Spatial Join Selectivity Using Power Laws

Christos Faloutsos[1], Bernhard Seeger[2], Agma Traina[3], Caetano Traina Jr.[4]

April, 2000
CMU-CS-00-124

School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA 15213-3890

---

[1] Department of Computer Science, Carnegie Mellon University - USA. Christos@cs.cmu.edu. This material is based upon work supported by the National Science Foundation under Grants No. IRI-9625428, DMS-9873442, IIS-9817496, and IIS-9910606, and by the Defense Advanced Research Projects Agency under Contract No. N66001-97-C-8517. Additional funding was provided by donations from NEC and Intel. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation, DARPA, or other funding parties.

[2] Fachbereich Mathematik und Informatik, Universität Marburg - Germany. Seeger@mathematik.uni-marburg.de. His work has been supported by Grant No. SE 553/2-1 from DFG (Deutsche Forschungsgemeinschaft).

[3] Department of Computer Science, University of São Paulo at São Carlos - Brazil . Agma@cs.cmu.edu. Her research was partially funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo - Brazil, under Grant 98/0559-7). On leave at Carnegie Mellon University.

[4] Department of Computer Science, University of São Paulo at São Carlos - Brazil Caetano@cs.cmu.edu. His research was partially funded by FAPESP (Fundação de Amparo à Pesquisa do Estado de São Paulo - Brazil, under Grant 98/05556-5). On leave at Carnegie Mellon University.

DISTRIBUTION STATEMENT A
Approved for Public Release
Distribution Unlimited

20000509 118

DTIC QUALITY INSPECTED 2

Keywords: spatial join, spatial join selectivity, spatial data mining, data mining,

Abstract

We discovered a surprising law governing the spatial join selectivity across two sets of points. An example of such a spatial join is "*find the libraries that are within 10 miles of schools*". Our law dictates that the number of such qualifying pairs follows a power law, whose exponent we call "pair-count exponent" (PC). We show that this law also holds for self-spatial-joins ("*find schools within 5 miles of other schools*") in addition to the general case that the two point-sets are distinct. Our law holds for many real datasets, including diverse environments (geographic datasets, feature vectors from biology data, galaxy data from astronomy).

In addition, we introduce the concept of the Box-Occupancy-Product-Sum (BOPS) plot, and we show that it can compute the pair-count exponent in a timely manner, reducing the run time by orders of magnitude, from quadratic to linear. Due to the pair-count exponent and our analysis (Law 1), we can achieve accurate selectivity estimates in constant time (O(1)) without the need for sampling or other expensive operations. The relative error in selectivity is about 30% with our fast BOPS method, and even better (about 10%), if we use the slower, quadratic method.

1. INTRODUCTION

Multi-dimensional and spatial database management systems (DBMS) have attracted a lot of interest. One of the most important operations in a spatial DBMS [GÜT94] is the spatial join, which is the counterpart to the equi-join in a relational DBMS.

The typical query is also called the 'all pairs' query or 'spatial distance join', as in the example, *'Estimate the number of schools that are within 5 miles from libraries'*. Spatial distance joins are considered to be among the most exxential joins in application areas, like data mining [CMN99] [NH94]. They are useful in multiple settings, such as the following.

- In geographic information systems (GIS) under the name of overlay queries: for example, *'Find all houses within 2 miles of a river'*.
- In urban planning, business planning, commercial intelligence: *'How many households are within 1 mile of our branches and from our competition's branches'*.
- In spatial data mining to detect correlations and test hypotheses: for example, *'Find 4-bedroom houses that are within 5 miles of a school'*, or *'How many luxury apartments are within 2 miles of a lake'* [NH94].
- In temporal data mining: *'Find economic embargos that were followed by war within a year'*, or *'Find network-switch failures that were within 5 seconds of a power surge'* [MTV95] [HKM+96].
- In multimedia and traditional databases: *'Find pairs of stock price changes that are within $10 of each other'* [FRM94].

The **spatial distance join** is defined using two spatial data sets, A and B, and a distance function L. For a given radius r, the spatial distance join computes $\{<a,b> \mid a \in A \text{ and } b \in B, L(a,b) \leq r\}$. A special case arises when the two datasets, A and B are identical. Such joins will be qualified as '**self** spatial joins'. We will use the term '**cross** spatial joins', when we need to emphasize that the two point sets are distinct. Otherwise, we will simply use the term 'spatial join' to denote a spatial distance join between two distinct datasets.

The goal of this work is to estimate the selectivity of spatial joins among two datasets as opposed to only one. The join selectivity represents the size of the resultant set of the spatial distance join divided by the size of the Cartesian product of the whole data. Estimation of the join selectivity is important for the following two reasons.

- An accurate estimation is necessary to optimize complex queries. Though there has been quite a lot of work done on how to estimate the selectivity of equi-joins, the problem of estimating the size of spatial joins has received only minimal attention up to now.
- In application areas like the ones mentioned earlier, the size of the spatial distance join (as a function of the radius) is important for evaluating the correlation between datasets. Note that it is generally too costly to obtain the size of the spatial join by simply computing the spatial distance join itself. Therefore, an accurate and inexpensive method is required to estimate the size of spatial distance joins.

Our main contribution is that we observe a 'power law', which holds for many pairs of real datasets. We show how to use this power law to accurately estimate the spatial join selectivities efficiently (in constant time, O(1)).

The rest of the paper is organized as follows. Section 2 presents the related work. Section 3 describes our main contribution, the pair-count exponent $\mathcal{P}$ and the fast way to estimate it, through the proposed box-occupancy-product-sum (BOPS). Section 4 discusses implementation and speed issues of the proposed methods. Section 5 gives experimental results, and Section 6 discusses issues for practitioners. Section 7 presents the conclusions.

2. RELATED WORK

There has been quite a lot of work on spatial joins recently. See, for example [ORE 86], [BKS 93], [LR 94], [PD 96], [KS 97], [APR+98] and [MP 99]. Most of the mentioned work has dealt with developing efficient methods to process spatial intersection joins for two-dimensional data sets [BSW 99] [DNS 91] [SK 96] with little emphasis on the estimation of selectivity. Recently, methods have also been examined and developed for processing spatial distance joins on multidimensional point sets [SSA 97], [KS 98]. The term "*similarity join*" has frequently also been used for spatial distance joins in the literature. For one-dimensional data, the spatial distance join corresponds to the 'band-join' [DNS 91].

Although not directly related to our spatial join selectivity, we mention earlier attempts to estimate the selectivity of range queries. Typical methods include the milestone 'uniformity and independence' assumptions [SAC+79]. Although simple to use in a query optimizer, these assumptions are pessimistic and unrealistic [CHR 84]. Modern methods include histograms [Poo97], kernel estimators [BKS99], wavelets [VW99], and hybrid methods using query feedback [KW 99]. Methods for selectivity estimation of range queries in spatial datasets use multi-dimensional histograms [TS 96], or arguments from the theory of fractals [BF 95]. It should be noted that most of these methods are susceptible to the 'dimensionality curse' [SIL 96] [SCO 92].

