Selectivity Estimation in Spatial Networks

E. Tiakas A.N. Papadopoulos A. Nanopoulos Y. Manolopoulos
Department of Informatics, Aristotle University
54124 Thessaloniki
Thessaloniki, Greece
{tiakas,papadopo,ananopou,manolopo}@csd.auth.gr

ABSTRACT

Modern applications requiring spatial network processing pose many interesting query optimization challenges. In many cases, query processing depends on the corresponding graph size (number of nodes and edges) and other graph parameters. This dependency may be local or global. In this paper, we present novel methods to estimate the number of nodes in regions of interest in spatial networks, towards predicting the space and time requirements of range queries. We examine all methods by using real-life and synthetic spatial networks. Experimental results show that the number of nodes can be estimated efficiently and accurately with small space requirements, thus providing useful information to the query optimizer.

Categories and Subject Descriptors

H.2.8 [**Database management**]: Database applications—*Spatial databases and GIS*

General Terms

Algorithms Performance

Keywords

estimation, optimization, spatial networks

1. INTRODUCTION

Every spatial network can be represented as a graph, where all spatial network's nodes and connections are represented as the graph's vertices and edges, respectively. Depending on the application this graph may be weighted, directed or un-directed. Thus, any spatial query into the original network can be executed to its corresponding graph representation G. Evidently, the performance of such queries is strongly related to the number of nodes and edges lying into the region of interest, which is a subgraph of G.

Several query processing techniques in spatial networks have been proposed for fundamental query types like window and k-nearest-neighbors [6]. However, when such queries are combined, query optimization techniques are important to increase efficiency. Therefore, estimations on factors that can affect the performance of such queries are crucial for query optimization purposes. More specifically, the number of vertices contained in a specific range, is an indication of the required computational time that will be required to store this part of the graph, as well as the time required to execute queries.

To the best of our knowledge, the estimation of the number of nodes in spatial networks has not been studied yet. This paper is a first attempt to design efficient techniques that can successfully estimate the number of vertices contained in a region of the spatial network graph determined by a starting vertex v_0 and a network-based distance e. Towards this goal, several different directions are taken, each one with different requirements and estimation accuracy. The studied methods are: (i) an estimation with global parameters, (ii) a local estimation using densities or kernels, and (iii) an estimation with binary encoding techniques.

2. RELATED WORK

In [11] the Power-Method has been presented, which can perform accurate estimations for spatial queries using only a simple formula with minimal computational cost, small space requirements and average relative error rate below 20%. However, this methodology can be applied only in Euclidean geometry data sets, using the L_2, L_∞ metrics.

In [12], the authors examined the performance of spatial range queries specifically in R-trees and variants. In particular, specific estimation formulae have been derived for the number of disk accesses using global parameters and local densities.

The notion of local density has been studied extensively in general and spatial data sets, but not in combination with spatial queries performance in spatial networks. There is a great number of density estimation proposals in bibliography, which are used in numerous estimation and approximation problems. A significant method in this direction is the Kernel Density Estimation Method and its variants [10, 2]. Kernel Density Estimators are used in many applications domains. For example, they are used in many clustering methods [4, 7], outlier detection [9], and visualization techniques [8]. Also, several corrections and variants of Kernel Density Estimations and Smoothing have appeared [3, 13]. However, all these applications are restricted to Euclidean spaces.

The rest of the paper is organized as follows. Section 3

SAC'08 March 16-20, 2008, Fortaleza, Ceará, Brazil
Copyright 2008 ACM 978-1-59593-753-7/08/0003 ...$5.00.

presents the proposed methods. Section 4 presents experimental results, whereas Section 5 concludes our work.

3. ESTIMATION APPROACHES

Here, we define the problem and present the proposed methods aiming at effective solutions for general graphs with specific properties. Table 1 depicts the basic notations being used for the rest of the work.

