Spatial Clustering of Structured Objects

Donato Malerba, Annalisa Appice, Antonio Varlaro, and Antonietta Lanza

Dipartimento di Informatica, Università degli Studi di Bari,
via Orabona, 4 - 70126 Bari - Italy
{malerba, appice, varlaro, lanza}@di.uniba.it

Abstract. Clustering is a fundamental task in Spatial Data Mining where data consists of observations for a site (e.g. areal units) descriptive of one or more (spatial) primary units, possibly of different type, collected within the same site boundary. The goal is to group structured objects, i.e. data collected at different sites, such that data inside each cluster models the continuity of socio-economic or geographic environment, while separate clusters model variation over the space. Continuity is evaluated according to the spatial organization arising in data, namely discrete spatial structure, expressing the (spatial) relations between separate sites implicitly defined by their geometrical representation and positioning. Data collected within sites that are (transitively) connected in the discrete spatial structure are clustered together according to the similarity on multi-relational descriptions representing their internal structure. CORSO is a novel spatial data mining method that resorts to a multi-relational approach to learn relational spatial data and exploits the concept of neighborhood to capture relational constraints embedded in the discrete spatial structure. Relational data are expressed in a first-order formalism and similarity among structured objects is computed as degree of matching with respect to a common generalization. The application to real-world spatial data is reported.

1 Introduction

Within both social and environmental sciences much of data is collected in a spatial framework, where data consists of measurements or observations of one or more attributes taken at specific sites which are spatially-referenced. This means that geometrical representation and relative positioning of sites are recorded to express the spatial organization arising in social and environmental data. A simple form of spatially referenced data is point data where observations are taken at fixed point sites of space and represented as triple $\{(x_i, y_i), z_i\}$, such that (x_i, y_i) references the location of a point i with respect to some coordinate system, while z_i is the vector of measured attributes observed at site i. However, operations and activities of private and public institutions generally deal with space in terms of areas (irregular partitions or regular grid) and not points.

Areal data can be represented as point data by identifying each area with its centroid [24], but this is restrictive when observations for an area are descriptive of one or more (spatial) primary units, possibly of different type, collected

S. Kramer and B. Pfahringer (Eds.): ILP 2005, LNAI 3625, pp. 227–245, 2005.
© Springer-Verlag Berlin Heidelberg 2005

within the same area boundary. In this case, data includes both attributes that relate to primary units or areas and attributes that refer to relations between primary units (e.g., contact frequencies between households) and between areal units (e.g., migration rates). Moreover, spatial-referencing poses a further degree of complexity due to the fact that the geometrical representation (point, line or polygon) and the relative positioning of primary units or areal units implicitly define spatial features (properties and relations) of different nature, that is, geometrical (e.g. area, distance), directional (e.g. north, south) and topological (e.g. crosses, on top) features. This relational information may be responsible for the spatial variation among areal units and it is extremely useful in descriptive modeling of different distributions holding for spatial subsets of data. An extra consequence is that observations across space cannot be considered independent due to the spatial continuity of events occurring in the space. Continuity of events over neighbor areas is a consequence of social patterns and environmental constraints that deal with space in terms of regions and allow to identify a mosaic of nearly homogeneous areas in which each patch of the mosaic is demarcated from its neighbors in terms of attributes levels. For instance, the spatial continuity of an environmental phenomenon such as air pollution may depend on the geographical arrangements of pollution sources. As a model for this spatial continuity, the regional concept encourages the analyst to exploit spatial correlation following from the first Law of Geography [22], according to which everything is related to everything else, but near things are more related than distant things. This means that primary units forming areal units of analysis will tend to be essentially identical members of same populations in nearby locations. In this spatial framework, relations among areal units of analysis are expressed in form of relational constraints that represent a discrete spatial structure arising in spatial data, while relations among primary units within an area model the spatial structure of each single areal unit of analysis.

Grouping connected areas to form clusters of homogeneous regions, i.e., spatial clustering, is a fundamental task of Spatial Data Mining. In this paper, we propose to represent the discrete spatial structure as a graph, where nodes are associated with relational descriptions of areal units to be clustered, while links express relational constraints which typically reflect spatial relations such as adjacency. In this way, discontinuity in the graph represents some obstacles in the space. Exploiting this graph-based representation, we present a clustering method, named CORSO (Clustering Of Related Structured Objects), that resorts to a multi-relational approach [2] to model homogeneity over relational structure embedded in spatial data and exploits the concept of graph neighborhood to capture relational constraints embedded in the graph edges. Units associated with (transitively) graph connected nodes are clustered together according to the similarity of their relational descriptions.

The paper is organized as follows. In the next section we discuss some related works. The method is presented in Section 3. Two applications of spatial clustering for topographic map interpretation and geo-referenced census data analysis are reported in Section 4, while conclusions are drawn in Section 5.

2 Background and Motivation

The problem of clustering spatial data has been investigated by some researchers, but while a lot of research has been conducted on detecting spatial clusters from point data, only few works deal with areal data. For instance, Ng and Han [18] have proposed to extend the k-medoid partitioning algorithm [12] to group point data in a set of k clusters. However, the k-medoid partitioning appears well suited only when spatial clusters are of convex shape and similar size, and the number k is reasonably a-priori estimated. Moreover, the method suffers from severe limitations when clustering large spatial dataset [5] due to the complexity of computing distance between medoid points representing each pair of clusters. These efficiency drawbacks are partially alleviated when adopting both proximity and density information to achieve high quality spatial clusters in a sub-quadratic time without requiring the user to a-priori specify the number of clusters [7]. Similarly, DBSCAN [6] exploits density information to efficiently detect clusters of arbitrary shape from point spatial data with noise. The key idea of density-based clustering is that for each point of a cluster, a neighborhood of a given radius has to contain a minimum number (cardinality) of data points. Neighborhood is determined according to the Euclidean distance. However, when observations concern areal units, Euclidean distance may not be appropriate to neighborhood determination. To this purpose, Sander et al. [21] have proposed GDBSCAN that generalizes DBSCAN in order to cluster not only point data but also spatially extended objects (lines or areas) taking into account both spatial and non spatial attributes when defining cardinality. Indeed, GDBSCAN extends the notion of neighborhood to any binary predicate that is symmetric and reflexive (e.g. distance, meet) and imposes a discrete spatial structure on data that guides the clustering detection. This discrete spatial structure can be equivalently modeled as links of a graph, namely neighborhood or proximity graph [23], whose nodes represent the units to be clustered. The graph-based representation of data, that is extensively used in pattern recognition [9], perfectly fits the spatial need of representing the relational constraints among spatial units to be clustered. In this perspective, it is clear that hybrid methods [17] which combine data clustering with graph-partitioning technique have some interesting applications properly in spatial clustering [8].