Analytical estimates of spatial distance join selectivities are few. The very recent work presented in [PMT 99] assumed the data are uniformly distributed in the address space. As mentioned earlier, the uniformity assumption was discredited long ago [CHR 84], [FK 94] as unrealistic and unfeasible. Our experiments in Section 5 indeed show that it is unrealistic. The cost model presented [TSS 98] was built for datasets not uniformly distributed datasets using R-tree-based structures.

In the next sections we proceed with our proposed solution. The major observation is that the selectivity of spatial distance joins follows a power law surprisingly well.

3. PROPOSED METHOD

Our main contribution and its corollaries are discussed below. The problem to be solved is the following.

Given: two point-sets A and B and a radius r

Find: the distribution of the count of pairs, as a function of the radius r.

That is, is this distribution Gaussian? Is it Poisson? Is it Weibul? It turns out that real datasets do not follow any of the traditional statistical distributions. Instead, we show that the distribution of the pair-wise distances follows a *power law*. Table 1 lists symbols used in this document. Next, we describe our power law, as well as several useful properties of its exponent.

3.1. Pair-count function and the PC exponent

We propose to study the probability distribution function of the number of pairs as a function of the distance between those pairs. Specifically, we define and study the pair-count function $PC_{A,B}(r)$, or simply $PC(r)$, of two point-sets A and B used in a spatial join query. It is defined as follows.

Definition 1: For two point-sets A and B, we define $PC_{A,B}(r)$ as the **pair-count function**, that is, the count of pairs within distance r or less. The first member of the pair should belong to point set A, and the second member to point set B.

$PC_{A,B}(r)$ = count(of A-B pairs, within distance $\leq r$)

Some observations are helpful:

- Our $PC(r)$ function roughly corresponds to the 'cumulative probability density function' from statistics.

• We typically omit the subscripts A, B for simplicity.

• The implied distance function can be any L_p norm. We use the $L_{infinity}$ norm unless otherwise specified. The reason is that all the upcoming results hold for any L_p norm, but the formulas are simpler for the $L_{infinity}$ norm.

• For a self spatial join (i.e., A== B) we *omit* the self-pairs, and we count each pair only *once*. That is, if there are N points in the set, we consider $N*(N-1)/2$ pairs. Again, the upcoming results can be easily adapted to handle any of the omitted cases.

For reasons that will soon be obvious, we define the concept of the pair-count plot:

Definition 2: The **pair-count plot**, or simply **PC-plot**, for two point sets A and B is the plot of $PC_{A,B}(r)$ versus r, in log-log scales.

Figure 1 presents (a) a pair-count plot for real datasets in linear scales, and (b) the same pair-count plot in log-log scales (b). The datasets are explained in Section 5. The question is whether functions obey any rules? It turns out that many of them indeed follow a law, specifically a power law, as we discuss next. The experiments we have done with many real datasets show that many of them result in a PC-plot that is almost linear (within 1.5% MLS error and typically less) for a suitable range of distances r (radius from r_1 to r_2). Considering this, we present our major result.

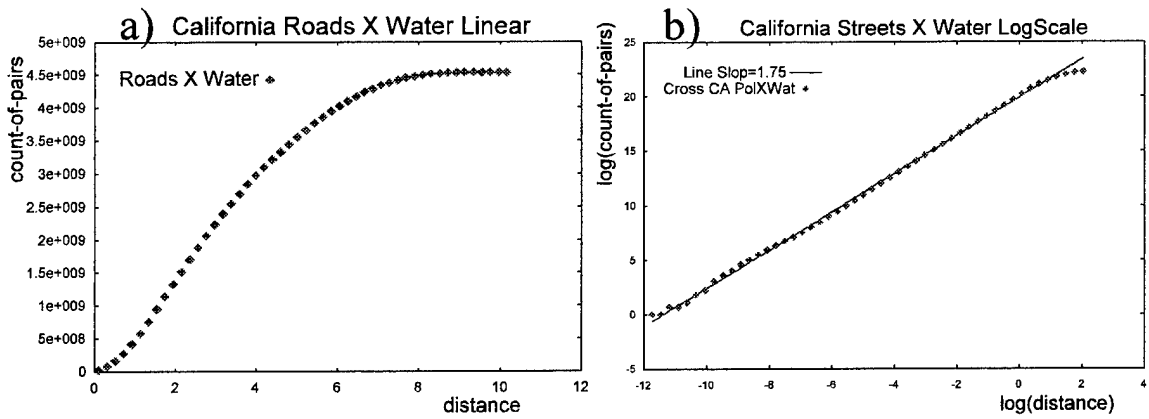

Figure 1 - The Pair-count plot of California datasets (CA-str cross joined with CA-wat) (a) linear scales, and (b) log-log scales

Law 1 (PAIR-COUNT): For several real datasets and for a usable range of scales, the pair-count $PC(r)$ of pairs within distance r or less follows a power law:

$$PC(r) = K \cdot r^{\mathcal{P}} \tag{1}$$

where K is a proportionality constant. Equivalently Definition 3 follows.

Definition 3: The exponent of the law is defined as the **pair-count exponent** $\mathcal{P}$ as

$$\mathcal{P} = \frac{\partial(\log(PC(r)))}{\partial(\log(r))} \tag{2}$$

Figure 1(b) shows the pair-count plot for the same pair of datasets as Figure 1(a) in log-log scales. The plots are clearly linear, for a significant range of scales. This range is usually most sought after for queries; we are not interested in radii much smaller or larger than the typical distances involved in the dataset.

Figure 2 shows PC-Plots and fitting lines for two cross-joins of California datasets, **a** streets cross joined with railroads and **b** streets cross joined with water. The description of these datasets and additional $PC(r)$ plots are shown later in Section 5, which deals with our experiments.

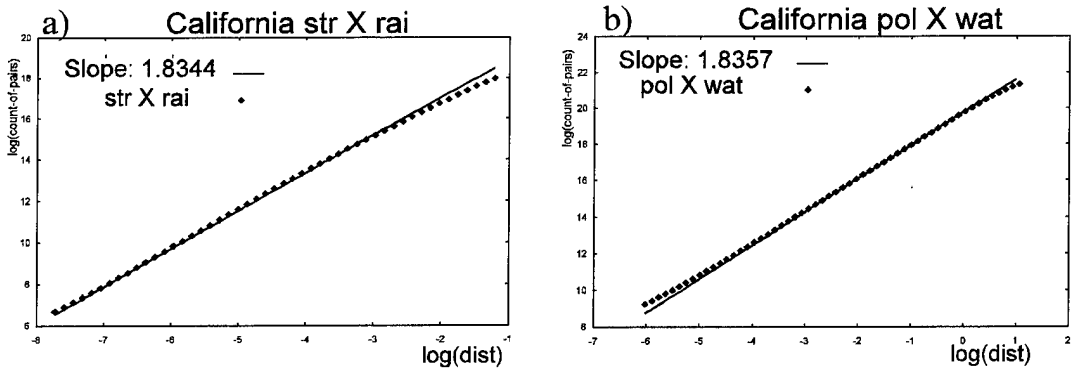

Figure 2 - PC-Plots and slopes of the fitting lines and the pair-count exponent $\mathcal{P}$ for two pairs of California datasets: **(a)** streets cross joined with railroads; **(b)** streets cross joined with water.

