a selection on the database is possible, allowing a tight integration of the generalized DBSCAN algorithm with a SDBMS.

We can now define density-connected sets, analogously to the definition of density-based clusters in (Ester et al. 1996), in a straightforward way.

Definition 3. (directly density-reachable) An object p is *directly density-reachable* from an object q wrt. *NPred*, *MinWeight* if

1) $p \in N_{NPred}(q)$ and

2) $MinWeight(N_{NPred}(q)) = true$ (core object condition).

Definition 4: (density-reachable) An object p is *density-reachable* from an object q wrt. *NPred, MinWeight* if there is a chain of objects $p_1, ..., p_n$, $p_1 = q$, $p_n = p$ such that p_{i+1} is directly density-reachable from p_i wrt. *NPred*, *MinWeight*.

Definition 5: (density-connected) An object p is *density-connected* to an object q wrt. *NPred*, *MinWeight* if there is an object o such that both, p and q are density-reachable from o wrt. *NPred*, *MinWeight*.

Density-reachability is a canonical extension of direct density-reachability. Density-reachability is transitive, but it is not symmetric. Figure 1 depicts the relations density-reachability and density-connectivity on a sample database of 2D points using a distance based neighborhood for the points and the default weighted cardinality. Although not symmetric in general, an important property of density-reachability is that it is symmetric for core objects. This holds because a chain from q to p can be "reversed" if also p is a core object since we require the neighborhood predicate to be reflexive and symmetric. Density-connectivity is a symmetric relation. For density-reachable objects, the relation of density-connectivity is also reflexive.

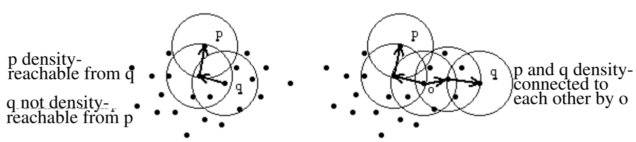

figure 1: Density-reachability and density-connectivity

Definition 6: (density-connected set) A *density-connected set C* wrt. *NPred, MinWeight* in D is a non-empty subset of D satisfying the following conditions:
1) Maximality: $\forall\, p, q \in D$: if $p \in C$ and q is density-reachable from p wrt. *NPred*, *MinWeight*, then $q \in C$.
2) Connectivity: $\forall p,q \in C$: p is density-connected to q wrt. *NPred*, *MinWeight*.

We can now define a clustering *CL* of a database D wrt. *NPred*, *MinWeight* based on density-connected sets as the set of all density-connected sets wrt. *NPred*, *MinWeight* in D, i.e. all clusters from a clustering *CL* are density-connected sets with regard to the same "parameters" *NPred* and *MinWeight. Noise* will then be defined relative to a given clustering *CL* of D, simply as the set of objects in D not belonging to any of the clusters of *CL*.

Definition 7: (clustering) A clustering *CL* of D wrt. *NPred*, *MinWeight* is a set of density-connected sets wrt. *NPred*, *MinWeight* in D, $CL = \{C_1, \ldots, C_k\}$, such that for all C: if C is a density-connected set wrt. *NPred*, *MinWeight* in D, then $C \in CL$.

Definition 8: (noise) Let $CL=\{C_1, \ldots, C_k\}$ be a clustering of the database D wrt. *NPred, MinWeight*. Then we define the *noise* in D as the set of objects in the database D not belonging to any density-connected set C_i, i.e.
$noise_{CL} = D \setminus (C_1 \cup \ldots \cup C_k)$.

The algorithm DBSCAN presented in (Ester et al. 1996) is based on two lemmata which can also be proven for the generalized notion of a cluster, i.e. a density-connected set. In the current context they state the following. Given the parameters *NPred* and *MinWeight*, we can discover a density-connected set in a two-step approach. First, choose an arbitrary object from the database satisfying the core object condition as a seed. Second, retrieve all objects that are density-reachable from the seed obtaining the density-connected set containing the seed. Furthermore, a density-connected set C wrt. *NPred, MinWeight* is uniquely determined by *any* of its core objects, i.e., each object in C is density-reachable from any of the core objects of C and, therefore, a density-connected set C contains exactly the objects which are density-reachable from an arbitrary core object of C.

Lemma 1: Let p be an object in D and $MinWeight(N_{NPred}(p)) = true$. Then the set
$O = \{o \in D \mid o$ is density-reachable from p wrt. $NPred, MinWeight\}$ is a density-connected set wrt. *NPred, MinWeight*.