However even when clustering takes into account relational constraints forming discrete spatial structure, all methods reported above suffer from severe limitations due to the single-table representation [2]. Data to be clustered is represented in a single table (or relation) of a relational database, such that each row (or tuple) corresponds to a single unit of the sample population and the columns correspond to both spatial and a-spatial properties of these units. This representation is clearly inadequate when describing observations concerning several (spatial) primary units, eventually of different types, which are naturally modeled as many data tables as the number of object types and interactions. Some methods for mining clusters on (multi-)relational data have been investigated by resorting to the field of relational data mining. For instance, RDBC [13] forms clusters bottom-up in an agglomerative fashion that uses the distance metric

introduced in [11] and handles relational representations with lists and other functional terms as well. In contrast, C0.5 [1] adopts logical decision trees for clustering purposes by choosing split literals that maximize the distance between two resulting subsets (clusters) of examples. However, differently from RDBC, distance in literal choice is in this case estimated according to a user-provided propositional distance.

Although these relational clustering methods present several interesting aspects, detecting spatial clusters is a more complex task. Indeed, relational clustering methods generally work in the learning from interpretation setting [20] that allows to mine examples and background knowledge stored as Prolog programs exploiting expressiveness of first-order representation during clustering detection. The interpretation corresponding to each example e given the background knowledge BK is here intended as the minimal Herbrand model of $e \wedge BK$ and the implicit assumption is that separate interpretations are independent. This leads to ignore relational constraints eventually relating separate interpretations (e.g. geographic contiguity of areal units). This problem also occurs in graph-based relational learning methods [10] where graphs appear as a flexible representation for relational domains. However, these methods generally continue to work in learning from interpretation settings and thus ignore relations among graphs representing separate examples. In contrast, we propose to combine a graph-based partitioning algorithm with a relational clustering method to mine both relational constraints imposing the discrete spatial structure and relational data representing structured objects (spatial unit) to be clustered.

3 The Method

In a quite general formulation, the problem of clustering structured objects (e.g., complex areal units), which are related by links representing persistent relations between objects (e.g., spatial correlation), can be defined as follows: *Given*: (i) a set of structured objects O, (ii) a background knowledge BK and (iii) a binary relation R expressing links among objects in O; *Find* a set of homogeneous clusters $\mathbf{C} \subseteq \wp(O)$ that is feasible with R.

Each structured object $o_i \in O$ can be described by means of a conjunctive ground formula (conjunction of ground selectors) in a first-order formalism, while background knowledge BK is expressed with first-order clauses that support some qualitative reasoning on O. In both cases, each basic component (i.e., *selector*) is a relational statement in the form $f(t_1, \ldots, t_n) = v$, where f is a function symbol or *descriptor*, t_i are terms (constant or variables) and v is a value taken from the categorical or numerical range of f.

Structured objects are then related by R that is a binary relation $R \subseteq O \times O$ imposing a discrete structure on O. In spatial domains, this relation may be either purely spatial, such as topological relations (e.g. adjacency of regions), distance relations (e.g. two regions are within a given distance), and directional relations (e.g. a region is on south of an other region), or hybrid, which mixes both spatial and non spatial properties (e.g. two regions are connected by a

road). The relation R can be described by the graph $G = (N_O, A_R)$ where N_O is the set of nodes n_i representing each structured object o_i and A_R is the set of arcs $a_{i,j}$ describing links between each pair of nodes $\langle n_i, n_j \rangle$ according to the discrete structure imposed by R. This means that there is an arc from n_i to n_j only if $o_i R o_j$. Let $N_R(n_i)$ be the *R-neighborhood* of a node n_i such that $N_R(n_i) = \{n_j |$ there is an arc linking n_i to n_j in $G\}$, a node n_j is *R-reachable* from n_i if $n_j \in N_R(n_i)$, or $\exists n_h \in N_R(n_i)$ such that n_j is R-reachable from n_h.

According to this graph-based formalization, a clustering $\mathbf{C} \subseteq \wp(O)$ is feasible with the discrete structure imposed by R when each cluster $C \in \mathbf{C}$ is a subgraph G_C of the graph $G(N_O, A_R)$ such that for each pair of nodes $\langle n_i, n_j \rangle$ of G_C, n_i is R-reachable from n_j, or vice-versa. Moreover, the cluster C is homogeneous when it groups structured objects of O sharing a similar relational description according to some similarity criterion.

CORSO integrates a neighborhood-based graph partitioning to obtain clusters which are feasible with R discrete structure and resorts to a multi-relational approach to evaluate similarity among structured objects and form homogeneous clusters. This faces with the spatial issue of modeling spatial continuity of a phenomenon over the space. The top-level description of the method is presented in Algorithm 1. CORSO embeds a saturation step (function *saturate*) to make explicit information that is implicit in data according to the given BK. The key idea

Algorithm 1. Top-level description of CORSO algorithm

```
1: function CORSO(O, BK, R, h − threshold) → CList;
2: CList ← ∅; O_BK ←saturate(O,BK); C ← newCluster( );
3: for each seed ∈ O_BK do
4:    if seed is UNCLASSIFIED then
5:       N_seed ← neighborhood(seed,O_BK,R);
6:       for each o ∈ N_seed do
7:          if o is assigned to a cluster different from C then
8:             N_seed = N_seed/o;
9:          end if
10:      end for
11:      T_seed ← neighborhoodModel(N_seed);
12:      if homogeneity(N_seed, T_seed) ≥ h − threshold then
13:         C.add(seed); seedList ← ∅;
14:         for each o ∈ N_seed do
15:            C.add(o); seedList.add(o);
16:         end for
17:         ⟨C, T_C⟩ ←expandCluster(C,seedList,O_BK,R,T_seed,h − threshold);
18:         CLabel=clusterLabel(T_C); CList.add(⟨C, CLabel⟩); C ← newCluster( );
19:      else
20:         seed ← NOISE;
21:      end if
22:   end if
23: end for
24: return CList;
```

is to exploit the R-neighborhood construction and build clusters feasible with R-discrete structure by merging partially overlapping homogeneous neighborhood units. Cluster construction starts with an empty cluster ($C \leftarrow newCluster()$) and chooses an arbitrary node *seed* from G. The R-neighborhood N_{seed} of the node *seed* is then built according to G discrete structure (function *neighborhood*) and the first-order theory T_{seed} is associated to it. T_{seed} is built as a generalization of the objects falling in N_{seed} (function *neighborhoodModel*). When the neighborhood is estimated to be an homogeneous set (function *homogeneity*), cluster C is grown with the structured objects enclosed in N_{seed} whcih are not yet assigned to any cluster. The cluster C is then iteratively expanded by merging the R-neighborhoods of each node of C (neighborhood expansion) when these neighborhoods result in homogeneous sets with respect to current cluster model T_C (see Algorithm 2.). T_C is obtained as the set of first-order theories generalizing the neighborhoods merged in C. It is noteworthy that when a new R-neighborhood is built to be merged in C, all the objects which are already classified into a cluster different from C are removed from the neighborhood. When the current cluster cannot be further expanded it is labeled with $CLabel$ and an unclassified seed node for a new cluster is chosen from G until all objects are classified. $CLabel$ is obtained by T_C (function *labelCluster*) to compactly describe C.

