An Efficient Data Ranking Technique based on Bicluster Score

S.Sundararajan1 and Dr.S.Karthikeyan2
1Asscoiate Professor and Head/MCA, SNS College of Technology, Coimbatore, Tamil Nadu, India - 641035.
2 Professor and Director, Department of Computer Applications, Karpagam University- Coimbatore, Tamil Nadu, India -641035.

ABSTRACT

The study and understanding of spatial data sets structured an important part of geostatistics and is, regrettably, highly human dependent (Genton and Furrer, 1998). For example, it is finely known with the intention of different individuals will acquire different approaches, yielding a large collection of distinct solutions. It is frequently the case where decision and experience play a key role in choosing the appropriate spatial interpolation technique for each individual case (England, 1990). This is somewhat due to the variety of existing spatial interpolation methods, which range from simple intuitive predictions to more sophisticated and complex procedures (Cressie, 1991). Approximating both rainfall at unwaged locations and mean area rainfall more an area (e.g. a catchment) based on the results of meteorological observations, motivated the development of gridded estimates of precipitation to provide inputs to spatially distributed hydrologic and management models.

Although there are numerous articles have been written that are concerned with spatial interpolation, there is little or no agreement among the authors on the superiority of some techniques over others. Additionally, the increasing interest in Geographic Information Systems (GIS) with their broad usage and popularity, made it crucial to simply investigate the credibility and applicability of the different ready-to-use spatial interpolation techniques that are embedded in those systems. Generated with that in mind, this work has also been inspired by the Journal of Geographic Information and Decision Analysis initiative's special edition on spatial interpolation (Spatial Interpolation Comparison SIC97).

Keywords: *spatial data mining, biculster score, data ranking, spatial query on R- trees*

INTRODUCTION

Data Mining (the analysis step of the Knowledge Discovery in Databases process, or KDD), a relatively young and interdisciplinary field in astronomy, business, computer science , economics, physics, social sciences and others is the process of discovering new patterns from large data sets involving methods from statistics and artificial intelligence but also database management. Spatial database systems manage large collections of geographic entity, which apart from spatial attributes contain spatial information and non-spatial information (e.g., name, size, type, price, location etc.). In this paper, an interesting type of partiality query, which select the best spatial location with respect to the excellence of conveniences in its spatial area. Given a set D of interesting objects (e.g., candidate locations), a top-k spatial preference query retrieves the k objects in D with the highest scores. The score of a location is defined by the kind of quality in its spatial locality. A customer may want to rank the contents of this database with respect to the quality of their locations, quantified by aggregating non-spatial characteristics of other features (e.g., restaurants, cafes, hospital, market, etc.) in the spatial neighbourhood of the flat (defined by a spatial range around it). Quality may be subjective and query-parametric. Fig. 1a illustrates the locations of an object data set D (hotels) in white, and two feature data sets: the set F1 (restaurants) in gray, and the set F2 (cafes) in black. Quality points are labelled by excellence values that can be obtained from rating providers (e.g., http://www.zagat.com/). For the ease of argument, the qualities are normalized to values in [0, 1]. The score T (p) of a hotel p is defined in terms of: 1) the maximum quality for each feature in the neighbourhood region of p, and 2) the aggregation of those qualities.