Proof: 1) O is non-empty: p is a core object by assumption. Therefore p is density-reachable from p. Then p is in O.
2) Maximality: Let $q_1 \in O$ and q_2 be density-reachable from q_1 wrt. *NPred*, *MinWeight*. Since q_1 is density-reachable from p and density-reachability is transitive wrt. *NPred*, *MinWeight*, it follows that also q_2 is density-reachable from p wrt. *Eps*, *MinWeight*. Hence, $q_2 \in O$. *3)* Connectivity: All objects in O are density-connected via object p. ❑

Lemma 2: Let C be a density-connected set wrt. *NPred, MinWeight* and let p be any object in C with $MinWeight(N_{NPred}(p)) = true$. Then C equals to the set $O = \{o \in D \mid o$ is density-reachable from p wrt. $NPred, MinWeight\}$.

Proof: 1) $O \subseteq C$ by definition of O. 2) $C \subseteq O$: Let $q \in C$. Since also $p \in C$ and C is a density-connected set, there is an object $o \in C$ such that p and q are density-connected via o, i.e. both p and q are density-reachable from o. Because both p and o are core objects, it follows that also o is density-reachable from p (symmetry for core objects). With transitivity of density-reachability wrt. *NPred*, *MinWeight* it follows that q is density-reachable from p. Then $q \in O$. ❑

Since we have proven that density-connected sets in their most general form have the same properties as density-based clusters, as far as the procedure for finding them is concerned, we can use the same algorithmic schema to detect both.

Note that GDBSCAN is similar to a simple form of region growing. Note also, that there are special instances of the schema GDBSCAN in which density connected sets are in fact clusters in a very common sense: if the default weighted cardinality is used, *MinWeight* set to 2, and an ε-neighborhood is used for *NPred* where ε corresponds to an NN-distance, then a clustering wrt. *NPred, MinWeight* is equivalent to the *level* in the *single-link hierarchy* (Sibson 1973) determined by the "critical distance" $D_{min} = \varepsilon$ if the points p in the set $noise_{CL}$ are considered as a single cluster as well.

```
Algorithm GDBSCAN (Generalized DBSCAN)
// Precondition: All objects in D are unclassified.
FORALL objects o in D DO:
   IF o is unclassified and wCard(N_NPred({o})) > 0
      call function expand_cluster to construct a density-
      connected set wrt. NPred, MinWeight containing o.

FUNCTION expand_cluster:
retrieve the neighborhood of o wrt. NPred, MinWeight;
IF MinWeight(N_NPred(o)) = false
   // i.e. o is not a core object
   mark o as noise and RETURN;
ELSE // i.e. o is a core object
   select a new cluster-id and mark all objects in N_NPred(o)
   with this current cluster-id;
   push all objects from N_NPred(o)\{o} onto the stack seeds;
   WHILE NOT seeds.empty() DO
      currentObject := seeds.top();
      retrieve the neighborhood of currentObject,
      // i.e. N_NPred(currentObject);
      IF MinWeight(N_NPred(currentObject)) = true
         select all objects in N_NPred(currentObject)
         which are not yet classified or are marked as
         noise,push the unclassified objects onto seeds
         and mark all of these objects with current cluster-id;
      seeds.pop();
RETURN
```

In (Ester et al. 1996) we argued that the good efficiency of DBSCAN is due to the fact that DBSCAN uses spatial access methods such as R*-trees, which efficiently support range queries to extract the ε-neighborhood of a point p. For small query regions the runtime complexity of a region query using R*-trees is $O(\log n)$. Since the regions used for DBSCAN are assumed to be very small compared to the size of the dataspace, we have in general an overall runtime complexity of $O(n \log n)$ where n is the number of points in the database.This analysis was confirmed by the experiments on real datasets reported in (Ester et al. 1996).

In the case of polygonal objects, spatial access methods can also be used to support efficient computation of the neighborhood of a polygon for some neighborhood predicate based on a topological relation like *intersects*. But because we cannot store polygons directly in such a spatial access structure, we have to use a multi-step filter-refinement procedure for the computation of the neighborhood (Brinkhoff et al. 1994). A further filter step becomes necessary if the neighborhood predicate *NPred* is defined as a combination of spatial and non-spatial attributes. The multi-step approach to spatial query proccessing for databases containing extended spatial objects also scales well with the size of the database (Brinkhoff et al. 1994).