Algorithm 2. Expand current cluster by merging homogeneous neighborhood

```
    function expandCluster(C, seedList, O_BK, R, T_C, h − threshold) → ⟨C, T_C⟩;
 2: while (seedList is not empty) do
      seed ← seedList.first(); N_seed ← neighborhood(seed,O_BK,R);
 4:   for each o ∈ N_seed do
        if o is assigned to a cluster different from C then
 6:       N_seed = N_seed/o;
        end if
 8:   end for
      T_seed ← neighborhoodModel(N_seed);
10:   if homogeneity(N_seed, {T_C, T_seed}) ≥ h − threshold then
        for each o ∈ N_seed do
12:       C.add(o); seedList.add(o);
        end for
14:     seedList.remove(seed); T_C ← T_C ∪ T_seed;
      end if
16: end while
    return ⟨C, T_C⟩;
```

This is different from spatial clustering performed by GDBSCAN, although both methods share the neighborhood-based cluster construction. Indeed, GDBSCAN retrieves all objects density-reachable from an arbitrary core object by building successive neighborhoods and checks density within a neighborhood by ignoring the cluster. This yields a density-connected set, where density is

efficiently estimated independently from the neighborhoods already merged in forming the current cluster. However, this approach may lead to merge connected neighborhoods sharing some objects but modeling different phenomena. Moreover, GDBSCAN computes density within each neighborhood according to a weighted cardinality function (e.g. aggregation of non spatial values) that assumes single table data representation. CORSO overcomes these limitations by computing density within a neighborhood in terms of degree of similarity among all relationally structured objects falling in the neighborhood with respect to the model of the entire cluster currently built. In particular, following the suggestion given in [16], we evaluate homogeneity within a neighborhood N_{seed} to be added to the cluster C as the average degree of matching between objects of N_{seed} and the cluster model $\{T_C, T_{seed}\}$. Details on cluster model determination, neighborhood homogeneity estimation and cluster labeling are reported below.

3.1 Cluster Model Generation

Let C be the cluster currently built by merging w neighborhood sets $N_1, \ldots, N_w$, we assume that the cluster model T_C is a set of first-order theories $\{T_1, \ldots, T_w\}$ for the concept C where T_i is a model for the neighborhood set N_i. More precisely, T_i is a set of first-order clauses: $T_i : \{cluster(X) = c \leftarrow H_{i1}, \ldots, cluster(X) = c \leftarrow H_{iz}\}$, where each H_{ij} is a conjunctive formula describing a sub-structure shared by one or more objects in N_i and $\forall o_i \in N_i$, $BK \cup T_i \models o_i$. Such model can be learned by resorting to the ILP system ATRE [14] that adopts a separate-and-conquer search strategy to learn a model of structured objects from a set of training examples and eventually counter-examples. In this context, ATRE learns a model for each neighborhood set without considering any counter-examples. The search of a model starts with the most general clause, that is,$cluster(X) = c \leftarrow$, and proceeds top-down by adding selectors (literals) to the body according to some preference criteria (e.g. number of objects covered or number of literals).

Selectors involving both numerical and categorical descriptors are handled in the same way, that is, they have to comply with the property of linkedness and are sorted according to preference criteria. The only difference is that selectors involving numerical descriptors are generalized by computing the closed interval that best covers positive examples and eventually discriminates from contour-examples, while selectors involving categorical descriptors with same function value are generalized by simply turning all ground arguments into corresponding variables without changing the corresponding function value.

3.2 Neighborhood Homogeneity Estimation

The homogeneity of a neighborhood set N to be added to the cluster C is computed as follows:

$$h(N, T_{C \cup N}) = \frac{1}{\#N} \sum_i h(o_i, T_{C \cup N}) = \frac{1}{\#N} \sum_i \frac{1}{w+1} \sum_j h(o_i, T_j), \qquad (1)$$

where $\#N$ is the cardinality of the neighborhood set N and $T_{C \cup N}$ is the cluster model of $C \cup N$ formed by both $\{T_1, \ldots, T_w\}$, i.e., the model of C and T_{w+1}, i.e., the model of N built as explained above. Since $T_j = H_{1j}, \ldots, H_{zj}$ $(z \geq 1)$ with each H_{ij} a conjunctive formula in first-order formalism, we assume that:

$$h(o_i, T_j) = \frac{1}{z} \sum_i fm(o_i, H_{ij}), \tag{2}$$

where fm is a function returning the degree of matching of an object $o_i \in N$ against the conjunctive formula H_{ij}. In this way, the definition of homogeneity of a neighborhood set $N = \{o_1, \ldots, o_n\}$ with respect to some logical theory $T_{C \cup N}$ is closely related to the problem of comparing (matching) the conjunctive formula f_i representing an object $o_i \in N$[1] with a conjunctive formula H_{ij} forming the model T_j in order to discover likenesses or differences [19]. This is a directional similarity judgment involving a *referent* R, that is the description or prototype of a class (cluster model) and a *subject* S that is the description of an instance of a class (object to be clustered). In the classical matching paradigm, the matching of S against R corresponds to compare them just for equality. In particular, when both S and R are conjunctive formulas in first-order formalism, matching S against R corresponds to check the existence of a substitution θ for the variables in R such that $S = \theta(R)$. This last condition is generally weakened by requiring that $S \Rightarrow \theta(R)$, where $\Rightarrow$ is the logical implication. However, the requirement of equality, even in terms of logical implication, is restrictive in presence of noise or variability of the phenomenon described by the referent of matching. This makes necessary to rely on a flexible definition of matching that aims at comparing two descriptions and identifying their similarities rather than equalities. The result of such a flexible matching is a number in the interval $[0, 1]$ that is the probability of precisely matching S against R, provided that some change described by θ is possibly made in the description R.