3.2. Properties of the pair-count exponent $\mathcal{P}$

The following observations show some of the interesting properties of the pair-count exponent $\mathcal{P}$.

- **Observation 1:** *The pair-count exponent $\mathcal{P}$ includes the "correlation fractal dimension" D_2 as a special case.*

  Justification: When the second dataset is identical to the first, the PC exponent is, by definition, equal to the "correlation fractal dimension" [BELUSSI_95]. Intuitively, this is the 'intrinsic' dimensionality of the dataset.

- **Observation 2:** *The pair-count exponent $\mathcal{P}$ is invariant to affine transformations, namely to translation,*

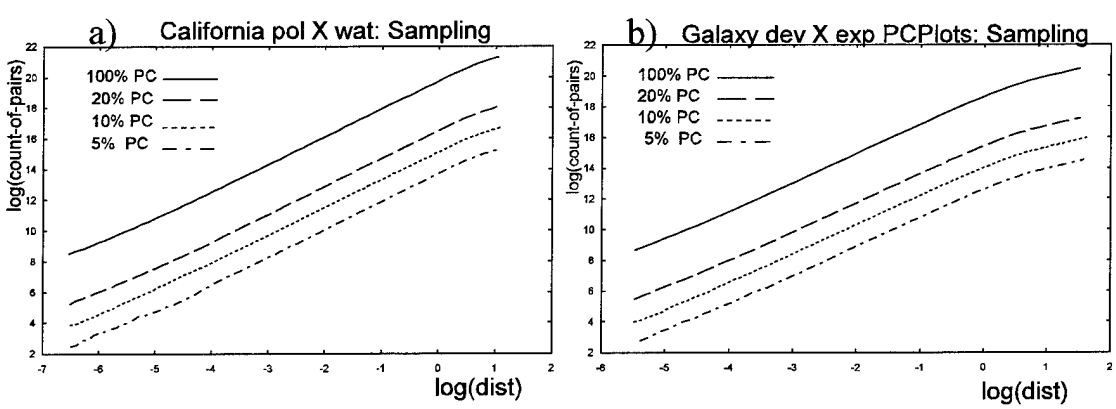

Figure 3 - Illustration of the effects of sampling on the pair-count exponent $\mathcal{P}$. The PC-plots for the full datasets and for 20%, 10% and 5% samples. (a) California pol X wat and (b) Galaxy dev and X exp.

rotation, and uniform scaling.

Justification: By 'uniform scaling' we mean that all the axes are scaled by the same amount. Translation and rotation do not affect the distances and thus leave the plots unchanged. Uniform scaling scales all the distances, and thus shifts the plot to the left or the right. Its slope, however, remains the same.

• **Observation 3:** *The pair-count exponent $\mathcal{P}$ is invariant to sampling.*

Justification: Sampling is useful when we deal with large datasets, although our upcoming BOPS algorithm can handle huge datasets even better. It is useful that our power law holds for subsets of our data. The intuitive argument is as follows. Consider a dataset A with N points and a sampling rate p_a ($0 \le p_a \le 1$), that is the sample has $N*p_a$ points. Similarly, let M be the number of points in dataset B, and let p_b be its sampling rate. Consider a point a_1 from the dataset A and let $a_1(r)$ be the number of its B-type neighbors within distance r. After sampling, it will have $p_a(r)*p_b$ neighbors on the average. Thus, the total number of pairs in the two samples within distance r will be the original $PC(r)$ times p_a*p_b on the average. This will not change the slope of the PC-plot: it will only lower the position of the plot, by $log(p_a*p_b)$.

Figure 3 shows the $PC(r)$ plots for two pairs of datasets. In (a) it shows California political cross joined with California water and in (b) it shows Galaxy-dev cross-joined with Galaxy-exp, as well as their 20%, 10% and 5% samples. Notice that the plots are linear, and those corresponding to samples are parallel to the full dataset. Tables 3 and 4 summarize their $\mathcal{P}$ values.

• **Observation 4:** *The pair-count exponent $\mathcal{P}$ is invariant to the L_p distance used.*

Justification: Consider the 'sphere' that each L_p metric defines (see Figure 4). Let $vol(p,r)$ be the volume of an n-dimensional L_p-'sphere' of radius r. For p=2, this is indeed a sphere; for p=infinity this is an n-dimensional cube, etc. Our power law states that the number of type-B neighbors of a type-A point grows as $r^{\mathcal{P}}$ or, equivalently it grows as volume$^{\mathcal{P}/E}$. Then, if $PC_p(r)$ denotes the number of neighbors within L_p distance r, we have:

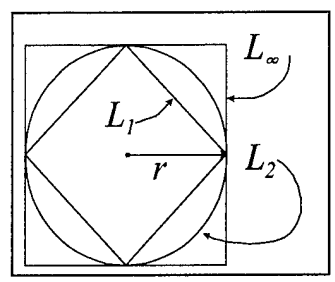

Figure 4 - The shapes of L_∞, L_1 and L_2 norms in 2-d.

$$PC_{AB}(r, L_P) = PC(r, L_\infty) \cdot \left(vol(p,r) / vol(p,r)\right)^{\mathcal{P}/E} \tag{3}$$

therefore, the number of pairs will only differ by a multiplicative constant for different values of p in the L_p

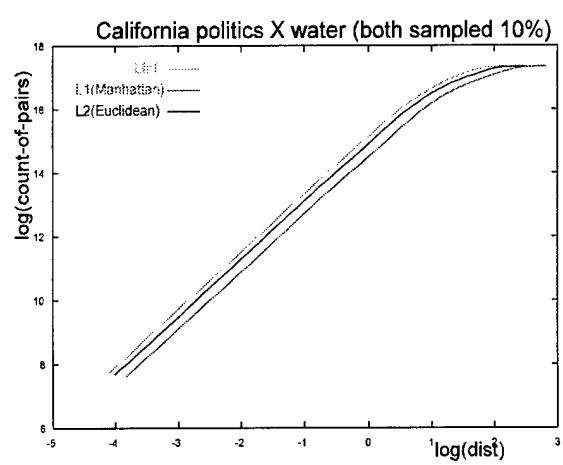

Figure 5 - Effects of the distance functions to obtain PC-plots.

metric. Figure 5 shows the effect of norm invariance on the cross join of two California datasets (political and water). It is clear that the three L_p metrics chosen result in parallel lines. Therefore, for the rest of this work, we will only focus on the L_{inf} metric. We can conclude that the pair-count exponent shows an intrinsic property of the two point-sets, and it is independent of the particular L_p distance function used to build the PC plot.