PROBLEM DEFINITION
Let G be a connected weighted graph (directed or not), with a set of nodes V_G and a set of weighted edges E_G. In this graph, we apply the network distance measure $d()$ where $d(u, v)$ denotes the shortest path distance between the nodes v, u of G. Giving a specific starting node $v_0 \in V_G$, and a desired distance e, we are interested in determining an estimator $\widetilde{N}(v_0, e)$, which can efficiently estimate the total number of nodes that can be reached from v_0 within a distance less than or equal to e:

$$N(v_0, e) = |\{v \in V_G : d(v, v_0) \leq e\}|$$

There are several parameters affecting the answer to this problem. The most significant ones are: (i) The selection of the starting node v_0 and its position on the graph. Different results can arise between starting nodes from dense regions and from sparse regions. (ii) The graph morphology and topology. Different results are obtained for uniform and non-uniform graphs. We define the notion of uniformity with the following properties: (a) all edge weights of G are close to the overall average weight, and (b) all node degrees of G are close to the overall average degree.

3.1 Global Parameters Estimation Method

The first proposed method is the most applicable in uniform or almost uniform graphs, and constructs an estimation formula $\widetilde{N}(v_0, e)$ containing only global graph parameters. The necessary definitions for these parameters are:

- Average weight: $\overline{w} = \frac{1}{|E_G|} \sum w(v_i, v_j)$
- Total degree: $\deg(v_i) = in\deg(v_i) + out\deg(v_i)$, if G is a directed graph. Otherwise, is defined as the simple degree of the node v_i.
- Average total degree: $\overline{deg} = \frac{1}{|V_G|} \sum\limits_i \deg(v_i) = \frac{2|E_G|}{|V_G|}$

Symbol	Description
G	The network graph, directed or not
V_G	The set of nodes in the graph G
E_G	The set of edges in the graph G
$\lvert V_G \rvert$	The number of nodes in the graph G
$\lvert E_G \rvert$	The number of edges in the graph G
v_0	The selected starting node
e	The selected radius in the region of v_0
$d(v_i, v_j)$	The network distance between nodes v_i, v_j
D_G	The diameter of G
$in\deg(v_i)$	The in-degree of node v_i
$out\deg(v_i)$	The out-degree of node v_i
$deg(v_i)$	The total degree of node v_i
$w(v_i,v_j)$	The weight of the edge (v_i,v_j)
$\overline{w}$	The average weight of all edge weights
$\overline{deg}$	The average total degree of all nodes
$N(v_0, e)$	The exact number of nodes into the e-range
$\widetilde{N}(v_0, e)$	The estimator of nodes into the e-range

Table 1: Frequently used symbols.

The Global Parameters Estimation method, defines a specific estimation formula for the number of nodes $N(v_0, e)$ lying into the e-range region of v_0. This formula use only the global graph parameters $\overline{w}$ and $\overline{deg}$:

$$\widetilde{N}(v_0, e) = \widetilde{N}(e) = \frac{\overline{deg}}{2} \cdot \frac{e}{\overline{w}} \cdot \left(\frac{e}{\overline{w}} + 1\right) + 1 \qquad (1)$$

This formula can be used in all graph types, but in many cases of non-uniform graphs the estimation error may be high. By studying further the graph types that Global Parameters Estimation Method returns always efficient estimation results, we concluded to interesting properties and graph classes. More specifically, it can be proven that Global Parameters Estimation Method returns definitely efficient estimation results if the following hold (proofs omitted):

1. The spatial network G is a uniform or almost uniform graph.
2. The starting node v_0 is selected from the region of G which is defined by the most central node v_c of G and distance radius equal to $\frac{D_G}{4}$, where D_G is the diameter of the graph G, thus: $d(v_0, v_c) \leq \frac{D_G}{4}$.
3. The range distances e are selected from the interval $[0, \frac{D_G}{4}]$, thus: $0 \leq e \leq \frac{D_G}{4}$.