The problem of computing flexible matching to compare structures is not novel. Esposito et al. [4] have formalized a computation schema for flexible matching on formulas in first-order formalism whose basic components (selectors) are the relational statements, that is, $f_i(t_1, \ldots, t_n) = v$, which are combined by applying different operators such as conjunction $(\wedge)$ or disjunction $(\vee)$ operator. In this work, we focus on the computation of flexible matching $fm(S, R)$ when both S and R are described by conjunctive formulas and $fm(S, R)$ looks for the substitution θ returning the best matching of S against R, as:

$$fm(S, R) = \max_{\theta} \prod_{i=1,\ldots,k} fm_{\theta}(S, r_i). \tag{3}$$

The optimal θ that maximizes the above conditional probability is here searched by adopting the branch and bound algorithm that expands the least cost partial path by performing quickly on average [4]. According to this formulation, fm_{θ}

[1] The conjunctive formula f_i is here intended as the description of $o_i \in N$ saturated according to the BK.

denotes the flexible matching with the tie of the substitution fixed by θ computed on each single selector $r_i \equiv f_{r_i}(t_{r_1}, \ldots, t_{r_n}) = v_{r_i}$ of the referent R where f_{r_i} is a function descriptor with either numerical (e.g. area or distance) or categorical (e.g. intersect) range. In the former case the function value v_{r_i} is an interval value ($v_{r_i} \equiv [a, b]$), while in the latter case v_{r_i} is a subset of values ($v_{r_i} \equiv \{v_1, \ldots, v_M\}$) from the range of f_{r_i}. This faces with a referent R that is obtained by generalizing a neighborhood of objects in O. Conversely for the subject S, that is, the description of a single object $o \in O$, the function value w_{s_j} assigned to each selector $s_j \equiv f_{s_j}(t_{s_1}, \ldots, t_{s_n}) = w_{s_j}$ is an exactly known single value from the range of f_{s_j}. In this context, the flexible matching $fm_\theta(S, r_i)$ evaluates the degree of similarity $fm(s_j, \theta(r_i))$ between $\theta(r_i)$ and the corresponding selector s_j in the subject S such that both r_i and s_j have the same function descriptor $f_r = f_s$ and for each pair of terms $\langle t_{r_i}, t_{s_i} \rangle$, $\theta(t_{r_i}) = t_{s_i}$. More precisely,

$$fm(s_j, \theta(r_i)) = fm(w_{s_j}, v_{r_i}) = \max_{v \in v_{r_i}} P(equal(w_{s_j}, v)). \tag{4}$$

The probability of the event $equal(w_{s_j}, v)$ is then defined as the probability that an observed w_{s_j} is a distortion of v, that is:

$$P(equal(w_{s_j}, v)) = P(\delta(X, v) \geq \delta(w_{s_j}, v)) \tag{5}$$

where X is a random variable assuming value in the domain D representing the range of f_r while *delta* is a distance measure. The computation of $P(equal(w_{s_j}, v))$ clearly depends on the probability density function of X. For categorical descriptors, that is, D is a discrete set with cardinality $\#D$, it has be proved [4] that:

$$P(equal(w, v)) = \begin{cases} 1 \text{ if } w_{s_j} = v \\ \#D - 1/\#D \text{ otherwise} \end{cases} \tag{6}$$

when X is assumed to have a uniform probability distribution on D and $\delta(x, y) = 0$ if $x = y$, 1 otherwise. Although similar results have been reported for both linear non numerical and tree-structured domains, no result appears for numerical domains. Therefore, we have extended definitions reported in [4] to make flexible matching able to deal with numerical descriptors and we have proved that:

$$fm(c, [a, b]) = \begin{cases} 1 & \text{if } a \leq c \leq b \\ 1 - 2(a - c)/(\beta - \alpha) & \text{if } c < a \wedge 2a - c \leq \beta \\ (c - \alpha)/(\beta - \alpha) & \text{if } c < a \wedge 2a - c > \beta \\ (\beta - c)/(\beta - \alpha) & \text{if } c > b \wedge 2b - c < \alpha \\ 1 - 2(c - b)/(\beta - \alpha) & \text{if } c > b \wedge 2b - c \geq \alpha \end{cases} \tag{7}$$

by assuming that X has uniform distribution on D and $\delta(x, y) = |x - y|$. A proof of formula 7 is reported in the Appendix A of this paper.

3.3 Cluster Labeling

A cluster C can be naturally labeled with T_C that is the set of first-order clauses obtained from the generalization of neighborhoods merged in C. Each first-order

clause is in the form $C \leftarrow s_1, \ldots, s_n$, where C represents the cluster label and each s_i denotes a selector in the form $f_i(t_{i_1}, \ldots, t_{i_l}) = v_i$. In this formalization, two selectors $s_1 : f_1(t_{1_1}, \ldots, t_{1_l}) = v_1$ and $s_2 : f_2(t_{2_1}, \ldots, t_{2_l}) = v_2$ are comparable according to some substitution θ when they involve the same descriptor ($f_1 = f_2 = f$) and each pair of terms $\langle t_{1_i}, t_{2_i} \rangle$ is unifiable according to θ, i.e., $t_{1_i}\theta = t_{2_i}\theta = t_i$ $(\forall i = 1 \ldots l)$. In this case, the selector $s : f(t_1, \ldots, t_l) = \{v_1\} \cup \{v_2\}$ is intended as a generalization for both s_1 and s_2. In particular, the selectors s_1 and s_2 are equal when they are comparable and $v_1 = v_2 = v$ such that the generalization of s_1 and s_2 is built as $s : f(t_1, \ldots, t_l) = v$. Similarly, the selector s_1 (s_2) is contained in the selector s_2 (s_1) when they are comparable and $v_1 \subseteq v_2$ ($v_2 \subseteq v_1$), while the generalization s is $f(t_1, \ldots, t_l) = v_2$ ($f(t_1, \ldots, t_l) = v_1$). Note that equality of selectors implies containment, but not vice-versa. Similarly, the first-order clauses $H_1 : C \leftarrow s_{1_1}, \ldots, s_{1_n}$ and $H_2 : C \leftarrow s_{2_1}, \ldots, s_{2_n}$ are comparable according to some substitution θ when each pair of selectors $\langle s_{1_i}, s_{2_i} \rangle$ is comparable according to θ. Hence, H_1 is equal (contained) to H_2 when s_{1_i} is equal (contained) to s_{2_i} for each $i = 1, \ldots, n$. In both these cases (equality and containment condition), the pair of first-order clauses H_1, H_2 can be replaced without lost of information with the first-order clause H that is the generalization of H_1, H_2 built by substituting each pair of comparable selectors $\langle s_{1_i}, s_{2_i} \rangle \in \langle H_1, H_2 \rangle$ with the generalization obtained as stated above. This suggests the idea of merging a pair of comparable first-order clauses H_1, H_2 in a single clause H by preserving the equivalence of coverage, that is: (i) for each structured object o with $H_1, H_2, BK \models o$ then $H, BK \models o$ and vice-versa, (ii) for each structured object o with $H_1, H_2, BK \not\models o$ then $H, BK \not\models o$ and vice-versa, where BK is a set of first-order clauses. The equivalence of coverage between $\{H_1, H_2\}$ and H is obviously guaranteed when H_1 is either equal or contained in H_2 or vice-versa, but this equivalence cannot be guaranteed when H_1 and H_2 are comparable first-order clauses but neither equality condition nor containment condition are satisfied.

