Selectivity Estimation of Multidimensional Queries Based on Point Synthesis

Matthew Gebski and Raymond K. Wong
National ICT Australia
and School of Computer Science & Engineering
University of New South Wales
NSW 2052, Australia
Email: {francg,wong}@cse.unsw.edu.au

Technical Report
UNSW-CSE-TR-0507
June 2005

SCHOOL OF COMPUTER SCIENCE & ENGINEERING
THE UNIVERSITY OF NEW SOUTH WALES

Abstract

An important problem in database systems is estimating the selectivity of multidimensional queries. While various approaches have been proposed for selectivity estimation for low dimensional spatial databases, many of these techniques have weaknesses for higher dimensional data. This paper presents a novel approach to estimate the selectivity of queries by generating points from an empirical distribution and testing these against a given query. This allows us to take small samples as a summary and generate points for selectivity estimation. Unlike histogram based approaches, our technique does not use containers to represent regions of the data. This alleviates problems that arise when container densities approach zero as dimensionality increases. Experiments show that our approach handles high levels of skew and very high dimensionality, and achieves higher accuracy than previous approaches.

1 Introduction

The problem of selectivity estimation involves the estimation of the size of a result set for a given query from a summary of a database. For instance, a user may be interested in determining the number of residents within a particular area who are single, earn between \$40,000–\$50,000 and work more than 80 hours a week. It may be prohibitively expensive to collect the data or difficult to maintain the entire database in main memory. As such, it is often more practical to first examine a summary of the data possibly followed by a more in depth examination.

This paper examines a technique for generating points from a sample of observations that can be used to help answer such queries over real valued attributes. Unlike many existing approaches, we do not use a bucket for our summary objects. Typically bucket based techniques severely underestimate the selectivity of queries for high dimensional datasets as the density becomes extremely low. Instead, our approach uses a set of sample points as a summary and does not suffer from this problem.

Furthermore, while many good techniques for selectivity estimation on uniformly distributed data exist — histogram techniques are well suited to modelling uniformly distributed data — these techniques do not cope well with skewed data and sparse regions. Therefore, we are motivated to address problems of this type.

Finally, through extensive experiments against a histogram based and the Power Law approaches over both real world and synthetic data, our proposed technique provides a more effective mechanism (in terms of summary size and accuracy) for summarizing high dimensional data.

Unlike most sampling approaches, we use the summary as a guide for estimating points. We aim to have the structure of these generated points consistent with the remainder of the full population. These points are then tested against the query in order to determine selectivity.

Let us consider a univariate, real valued data set, D drawn from a standard normal as depicted in Figure 1 and a query, Q, expressed by the regions enclosed within the dashed lines. Furthermore, assume we have a function $\Delta(X)$ that calculates the expected distance between each point in a sequence of points X and its subsequent point.

In our example, we randomly partition D into two disjoint subsets as in, S_1 and S_2 so that for each point p, $P(p \in S_1) = P(p \in S_2)$. Now, we would expect the value of $\Delta(S_1)$ to be approximately $2 * \Delta(D)$ (the total area is the same, but the number of points within this area has been halved). Furthermore, if we wanted to estimate the location of the points in S_2 solely from S_1 and the knowledge that we have taken a fifty percent sample, we could do it naively as follows. Initially, iterate through S_1', the sequence formed by sorting S_1. Following this, for each point, add a point between it and the following point. The black points in Figure 1 show S_1, while the white points are the estimated points — S_1'. Executing the query on S_1 will give an answer of zero, while executing the query on $S_1 \cup S_1'$ gives the correct answer of 1.

Of course, this approach can not be directly applied to high dimensional data directly, but illustrates the concept behind our approach. Our aim is to estimate points in a similar, albeit more sophisticated, manner to allow us to predict the selectivity of queries from a small sample without requiring storage of the full population. This differs significantly from other sampling approaches which concentrate on methods of improving the quality of the sample.

Our synthesis technique is heavily influenced by statistical goodness of fit tests. We choose points in such a way to ensure a high goodness of fit with our initial empirical sample, assisting us in estimating the location of unsampled points.

The remainder of this paper is organized as follows. In Section 2, we begin with an analysis of related approaches and background knowledge required for our approach. This is followed

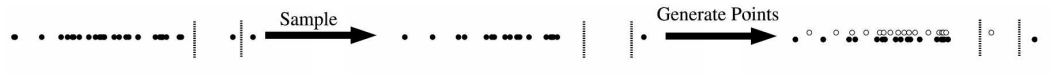


Figure 1: Black represents data from the original data set. White represents generated points. Dashed lines represent a query.

by an outline of our point synthesis technique along with algorithms detailing how it can be applied to selectivity estimation in Section 3. Section 4 provides details of our experimental analysis. We conclude in Section 5 with a discussion and future directions in which our work may lead.