A significant advantage of the Global Parameters Estimation method is that there is no need for additional space (in memory or disk) for bookkeeping. Moreover, the estimator $\widetilde{N}(v_0, e)$, can be computed instantly.

3.2 Local Densities Estimation Method

The second proposed method is the most applicable in both uniform and non-uniform graphs. To achieve efficient estimation results, it defines the notion of local densities. Using local densities, this method can include the affection of sparse and dense regions to the estimation procedure. The local densities are computed in a preprocessing step and kept in memory. The most representative local density factors we use, the best estimation results we will have. More specifically, to derive the new formula we need the following definitions:

Local Node Density: Let G be a graph with a set of nodes V_G. Then, for every node $v \in V_G$ we define its local node density AN_v to be a positive real number representing the local density of node v based on the node distribution into the spatial area of the graph.

Normalized Local Node Density: Let G be a graph with a set of nodes V_G, and local node densities $AN_v, \forall v \in V_G$. Then for every node $v \in V_G$ we define its normalized local node density UN_v to be a real number into the interval [0,1], given by:

$$UN_v = \frac{AN_v}{\max\{AN_{v_i}, v_i \in V_G\}} \qquad (2)$$

A significant property of the normalized local node densities is: *Nodes with normalized local densities close to 1, lie on* *dense regions of the graph, and nodes that have normalized local densities close to 0, lie on sparse regions of the graph.*

Following all the previous definitions, the proposed new estimation formula is an extension of the global parameters estimation formula adding the normalized local nodes densities of v_0 as global factors:

$$\widetilde{N}(v_0, e) = UN_{v_0} \cdot \left(\frac{\overline{deg}}{2} \cdot \frac{e}{\overline{w}} \cdot \left(\frac{e}{\overline{w}} + 1 \right) + 1 \right) \qquad (3)$$

The local node densities AN_v of all graph nodes can be computed with any known method, but different estimation results may arise between method selections in these computations. Here, we propose two different methods for the calculation of the local densities which can give efficient estimation results: (a) the Local Counting Density Estimators and (b) Kernel Density Estimators.

3.2.1 Local Counting Density Estimators

The main strategy on Local Counting Density Estimators, is to count the exact number of nodes on every node v of the spatial graph G, within a small local region of the node v with a global constant range radius e_c. Therefore, the Local Counting Density Estimators method defines the local node densities as follows: $AN_v = |\{v_i \in V_G : d(v, v_i) \leq e_c\}|$. The normalized local node densities are computed by Equation (2). This approach is simple and can give efficient estimation results if the global constant radius e_c is calibrated correctly. This constant affects all the local densities values. We propose to use e_c values that are small multiples of the average weight $\overline{w}$ of the graph. Varying this constant, we can change the accuracy of the estimator $\widetilde{N}(v_0, e)$. However, we must never use values $e_c \leq \overline{w}$, because we will underestimate many local densities of the graph.

3.2.2 Kernel Density Estimators

The Kernel Density Estimators are non-parametric density estimators which they smooth out the contribution of each observed object over a local neighborhood of this object. We applied the Kernel Density Estimators to our problem, where the observed objects now are the nodes of the spatial network graph G, and their contribution to each other is their network distance (shortest path distance). Any known kernel function $K(x)$ can be used, but we propose using the normal (Gaussian) kernel, for better estimation results: $K(X) = \frac{1}{\sqrt{2\pi}} e^{-\frac{1}{2}x^2}$. Now the local node and edge densities are defined as follows: $AN_v = \sum_{v_i \in V_G} K\left(\frac{d(v, v_i)}{h}\right)$. The normalized local node densities are computed by the Equation (2). The parameter $h > 0$ is the bandwidth of the kernel function, which is a global constant used for smoothing. Varying this constant, we can change the accuracy of the estimator $\widetilde{N}(v_0, e)$.