Example 1: Let us consider the pair of comparable first-order clauses:

$H_1 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..10], type(X_2) = street$

$H_2 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [3..7], type(X_2) = river$

where neither H_1 is equal to H_2 nor $H_1(H_2)$ is contained in $H_2(H_1)$. The first-order clause obtained by generalizing pairs of comparable selectors in both H_1 and H_2, is $H : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [3..10], type(X_2) = \{street, river\}$, where $H \models o$ with $o : distance(X_1, X_2) = 3 \wedge type(X_2) = street$, but neither $H_1 \models o$ nor $H_2 \models o$.

The requirement of equality between H_1 and H_2 can be relaxed while preserving equivalence of coverage with respect to the generalization H. Indeed, when

$H_1 : C \leftarrow s_1(\_) = v_1, \ldots s_k(\_) = v_k, \ldots, s_n(\_) = v_n$

$H_2 : C \leftarrow s_1(\_) = v_1, \ldots s_k(\_) = w_k, \ldots, s_n(\_) = v_n$

are comparable first-order clauses differing only in the function value of a single selector (i.e. s_k), the first-order clause:

$H : C \leftarrow s_1(\_) = v_1, \ldots s_k(\_) = \{v_k\} \cup \{w_k\}, \ldots, s_n(\_) = v_n$
continues to preserve the equivalence of coverage with $\{H_1, H_2\}$.

Example 2: Let us consider the pair of comparable first-order clauses:

$H_1 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [3..7], type(X_2) = street,$
$length(X_2) = [3, 5]$
$H_2 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [3..7], type(X_2) = street,$
$length(X_2) = [7, 10]$

which differ only in the value of a single selector (length), the first-order clause obtained by generalizing the pairs of comparable selectors in both H_1 and H_2 is:

$H : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [3..7], type(X_2) = street,$
$length(X_2) = [3, 5] \cup [7, 10]$

that is equivalent in coverage to the pair $\{H_1, H_2\}$.

Following this idea, it is possible to compactly describe the cluster theory T_C finally associated to a cluster C by iteratively replacing pairs of comparable first-order clauses H_1, H_2 with the generalization H, when H results equivalent in coverage to $\{H_1, H_2\}$ (see Algorithm 3.).

Algorithm 3. Build a compact theory to describe a cluster C

```
1: function clusterLabel(T_C) → T'_C;
2: T'_C ← ∅
3: merge ← false;
4: while T_C is not empty do
5:    H is a first-order clause in T_C;
6:    T_C = T_C/H;
7:    for each H' ∈ T_C do
8:       if H and H' are generalizable without lost of information then
9:          H = generalize(H,H'); T_C = T_C/H'; merge = true;
10:      end if
11:   end for
12:   T'_C = T'_C ∪ H;
13: end while
14: if merge is true then
15:   T'_C ←clusterLabel(T'_C);
16: end if
17: return T'_C;
```

Example 3: Let us consider T_C that is the set of first-order clauses including:

$H_1 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..10], color(X_2) = red$
$H_2 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..6], color(X_2) = blue$
$H_3 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..10], color(X_2) = blue$
$H_4 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [6..10], area(X_2) in [30..40]$

T_C can be transformed in the set of first-order clauses:

$H'_1 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..10], color(X_2) = \{red, blue\}$
$H'_2 : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [6..10], area(X_2) in [30..40]$

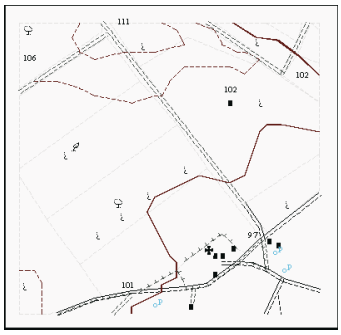
where H_1' results by firstly merging H_1 and H_3, which are comparable and differ only in the function value of a selector ($color(X_2) = red$ vs $color(X_2) = blue$), and obtaining $H_{13} : cluster(X_1) = c \leftarrow distance(X_1, X_2) = [5..10], color(X_2) = \{red, blue\}$ and then merging H_{13} and H_2 since H_2 is contained in H_{13}.

4 The Application: Two Case Studies

In this section, we describe the application of CORSO to two distinct real-world problems, namely topographic map interpretation and geo-referenced census data analysis. In the former problem, a topographic map is treated as a grid of square cells of same size, according to a hybrid tessellation-topological model such that adjacency among cells allows map-reading from a cell to one of its neighbors in the map. For each cell, geographical data is represented as humans perceive it in reality, that is, geometric (or physical) representation and thematic (or logical) representation. Geometric representation describes the geographical objects by means of the most appropriate physical entity (point, line or region), while thematic representation expresses the semantics of geographical objects (e.g., hydrography, vegetation, transportation network and so on), independently of their physical representation. Spatial clustering aims at identifying a mosaic of nearly homogeneous clusters (areas) including adjacent cells in the map such that geographical data inside each cluster properly models the spatial continuity of some morphological environment within the cluster region, while separate clusters model spatial variation over the entire space. In the second problem, the goal is to perform a joint analysis of both socio-economic factors represented in census data and geographical factors represented in topographic maps to support a good public policy. In this case, spatial objects are territorial units for which census data are collected as well as entities of geographical layers such as urban and wood areas. Spatial partitioning of CORSO is compared with the first-order clustering performed with logical decision trees [1], which are able to manage relational structure of spatial objects but ignore relations imposed with discrete spatial structure. The empirical comparison with GDBSCAN was not possible since the system is not publicly available. However, CORSO clearly improves GDBSAN clustering that is not able to manage complex structure of spatial data. In both applications, running time of CORSO refers to execution performed on a 2 Ghz IBM notebook with 256 Mb of RAM.