2 Background

2.1 Problem Description and Notation

Our goal is to determine the estimated result size of a query over a database without explicitly answering the query. Let D be a database containing tuples $X = \{x_1, \ldots, x_d\}$ where d is the number of attributes or dimensions. If query $Q = \langle Q_{min}, Q_{max} \rangle$ where $Q_{min} = \{q_{min_1}, \ldots, q_{min_d}\}$ and $Q_{max} = \{q_{max_1}, \ldots, q_{max_d}\}$. We are interested in all entries $X \in D$ such that $q_{min_i} < x_i < q_{max_i}$ where $1 \leq i \leq d$. Alternatively, we can represent the query as a range query in which Q is a single point and Q return all points within radius r for some distance function δ, that is all points $X \in D$ such that $\delta(X, Q) < r$. We refer to size of the result set of a query is the selectivity of the query.

Using R to denote the selectivity of a query, we are interested in a function to estimate this based on a summary of D. We measure the efficacy of our approach firstly based on the relative error $Err_{rel} = 1 - R_{est}/R$. In this paper, we do not use the absolute relative error, as it is important to be able to discriminate between overestimation and underestimation.

2.2 Related Work

There has been extensive research into selectivity estimation. In the XML literature, Markov chains [1] and bloom filters [22] are used to estimate the selectivity of path expressions. In the 1980s, Piatetsky-Shapiro and Connell's proposed an approach for univariate data by controlling the depth, as opposed to the width, of buckets [14]. Muralikrishna and DeWitt [13] extend this approach to handle data with multiple attributes, again maintaining a constant depth and varying the size of the buckets. While these concepts are relevant to selectivity estimation for spatial and multidimensional range queries — particularly, the use of summary histograms — as Acharya et al noted [2], these approaches are not directly applicable to spatial problems. One of the first approaches, Acharya's MinSkew technique was motivated by the need in spatial domains to handle varying counts and depths in addition to varying bucket size.

As with relational selectivity approaches, cell or bucket based histogram techniques are popular for spatial selectivity estimation [2, 5, 7, 13, 15]. The number of points contained within a cell is recorded as well as the location of the cell. For a bucket B of density B_d, when a query is performed, we compute the selectivity as $\sum B_d B_o$, where B_o is the percentage overlap of the query and bucket B.

Construction of a histogram involves dividing buckets at each iteration to improve the quality of the partitioning. Division continues until every bucket contains only one point, a maximum number of buckets is reached or some measure of goodness is met. Typically, the

split is designed to lead to a reduction of the overall badness of the partitioning; MinSkew for instance uses the sum of variances for each bucket as a measure of badness. Other histogram based techniques include the use of kernel estimators to help reduce bucket skew [7].

Another important histogram improvement is the wavelet based histograms by Vitter and Wang [20, 19, 21] who use wavelet decomposition to help represent the underlying distribution. This was later extended by Matias, et al [11] to allow for modifications to the underlying distribution using a probabilistic counting approach and was shown experimentally to be very effective for histograms for one dimension.

While histogram based approaches are accurate for data of lower dimensionality, they do not scale well to higher dimensionalities. Consider a bucket with 160 points covering an area of 4^2 in two dimensional space, B_d is 10; for three dimensions, the bucket's volume is 4^3 and B_d drops to 2.5, for 5 dimensions, B_d is approximately 0.15. B_d approaches zero as the dimensionality increases. This results in a very large underestimation for selectivity for high dimensional data.

Sampling has been repeatedly used as a data reduction technique for database problems. For selectivity estimation, approaches have focused on density estimation where biased samples are taken with an aim to more heavily sample from dense regions of the data. Wu et al [23] suggested using the cumulative distribution function (CDF) for density estimation. Kolios et al [8] use kernel smoothing with application to selection of dense regions to aid in clustering algorithms. Similar techniques (although presented with less direct applicability), such as Fourier series, are outlined by Silverman [17], who notes that selecting samples to model a single normally distributed cluster in ten dimensions would require close to one million points. Chaudhuri et al [4] proposed a two tier sampling based system. Samples are Initially constructed from the database with a predetermined workload., during the query answering phase, queries are modified to use the sample if they are sufficiently similar, allowing the system to return the expected size of the result set in addition to the expected error.

Tao et al [18] build on the work of Faloutsos et al. [6] who suggest that the *paircount* metric, that is, the distance of each point to its k nearest neighbors, can be accurately modelled using a power law. A number of seed points are chosen in dense regions of the data; for each seed point p, the local exponent factor, n_p, and local constant, c_p are calculated. The selectivity is calculated as $c_p * r^{n_p}$, r is the radius of the query from p. A benefit of the Power Law approach is that once seed points have been selected, the time taken to estimate the selectivity is close to constant. Power Law outperforms the histogram approaches and allows a slightly higher dimensionality. However, there are limitations with regards to the locations of the seed points, relative distance of the center of a query to the seed points and the radius of the query which result in varying accuracy.

2.3 Goodness of fit

Goodness of fit tests are statistical tests used to determine the closeness of a sample and a known distribution. Common tests are typically univariate and do not extend well to multivariate data. However, there have been a number of very interesting multivariate goodness of fit tests. Maa, et al. [9], present a method using interpoint distance distributions for non-parametric comparisons of empirical distributions. We provide a summary of this multidimensional approach for the reader.