In both Local Densities estimation methods, the required space for keeping all normalized local node and edge densities after pre-computations is: $2 \cdot |V_G| \cdot SizeOf(real)$. If we keep the normalized local node densities UN_{v_0} in memory, then the estimation values of $\widetilde{N}(v_0, e)$ can be computed instantly without any time complexity cost during the estimation calculation.

3.3 Binary Encoding Estimation Method

The following proposed method is applicable in all type of graphs, and the selection of both parameters e and v_0 is completely free: the radius e can be any non-negative real number ($e \geq 0$), and the starting node v_0 can be any node of the graph G.

The rationale of this method is based onto the following observations. In the previous sub-section we have seen that all local densities pre-computations require some or all of the shortest path distances between the graph nodes. This happens for any selection of density estimators (Kernel, Counting, etc.). Therefore, a process that could encode all shortest path distances into node labels in a preprocessing step would be helpful, because all computations of network distances would be avoided.

An important research work in this direction is [1], were the authors present an efficient encoding technique, based on hypercube embedding, for assigning labels to all graph nodes, such that the network distance of every two nodes can be calculated directly by their labels. More specifically, after the encoding process, the node labels are binary code numbers and the network distance of every two nodes can be approximated efficiently by the Hamming distance between their corresponding codes. This binary encoding technique and its algorithms require a unitary weighted graph (all edge weights are equal to 1), therefore to apply this technique to our methodology, we must transform the spatial network to such a graph. The main task is that during this transformation all network distances on the final graph must approximate (as accurately as possible) the corresponding original distances. For this purpose, we propose the following preprocessing steps:

- We select a constant positive real value w_u into the interval $0 < w_u < \overline{w}$, which represents the selected unitary distance in the graph G. This constant affects the precision on distances approximations. If it takes values close to $\overline{w}$ then we will have low precision and a high estimation error. If instead it takes values close to 0 (but never equal to 0), then we will have high precision and an almost zero estimation error.

- In the next step, we transformation the graph G into a new graph $G' = T(G)$, by dividing all weights $w(v_i, v_j)$ with the selected unitary distance measure w_u and by adding intermediate nodes. More specifically, to produce the graph G', in every edge (v_i, v_j) of G we add a number of intermediate nodes, which is equal to the closest integer of $\frac{w(v_i, v_j)}{w_u}$ minus one. If we denote A_G the set of all added new nodes, then the graph G' will have $V_{G'} = V_G \cup A_G$, therefore $|V_{G'}| = |V_G| + |A_G|$ and $|E_{G'}| = |E_G| + |A_G|$, because the number of all new edges on G' is exactly the same with all added intermediate nodes. Finally, all edge weights on the new constructed graph G' will be equal to 1.

Now that we have construct the required unitary weighted graph G', we can execute the encoding algorithms of [1], and construct the final binary codes. These binary codes have a length of k-bits, where k equals to the number of edges contained into the perimeter of the graph G'. According to [1] this length is $O(\sqrt{n})$ where n is the number of all nodes on G', thus: $k = O(\sqrt{|V_{G'}|}) = O(\sqrt{|V_G| + |A_G|})$.

After all these preprocessing steps, we can have the final binary encoding in all nodes of our basic graph G. Let us now present the necessary definitions for the estimation:

- We denote by C_v the binary code of node v derived by its label, for every $v \in V_G$.
- We denote by $H(c_{v_i}, c_{v_j})$ the Hamming distance between the binary codes c_{v_i}, c_{v_j}, which is equal to the number of bit positions where the codes c_{v_i}, c_{v_j} differ.
- We denote by $u(x)$ the simple function that returns 1 if $x \geq 0$ and 0 otherwise.