4.1 Topographic Map Interpretation

In this study we discuss two real-world applications of spatial clustering to characterize spatial continuity of some morphological elements over the topographic map of the Apulia region in Italy. The territory considered in this application covers 45 km^2 from the zone of Canosa in Apulia. The examined area is segmented into square areal units of 1 Km^2 each. Thus, the problem of recognizing spatial continuity of some morphological elements in the map is reformulated as the problem of grouping adjacent cells resulting in a morphologically homogeneous area, that is, a problem of clustering spatial objects according to the



contain(c, f2) = true, …,
contain(f, f70) = true,
type_of(c) = cell, …,
type_of(f4) = vegetation,…,
subtype_of(f2) = grapewine,…,
subtype_of(f7) = cart_track_road,…,
part_of(f4, x4),
part_of(f7, x5), part_of(f7_x6),…,
extension(x7) = 111.018,…,
extension(x33) = 1104.74,
line_to_line(x7, x68) = almost_parallel, …,
point_to_region(x4, x21) = inside,
point_to_region(x4, x18) = outside,…,
line_to_region(x8, x27) = adjacent, …

Fig. 1. First-order description of a cell extracted from topographic chart of Apulia

discrete spatial structure imposed by the relation of "adjacency" among cells. Since several geographical objects, eventually belonging to different layers (e.g., almond tree, olive tree, font, street, etc) are collected within each cell, we apply algorithms derived from geometrical and topological reasoning [15] to obtain cell descriptions in first-order formalism (see Figure 1). For this task, we consider descriptions including spatial descriptors encompassing geometrical properties ($area$, $extension$) and topological relations ($regionToRegion$, $lineToLine$, $pointToRegion$) as well as non spatial descriptors ($typeOf$, $subtypeOf$). The descriptor $partOf$ is used to define the physical structure of a logical object. An example is: $typeOf(f_1) = font \wedge partOf(f_1, x_1) = true$, where f_1 denotes a font which is physically represented by a point referred with the constant x_1. Each cell is here described by a conjunction of 946,866 ground selectors in average. To support some qualitative reasoning, a spatial background knowledge (BK) is expressed in form of clauses. An example of BK we use in this task is:

$fontToParcel(Font, Culture) = Relation \leftarrow typeOf(Font) = font,$
$partOf(Font, Point) = true, typeOf(Parcel) = parcel,$
$partOf(Parcel, Region) = true, pointToRegion(Point, Region) = Relation$

that allows to move from a physical to a logical level in describing the topological relation between the point that physically represents the font and the region that physically represents the culture and that are, respectively, referred to as the variables $Font$ and $Culture$. The specific goal of this study is to model the spatial continuity of some morphological environment (e.g. cultivation setting) within adjacent cells over the map. This means that each cluster covers a contiguous area over the map where it is possible to observe some specific environment that does not occur in adjacent cells not yet assigned to any cluster. It is noteworthy that granularity of partitioning changes by varying homogeneity threshold (see Figure 2). In particular, when $h - threshold = 0.95$, CORSO clusters adjacent cells in five regions in 1821 secs. Each cluster is compactly labeled as follows:

$C_1 : cluster(X_1) = c_1 \leftarrow containAlmondTree(X_1, X_2) = \{true\},$
$cultivationToCulture(X_2, X_3) = \{outside\},$
$areaCulture(X_3) = [328..420112], fontToCulture(X_4, X_3) = \{outside\}.$

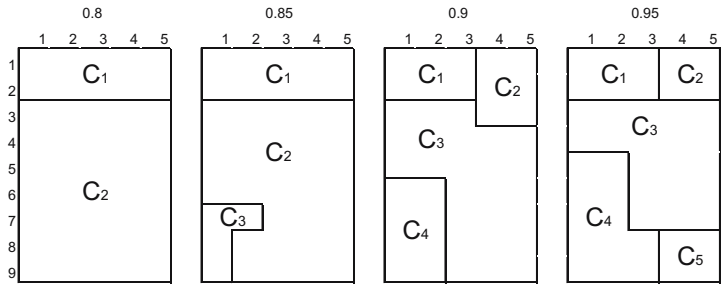


Fig. 2. Spatial clusters detected on map data from the zone of Canosa by varying $h - threshold$ value in {0.8,0.85,0.9,0.95}

$C_2 : cluster(X_1) = c_2 \leftarrow containAlmondTree(X_1, X_2) = \{true\}$,
$cultivationToCulture(X_2, X_3) = \{inside\}, areaCulture(X_3) = [13550..187525], areaCulture(X_3) = [13550..187525]$,
$cultivationToCulture(X_2, X_4) \in \{outside\}$.
$C_3 : cluster(X_1) = c_3 \leftarrow containGrapevine(X_1, X_2) = \{true\}$,
$cultivationToCulture(X_2, X_3) = \{inside\}, areaCulture(X_3) = [13550..212675], cultivationToCulture(X_2, X_4) = \{outside\}$.
$cluster(X_1) = c_3 \leftarrow containGrapevine(X_1, X_2) = \{true\}$,
$cultivationToCulture(X_2, X_3) = \{outside\}, areaCulture(X_3) = [150..212675], cultivationToCulture(X_2, X_4) = \{outside, inside\}$.
$C_4 : cluster(X_1) = c_4 \leftarrow containStreet(X_1, X_2) = \{true\}$
$streetToCulture(X_2, X_3) = \{adjacent\}, areaCulture(X_3) = [620..230326], cultureToCulture(X_3, X_4) = \{outside, inside\}$.
$C_5 : cluster(X_1) = c_5 \leftarrow containOliveTree(X_1, X_2) = true$,
$cultivationToCulture(X_2, X_3) \in \{outside\}, areaCulture(X_3) \in [620..144787], oliviToParcel(X_2, X_4) = \{outside\}$.

Notice that each detected cluster effectively includes adjacent cells sharing a similar morphological environment, while separate clusters describe quite different environments. Conversely, the logical decision tree mined on the same data divides the territory under analysis in twenty different partitions where it is difficult to recognize the continuity of any morphology phenomenon.