Given two observations from sample $\mathbf{X} \sim G$ and $\mathbf{Y} \sim F$, a univariate function of the interpoint distances h can be used to determine if $F = G$. This is done by examining the within sample comparisons $h(\mathbf{X}_1, \mathbf{X}_2)$ and $h(\mathbf{Y}_1, \mathbf{Y}_2)$ as well as showing that $h(\mathbf{X}_1, \mathbf{X}_2) = (\mathbf{X}_3, \mathbf{Y}_3)$.

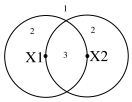


Figure 2: The areas corresponding to h_1, h_2, h_3

This is further developed by Pearl and Bartoszynski [3] to provide a high powered multivariate goodness of fit test. The test is based on three functions:

- $h_1(\mathbf{X_1}, \mathbf{X_2}) =$
  $P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) < min[\delta(\mathbf{X_1}, \mathbf{Y_1}), \delta(\mathbf{X_2}, \mathbf{Y_1})])$
  $+\frac{1}{2}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_1}, \mathbf{Y_1}) < \delta(\mathbf{X_2}, \mathbf{Y_1}))$
  $+\frac{1}{2}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_2}, \mathbf{Y_1}) < \delta(\mathbf{X_1}, \mathbf{Y_1}))$
  $+\frac{1}{3}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_1}, \mathbf{Y_1}) = \delta(\mathbf{X_2}, \mathbf{Y_1}))$
- $h_2(\mathbf{X_1}, \mathbf{X_2}) =$
  $P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) > max[\delta(\mathbf{X_1}, \mathbf{Y_1}), \delta(\mathbf{X_2}, \mathbf{Y_1})])$
  $+\frac{1}{2}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_1}, \mathbf{Y_1}) > \delta(\mathbf{X_2}, \mathbf{Y_1}))$
  $+\frac{1}{2}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_2}, \mathbf{Y_1}) > \delta(\mathbf{X_1}, \mathbf{Y_1}))$
  $+\frac{1}{3}P_Y(\delta(\mathbf{X_1}, \mathbf{X_2}) = \delta(\mathbf{X_1}, \mathbf{Y_1}) = \delta(\mathbf{X_2}, \mathbf{Y_1}))$
- $h_3(\mathbf{X_1}, \mathbf{X_2}) = 1 - (h_1 + h_2)$

Where $\mathbf{X}_1$ and $\mathbf{X}_2$ are two points from $\mathbf{X} \sim F$ and $\mathbf{Y}_1$ is from G; h_1, represents the likelihood of the line $\mathbf{X_1X_2}$ being the shortest side of the triangle i.e. $\mathbf{Y}_1$ falling within area 3 in Figure 2. $\mathbf{X_1X_2Y_1}$, h_2 the second longest side, area 2 in Figure 2, and h_3 the longest side of the triangle, $\mathbf{Y}_1$ in area 3 in Figure 2. This is extended to calculate the probability of a point lying at least $\delta(\mathbf{X_1}, \mathbf{X_2})$ from both $\mathbf{X_1}$ and $\mathbf{X_2}$, U_1, one of $\mathbf{X_1}$ or $\mathbf{X_2}$, U_2, or neither, U_3.

We calculate the measure: $U_k = \frac{1}{\binom{n}{2}} \sum_{i<j} h_k(X_i, X_j)$ for k = 1,2,3. These values, U_1, U_2, U_3, form the basis of the goodness of fit test. Of course, it is not possible to calculate this value for discrete value distributions (such as the empirical distribution for a spatial database). For discrete data, we can use the ratios of counts for the likelihood of selecting a point from each region. That is, $U_k = \frac{1}{\binom{n}{2}} \sum_{i<j} \sum_{m=1}^{n} I_k[Y_m(i,j)|X_i, X_j]$ — $I_k[Y_m(i,j)|X_i, X_j] = 1$ when $Y_m(i,j)$ falls in region k, otherwise it is 0. Importantly, they note that if $F = G$, $E(U_k) = 1/3$ for $k = 1, 2, 3$. There are some additional results that are useful for consideration of univariate data, however in the context of selectivity estimation, we are interested in data with higher dimensionality.

3 Point Estimation

Our approach to selectivity estimation takes a sample from an initial data set to be summarized and uses this sample as a summary from which points not included in the summary points are estimated. These estimated points are then tested against the query in order to determine the final estimate of selectivity. While the choice of sampling function does affect the outcome of our approach — the more representative the sample, the higher the quality of the resulting estimation — for the remainder of the paper, the focus is on generating the points as opposed to techniques designed to improve the quality of the samples.

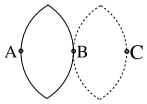


Figure 3: Inserting a point in $R_3(A, B)$ affects $R_3(A, C)$, $R_2(B, C)$ in addition to $R_3(A, B)$

Principally, points are generated to fall within the three regions corresponding to the triangle edges used for Bartoszynski's goodness of fit test. That is, for two points p_i and p_j, the regions:

- $R_1(p_i, p_j)$ — the region at least $\delta(p_i, p_j)$ from both p_i and p_j — $p_i p_j$ is the shortest side of the triangle.
- $R_2(p_i, p_j)$ — the region within distance $\delta(p_i, p_j)$ from *either* p_i or p_j — $p_i p_j$ is the second longest side of the triangle.
- $R_3(p_i, p_j)$ — the region within distance $\delta(p_i, p_j)$ from both p_i and p_j — $p_i p_j$ is the longest side of the triangle.