4. IMPLEMENTATION AND SPEED ISSUES

By the definition of the 'pair-count exponent', we need to estimate the pair-counts for several distances r. Each of them requires $O(N*M)$ operations, which are quadratic on the size of the input datasets. This is prohibitive for large datasets. The question becomes: how we can accelerate the computation of $\mathcal{P}$. This is precisely the topic of this Section.

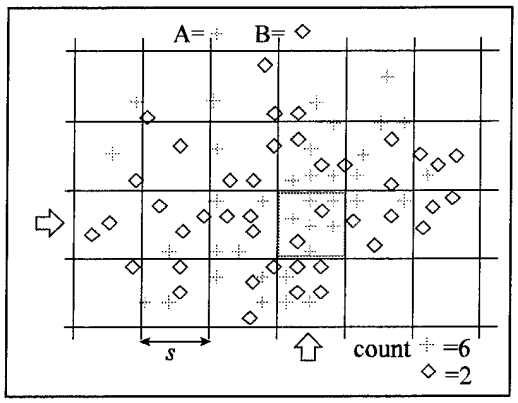

Figure 6 - A grid superimposed over a point-set to count C_{Ai} and C_{Bi}

4.1. A faster way to compute the 'pair-count exponent' $\mathcal{P}$

Here we give a Lemma, which computes of the pair-count exponent $O(N+M)$ and thus performs dramatically faster for huge datasets. A crucial concept that we introduce is the *Box-Occupancy- Product-Sum* (BOPS), which is defined as follows. Consider the address space of two point-sets in a n-dimensional space, and impose an n-grid with grid-cells of side s (or, equivalently, radius $r=s/2$). Focusing on the i-th cell, let $C_{A,i}$, $C_{B,i}$ be the counts ('occupancies') of points from the first and from the second point-set, respectively, as illustrated in Figure 6.

Definition 2: The **"Box-Occupancy-Product-Sum" (BOPS)** of a grid with cell side s is defined as the sum of products of occupancies as

$$BOPS(s) = \sum_i C_{A,i} * C_{B,i} \tag{4}$$

and the **BOPS plot** is the plot of $BOPS(s)$ as a function of the grid side s, in log-log scales.

Lemma 1 (BOPS): The pair-count exponent $\mathcal{P}$ for a given radius is equal to the box-occupancy- product-sum (BOPS) for the doubled radius; that is

$$PC(s/2) \approx BOPS(s) \tag{5}$$

***Proof*:** The fundamental assumption is that the densities of points are smooth functions. Thus, if a point p_1 of set A has x neighbors from the set B within radius r, so does a close-by neighbor p_2 that also belongs to set A.

Thus, for a given cell side s and another given cell (say, the i-th one), consider one of the points of the set A. This point has a number of neighbors proportional to $C_{B,i}$ neighbors from the set B within radius $s/2$. Thus, the i-th cell contributes with

$$C_{A,i} * C_{B,i} \tag{6}$$

pairs. Adding up the contributions of all the cells, we have

$$PC(s/2) = \sum_i C_{A,i} \cdot C_{B,i} \tag{7}$$

which completes the proof.

QED

Corollary: The BOPS follows a power law with its exponent equal to the "pair-count exponent".

$$BOPS(s) = s^p \tag{8}$$

Proof: Trivial, from Lemma 1 and Law 1.

QED

We are going to use the estimation $PC(r) = BOPS(2r)$ for the rest of this work. The 'BOPS' Lemma has important efficiency implications which are vital for large datasets. Next we show how to use this Lemma for fast selectivity estimations.

4.2. Algorithms

The problem is defined as follows.

Given two point-sets A and B in n-dimensional space,

Estimate their pair-count exponent $\mathcal{P}$ and the proportionality constant K.

We developed a *single-pass* algorithm to obtain the BOPS plot. Specifically, the algorithm is linear $O(N+M)$ over the total number of points in both datasets. If l is the number of points that we want in the BOPS plot (ie., number of grid-sizes), then the complexity of our algorithm is $O((N+M)*l*n)$, where n is the dimensionality of the input point-sets. Below is a brief algorithm to generate the BOPS-plot and the estimate of the pair-count exponent.

4.3. Estimation of selectivity

Here we describe exactly how to estimate the spatial join selectivities, exploiting our two major observations, the pair-count law and the BOPS lemma. More specifically, the problem is as follows.

Given two point-sets A and B, and a radius r,

Estimate the count of pairs $PC(r)$.

We distinguish the following methods, depending on what else we are given:

- **PC plot estimation**: Through previously kept statistics on the PC plot, suppose that we already know the pair-count exponent $\mathcal{P}$ and the proportionality constant K. Then we estimate immediately the PC plot as

$PC(r) = K * r^{\mathcal{P}}$

• **BOPS plot estimation**: We assume that we are given only the dataset, without any statistics about the data. Then, we generate the BOPS plot for several values of grid-side s, and we estimate the slope $\mathcal{P}$ and the constant K, as explained in the algorithm in Figure 7. Notice that we not only obtain our estimate, but we also provide $\mathcal{P}$ and K for future upcoming queries.

```
Without loss of generality, due to Observation 2,
Normalize the address space of the datasets to the unit hyper-
cube;
For each desirable grid-size s=1/2^j, j= 1, 2, ..., l;
    For each point a of dataset A
        Decide which grid cell it falls in (say, the i-th cell);
        Increment the count C_A,i;
    For each point b of dataset B
        Decide which grid cell it falls in (say, the i-th cell);
        Increment the count C_B,i;
    Compute the sum of product occupancies ;
                    BOPS(s) = ∑C_A,i *C_B,i
    Print the values of log(s/2) and log(BOPS(s)) as the BOPS-
        plot;
Perform a linear interpolation and report the slope P and the;
proportionality constant K.
```

Figure 7 - Algorithm for calculating BOPS plots.

An obvious trick to approximate the BOPS plot is to do sampling first. We discuss its relative merit in Section 5.