The estimation formula of the binary encodings method finally is:

$$\widetilde{N}(v_0, e) = \sum_{v_i \in V_G} u\left(e - \frac{H(c_{v_0}, c_{v_i}) \cdot w_u}{2}\right)$$

We have proven (proof omitted) that the final binary codes have length equal to:

$$k = O(\sqrt{|V_G| + |E_G| \cdot \frac{\overline{w} - w_u}{w_u}}) \tag{4}$$

A significant observation regarding Equation (4) is that the unitary weight parameter w_u affects the length of the final binary encoding k. As w_u take values close to 0, k becomes larger, and therefore we need more space for the encoding. Thus, there is a trade-off between required space and estimation accuracy of this method. According to (4), we need an $O(|V_G| \cdot \sqrt{|V_G| + |E_G| \cdot \frac{\overline{w}-w_u}{w_u}})$ space to store all node binary codes. Thus, if we have not enough space, we can calibrate the w_u value to the available space.

If we keep the needed binary encoding in memory, then the required time for computations of estimator $\widetilde{N}(v_0, e)$, will be only the time of calculations of this formula, which is $O(|V_G|)$. However, this time cost is negligible, because the final formula has only binary and basic register operations.

4. EXPERIMENTS AND RESULTS

In this section we present experimental results for all the proposed estimation methods. We have used several real and synthetic spatial networks and we have tested the proposed methods for several parameter values. For brevity, we present only a small set of representative results, which depict the most significant performance aspects and trade-offs of the proposed methods. The data-sets used are as follows:

- **UN**: A synthetic, almost uniform, un-directed and weighted spatial network with 10,000 nodes of degree 4 (except from a small fraction of randomly selected nodes with degrees 2 and 3), and 19,800 edges with weights having randomly selected values from 12 to 18. Its global parameters are: $\overline{w} = 14.98591$, $\overline{deg} = 3.96$, and $D_G = 2,666.25$.
- **OL**: The real road network of Oldenburg, with 6,105 nodes and 7,035 edges [5]. Its global parameters are: $\overline{w} = 73.67902$, $\overline{deg} = 2.304668$, and $D_G = 12,985.97$.
- **SF**: The real road network of San Francisco, with 174,956 nodes and 223,001 edges [5]. Its global parameters are: $\overline{w} = 8.782676$, $\overline{deg} = 2.549224$, and $D_G = 16,828.54$.

In all presented experimental results, we have selected a large number of random nodes from these graphs (10% of the whole node sets) as representatives starting nodes v_0, we have performed e-range queries varying e from 0 to $\approx \frac{D_G}{4}$ (which is 50% of the graph radius), with a small increasing step (5 in UN and 25 in OL and SF), and we have computed and recorded the real $N(v_0, e)$ values. Then, we computed and recorded the corresponding estimation results $\widetilde{N}(v_0, e)$ in all proposed methods with the following parameter setups for the three data-sets: (i) $e_c \approx 3\overline{w}$, $10\overline{w}$, $20\overline{w}$, (ii) $h \approx \frac{1}{\overline{w}}$, 1, $\frac{\overline{w}}{3}$, $\overline{w}$, (iii) $w_u \approx \frac{2\overline{w}}{3}$, $\frac{\overline{w}}{3}$, 1.

Next, we have calculated all average real $N_{avg}(e)$ and estimation $\widetilde{N}_{avg}(e)$ recorded values by v_0 for every range e. These averages are functions of e. The basic accuracy measure used in all estimation methods and selections is defined as: $error[N](e) = \frac{|N_{avg}(e) - \widetilde{N}_{avg}(e)|}{N_{avg}(e)}$. Again this measure is a function of e, which we call *estimation error function*.