Points are chosen in a manner designed to keep the expectation of a point falling into any of these three regions equal over the whole data set, for a generated point, $E[p' \in R_1(p_i, p_j)] = E[p' \in R_2(p_i, p_j)] = E[p' \in R_3(p_i, p_j)] = 1/3$ for all p_i, p_j pairs. During the point placement stage, placing a point p' for $R_x(p_i, p_j)$ must fall within either $R_1(p_a, p_b)$, $R_2(p_a, p_b)$, $R_3(p_a, p_b)$ for all other points $a \neq b$ and we must consider how these regions will be affected.

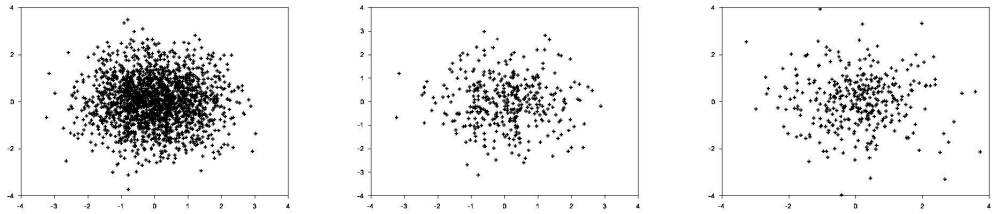
We now suggest three techniques for generating points. The first of these is what we would consider the the naive approach — select pairs and then regions randomly. The second is a holistic approach in which we look at the impact of point placement over the whole data set and the associated regions. The third approach is a heuristic using nearest neighbor pairs to guide the placement.

3.1 Naive Placement

We first present the naive approach and demonstrate why it is inappropriate for this problem. With this technique, each iteration involves selecting two points p_i and p_j, $i \neq j$, at random from S. A region R_x is then selected from $\{R_1(p_i, p_j), R_2(p_i, p_j), R_3(p_i, p_j)\}$ with $P(R_1) = P(R_2) = P(R_3)$. Once R_x has been determined, a point p' is synthesized within R_x. This process is then repeated with another pair p_i, p_j. A variant is to generate a point for all $\binom{n}{2}$ p_i, p_j combinations rather than selecting pairs at random. If the sample size is relatively small, it may be useful to take pairs without replacement from the pool to prevent adversely affecting the final set of synthesized points.

We should consider that inserting a p' in region $R_1(p_i, p_j)$ affects regions of other points and we must consider all other R_1s, R_2s and R_3s that will contain p'. Consider Figure 3, if we insert a p' in region $R_3(A, B)$, as well as contributing to $R_3(A, B)$, x will affect $R_2(B, C)$ and $R_3(A, C)$.

We define a *balanced synthesis* is a set of generated points such that $|R_a| : |R_b| < (1 + \epsilon)$ and $|R_b| : |R_a| < (1 + \epsilon)$ for all pairs $a \neq b$. An *unbalanced synthesis* is any synthesis that is not balanced. For obvious reasons, our point generation techniques aim at producing balanced syntheses. Our two main approaches for this are our holistic approach and our nearest neighbor approach, both of which we now describe.



(a) Initial data set D (b) 20% sample S (c) Generated points S'

Figure 4: An example of points generated from a sample

3.2 Holistic Approach

We now consider our holistic approach, which considers how each generated point affects the regions of the sampled points. As we choose points in a more intelligent manner, the outcome is that we do not suffer from unbalanced syntheses. In a similar manner as the naive approach, we have two variants, a randomized and an all-pairs approach. However, when selecting points for the holistic approach, points are weighted based on the impact they are likely to have on the regions of the other points.

During each iteration, bias is placed towards selecting points that either a) have little effect on the balance of the synthesis or b) may have a large effect, but can be counterweighted by choices made during later iterations. The random holistic approach can be seen in Algorithm 1 For the former, we can choose to reject any (p_i,p_j) pair whose affect is greater than ϵ. For the latter, we can begin choosing points randomly. As the ratios of the regions become skewed, a bias is placed on discarding generated points that increase the skew. We repeat this until a sufficient number of points have been chosen.

For the holistic all-pairs technique, initially, all $\binom{n}{2}$ pairs are considered. For each pair considered, the pair is ranked by examining how much a generated point will affect the regions of other pairs. To generate all the points, a pair is chosen at random, and a point generated that maintains balance. As opposed to simply reconsidering previously used pairs as in the randomized approach, or discarding pairs that do not allow 'useful' synthesized points, at each iteration, we are able to choose a point that affects all regions appropriately. As with the randomized approach, we can choose to select points either to consistently maintain a balanced synthesis, or allow the synthesis to become unbalanced and adjust it at a later stage.

Due to the large number of combinations, this approach may be too slow as we must consider $\binom{n}{2}^2$ pairs to have points generated as well as regions affected by the generated points. However, our nearest neighbor heuristic provides very high accuracy with significantly improved performance - the time taken is proportional to the cost of finding the k nearest neighbors. Indexing structures allow these to be found in $O(|S|^2)$ time for an arbitrary number of dimensions or $O(|S|log|S|)$ time for a low number of dimensions. Figure 4 is an example of the process we follow.