a) Range score
b) Influence score

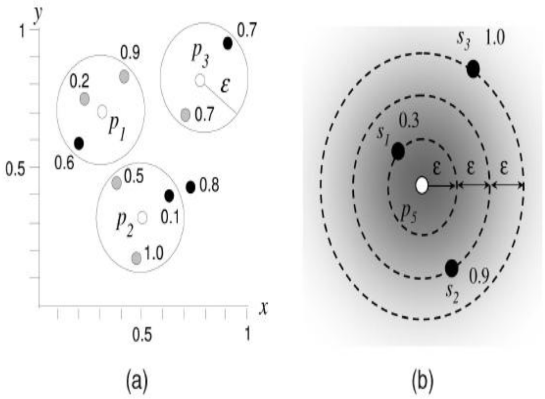


Fig1. Examples of top-k spatial preference queries

A simple score instance, called the range score, binds the neighborhood region to a circular region at p with radius (shown as a circle), and the aggregate function to SUM. For instance, the maximum quality of gray and black points within the circle of p1 are 0.9 and 0.6, respectively, so the score of p1 is T (p1) = 0:9 + 0:6 = 1:5. Similarly, we obtain T (p2) = 1:0 + 0:1 = 1:1 and T (p3) = 0:7 +0:7 = 1:4. Hence, the hotel p1 is returned as the top result. In detail, the semantics of the combined function is relevant to the Customer's query. The SUM function challenges to set of scales the overall qualities of all features. For the MIN function, the top result becomes p3, with the score T (p3) = min {0:7, 0:7} = 0.7. It ensures that the top result has reasonably high qualities in all features. For the MAX function, the top result is p2, with (p2) = max {1:0, 0:1} = 1:0. It is used to optimize the superiority in exacting feature, but not necessarily all of them. The neighborhood region in the above spatial preference query can also be defined by other score functions. An important score is the influence score. Now, the score of a cafe $S_i$computed by multiplying its quality with the weight 2^{-j}, where j is the order of the smallest circle containing S_i. For example, the scores of $S_1; S_2$; and S_3 are $0.3/2^1 = 0{:}15,\ 0.9/2^2 = 0{:}225, and\ 1.0/2^3 = 0{:}125$, respectively. The influence score of p5 is taken as the highest value (0.225). Traditionally, there are two types of ranking lands: First one spatial ranking, which orders the objects based on their distance and score from a reference feature, and second Non spatial ranking, which orders the objects by an combined method on their non-spatial values. Our top-k spatial preference query integrates these two types of ranking in an spontaneous way. A brute-force approach for evaluating it is to calculate the scores of all objects in D and select the top-k land. This technique, however, is expected to be very costly for large input data sets. In this paper, we propose alternative techniques that aim at minimizing the I/O accesses to the object and feature data sets, while being also computationally efficient. Our techniques apply on spatial-separation access functions and work out score bounds for the objects indexed by them, which are used to effectively trim the try to find space. Specifically, we contribute the branchand-bound (BB) algorithm and the feature join (FJ) algorithm for efficiently processing the top-k spatial preference query.

An attention to location, spatial interaction, spatial structure and spatial processes lies at the heart of research in several sub disciplines in the social sciences. Empirical studies in these fields routinely employ data for which locational attributes (the "where") are an important source of information. Such data typically consist of one or a few cross sections of observations for either micro-units, such as households, store sites, settlements, or for aggregate spatial units, such as electoral districts, counties, states or even countries. Observations such as these, for which the absolute location and/or relative positioning (spatial arrangement) is taken into account, are referred to as spatial data. In the social sciences, they have been utilized in a wide range of studies, such as archeological investigations of ancient settlement patterns (e.g., in Whitley and Clark, 1985, and Kvamme, 1990), sociological and anthropological studies of social networks (e.g., in White et al., 1981, and Doreian et al., 1984), demographic analyses of geographical trends in mortality and fertility (e.g., in Cook and Pocock, 1983, and Loftin and Ward, 1983), and political models of spatial patterns in international conflict and cooperation (e.g., in O'Loughlin, 1985, and O'Loughlin and Anselin, 1991). Furthermore, in urban and regional economics and regional science, spatial data are at the core of the field and are studied to model the spatial structure for a range of socioeconomic variables, such as unemployment rates (Bronars and Jansen, 1987), household consumer demand (Case, 1991), and prices for gasoline (Haining, 1984) or housing (Dubin, 1992).

The locational attributes of spatial data (i.e., for the settlements, households, regions, etc.) are formally expressed by means of the geometric features of points, lines or areal units (polygons) in a plane, or, less frequently, on a surface. This spatial referencing of observations is also the salient feature of a Geographic Information System (GIS), which makes it a natural tool to aid in the analysis of spatial data. I return to this issue in more detail below.