5. EXPERIMENTS

We implemented our method and checked whether the power law holds for different data sets. For the sake of clarity we named the datasets used in the experiments. Point-sets come in groups; thus, each dataset is

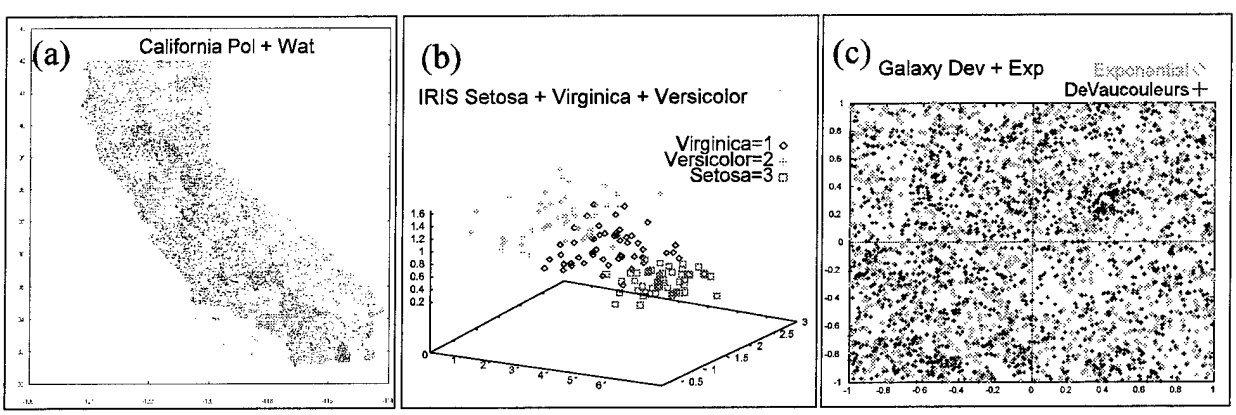

Figure 8 - Real data used in the experiments. **(a)** California: CA-pol and CA-wat, (2-dimensional point-sets), **(b)** Iris: setosa, versicolor and virginica (4-dimensional point-sets) and **(c)** Galaxy: class dev and exp (2-dimensional point-sets).

characterized by its group name, a dash '-' and the dataset name. Their characteristics are as follows.

California - Two-dimensional sets of points, they refer to geographical coordinates in California (see Figure 8(a)). The four files contain data features from streets (CA-str with 62,933 points), railways (CA-rai with 31,059 points), political borders (CA-pol with 46,850 points), and natural water systems (CA-wat, with 1172,066 points) [CEN 89].

Iris - This set contains three files, each of which describes a few properties of a specific flower type of Iris. The points are 4-dimensional (sepal length, sepal width, petal length, petal width); the species are 'virginica', 'versicolor' and 'setosa', and there are 50 points from each species. This is a well-known dataset in the literature of machine learning and statistics, which we obtained from the UC-Irvine Repository (see Figure 8(b)).

Galaxy - Galaxies come from the SLOAN telescope: (x,y) coordinates, plus class label (see Figure 8(c)). There are 82,277 in the 'dev' class (deVaucouleurs), and 70,405 in the 'exp' class (exponential).

Eigenfaces - Two datasets ('lyf' with 11,900 points; and 'tyf' with 3,456 points) come from the Informedia project [WKS+96] at Carnegie Mellon University. Each face was processed with the eigenfaces method [TP91], resulting in 16-dimensional points.

Our experiments are designed to answer the following questions.

- How often do real datasets follow the proposed power law?
- How good is the linear fit?
- How accurate is our 'box-occupancy-product-sum' Lemma?
- What are the effects on sampling and affine transformations on them ?
- How fast is the BOPS method, compared to other estimations of $PC(r)$?

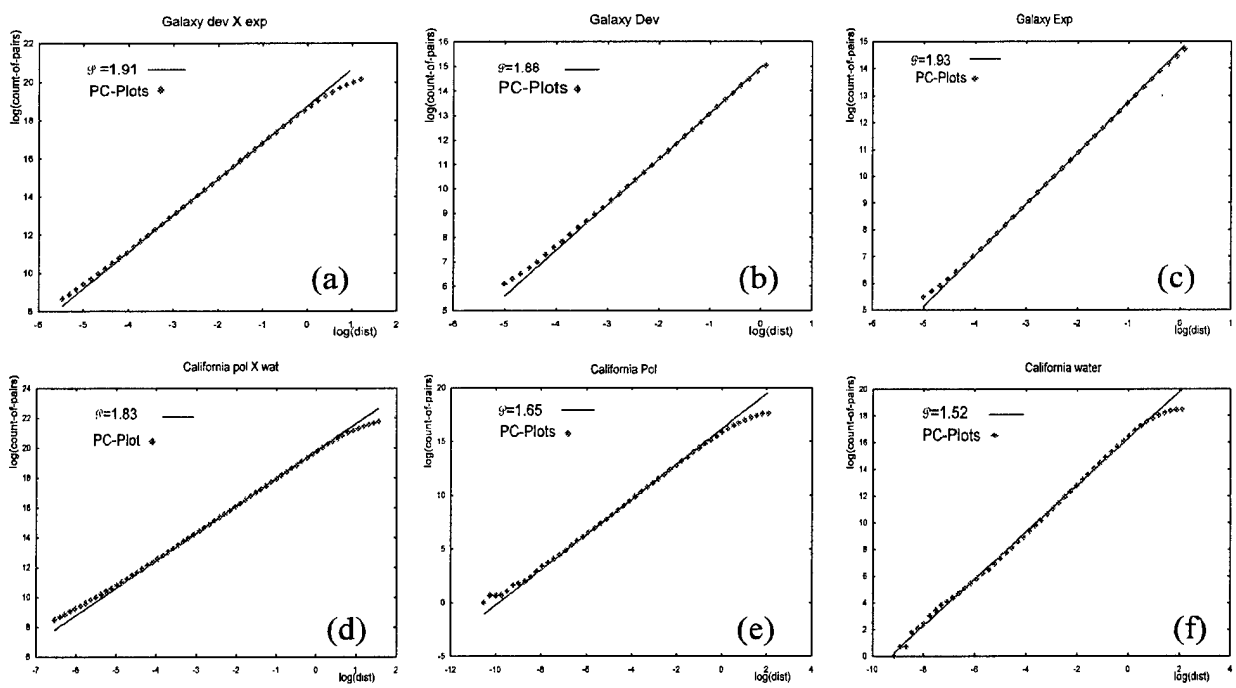

Figure 9 - PC plots and the pair-count exponents ℘ of geographical data. First row: Galaxy datasets (a) cross join of 'dev' and 'exp', (b) self join of 'dev', (c) self join of 'exp'. Second row California datasets (d) cross join of CA-pol and CA-wat, (e) self join of CA-pol, (f) self join of CA-wat.

5.1. Accuracy of '*PC*' Law

We present our experiments in two groups, two-dimensional geographical datasets (California and Galaxy data), and higher-dimensionality ones (Iris, Eigenfaces).