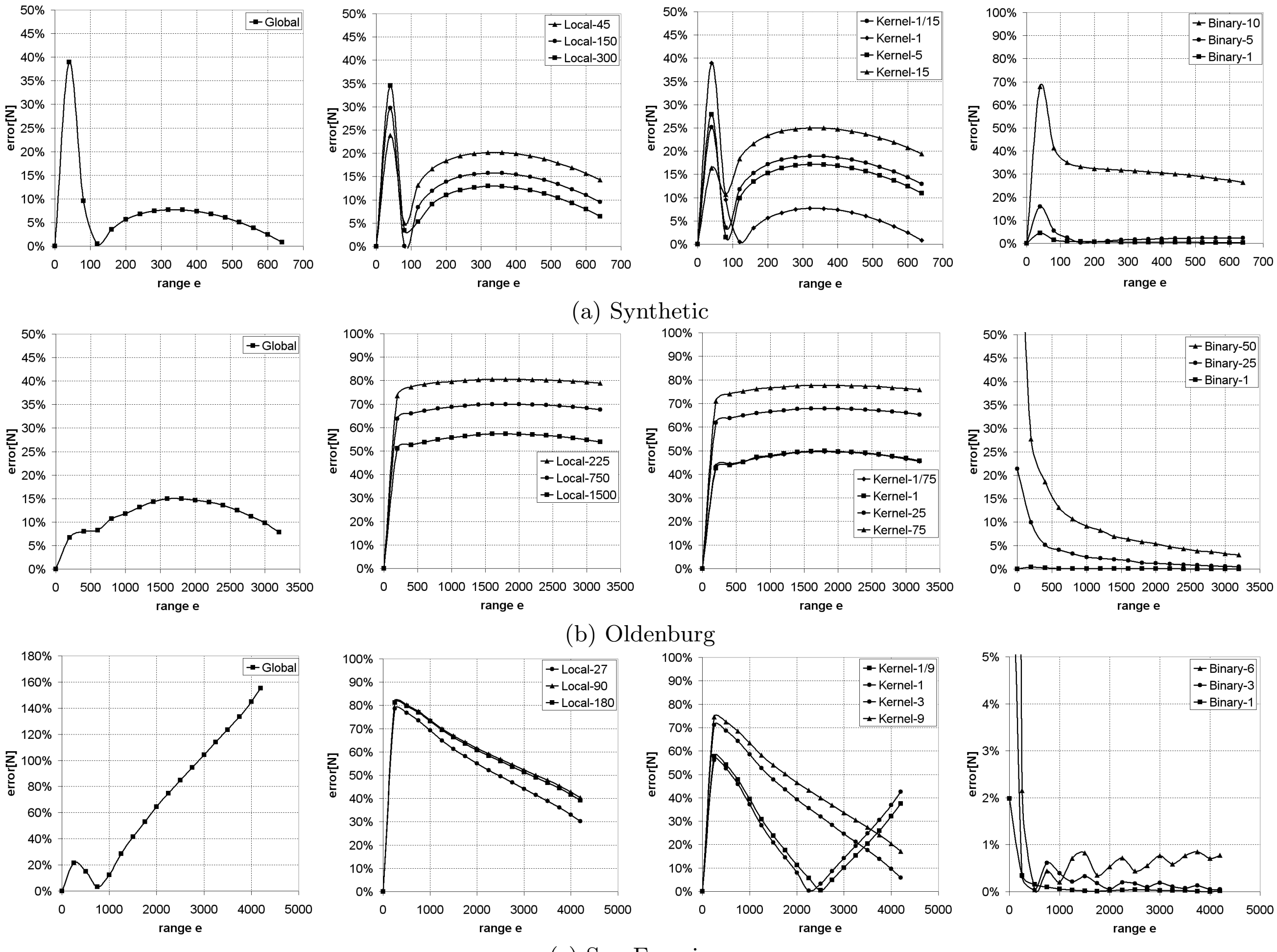
Figure 3a depicts the estimation error for the selected parameters of all proposed methods in UN network, where we observe that: (i) Global method offers good estimation results, due to the uniformity of the graph. (ii) By varying the densities radius e_c and the bandwidth h, we can improve the estimation accuracy of the Local methods. (iii) The Binary Encoding method returns accurate estimations with unitary distances equal to $\frac{\overline{w}}{3}$ or bellow. However, by decreasing the unitary distance w_u, the required space for the binary encoding becomes more significant, and therefore, the choice of $w_u \approx \frac{\overline{w}}{3}$ is adequate because it balances estimation accuracy with small space requirements.