The crucial role of location for spatial data, both in an absolute sense (coordinates) and in a relative sense (spatial arrangement, distance) has major implications for the way in which they should be treated in statistical analysis, as discussed in detail in Anselin (1990a). Indeed, location gives rise to two classes of so called spatial effects: spatial dependence and spatial heterogeneity. The first, often also referred to as spatial autocorrelation or spatial association, follows directly from Tobler's (1979) First Law of Geography, according to which "everything is related to everything else, but near things are more related than distant things." As a consequence, similar values for a variable will tend to occur in nearby locations, leading to spatial clusters. For example, a high crime neighborhood in an inner city will often be surrounded by other high crime areas, or a low income county in a remote region may be neighboring other low income counties. This spatial clustering implies that many samples of geographical data will no longer satisfy the usual statistical assumption of independence of observations.

A major consequence of the dependence in a spatial sample is that statistical inference will not be as efficient as for an independent sample of the same size. In other words, the dependence leads to a loss of information[1] Roughly speaking, and everything else being the same, this will be reflected in larger, variances for estimates, lower significance levels in tests of hypotheses and a poorer fit for models estimated with data from dependent samples, compared to independent samples of the same size. I will refer to this aspect of spatial dependence in the rest of the paper as a nuisance. The loss in efficiency may be remedied by increasing the sample size or by designing a sampling scheme that spaces observations such that their interaction is negligible. Alternatively, it may be taken into account by means of specialized statistical methods. In this paper, I will focus on the latter. When spatial dependence is considered to be a nuisance, one only wants to make sure that the interpretation of the results of a statistical analysis are valid. One is thus not really interested in the source of the spatial association, i.e., in the form of the spatial interaction, the characteristics of the spatial structure, or the shape of the spatial and/or social processes that led to the dependence. When the latter is the main concern, I will use the term substantive spatial dependence instead.

The second type of spatial effect, spatial heterogeneity, pertains to the spatial or regional differentiation which follows from the intrinsic uniqueness of each location. This is a special case of the general problem of structural instability. As is well known, in order to draw conclusions with a degree of general validity from the study of a spatial sample, it is necessary that this sample represents some type of equilibrium. In the analysis of cross-sectional data in the social sciences this assumption is typically made. However, this assumption is considered with respect to the time dimension only, and systematic instability or structural variation that may be exhibited across different locations in space is mostly ignored. Such spatial heterogeneity may be evidenced in various aspects of the statistical analysis: it may occur in the form of different distributions holding for spatial, subsets of the data, or more simply, in the form of different means, variances or other parameter values between the subsets. I will refer to discrete changes over the landscape, such as a difference in mean or variance between inner city and suburb, or between northern and southern states as spatial regimes, where each regime corresponds to a well-defined subset of locations. Alternatively, I will call a continuous variation with location spatial drift. This would be the case if the parameters of a distribution vary in a smooth fashion with location, for example, when their mean follows a polynomial expression in the x and y coordinates (this is referred to as a trend surface). As is the case for spatial dependence, spatial heterogeneity can also be considered either as a nuisance or as substantive heterogeneity

SPATIAL DATA ANALYSIS

In Anselin and Griffith (1988), it is shown in some detail how the results of data analyses may become invalid if spatial dependence and/or spatial heterogeneity are ignored. Consequently, specialized techniques must be used instead of those that follow the standard assumptions of independence and homogeneity. By now, a large body of such techniques has been developed, which appears in the literature under the rubrics of spatial statistics, geostatistics, or spatial econometrics. The differences between these "fields" are subtle and to some extent semantic. Spatial statistics is typically considered to be the most general of the three, with geostatistics focused on applications in the physical (geological) sciences, and spatial econometrics finding application in economic modeling.