5.1.1 - Geographical datasets

The immediate application for the pair-count exponent is to estimate the selectivities for cross spatial joins. Thus, the natural candidates to show that this method works are geographical datasets. Figure 9 shows the pair-count exponent for California and Galaxy datasets, and it can be seen that the PC plots are linear for a suitable range of r. The slopes of the fitting lines are also shown, and these give us the proportionality constant that will be used to estimate the selectivities in cross or self joins.

5.1.2 - Higher Dimensional datasets

Figure 10 presents the PC-plots, the fitting lines and the pair-count exponent $\mathcal{P}$ for the Eigenfaces datasets which are 16-dimensional data. It can be seen that our power law remains quite accurate for high-dimensional datasets. Recurring conclusions from all the above experiments are:

1. The linear fit implied by our 'pair-count' law is extremely precise, for a wide variety of diverse datasets.
2. For self-joins, as well as for cross-joins, the correlation coefficient of the fit is at least 0.995 (where '1' is the value of perfect linear correlation).
3. Especially for the high-dimensional datasets, the self-join exponent is significantly lower than the embedding dimensionality of the data. For example, in Eigenfaces, the intrinsic dimensionality is between 4.5 to 6.7 (values of $\mathcal{P}$ varies from 4.49 for self-join of 'lyf' to 6.73 for the cross-join of 'tyf' and 'lyf'), while the embedding dimensionality E was 16. This implies that these n-dimensional points are not even close to being uniformly distributed (if they were, then $\mathcal{P} = 16$). Thus, any analysis making the uniform assumption will be very inaccurate, since the dimensionality of the data ($\mathcal{P}$ or E) is in the exponent!

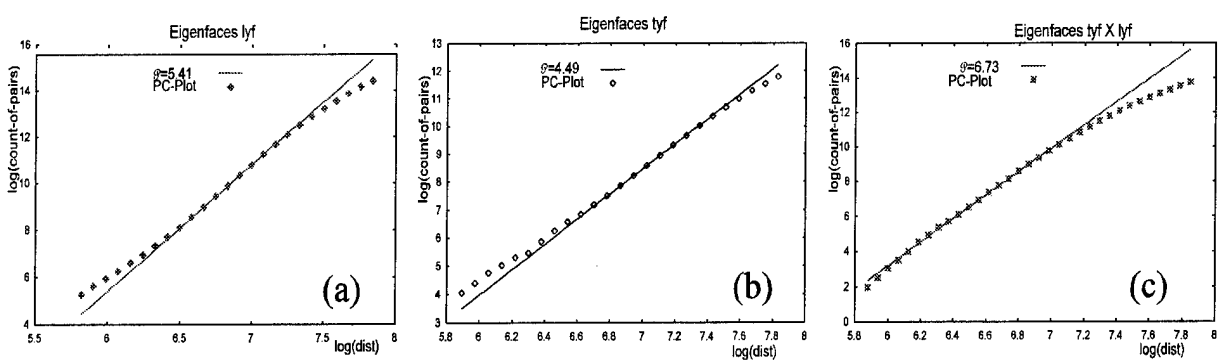

Figure 10 - PC Plots and the pair-count exponent $\mathcal{P}$ of the Eigenfaces datasets, (a) self join of 'lyf' dataset, (b) self join of 'tyf' dataset, (c) cross join of 'lyf' and 'tyf' datasets.

5.2. Sampling

We present further experiments in order to illustrate Observation 3, which states that PC plots are invariant to sampling. Figure 11 presents the pair-count exponents obtained from PC plots (points) and BOPS plots (lines).

All plots are clearly parallel. Table 2 shows the results for the Galaxy and California datasets when the pair-count exponent was calculated for self-joins. Sampling clearly has negligible effects on the PC exponent. Table 3 shows the results for the same datasets using the pair-count exponent obtained from PC plots and from BOPS plots.

Conclusions from the above experiments are as follows.

1). The pair-count exponent $\mathcal{P}$ is practically unaffected by sampling, for reasonable sample sizes (e.g., equal or higher than 10%).

2). Whatever the sampling rate, the corresponding BOPS plot on the samples is very close to the pair-count plot of the samples. This means that whatever the time that sampling can save, BOPS applied on the samples will outperform, with practically the same accuracy.

The estimation of $\mathcal{P}$ obtained from BOPS results on relative error practically always less than 5%. Only when the sampled size of a dataset is very small, the BOPS plot results in a 9% error; indeed, 9% of error is also a reasonable value.

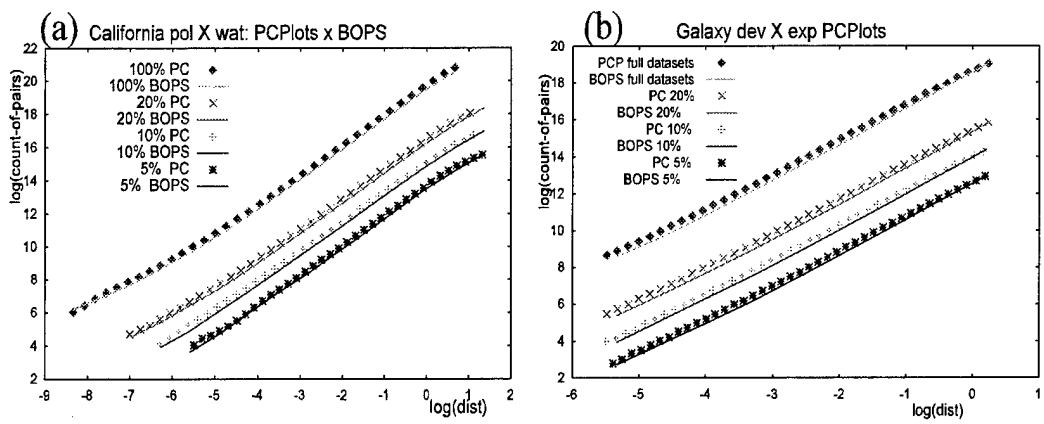

Figure 11 - PC-plots and corresponding BOPS plots for (a) California datasets; (b) Galaxy datasets. Both plots are shown for the full datasets and three levels of sampling.

	Galaxy		California		
Sampling rate	dev	exp	pol	wat	str
100%	1.876	1.928	1.650	1.529	1.838
20%	1.875	1.932	1.643	1.562	1.701
10%	1.873	1.952	1.631	1.694	1.661
5%	1.880	2.146	1.515	1.711	1.623

Table 2: The pair-count exponents $\mathcal{P}$ for samples of Galaxy ('dev' and 'exp') and California (CA_pol, CA_wat and CA_str) datasets for self-joins.