3.3 Nearest Neighbor

At the beginning of the nearest neighbor approach, the nearest neighbors for each point are found. For each of these neighbors, we select a region is selected and used for point generation. An alternative would be to simply use the nearest neighbor, $p_{1_{nn}}$ of each point and generated a number of points between p and $p_{1_{nn}}$. However, problems may arise where fluctuations induced

Algorithm 1 Synthesizes points using the random-holistic variant

GENERATE-POINTS $(S, \epsilon, size)$

1: $Generated \leftarrow \emptyset$
2: $count_{R_1} \leftarrow count_{R_2} \leftarrow count_{R_3} \leftarrow 0$
3: **while** $|Generated| < size$ **do**
4: Select a pair (p_1, p_2) from S
5: Select $R_x(p_1, p_2)$ at random where $x = 1$, 2 or 3
6: Generate p' for $R_x(p_1, p_2)$
7: Calculate affected regions over the whole sample, $R'_k(p_1, p_2)$ for $k = 1, 2, 3$
8: $Ratios \leftarrow$ Calculate the ratios of $count_{R_1} + |R'_1(p_1, p_2)|$, $count_{R_2} + |R'_2(p_1, p_2)|$, $count_{R_3} + |R'_3(p_1, p_2)|$
9: **if** $Ratios < \epsilon$ **then**
10: Update $count_{R_1}$, $count_{R_2}$, $count_{R_3}$ to account for modified $R'_k(p_1, p_2)$ for $k = 1, 2, 3$
11: $Generated \leftarrow Generated \cup \{p'\}$
12: **return** *Generated*

by sampling severely influence the synthesis. While the nearest neighbor approach does not guarantee that the points will necessarily fall into the regions to provide a good balance of region sizes, in practice we find that the synthesis is close to being balanced, although this is distribution dependent. Algorithm 2 describes the approach for generating a nearest neighbor synthesis.

One advantage of using the nearest neighbor approach for synthesis is the ability to use the location of points to provide information for other algorithms that require the knowledge of nearest neighbors. We can perform nearest neighbor queries on a sample and because we know the position of generated points, use these to satisfy other queries such as distance to the furthest of a point's k nearest neighbors without having to determine all nearest neighbors from the full data set.

3.4 Practical Issues

Insertion of points can be implemented quite easily for most cases when Euclidian is the distance measure. The simplest case is to select the region at random for the two points, and randomly place a point along the axis defined by the two points. To generate points randomly within the hyperspheres for the two points, firstly to place a point on the surface of the hypersphere, an n-dimensional Gaussian is generated and then normalized [12, 10]. Secondly, a random distance d is chosen, we take the n-th root and move the point from the surface towards the center accordingly.

Algorithm 2 Generates a new set of point based on the location of each point's nearest neighbors

GENERATE-NN-POINTS (S, k)

1: $Generated \leftarrow \emptyset$
2: **for** $p_1 \in S$ **do**
3: $\quad P_{1NN} = FindKNearest(p_1, k)$
4: $\quad$ **for** $p_2 \in P_{1NN}$ **do**
5: $\quad\quad$ Select $R_x(p_1, p_2)$ at random where $x = 1$, 2 or 3
6: $\quad\quad$ Generate p' for $R_x(p_1, p_2)$
7: $\quad\quad Generated \leftarrow Generated \cup \{p'\}$
8: **return** *Generated*

4 Experimental Evaluation

4.1 Experimental Structure

A comparison is made between the point synthesis approach, the MinSkew histogram algorithm and Tao's Power Law technique. MinSkew was chosen due having been extensively used for comparison in previous work [18, 5, 7]. Tao's Power Law based approach was chosen as a state of the art non-bucket based technique that contrasts with the choice of MinSkew.

When assessing MinSkew, we used hyper-rectangles to simplify the process of determining overlap between the query and histogram buckets. However, there are difficulties involved in generating rectangular test queries due to the high dimensionality. While it is possible to select queries by hand for a small number of queries, this technique is not practical for a large number of queries. The approach of creating a sphere around a point's k nearest neighbours and using the bounding box is inappropriate as the ratio of the size of the sphere and the size of its bounding box approaches zero as dimensionality increases. To the best of our knowledge, there is no standard approach for generating high dimensional rectangular queries of a particular selectivity.

For the experiments presented here, a very large number of spheres of varying size are generated (typically 50,000 spheres). The bounding box is computed and the selectivity recorded, when a query of a certain size is required, it is recalled. For a sphere of 2,000 points, there may be 2,500 points in the hyper-rectangle and this rectangle can then be used for any query requiring 2,500 points. The reader may notice that for each set of experiments, there are two sets of results reported — the first contains MinSkew against our synthesis technique using rectangular shaped queries. The second contains power law against synthesis using spheres. Due to the way points have been generated, our approach can handle any query shape, rectangular, spherical or even hexagonal queries.