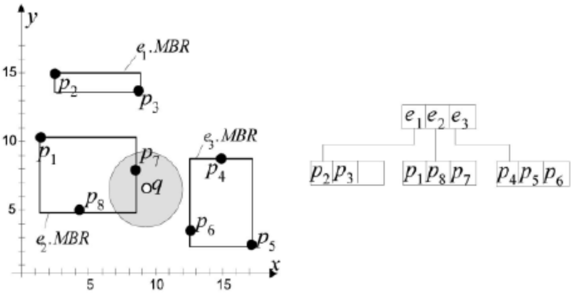
A useful taxonomy for spatial data analysis was recently suggested by Cressie (1991). He distinguishes between three broad classes of spatial data and identifies a set of specialized techniques for each. Crressie’s taxonomy consists of lattice data (discrete variation over space, with observations associated with regular or irregular areal units), geostatistical data (observations associated with a continuous variation over space, typically in function of distance), and point patterns (occurrences of events at locations in space). In the remainder of this paper, I will focus exclusively on the first category (lattice data), due to space limitations, but also because I have found it to be the most appropriate perspective for applications in the social sciences that utilize GIS. I chose not to discuss geostatistics, since the requirement of continuous variation with distance in an isotropic space is typically not satisfied by spatial samples in the social sciences. Such samples are mostly limited to data for areal units, which are often defined in a rather arbitrary fashion, making an assumption of continuity tenuous at best. Recent reviews of geostatistical techniques can be found in Davis (1986), Isaaks and Srivastava (1989), Webster and Oliver (1990), and Cressie (1991). In contrast to the geostatistical data viewpoint, point patterns represent a very appropriate perspective for the study of many phenomena in the social sciences, such as the analysis of the spatial arrangement of settlements, of store locations, occurrences of crime, infectious diseases, etc. I elected not to discuss them in this paper because their study does not require much in terms of the functionality of a GIS, once the coordinates of the locations have been determined. A very readable introduction to point pattern analysis is given in Boots and Getis (1988) and Upton and Fingleton (1985). More advanced treatments can be found in Getis and Boots (1978), Ripley (1981) and Diggle (1983), as well as in Cressie (1991).

Unfortunately, the need for specialized spatial data analysis techniques is not commonly appreciated in empirical work, as illustrated by an analysis of the contents of recent journal issues in regional science and urban economics in Anselin and Hudak (1992). [2] Over 200 empirical articles were reviewed, of which slightly more than one fifth employed spatial data, roughly evenly divided between purely cross-sectional and pooled cross-section and time series data. Of those, only one considered spatial dependence in a rigorous fashion. This absence of a strong dissemination of the methodological findings to the practice of empirical research is often attributed to the lack of operational software for spatial data analysis, e.g., as argued in Haining (1989, p. 201). While this may have been the case in the past, several recent efforts have added features for spatial analysis to many existing statistical and econometric software packages, in the form of macros and special subroutines. A small number of dedicated spatial data analysis software packages have become available as well, which should greatly facilitate the use of these techniques by a wider range of social scientists.

SPATIAL QUERY EVALUATION ON R-TREES

The most popular spatial access method is the R-tree [3], which indexes minimum bounding rectangles (MBRs) of objects. Fig. 2 shows a set D=(p1, . . . , p8) of spatial objects(e.g., points) and an R-tree that indexes them. R-trees can efficiently process main spatial query types, including spatial range queries, nearest neighbor queries, and spatial joins.

Given a spatial region W, a spatial range query retrieves from D the objects that intersect W. For instance, consider a range query that asks for all objects within the shaded area in Fig. 2. Starting from the root of the tree, the query is processed by recursively following entries, having MBRs that intersect the query region.



(a) (b)

Fig.2. Spatial queries on R-trees. a) MBRs
b)R-tree representation

For instance, e_1 does not intersect the query region, thus the subtree pointed by e_1 cannot contain any query result. In contrast, e_2 is followed by the algorithm and the points in the corresponding node are examined recursively to find the query result p_7. A nearest neighbor query takes as input a query object q and returns the closest object in D to q. For instance, the nearest neighbor of q in Fig. 2 is p_7. Its generalization is the k-NN query, which returns the k closest objects to q, given a positive integer k. NN (and k-NN) queries can be efficiently processed using the best-first (BF) algorithm of [4], provided that D is indexed by an R-tree. A min-heap H, which organizes R-tree entries based on the (minimum) distance of their MBRs to q is initialized with the root entries. In order to find the NN of q in Fig. 2, BF first inserts to H entries $e_1 e_2, e_3$, and their distances to q. Then, the nearest entry e_2 is retrieved from H and objects p1, p7, p8 are inserted to H. The next nearest entry in H is p7, which is the nearest neighbor of q. In terms of I/O, the BF algorithm is shown to be no worse than any NN algorithm on the same R-tree [4].