Sampling rate	Galaxy dev x exp		California pol x wat		California pol x str	
	$\mathcal{P}$ from PC	$\mathcal{P}$ from BOPS	$\mathcal{P}$ from PC	$\mathcal{P}$ from BOPS	$\mathcal{P}$ from PC	$\mathcal{P}$ from BOPS
100%	1.915	1.963	1.835	1.819	1.783	1.743
20%	1.915	1.963	1.833	1.825	1.776	1.759
10%	1.902	1.965	1.839	1.816	1.783	1.715
5%	1.918	1.736	1.856	1.786	1.752	1.725

Table 3: The pair-count exponent $\mathcal{P}$ values (PC and BOPS) for joins on sampled data from Galaxy ('dev' and 'exp') datasets and also on California_pol, California_wat, California_str datasets.

5.3. Accuracy of Selectivity Estimations

We see that the pair-count Law is obeyed (Figures 9 and 10). We also have just seen (Figure 11 and Table 3) that our BOPS Lemma leads to very close approximations for the pair-count exponent. The question now becomes how precise the selectivity estimation *PC(r)* can be by using,

(a) our Law 1 and

(b) our estimates from BOPS.

Table 4 shows the relative error for the selectivities calculated by $\frac{PC(r)-\hat{P}C(r)}{PC(r)}$, and we report the geometric average values for several values of *r*. The top row estimates $\hat{P}C(r)$ as follows.

Step (a): Compute the PC plot.

Step (b): Fit the line to obtain the estimation.

In order to measure the relative error in estimating the selectivities of queries, we compared pair-count exponent methods to the real prediction given by Law 1. Table 4 presents the geometric average of the relative error of the PC plot by the pair-count exponents $\mathcal{P}$ when we compare the values obtained from PC and BOPS plots with the actual figures given by Law 1.

	Galaxy			California		
	dev x exp	dev x dev	exp x exp	pol x wat	pol x pol	wat x wat
PC plot estimation	0.02	0.01	0.02	0.02	0.02	0.06
BOPS plot estimation	0.13	0.24	0.25	0.16	0.30	0.34

Table 4 - Geometric average of the relative error of selectivity estimation.

Datasets		PC-Plot (time in sec.)	BOPS (time in sec.)
California	pol x wat (100% of data)	7,752.50	3.44
	pol x wat (10% of data)	73.36	0.5
	str x rai (100% of data)	4,434.27	2.55
	str x rai (10% of data)	42.64	0.47
	pol x str (100 % of data)	7,664.28	3.44
	pol x str (10% of data)	66.58	0.53
Galaxy	dev x exp (100% of data)	13,078.38	5.27
	dev x exp (10% of data)	126.98	0.72
Iris	setosa x virginica	5.32	0.01
	virginica x versicolor	4.98	0.01

Table 5 - Clock time in seconds to obtain the pair-count exponent by PC-plots and BOPS-plots.

5.4. Timing results

The question now becomes: (a) how long it takes to estimate the PC exponent with the PC plot and (b) how long it takes to obtain the estimation from the BOPS plot. Table 5 reports the wall clock times for each plot on an Intel Pentium II 450 MHz, running Windows NT. Both methods were implemented in C++ language.

We can see in Table 5 that there is a huge difference in the CPU time when calculating the PC plots and BOPS plots. Calculating the pair-count exponent using BOPS method save orders of magnitude. Moreover, BOPS plots give a fast and accurate approximation of $\mathcal{P}$. Sampling also gives a close approximation of $\mathcal{P}$, but is much more time-consuming because all the dataset must be scanned in order to generate the sample before to apply the PC plot. When we compare the time needed to obtain the pair-count exponent for a dataset sampled to 10% of the data (a limit to preserve the accuracy of the estimation), BOPS still remains much faster than sampling technique, from 5.27 seconds for the whole dataset for BOPS to 2.11 minutes for a 10% sampling for the PC plot.

Table 5 reports the times needed to build each plot for several pairs of datasets. It also shows the times, when only samples are fed into the two algorithms. The sampling rate is reported on each row, and it is the same for both datasets. The observations are the following:

1). Our BOPS method is up to four order of magnitude faster.

2). In fact, BOPS on the full sets is <u>still</u> faster than the PC plots on the samples (10% sampling rate), up

to 20 times! Thus, we conclude that the BOPS plot is a fast and accurate tool for selectivity estimation of spatial joins.

6. DISCUSSION

Our discussion addresses two questions, which are

a) How often should we expect the 'pair-count' law to hold?

b) How can we use it to do other extrapolations?

6.1. How often?

We mention that power laws regularly occur in real datasets. In fact, our 'pair- count' law is obeyed by the self-join of any self-similar dataset, in which case the 'pair-count' exponent is exactly the correlation fractal dimension D_2 of that dataset. It is well-known that vast majority of real datasets are self-similar [BF 95], coastlines, with fractal dimension 1.1-1.3, stock prices (fractal dimension = 1.5), rain patches (fractal dimension = 1.3), brain surface of mammals (fractal dimension = 2.6-2.7). As we have just seen, the same is true for the self-joins of our real datasets (1.9 for the GALAXY datasets, 1.5-1.8 for the CA datasets, 1.9-2.9 for the 4-dimensional IRIS datasets, and 4.5-5.4 for the 16-dimensional Eigenfaces datasets).

6.2. Other extrapolations

There is a wealth of estimations that we can perform whenever a pair of real datasets obeys the pair-count law, and the invariant properties of the pair-count exponent $\mathcal{P}$. One extrapolation is to estimate the minimum distance r_{min} between the closest pair of points. The formula is

$$PC(r_{\min}) = 1 = K r_{\min}^{\mathcal{P}}$$
$$r_{\min} = K^{-1/\mathcal{P}} \tag{11}$$

The justification comes straightforward from Law 1. We can also estimate the distance r_c of the c-th closest pair and the formula is

$$PC(r_c) = K r_c^{\mathcal{P}} \tag{12}$$

Additional extrapolations can be performed for subsets and supersets of the two original datasets since the pair-count exponent $\mathcal{P}$ is not affected by sampling.

7. CONCLUSIONS

The main contribution of this work is the identification of a power law, namely the 'pair-count' law. This is the *first and only* published law that governs the distribution of pair-wise distances between two real, n-dimensional point-sets. This law leads to the estimation of spatial join selectivities through a simple formula, which is extremely accurate, less than 9% of error. Given the pair-count exponent $\mathcal{P}$, the selectivity estimations can be performed in constant time ($O(1)$) without the need for sampling or any other costly operations. Additional contributions include the following:

• The identification of several invariant properties of the pair-count exponent $\mathcal{P}$. It is invariant to rotation, translation, scaling, sampling. Moreover, this holds for any L_p norm.

• Efficiency issues: the introduction of the BOPS concept (box-occupancy-product-sum). It allows a fast estimation of the pair-count exponent $\mathcal{P}$. Its response time is *orders of magnitude* better than the straightforward estimation using the pair-count function $PC(r)$. Thanks to the BOPS plot, the whole concept of the pair-count exponent becomes practical. In fact, our method used on the full sets, is still significantly faster than the PC plots on samples.