3. Finding Interesting Regions for Trend Detection in a Geographic Information System

A *geographic information system* is an information system to manage data representing aspects of the surface of the earth together with relevant facilities such as roads or houses. In this section, we introduce a geographic database (*Bavaria database*) providing spatial and non-spatial information on Bavaria with its administrative units such as communities, its natural facilities such as the mountains and its infrastructure such as roads. The database contains the ATKIS 500 data (Atkis 1996) and the Bavarian part of the statistical data obtained by the German census of 1987. We use the SAND (Spatial And Non-spatial Database) architecture (Aref & Samet 1991): the spatial extension of all objects (e.g. polygons and lines) is stored and manipulated using an R*-tree (Brinkhoff et al. 1990), the non-spatial attributes of the communities (54 different attributes such as the rate of unemployment and the average income) are managed by a relational database management system.

The Bavaria database may be used, e.g., by economic geographers to discover different types of knowledge. In the following, we discuss the tasks of spatial classification and spatial trend detection.

Spatial classification should discover rules predicting the class membership of some object based on the spatial and non-spatial attributes of the object and its neighbors. The object may also be a density-connected set of objects, e.g. an agglomeration of several close cities, and the following spatial classification rule may be discovered:

```
if there is some agglomeration of cities,
then this agglomeration neighbors a highway
(confidence 75%)
```

A *trend* has been defined as a temporal pattern in some time series data such as network alarms or occurrences of recurrent illnesses (Berndt & Clifford 1996), e.g. “rising interest rates”. We define a *spatial trend* as a pattern of systematic change of one or several non-spatial attributes in 2D or 3D space.

To discover spatial trends of the economic power, an economic geographer may proceed as follows. Some non-spatial attribute such as the rate of unemployment is chosen as an indicator of the economic power. In a first step, areas with a locally minimal rate of unemployment are determined which are called *centers,* e.g. the city of Munich. The theory of central places (Christaller 1968) claims that the attributes of such centers influence the attributes of their neighborhood to a degree which decreases with increasing distance. E.g., in general it is easy to commute from some community to a close by center thus implying a low rate of unemployment in this community. In a second step, the theoretical trend of the rate of unemployment in the neighborhood of the centers is calculated, e.g.

```
when moving away from Munich,
the rate of unemployment increases
(confidence 86%)
```

In a third step, deviations from the theoretical trends are discovered, e.g.

```
when moving away from Munich in south-west
direction,
then the rate of unemployment is stable
(confidence 97%)
```

The goal of the fourth step is to explain these deviations. E.g. if some community is relatively far away from a center, but is well connected to it by train, the rate of unemployment in this community is not as high as theoretically expected.

We conjecture that this process of trend detection is relevant not only for economic geography but also for a broader class of applications of geographic information systems, e.g. for environmental studies. The steps are summarized as follows and are illustrated by figure 2:

(1) discover centers
i.e. local extrema of some non-spatial attribute(s)

(2) determine the trend of some non-spatial attribute(s)
when moving away from the centers (theoretical as well as observed trend)

(3) discover deviations
of the observed trend from the theoretical trend

(4) explain the deviations
by other spatial objects (e.g. by some infrastructure) in that area and direction.

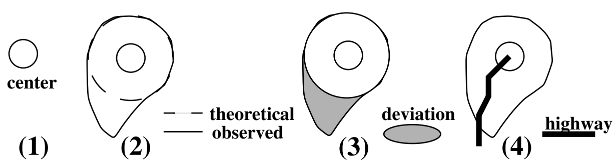

figure 2: The steps of trend detection in a geographic information system

In the following, we present a simple method for detecting spatial trends based on GDBSCAN. GDBSCAN is used to extract density-connected sets of neighboring objects having a similar value of the non-spatial attribute(s). In order to define the similarity on an attribute, we partition its domain into a number of disjoint classes and consider the values in the same class as similar to each other. The sets with the highest or lowest attribute value(s) are most interesting and are called *influence regions,* i.e. the maximal neighborhood of a center having a similar value in the non-spatial attribute(s) as the center itself. Then, the resulting influence region is compared to the circular region representing the theoretical trend to obtain a possible deviation. Different methods may be used to accomplish this comparison, e.g. difference-based or approximation-based methods. A *difference-based method* calculates the difference of both, the observed influence region and the theoretical circular region, thus returning some region indicating the location of a possible deviation (see figure 2). An *approximation-based method* calculates the optimal approximating ellipsoid of the observed influence region. If the two main axes of the ellipsoid significantly differ in length, then the longer one is returned indicating the direction of a deviation.