The aggregate R-tree (aR-tree) [5] is a variant of the R-tree, where each non leaf entry augments an aggregate measure for some attribute value (measure) of all points in its sub tree. As an example, the tree shown in Fig. 2 can be upgraded to a MAX aR-tree over the point set, if entries e1,e2,e3 contain the maximum measure values of sets (p2,p3), (p1,p8, p7), (p4, p5, p6), respectively. Assume that the measure values of p4, p5, p6 are 0.2, 0.1, and 0.4, respectively. In this case, the aggregate measure augmented in e3 would be max (0.2, 0.1, 0.4) = 0.4. In this paper, we employ MAX aR-trees for indexing the feature data sets (e.g., restaurants), in order to accelerate the processing of top-k spatial preference queries.

Given a feature data set F and a multidimensional region R, the range top-k query selects the tuples (from F) within the region R and returns only those with the k highest qualities. Hong et al. [6] indexed the data set by a MAX aR-tree and developed an efficient tree traversal algorithm to answer the query. Instead of finding the best k qualities from F in a specified region, our (range score)query considers multiple spatial regions based on the points from the object data set D, and attempts to find out the best k regions (based on scores derived from multiple feature data sets Fc).

SPATIAL PREFERENCE QUERIES

It formally defines the top-k spatial preference query problem and describes the index structures for the data sets. Section 3.2 studies two baseline algorithms for processing the query. Section 3.3 presents an efficient branch-and-bound algorithm for the query, and its further optimization is proposed in Section 3.4. Section 3.5 develops a specialized spatial join algorithm for evaluating the query [7]. Finally, Section 3.6 extends the above algorithms for answering top-k spatial preference queries involving other aggregate functions.

4.1. Definitions and Index Structures

Given an object data set D and m feature data sets F1,F2 . . . Fm, the top-k spatial preference query retrieves the k points in D with the highest score. Here, the overall score of an object point p ∈ D is defined as

$$T(p) = AGG\{T_c(p) | c \in [1, m]\} \quad (1)$$

Where AGG is an aggregate function (e.g.: SUM, MIN, MAX etc)

$T_c(p)$ is the c^{th} component score of p with respect to the neighborhood condition and m is the number of feature data sets. The cth component score of p i.e., $T_c(p)$ can be computed as follows

$$T_c(p) = \max\left(\{w(s) \middle| s \in F_c \wedge dist(p,s) \leq \epsilon\} U \{0\}\right). \tag{2}$$

4.2. Algorithms

We develop various algorithms for processing top-k spatial preference queries. We first introduce a brute-force solution that computes the score of every point $p \in D$ in order to obtain the query results [8]. Then, we propose a group evaluation technique that computes the scores of multiple points concurrently.

1) Simple Probing Algorithm

For a point $p \in D$, where not all its component scores are known, its upper bound score $T_u(p)$ defined as

$$T_u(p) = \sum_{c=1}^{m} \{T_c(p), \ \ if \ T_c(p) \ is \ known \tag{3}$$

1, otherwise

It is guaranteed that the upper bound $T_u(p)$ is greater than or equals to the actual score $T(p)$.

Algorithm 1 is a pseudo code of the simple probing (SP) algorithm, which retrieves the query results by computing the score of every object point. The algorithm uses two global variables: W_k is a minheap for managing the top-k results and γ represents the top-k score so far (i.e., lowest score in Wk). Initially, the algorithm is invoked at the root node of the object tree (i.e., N =D.root)[9]. The procedure is recursively applied (at Line 4) on tree nodes until a leaf node is accessed. When a leaf node is reached, the component score $T_c(e)$ (at Line 8) is computed by executing a range search on the feature tree Fc for range score queries. Lines 6-8 describe an incremental computation technique, for reducing unnecessary component score computations. In particular, the point e is ignored as soon as its upper bound score $T_u(e)$ (see (3)) cannot be greater than the best-k score γ. The variables W_k and γ are updated when the actual score $T(e)$ is greater than γ.