All our measurements are in terms of relative error, calculated as $E_{rel} = \frac{(s-s')}{s}$, where s is the true result size and s' is the size of the estimated selectivity. The relative error is not defined if the true selectivity is zero and for this reason included no queries with a result size of zero. Our approach uses the axis based method of placing points into the R_1, R_2 and R_3 regions. As point synthesis is randomized, our simulations are repeated 200 times using the same query and varying the sample. Graphs represent the mean of all results obtained for each query and error bars represent one standard deviation from the mean.

4.2 Datasets

4.2.1 Synthetic Data

Due to space limitations, we are unable to report results for a large number of synthetic distributions. We focus on:

1. A single multivariate normal distribution with 100,000 instances.
2. A "pimple" data set comprised of twenty multivariate normals each with different mean and variance each with 10,000 instances. This set is designed to represent regions of varying densities and allow for queries that overlap multiple distributions; for instance we may see multiple distributions in medical trial data where we have a experimental and control group.

All values are normalized in the range [0, 1].

4.2.2 Real Data

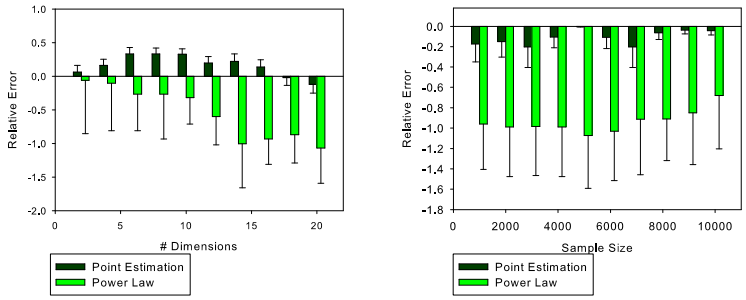
We also look at a number of data sets comprised of real world observations. For each of the real data sets, we normalize the data into the range [0,1].

- Microarray — NIEHS (National Institute of Environmental Health Sciences) microarray data (Global Uterine Genomics in vivo: Microarray Evaluation of the Dstrogen Receptor Alpha-growth Factor Cross-talk Mechanism). This set contains approximately 80 numerical attributes with slightly under ten thousand instances. Although smaller than some of the other data sets, selectivity estimation for data of such high dimensionality is difficult. We create subsets of 20, 40, 60 attributes and also the full set for our analysis.
- Corel Histogram — This set obtained from the UCI KDD archive [16] contains layout information for color histogram layout. The data is comprised of 66616 instances each with 32 real valued attributes.
- Forest Cover — This set is also from the UCI KDD archive and contains information on the forest cover type from the US Forest Service.

4.3 Results

4.3.1 Accuracy for Synthetic Data

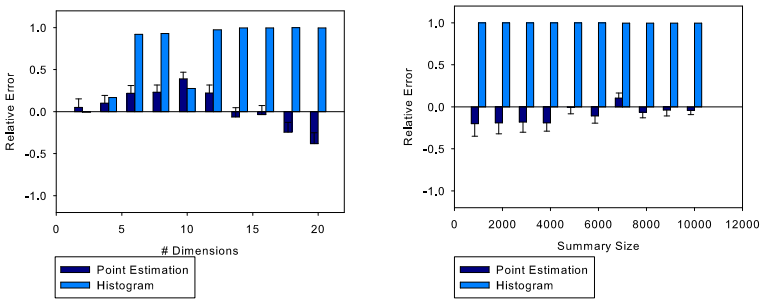
Varying Dimensionality: Figure 5a and Figure 6a demonstrate the effect of varying dimensionality for the three approaches for normally distributed data. Both the sample size and true selectivity used for these experiments were set to 1,000 points (1% of the database). It is clear that MinSkew and Power Law algorithms provide better accuracy for the lower dimensional data. However, for higher dimensionality, the point estimation technique outperforms both of the existing techniques. MinSkew severely underestimates the selectivity for higher dimensions, this is consistent with what we would expect due to the increase in bucket size and corresponding decrease in bucket density. Power Law tends to overestimate the selectivity, this overestimation is slight for sets with lower dimensionality, for sets with 10 or more attributes, this error is quite prohibitive. In contrast, the error for point synthesis is approximately 20% for 5–15 dimensions and after approximately 15 dimensions, begins to overestimate the selectivity. The error for point synthesis becomes less than -1.0 after approximately 30 dimensions.



(a) Varying # dimensions

(b) Varying sample size - 16 dimensions

Figure 5: Normally Distributed Data - Relative error for Power Law and Synthesis



(a) Varying # dimensions

(b) Varying sample size - 16 dimensions

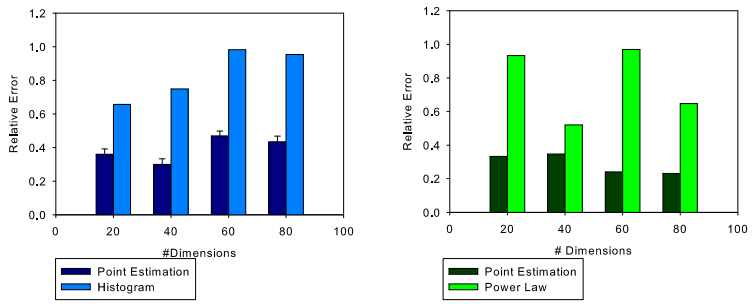
Figure 6: Normally Distributed Data - Relative error for MinSkew and Synthesis

Varying Sample Size: The second set of experiments for the Gaussian data demonstrate the effect of varying the sample size for queries with true selectivity of 1,000 points (again 1%) and 16 attributes. Figure 5b shows the results for the Power Law method while Figure 6b shows the results for MinSkew. As withh the previous experiments for high dimensional data, MinSkew estimates very close to 0 for most queries with a marginal improvement as the sample size approaches 10%. Power Law received a greater benefit from the increase in sample size with the error for 10% slightly greater than 50%. Point Synthesis however outperforms both approaches even with the smaller sample sizes.