Figure 3b depicts the estimation error for the selected parameters of all proposed methods in OL network, where we observe that: (i) Global method returns good estimation results, due to the almost uniform node distribution of the graph. (ii) By varying the densities radius e_c and the bandwidth h, we can improve the estimation accuracy of the Local methods. (iii) Again, the Binary method returns accurate estimations when $w_u \approx \frac{\overline{w}}{3}$ or less.

Figure 3c depicts the estimation error for the selected parameters of all proposed methods in SF network, where we observe that: (i) The Global method returns inaccurate estimations because SF is a completely non-uniform graph. (ii) Local methods return better estimation results, where Kernel Densities estimators are sensitive to bandwidth selections. (iii) Again, the Binary method returns accurate estimations when $w_u \approx \frac{\overline{w}}{3}$ or less.

5. CONCLUSIONS

In this paper, we have presented methods to estimate the number of vertices that are lying within a distance e from a vertex. Three different methods have been studied. We have given estimation equations for all methods, as well as specific space and time bounds. We have applied the proposed methods in both synthetic and real spatial networks, and we have given performance results. We conclude that: (a) Global Parameters Estimation method performs efficient estimations in uniform or almost uniform graphs, (b) Local Densities methods offer better estimations in non-uniform graphs, and (c) the Binary Encoding method offers the most accurate estimations in all graph types, with small space requirements which can be adjusted. An interesting direction for future work is to study the selectivity estimation for the number of edges in spatial networks.



(a) Synthetic

(b) Oldenburg

(c) San Francisco

Figure 1: Estimation results.

6. REFERENCES

[1] S. Gupta, S. Kopparty, C. Ravishankar. "Roads, Codes, and Spatiotemporal Queries" *Proceedings of the 23th ACM Symposium on Principles of Database Systems*, June 2004, Paris, France.

[2] J. N. Hwang, S. R. Lay and A. Lippman. " Nonparametric multivariate density estimation: a comparative study." Transactions on Signal Processing, 42(10):2795-2810, October 1994

[3] M.C. Jones: "Simple boundary correction for kernel density estimation", *Statistics and Computing*, 3, 135-146, 1993.

[4] M. Klusch, S. Lodi, G. Moro: "Distributed Clustering Based on Sampling Local Density Estimates", *Proceedings of International Joint Conference On Artificial Intelligence*, 2003.

[5] R-Tree Portal, *http://www.rtreeportal.org/main.html.*

[6] J. Sankaranarayanan, H. Alborzi, H. Samet: "Efficient Query Processing on Spatial Networks", *Proceedings 13th ACM International Symposium on Geographic Information Systems (GIS)*, pp.200-209, Bremen, Germany, 2005.

[7] E. Schikuta: "Grid-Clustering: An efficient hierarchical clustering method for very large data sets", *Proceedings of the 13th International Conference on Pattern Recognition*, pp.101-105, 1996.

[8] D.W. Scott: *"Multivariate Density Estimation: Theory, Practice and Visualization"*, John Wiley and Sons Inc., 1992.

[9] X. Shen, S. Agrawal: "Kernel Density Estimation for An Anomaly Based Intrusion Detection System", *International Conference on Machine Learning; Models, Technologies, Applications*, p.161-167, 2006.

[10] B. W. Silverman: *"Density Estimation for Statistics and Data Analysis"*, Chapman and Hall, 1986.

[11] Y. Tao, C. Faloutsos, D. Papadias: "Spatial Query Estimation without the Local Uniformity Assumption", *Geoinformatica*, No.10, pp.261-293, 2006.

[12] Y. Theodoridis, T. Sellis: "A Model for the Prediction of R-Tree Performance", *Proceedings of the 15th ACM Symposium on Principles of Database Systems*, 1996.

[13] M.P. Wand, M.C. Jones: *"Kernel Smoothing"*, Chapman and Hall, 1995.