Algorithm1. Simple Probing Algorithm

Algorithm $SP(Node\ N)$

```
1) for each entry $e \in N$ do
2) If $N$ is nonleaf then
3) read the child node $N'$ pointed by $e$;
4) $SP(N')$;
5) else
6) for $c = 1$ to $m$ do
7) If $T_u(e) > \gamma$ then
   //if upper bound is greater than $\gamma$

8) compute $T_c(p)$ using tree $Fc$; update $T_u(e)$;
9) If $T(e) > \gamma$ then
10)     Update $W_k$ and $\gamma$ by e;
```

Drawbacks

1) it is very expensive because it comutes score for all objects.

2) No concurrency
3) It is not efficient method for larger input data sets.

2) *Group Probing Algorithm*

Due to separate score computations for different objects, SP is inefficient for large-object data sets. In view of this, we propose the group probing (GP) algorithm, a variant of SP that reduces I/O cost by computing scores of objects in the same leaf node of the R-tree concurrently. In GP, when a leaf node is visited, its points are first stored in a set V and then their component scores are computed concurrently at a single traversal of the Fc tree[10].

We now introduce some distance notations for MBRs. Given a point p and an MBR e, the value $mindist(p, e)$ [4] denotes the minimum possible distance between p and any point in$e$. Similarly, given two MBRs e_a and e_b, the value $mindist(e_a, e_b)$ denotes the minimum possible distance between any point in ea and any point in eb**.**

Algorithm 2 shows the procedure for computing the c^{th} component score for a group of points. Consider a subset Vof D for which we want to compute their component score at feature tree Fc.

Initially, the procedure is called with N being the root node of Fc. If e is a nonleaf entry and its mindist from some point $p\epsilon V$ is within the range, then the procedure is applied recursively on the child node of e, since the subtree of Fc rooted at e may contribute to the component score of p. In case e is a leaf entry (i.e., a feature point), [11] the scores of points in V are updated if they are within distance ϵ from e.

Algorithm 2. Group Probing Algorithm

```
algorithm $GP(Node\ N, Set\ V\ , Value\ c, Value\ \epsilon\ )$
for each entry $e\ \epsilon\ N$ do
    If $N$ is nonleaf then
        If $\epsilon V$ , $mindist(p, e) \leq \epsilon$ then
            read the child node $N'$ pointed by $e$;
            $GP(N', V\ , c, \epsilon\ )$;
    else
        for each $p\ \epsilon\ V$ such that $dist(p, e) \leq \epsilon$ do
            $T_c(p) = max\{T_c(p), w(e)\}$;
```

Drawbacks

1. It is also expensive because it computes score for all objects but concurrently.

4.3. Branch-and-Bound Algorithm

GP is still expensive as it examines all objects in D and computes their component scores. We now propose an algorithm that can significantly reduce the number of objects to be examined [12]. The key idea is to compute, for nonleaf entries e in the object tree D, an upper bound $T_u(p)$ of the score $T(p)$ for any point p in the subtree of e. If $T_u(e) \leq \gamma$ then we need not access the subtree of e, thus we can save numerous score computations.

Algorithm 3 is a pseudocode of our BB algorithm, based on this idea. BB is called with N being the root node of D. If N is a nonleaf node, Lines 3-5 is used to compute the scores $T(e)$ for nonleaf entries e concurrently. Recall that $T_u(e)$ is an upperbound score for any point in the subtree of e. If $T_u(e) \leq \gamma$, then the subtree of e

cannot contain better results than those in Wk and it is removed from V. In order to obtain points with high scores early, we sort the entries in descending order of $T(e)$ before invoking the above procedure recursively on the child nodes pointed by the entries in$V$. If N is a leaf node, we compute the scores for all points of N concurrently and then update the set Wk of the top-k results [13]. Since both Wk and γ are global variables, their values are updated during recursive call of BB.

Algorithm 3. Branch-and-Bound Algorithm

```
$Wk = new\ min - heap\ of\ size\ k\ (initially\ empty); = 0;$

Algorithm $BB(Node\ N)$

$V = \{e \mid e \epsilon N\}$;
If $N$ is nonleaf then
    for $c = 1\ to\ m$ do
        compute $T_c(e)$ for all $e \epsilon V$ concurrently;
    remove entries e in V such that $T_u(e) \leq \gamma$;
    sort entries $e \epsilon V$ in descending order of $T(e)$;
    for each entry $e \epsilon V$ such that $T_u(e) > \gamma$ do
        read the child node $N'$ pointed by $e$;
        $BB(N')$;
else
    for $c = 1\ to\ m$ do
        compute $T_c(e)$ for all $e \epsilon V$ concurrently;
    remove entries $e\ in\ V$ such that $T_u(e) \leq \gamma$;
    update Wk (and $\gamma$) by entries in V ;
```

Advantages

1) It reduces number of objects to be examined.
2) It is efficient than SP and GP algorithms.