4.2 Geo-referenced Census Data Analysis

In this application, we consider both census and digital map data concerning North West England (NWE) area that is decomposed into censual sections or wards for a total of 1011 wards. Census data is available at ward level and provides some measures of deprivation level in the ward according to index scores that combine information provided by 1998 Census. We consider Jarman Underprivileged Area Score that is designed to measure the need for primary care, the indices developed by Townsend and Carstairs that is used in health-related analysis, and the Department of the Environment's Index (DoE) that is used

in targeting urban regeneration funds. The higher the index value the more deprived a ward is. Spatial analysis of deprivation distribution is enabled by the availability of vectorized boundaries of the 1010 census wards as well as by other Ordnance Survey digital maps of NWE, where several interesting layers are found, namely urban zones (including 384 large urban areas and 2232 small urban areas) and wood zones (including 859 woods). In particular, we focus attention on investigating continuity of socio-economic deprivation joined to geographical factors represented in linked topographic maps.

Both ward-referenced census data and map data are stored in an Object-Relational spatial database, i.e., Oracle Spatial 9i database, as a set of spatial tables, one for each layer. Each spatial table includes a geometry attribute that allows storing the geometrical representation (i.e. urban and wood zones are described by lines while wards are described by polygons) and the positioning of a spatial object with respect to some reference system. We adopt a topological algorithm based on the 9-intersection model [3] to detect both adjacency relation between NWE wards (i.e. wards which share some boundary) and overlapping relation between wards and urban areas (or woods). The former imposes a discrete spatial structure over NWE wards such that only adjacent wards may be grouped in the same cluster while the latter contributes to define the spatial structure embedded in each ward not only in terms of observed values of deprivation scores but also extension of urban areas and/or woods overlapping each ward. No BK is defined for this problem.

Granularity of partitioning changes when varying the value of $h-threshold$, that is, CORSO detects 79 clusters with $h-threshold = 0.80$, 89 clusters with $h-threshold = 0.85$, 122 clusters with $h-threshold = 0.90$ and 163 clusters with $h-threshold = 0.95$. In particular, when $h-threshold = 0.95$, CORSO clusters NWE area in 2160 secs and identifies adjacent regions modeling differently relational patterns involving deprivation and geographical environment. For instance, by analyzing these spatial clusters, we discover three adjacent areas, namely C_1, C_2 and C_3 compactly labeled as follows:

$C_1 : cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-4.7..-0.6],$
$doe(X_1) = [-12.4..2.7], carstairs(X_1) = [-4.5..-0.9],$
$jarman(X_1) = [-32.7..7.5], overlapped\_by\_wood(X1, X2) = true.$
$cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-5.4..-2.3],$
$doe(X_1) = [-10.9..-0.5], carstairs(X_1) = [-4.2..-1.6],$
$jarman(X_1) = [-22.8..0.6], overlapped\_by\_wood(X1, X2) = true.$
$cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-5.4..-3.2],$
$doe(X_1) = [-8.8..-2.1], carstairs(X_1) = [-4.4..-2.5],$
$jarman(X_1) = [-22.8..-2.4], overlapped\_by\_wood(X1, X2) = true.$
$C_2 : cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-2.0..0.6],$
$doe(X_1) = [-4.2..1.6], carstairs(X_1) = [-2.6..2.1],$
$jarman(X_1) = [-9.7..8.8], overlapped\_by\_largeUrbArea(X1, X2) = true.$
$cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-2.7..2.8],$
$doe(X_1) = [-4.2..4.0], carstairs(X_1) = [-2.2..2.7],$
$jarman(X_1) = [-8.8..21.3], overlapped\_by\_largeUrbArea(X1, X2) = true$

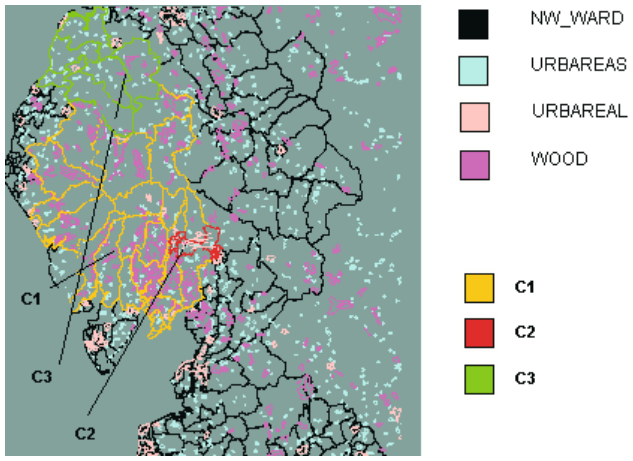


Fig. 3. Spatial clusters detected on NWE with $h - threshold = 0.95$

$C_3 : cluster(X_1) = c_1 \leftarrow townsend(X_1) = [-3.4..0.4],$
$doe(X_1) = [-8.2.. - 0.2], carstairs(X_1) = [-3.7..0.6],$
$jarman(X_1) = [-27.7.. - 1.5],$
$overlapped\_by\_smallUrbArea(X1, X2) = true.$

C_1, C_2 and C_3 cover adjacent areas with quite similar range value for deprivation indexes but C_1 models the presence of woods while C_2 and C_3 model the presence of small urban areas and large urban areas, respectively. Discontinuity of geographical environments modeled by these clusters is confirmed by visualizing map data about the area (see Figure 3).

The logical decision tree mined on the same data discovers 58 different clusters. Clusters are built by minimizing the distance among relational descriptions of wards. However, the discrete structure imposed by the adjacency relation is ignored. Hence, wards which are not connected in the graph imposed by the adjacency relation are clustered together.

5 Conclusions

This paper presents a novel approach to discover clusters from structured spatial data taking into account relational constraints (e.g. spatial correlation) forming the discrete spatial structure. We represent this discrete spatial structure as a graph such that the concept of graph neighborhood is exploited to capture relational constraints embedded in the graph edges. Moreover, we resort to a relational approach to mine data scattered in multiple relations describing the structure that is naturally embedded in spatial data. As a consequence, only spatial units associated with (transitively) graph connected nodes can be clustered together according to judgment of similarity on relational descriptions representing their internal (spatial) structure. As future work, we intend to integrate

CORSO in a spatial data mining system that is able to extract both the spatial structure and the structure of spatial objects from a spatial database, cluster these spatial objects coherently with the extracted spatial structure and visualize discovered clusters. We also plan to employ CORSO for air pollution analysis.

Acknowledgment

The work presented in this paper is partial fulfillment of the research objective set by the ATENEO-2004 project on "Metodi di Data Mining Multi-relazionale per la scoperta di conoscenza in basi di dati".