• Experiments on many, diverse datasets. The experiments show that (a) the pair-count law holds for a surprisingly large number of real datasets and (b) that our BOPS approximation is highly accurate. The error is less than 9% for the pair-count exponent $\mathcal{P}$ and less than 35% for the selectivity estimation.

Future research could focus on the discovery of additional power laws in real, spatial datasets, as well as on explaining the reasons why these laws hold.

8. ACKNOWLEDGMENTS

We are grateful for the use of Iris datasets from the UC-Irvine Repository. We would like also to thank Bob Nichol for the Galaxy datasets and the Informedia research group at Carnegie Mellon Univeristy for the Eigenfaces datasets used in this paper.

9. REFERENCES

[APR+98] L. Arge, O. Procopiuc, S. Ramaswamy, T. Suel, J.S. Vitter - *"Scalable Sweeping-Based Spatial Join"*. VLDB 1998, pp. 570-581.

[BF95] A.Belussi and C. Faloutsos - *"Estimating the Selectivity of Spatial Queries Using the 'Correlation' Fractal Dimension"*. VLDB 1995, pp. 299-310.

[BKS 93] T. Brinkhoff, H. P. Kriegel, B. Seeger - *"Efficient Processing of Spatial Joins using R-trees"*, SIGMOD 1993. pp. 237-246.

[BSW 99] J. Van den Bercken, B. Seeger, P. Widmayer - *"The Bulk Index Join: A Generic Approach to Processing Non-Equijoins"*. ICDE 1999, pp. 257.

[CEN 89] Bureau of the Census - *Tiger/Line Precensus Files: 1990 technical documentation.* Bureau of the Census. Washington, D.C. 1989.

[CHR 84] S. Christodoulakis - *"Implications of Certain Assumptions in Database Performance Evaluation"*. TODS 9(2), 1984, pp. 163-186.

[CMN99] S. Chaudhuri, R. Motwani, V. R. Narasayya - *"On Random Sampling over Joins"*. SIGMOD 1999, pp. 263-274.

[DNS91] D. J. DeWitt, J. F. Naughton, D. A. Schneider - *"An Evaluation of Non-Equijoin Algorithms"*. VLDB 1991, pp. 443-452.

[FJS97] C. Faloutsos, H.V. Jagadish and N. Sidiropoulos - *"Information Recovery from Partial data"*. Tech. Report ISR-TR-97-7, Inst. For Systems Research, Univ. of Maryland, College Park, MD, 1997.

[FK94] C. Faloutsos, I. Kamel - "Beyond Uniformity and Independence: Analysis of R-trees Using the Concept of Fractal Dimension". PODS 1994, pp. 4-13.

[FRM 94] C. Faloutsos, M. Ranganathan, Y. Manolopoulos - *"Fast Subsequence Matching in Time-Series Databases"*. SIGMOD 1994, pp. 419-429.

[GÜN 93] O. Günther - *"Efficient Computation of Spatial Joins"*. ICDE 1993, pp. 50-59.

[GÜT 94] R. H. Güting - *"An Introduction to Spatial Database Systems"*. The VLDB Journal. 3(4). October 1994. pp. 357-399.

[HKMRT 96] K. Hätönen, M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen - *"Knowledge Discovery from Telecommunication Network Alarm Databases"*. ICDE 1996, pp.115-122.

[KS 97] N. Koudas, K.C. Sevcik, - *"Size Separation Spatial Join"*. SIGMOD 1997, 324-335.

[KS 98] N. Koudas, K. C. Sevcik, - *"High Dimensional Similarity Joins: Algorithms and Performance Evaluation"*. ICDE 1998, pp. 466-475.

[KW 99] A. Christian Kvnig, G. Weikum - *"Combining Histograms and Parametric Curve Fitting for Feedback-Driven Query Result-size Estimation"*.VLDB 1999, pp.423-434.

[LR 94] M.-L.Lo, C. V. Ravishankar - *"Spatial Joins using Seeded Trees"*. SIGMOD 1994, pp. 209-220.

[MP 99] N. Mamoulis, D. Papadias - *"Integration of Spatial Join Algorithms for Processing Multiple Inputs"*. SIGMOD 1999. pp.1-12.

[MTV 95] H. Mannila, H. Toivonen, A. I. Verkamo - *"Discovering Frequent Episodes in Sequences"*. KDD 1995, pp.210-215.

[NH 94] R. T. Ng, J, Han - *"Efficient and Effective Clustering Methods for Spatial Data Mining"*. VLDB 1994, pp. 144-155.

[ORE 86] J. Orenstein, - *"Spatial Query Processing in an Object-Oriented Database System"*. SIGMOD 1986, pp. 326-33.

[PD 96] J. M. Patel, D. J. DeWitt, - *"Partition Based Spatial-Merge Join"*. SIGMOD 1996, pp. 259-270.

[POO 97] V. Poosala - *"Histogramm-based estimation techniques in databases"*. PhD thesis, Univ. of Wisconsin-Madison, 1997.

[PMT 99] D. Papadias, N. Mamoulis, Y. Theodoridis - *"Processing and Optimization of Multiway Spatial Joins Using R-Trees"*. PODS 1999, pp 44-55.

[SAC+ 79] P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, T. T. Price, - *"Access Path Selection in a Relational Database Management System"*. SIGMOD 1979, pp. 23-34.

[SCO 92] D. W. Scott - *Multivariate Density Estimation*, Wiley & Sons 1992.

[SIL 96] B. W. Silverman - *Density Estimation for Statistics and Data Analysis*. Chapman & Hall 1986.

[SK 96] K. C. Sevcik, N. Koudas - *"Filter Trees for Managing Spatial Data over a Range of Size Granularities"*. VLDB 1996, pp.16-27.

[SSA 97] K. Shim, R. Srikant, R.Agrawal - *"High-Dimensional Similarity Joins"*. ICDE 1997. pp. 301-311.

[TP 91] M. Turk and A. Pentland - "Eigenfaces for Recognition". Journal of cognitive Neuroscience, vol 3(1), 1991, pp. 71-86.

[TS 96] Y. Theodoridis, T. K. Sellis - *"A Model for the Prediction of R-tree Performance"*. PODS 1996, pp.161-171.

[TSS 98] Y. Theodoridis, E. Stefanakis, T. K. Sellis - *"Cost Models for Join Queries in Spatial Databases"*. ICDE 1998, pp. 476-483.

[WKS+ 96] H. D. Wactlar, T. Kanade, M.A. Smith and S. M. Stevens - *"Intelligente Access to Digital Video: Informedia Project"*. IEEE Computer, vol 29(3), pp. 46-52, May 1996.

[VW 99] J. S. Vitter, M. Wang - *"Approximate Computation of Multidimensional Aggregates of Sparse Data Using Wavelets"*. SIGMOD 1999, pp. 193-204.