THE MAHALANOBIS DISTANCE STATISTIC

The Mahalanobis distance statistic (D2) represents the standardized squared distance between the covariate values for a given sample and the mean vector of these covariates for the occupied locations used to build the model (hereafter, training data)[14]. In the context of habitat modeling, a D2 value is computed for each map cell in the study area based on the value of the habitat covariates under consideration in that cell, relative to the average values of those covariates in the training data as follows:

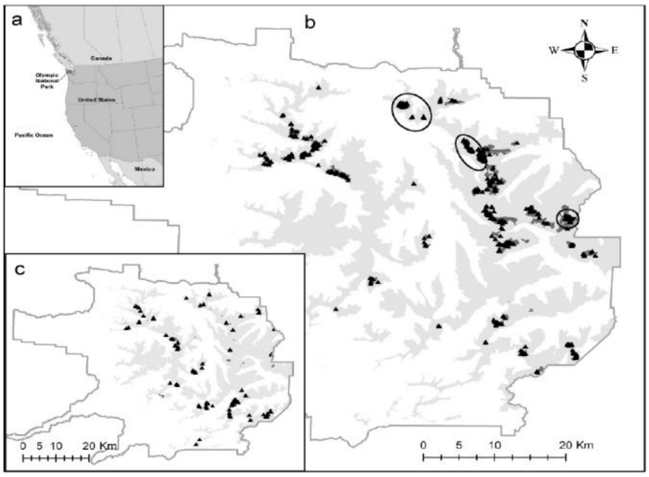


Fig1. (a) Location of Olympic National Park, Washington, USA. (b) Polygons of habitat known to be occupied by Olympic marmots in 2002–2005(dark gray shading) and 376 point locations used in development of habitat

models for the species (black triangles); intensive study sites are circled. (c)Polygons known to be abandoned (dark gray shading) and the 114 abandoned point locations (black triangles) used to test the habitat model. Areas within the park 1,300 m elevation are shown with light gray handling in (b) and (c).

$$D^2 = \left(\underline{\hat{\mu}} - \underline{x}\right)' \hat{\Sigma}^{-1} \left(\underline{\hat{\mu}} - \underline{x}\right),$$

where mˆ and Sˆ are, for the habitat covariates under consideration, the vector of the mean values and the variance–covariance matrix at presence locations, respectively. The variable x is the vector of values for each habitat variable for a given cell. Cells with smaller D2 values have habitat values more similar to the average of the training data and so should be more likely to be occupied. The D2 values are continuous with a minimum of zero. If the training data meet the assumption of multivariate normality, then the D2 values are chi-square distributed and can be rescaled to probabilities. Even when this assumption is violated, there is a monotonic relationship between the D2 values and dissimilarity from the mean, with equal scores being equally distant from the mean in multivariate space. Thus, D2 values rank habitat in terms of suitability rather than providing a probability of occupancy for each map cell.

Follow-up surveys guided by model predictions can provide estimates of probability of occupancy (Boetsch et al. 2003). For defining suitable habitat, a threshold D2 value is usually identified. Map cells with D2 values lower than that threshold are considered suitable for the study organism and the remaining cells are considered unsuitable (Thatcher et al. 2006). The threshold may be set so that all occupied points are classified as being within suitable habitat or such that some lesser proportion of the occupied locations are classified as suitable (Podruzny et al. 2002, Boetsch et al. 2003, van Manen et al. 2005, Thatcher et al. 2006,Thompson et al. 2006). When the proportion of occupied map cells with D2 values below the threshold is much greater than the proportion of random map cells with D2 values below that same value, or when distribution of D2 scores of occupied test locations is similar to those of training data, models are considered to perform well (Boetsch et al. 2003, Browning et al. 2005, van Manen et al. 2005).