References

1. L. De Raedt and H. Blockeel. Using logical decision trees for clustering. In S. Džeroski and N. Lavrač, editors, *Inductive Logic Programming, 7th International Workshop*, volume 1297, pages 133–140. Springer-Verlag, 1997.
2. S. Džeroski and N. Lavrač. *Relational Data Mining*. Springer-Verlag, 2001.
3. M. Egenhofer. Reasoning about binary topological relations. In *Symposium on Large Spatial Databases*, pages 143–160, 1991.
4. F. Esposito, D. Malerba, and G. Semeraro. Flexible matching for noisy structural descriptions. In *International Joint Conference on Artificial Intelligence*, pages 658–664, 1991.
5. M. Ester, H. P. Kriegel, and J. Sander. Algorithms and applications for spatial data mining. *Geographic Data Mining and Knowledge Discovery*, 5(6), 2001.
6. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Knowledge Discovery in Databases*, pages 226–231, 1996.
7. V. Estivill-Castro and M. E. Houle. Robust distance-based clustering with applications to spatial data mining. *Algorithmica*, 30(2):216–242, 2001.
8. V. Estivill-Castro and I. Lee. Fast spatial clustering with different metrics and in the presence of obstacles. In *International Symposium on Advances in geographic information systems*, pages 142–147. ACM Press, 2001.
9. E. Hancock and M. Vento. *Graph Based Representations in Pattern Recognitions*. Springer-Verlag, 2003.
10. L. Holder and D. Cook. Graph-based relational learning: Current and future directions. *SIGKDD Explorations Special Issues on Multi-Relational Data Mining*, 5(1):90–93, 2003.
11. T. Horvath, S. Wrobel, and U. Bohnebeck. Relational instance-based learning with lists and terms. *Machine Learning*, 43(1/2):53–80, 2001.
12. L. Kaufmann and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis*. John Wiley, 1990.
13. M. Kirsten and S. Wrobel. Relational distance-based clustering. In *Inductive Logic Programming, 8th International Conference*, volume 1446, pages 261–270. Springer-Verlag, 1998.
14. D. Malerba. Learning recursive theories in the normal ilp setting. *Fundamenta Informaticae*, 57(1):39–77, 2003.

15. D. Malerba, F. Esposito, A. Lanza, F. A. Lisi, and A. Appice. Empowering a gis with inductive learning capabilities: The case of ingens. *Journal of Computers, Environment and Urban Systems, Elsevier Science*, 27:265–281, 2003.
16. D. Mavroeidis and P. Flach. Improved distances for structured data. In T. Horváth and A. Yamamoto, editors, *Inductive Logic Programming, 13th International Conference*, volume 2835, pages 251–268. Springer-Verlag, 2003.
17. A. M. Neville, J. and J. D. Clustering relational data using attribute and link information. In *Text Mining and Link Analysis Workshop, 18th International Joint Conference on Artificial Intelligence*, 2003.
18. R. T. Ng and J. Han. Efficient and effective clustering methods for spatial data mining. In J. Bocca, M. Jarke, and C. Zaniolo, editors, *Very Large Data Bases, 20th International Conference*, pages 144–155. Morgan Kaufmann Publishers, 1994.
19. D. Patterson. *Introduction to Artificial Intelligence and expert systems.* Prentice-Hall, 1991.
20. L. D. Raedt and S. Dzeroski. First-order jk-clausal theories are pac-learnable. *Artificial Intelligence*, 70(1-2):375–392, 1994.
21. J. Sander, E. Martin, H.-P. Kriegel, and X. Xu. Density-based clustering in spatial databases: The algorithm gdbscan and its applications. *Data Mining and Knowledge Discovery*, 2(2):169–194, 1998.
22. W. Tobler. Cellular geography. In S. Gale and G.Olsson, editors, *Philosophy in Geography*, 1979.
23. G. Toussaint. Some unsolved problems on proximity graphs. In D. Dearholt and F. Harary, editors, *First Workshop on Proximity Graphs*, 1991.
24. M. Visvalingam. Operational definitions of area based social indicators. *Environment and Planning*, A(15):831–839, 1983.

A Appendix

Let us recall definitions (4) and (5) and apply them to numerical case. We have:

$$fm(c,[a,b]) = \max_{v\in[a,b]} P(equal(c,v)) = \max_{v\in[a,b]} P(\delta(X,v) \geq \delta(c,v))$$

By assuming that X has a uniform distribution on domain $D = [\alpha, \beta]$ with density function $f_D(x) = 1/(\beta - \alpha), \forall x \in D$ and fixing $\delta(x,y) = |x - y|$, $P(\delta(X,v) \geq \delta(c,v))$ can be rewritten as $P(|X - v| \geq |c - v|)$ that is maximized when minimizing $|c - v|$.
If $a \leq c \leq b$ then $\max_{v\in[a,b]} P(|X - v| \geq |c - v|) = P(|X - v| \geq |c - c|) = 1$.
If $c < a$ then $\max_{v\in[a,b]} P(|X - v| \geq |c - v|)$ is written as $\max_{v\in[a,b]} P(|X - v| \geq v - c)$.
Since the maximum of $P(|X - v| \geq v - c)$ is obtained for $v = a$, we have that $\max_{v\in[a,b]} P(|X - v| \geq v - c) = P(|X - a| \geq a - c) = P(X - a \geq a - c) + P(X - a \leq c - a) = P(X \geq 2a - c) + P(X \leq c)$ where:

1. $P(X \geq 2a - c) = \int_{\beta}^{2a-c} 1/(\beta - \alpha)dx = (\beta - 2a + c)/(\beta - \alpha)$ if $2a - c \leq \beta$, 0 otherwise;
2. $P(X \leq c) = (c - \alpha)/(\beta - \alpha)$.

Hence, we obtain that:

$$\max_{v\in[a,b]} P(|X-v| \geq v-c) = \begin{cases} 1-2(a-c)/(\beta-\alpha) & \text{if } c<a \wedge 2a-c \leq \beta \\ (c-\alpha)/(\beta-\alpha) & \text{if } c<a \wedge 2a-c > \beta \end{cases}$$

If $c > b$ then $\max_{v\in[a,b]} P(|X-v| \geq |c-v|)$ can be equivalently written as $\max_{v\in[a,b]} P(|X-v| \geq c-v)$ that is obtained for $v=b$. Therefore, $\max_{v\in[a,b]} P(|X-v| \geq c-v) = P(|X-b| \geq c-b) = P(X-b \geq c-b) + P(X-b \leq b-c) = P(X \geq c) + P(X \leq 2b-c)$. We have that:

$$\max_{v\in[a,b]} P(|X-v| \geq c-v) = \begin{cases} (\beta-c)/(\beta-\alpha) & \text{if } c>b \wedge 2b-c < \alpha \\ 1-2(c-b)/(\beta-\alpha) & \text{if } c>b \wedge 2b-c \geq \alpha \end{cases}$$