Pimple Dataset: Due to space limitations, we do not present graphs for our experiments on the pimples data sets. The results for all approaches are worse than those for the previously mentioned for normally distributed data. Point Synthesis achieves close to 25% error for six dimensions and is close to 40% for dimensions 10–30 — after 40 dimensions, the error is less than -1.

4.3.2 Accuracy for Microarray data

While the full population size for other data sets supported summaries that were very small in comparison to the size of the data set, the size of the microarray forces us to use approximately



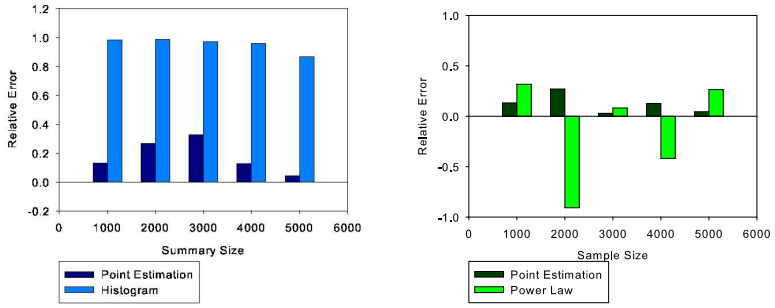
(a) Histogram vs Synthesis (b) Power Law vs Synthesis

Figure 7: NIEHS Microarray Data - Relative error for varying dimensionality

10% of the data. This means samples were 800 points for all experiments.

The microarray data results are very favorable to our approach. The histogram technique has difficulty representing the data given the high dimensionality, although is 'assisted' by the density of the data. In contrast, the point estimation returns consistently good results, especially when considering the high dimensionality of the data. Accuracy for Power Law is somewhat more varied with the estimate being very heavily affected by the particular query and how it is placed in relation to the sample points.

4.3.3 Accuracy for Corel Histogram data



(a) Histogram vs Synthesis (b) Power Law vs Synthesis

Figure 8: Corel Layout — Relative error for varying sample size — 32 dimensions

Against both existing techniques, synthesis provides higher level of accuracy. We should note that Power Law again consistently estimates the result to be much larger than the true selectivity of the query. Further analysis of the particular queries leading to this scenario indicate the overestimation to arise from queries with a large radius. Power Law raises the radius of the query to the local exponent which in sparse areas of a high dimensional data set can lead to gross overestimation of the query. As well as providing better accuracy, point synthesis benefits more in terms of accuracy gains when relaxing the constraints on the size of the sample than either MinSkew or Power Law.

4.3.4 Forest Cover

Our experiments have focused on selectivity estimation for high dimensional data that is hard to model with existing techniques. In our introduction, we considered that data that is uniform, or close to uniform, can be efficiently processed by using methods such as MinSkew and for this reason have so far not considered any data sets of this style. As uniform data by its nature is relatively consistant in terms of density, it exhibits the ideal properties for histogram based techniques. For uniform data, we would expect the point synthesis technique to require much larger summaries in comparison.

For completeness, we consider a 'uniform' set — the forest cover database that has been shown to perform well with Kollios' GenHist histogram technique using kernel estimators [7]. Analysis of the forest cover database is designed to show that unlike many real world datasets, point synthesis is *not* suitable for such datasets with large uniform regions, while existing techniques techniques are ideal.

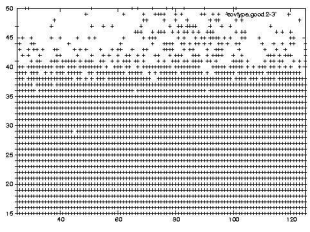


Figure 9: A portion of the Forest Cover data set

We use all 581011 and choose 6 dimensions from the forest cover set — Elevation, Aspect, Slope, Horizontal Distance To Hydrology, Vertical Distance To Hydrology, Horizontal Distance To Roadways. For a query with selectivity of 20,000 and 3,000 sample points/buckets, MinSkew has about 30–50% relative error depending on the exact query, while point synthesis is approximately 90%, Kollios reports marginally lower results, 5–10%, for the GenHist kernel histogram technique. When increasing the sample size to 20,000 buckets, MinSkew has between 5–10%. This contrasts sharply with results for other distributions, however consider Figure 9 which contains a portion of Slope against Horizontal Distance To Hydrology; histogram based techniques are better suited to exploit the uniformity of the data.

4.3.5 Summary of Experiments

There are several key results related to our approach that our experiments have repeatedly highlighted. The point synthesis approach is able to handle data with very high dimensionality, up to 80 dimensions for the microarray data which is much higher than current approaches are able to accurately process. We also note that the synthesis technique is better able to exploit increases in sample size.