CONCLUSION

In this paper, we have studied top-k spatial preference queries, provides a novel type of ranking for spatial objects based on qualities of features in their neighborhood. The neighborhood of an object p is captured by the scoring function: (i) the range score restricts the neighborhood to a crisp region centered at p, whereas (ii) the influence score relaxes the neighborhood to the whole space and assigns higher weights to locations closer to p. We presented five algorithms for processing top-k spatial preference queries. The baseline algorithm SP computes the scores of every object by querying on feature datasets. The algorithm GP is a variant of SP that reduces I/O cost by computing scores of objects in the same leaf node concurrently. The algorithm BB derives upper bound scores for non-leaf entries in the object tree, and prunes those that cannot lead to better results [15]. The algorithm BB* is a variant of BB that utilizes an optimized method for computing the scores of objects (and upper bound scores of non-leaf entries). The algorithm FJ performs a multi-way join on feature trees to obtain qualified combinations of feature points and then search for their relevant objects in the object tree.

Based on our experimental findings, BB* is scalable to large datasets and it is the most robust algorithm with respect to various parameters. However, FJ is the best algorithm in cases where the number m of feature datasets is low and each feature dataset is small. In the future, we will study the top-k spatial preference query on road network, in which the distance between two points is defined by their shortest path distance rather than their Euclidean distance. The challenge is to develop alternative methods for computing the upper bound scores for a group of points on a road network.

REFERENCES

[1] M.L. Yiu, X. Dai, N. Mamoulis, and M. Vaitis, “Top-k Spatial Preference Queries,” Proc. IEEE Int‟l Conf. Data Eng. (ICDE),2007.

[2] Anselin, Luc (1988a). Spatial Econometrics, Methods and Models (Dordrecht, Kluwer Academic). Anselin, Luc (1988b). Model validation in spatial econometrics: a review and evaluation of alternative approaches, International Regional Science Review 11, 279-316.

[3] Guttman, “R-Trees: A Dynamic Index Structure for Spatial Searching,” in SIGMOD, 1984.

[4] G. R. Hjaltason and H. Samet, “Distance Browsing in Spatial Databases,” TODS, vol. 24(2), pp. 265–318, 1999.

[5] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, “Efficient OLAP Operations in Spatial Data Warehouses,” in SSTD, 2001.

[6] S. Hong, B. Moon, and S. Lee, “Efficient Execution of Range Topk Queries in Aggregate R-Trees,” IEICE Transactions, vol. 88-D, no. 11, pp. 2544–2554, 2005.

[7] Cliff, AD. and J.K. Ord (198 1). Spatial Processes, Models and Applications (London, Pion)

[8] Griffith, Daniel A. (1987). Spatial Autocorrelation, A Primer (Washington, D.C., Association of American Geographers).

[9] D. Papadias, P. Kalnis, J. Zhang, and Y. Tao, “Efficient OLAP Operations in Spatial Data Warehouses,” in SSTD, 2001.

[10] T. Xia, D. Zhang, E. Kanoulas, and Y. Du, “On Computing Top-t Most Influential Spatial Sites,” in VLDB, 2005.

[11] Y. Du, D. Zhang, and T. Xia, “The Optimal- Location Query,” Proc.Int‟l Symp. Spatial and Temporal Databases (SSTD), 2005.

[12] Openshaw, S., 1989. Learning to live with error in spatial databases. In: Goodchild, M.F. and Gopal, S. (eds.), Accuracy of Spatial Databases. London: Taylor & Francis, pp. 263-276.

[13] Dunn, J. E., and L. Duncan. 2000. Partitioning Mahalanobis D2 to sharpen GIS classification. Pages 195–204 in C. A. Brebbia and P. Pascolo, editors. Management information systems 2000: GIS and remote sensing. WIT Press, Southhampton, United Kingdom.

[14] Browning, D. M., S. J. Beaupre, and L. Duncan. 2005. Using partitioned Mahalanobis D2(K) to formulate a GIS-based model of timber rattlesnake hibernacula. Journal of Wildlife Management 69:33–44.