GDBSCAN can be used to extract the influence regions from an SDBS. We define *NPred* as "intersect(X,Y) $\wedge$ attr-class(X) = attr-class(Y)" and use the default cardinality. Furthermore, we set *MinCard* to 2 in order to exclude sets of less than 2 objects. Figure 3 depicts the influence regions in the Bavaria database wrt. high average income detected by GDBSCAN some of which are discussed in the following. .

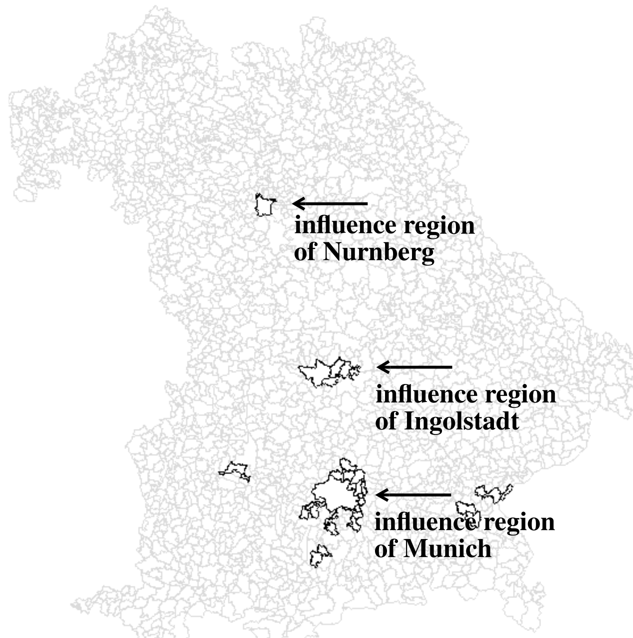

figure 3: Influence regions wrt. average income extracted from the Bavaria database

The influence region of Nurnberg is circle-shaped showing no significant deviation. The influence region of Ingolstadt is elongated indicating a deviation in west-east direction caused by the river Danube traversing Ingolstadt in this direction. Figure 4 shows the approximating ellipsoid together with the significantly longer main axis in west-east direction.

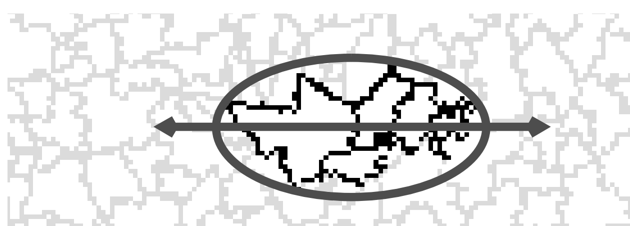

figure 4: Approximating ellipsoid of the influence region of Ingolstadt

The influence region of Munich has four significant deviations from the theoretical region (in the NE, SW, S and SE). Figure 5 illustrates the difference between the observed influence region and the theoretical circular region. These areas coincide with the highways originating from Munich.

.

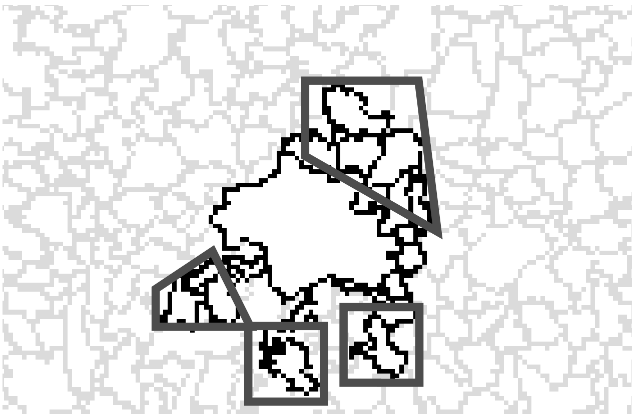

figure 5: Difference between the observed and the theoretical influence region of Munich

4. Conclusions

In this paper, we presented the algorithm GDBSCAN which is a generalized version of DBSCAN (Ester et al. 1996) to cluster spatial objects considering both spatial and non-spatial attributes. GDBSCAN can use any notion of a neighborhood of an object provided that the definition of the neighborhood is based on a binary predicate which is symmetric and reflexive. Instead of the set theoretic cardinality of the neighborhood of an object it can use measures for the "cardinality" of that neighborhood taking into account the non-spatial attributes.