5 Conclusions and Future Work

The point synthesis approach we have presented forms a good base for selectivity estimation for multidimensional range queries. It allows accurate estimation for real data sets with high dimensionality and does not suffer from many of the structures such as heavy skew that affects other approaches.

There are a number of avenues that we are pursuing in relation to the research presented in this paper. Firstly, we are interested in more efficient methods for calculating the effect of point generation on other regions for the holistic technique. Heavily linked to improving the holistic technique is the problem of devising an algorithm or heuristic to facilitate efficient computation of regions affected by point synthesis. Other interesting problems include the generation of data from sources containing categorical attributes and inference for temporal processing.

Acknowledgements

The authors would like to thank Prof. Yufei Tao for providing an implementation of the Power Law approach to assist with our experimental evaluation.

References

[1] Ashraf Aboulnaga, Alaa R. Alameldeen, and Jeffrey F. Naughton. Estimating the selectivity of xml path expressions for internet scale applications. In *VLDB*, pages 591–600, 2001.

[2] Swarup Acharya, Viswanath Poosala, and Sridhar Ramaswamy. Selectivity estimation in spatial databases. In *SIGMOD*, pages 13–24, 1999.

[3] Robert Bartoszynski, Dennis K. Peal, and John Lawrences. A multidimensional goodness-of-fit test based on interpoint distances. *J. of the Am. Stat. Ass. - Theory and Methods*, 92(438):577–586, June 1997.

[4] Surajit Chaudhuri, Gautam Das, and Vivek Narasayya. A robust, optimization-based approach for approximate answering of aggregate queries. In *SIGMOD*, pages 295–306, 2001.

[5] Amol Deshpande, Minos Garofalakis, and Rajeev Rastogi. Independence is good: dependency-based histogram synopses for high-dimensional data. In *SIGMOD*, pages 199–210, 2001.

[6] Christos Faloutsos, Bernhard Seeger, Agma Traina, and Jr. Caetano Traina. Spatial join selectivity using power laws. In *SIGMOD*, pages 177–188, 2000.

[7] Dimitrios Gunopulos, George Kollios, Vassilis J. Tsotras, and Carlotta Domeniconi. Approximating multi-dimensional aggregate range queries over real attributes. In *SIGMOD*, pages 463–474, 2000.

[8] G. Kollios, D. Gunopulos, N. Koudas, and S. Berchtold. Efficient biased sampling for approximate clustering and outlier detection in large datasets. *IEEE Transactions on Knowledge and Data Engineering*, 15(5):1170–1187, 2003.

[9] JenFue Maa, Dennis K. Pearl, and Robert Bartoszynski. Reducing multidimensional two-sample data to one-dimensional interpoint comparisons. *The Annals of Stat.*, 24(3):1069–1074, 1996.

[10] G. Marsaglia. Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, 43:645–646, 1972.

[11] Yossi Matias, Jeffrey Scott Vitter, and Min Wang. Dynamic maintenance of wavelet-based histograms. In *VLDB*, pages 101–110, 2000.

[12] Mervin E. Muller. A note on a method for generating points uniformly on n-dimensional spheres. *Commun. ACM*, 2(4):19–20, 1959.

[13] M. Muralikrishna and David J. DeWitt. Equi-depth multidimensional histograms. In *SIGMOD*, pages 28–36, 1988.

[14] Gregory Piatetsky-Shapiro and Charles Connell. Accurate estimation of the number of tuples satisfying a condition. In *SIGMOD*, pages 256–276, 1984.

[15] Viswanath Poosala and Yannis E. Ioannidis. Selectivity estimation without the attribute value independence assumption. In *VLDB*, pages 486–495, 1997.

[16] C.L. Blake S. Hettich and C.J. Merz. UCI repository of machine learning databases, 1998.

[17] B. Silverman. *Density Estimation for Statistics and Data Analysis*. 1986.

[18] Yufei Tao, Christos Faloutsos, and Dimitris Papadias. The power-method: a comprehensive estimation technique for multi-dimensional queries. In *CIKM*, pages 83–90, 2003.

[19] Jeffrey Scott Vitter and Min Wang. Approximate computation of multidimensional aggregates of sparse data using wavelets. In *SIGMOD*, pages 193–204, 1999.

[20] Jeffrey Scott Vitter, Min Wang, and Bala Iyer. Data cube approximation and histograms via wavelets. In *CIKM*, pages 96–104, 1998.

[21] Min Wang, Jeffrey Scott Vitter, Lipyeow Lim, and Sriram Padmanabhan. Wavelet-based cost estimation for spatial queries. In *SSTD*, pages 175–196, 2001.

[22] Wei Wang, Haifeng Jiang, Hongjun Lu, and Jeffrey Xu Yu. Bloom histogram: Path selectivity estimation for xml data with updates. In *VLDB*, pages 240–251, 2004.

[23] Yi-Leh Wu, Divyakant Agrawal, and Amr El Abbadi. Applying the golden rule of sampling for query estimation. *SIGMOD Record*, 30(2):449–460, June 2001.