Furthermore, we used GDBSCN to find interesting regions for trend detection in a geographic database on Bavaria. A spatial trend was defined as a pattern of systematic change of one or several non-spatial attributes in 2D or 3D space. We discussed how the discovered knowledge can be useful for economic geographers.

In the future, we will investigate the use of density-connected sets for other KDD tasks such as classification as indicated in section 3. It is also interesting to explore methods for discovering correlations between density-connected sets detected in the same database using different non-spatial attributes.

Acknowledgments

We thank Henning Brockfeld (Institute of Economic Geography, University of Munich) for introducing us into the KDD needs of economic geographers. Sascha Hosse (Institute of Computer Science, University of Munich) did the time consuming job of creating the Bavaria database from the ATKIS 500 and census data.

5. References

Aref W.G., and Samet H. 1991. Optimization Strategies for Spatial Query Processing. *Proc. 17th Int. Conf. on Very Large Data Bases*, 81-90, Barcelona, Spain.

ATKIS 500. 1996. Bavarian State Bureau of Topography and Geodasy, CD-Rom.

Berndt D. J., and Clifford J. 1996. Finding Patterns in Time Series: A Dynamic Programming Approach. In Fayyad U., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (eds.): *Advances in Knowledge Discovery and Data Mining*, AAAI Press / The MIT Press, 229-248.

Beckmann N., Kriegel H.-P., Schneider R., and Seeger B. 1990. The R*-tree: An Efficient and Robust Access Method for Points and Rectangles. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Atlantic City, NJ, 322-331.

Brinkhoff T., Kriegel H.-P., Schneider R., and Seeger B. 1994. Efficient Multi-Step Processing of Spatial Joins. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, Minneapolis, MN, 197-208.

Christaller W. 1968. *Central Places in Southern Germany.* (in German), Wissenschaftliche Buchgesellschaft.

Ester M., Kriegel H.-P., and Xu X. 1995. A Database Interface for Clustering in Large Spatial Databases. *Proc. 1st Int. Conf. on Knowledge Discovery and Data Mining*, Montreal, Canada, AAAI Press, 94-99.

Ester M., Kriegel H-P, Sander J. and Xu X. 1996. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, Oregon, 226-231.

Fayyad U., Piatetsky-Shapiro G., and Smyth P. 1996. Knowledge Discovery and Data Mining: Towards a Unifying Framework. *Proc. 2nd Int. Conf. on Knowledge Discovery and Data Mining*, Portland, Oregon, .82-88.

Gueting R.H. 1994 . An Introduction to Spatial Database Systems. *Special Issue on Spatial Database Systems of the VLDB Journal*, Vol.3, No.4, October 1994.

Koperski K., and Han J. 1995. Discovery of Spatial Association Rules in Geographic Information Databases. *Proc. 4th Int. Symp. on Large Spatial Databases (SSD'95)*, Portland, Maine, August 1995, 47-66.

Knorr E.M., and Ng R.T. 1996. Finding Aggregate Proximity Relationships and Commonalities in Spatial Data Mining. *IEEE Trans. on Knowledge and Data Engineering*, Vol. 8, No. 6, Dec. 1996, 884-897.

Lu W., Han J., and Ooi B.C. 1993. Discovery of General Knowledge in Large Spatial Databases. *Proc. Far East Workshop on Geographic Information Systems*, Singapore, June 1993, 275-289.

Ng R.T., and Han J. 1994. Efficient and Effective Clustering Methods for Spatial Data Mining. *Proc. 20th Int. Conf. on Very Large Data Bases*, Santiago, Chile, 144-155.

Sander J., Ester M., Kriegel H-P, and Xu X. 1997. Density-Based Clustering in Spatial Databases: The Algorithm DBSCAN and its Applications. submitted for journal publication.

Sibson R. 1973. SLINK: an optimally efficient algorithm for the single-link cluster method . *The Computer Journal* 16(1): 30-34.

Zhang T., Ramakrishnan R., and Linvy M. 1996. BIRCH: An Efficient Data Clustering Method for Very Large Databases. *Proc. ACM SIGMOD Int. Conf. on Management of Data*, 103-104.