Practical and efficient geometric ε-approximations [*]

Hüseyin Akcan[†] Hervé Brönnimann[‡] Robert Marini[§]

Abstract. We adapt an algorithm for computing a deterministic sample in a set system to compute ε-approximations for certain geometric set systems. We give algorithms to evaluate the quality of our samples. Our implementation shows that our deterministic samples, although more costly to obtain, significantly outperform random samples in quality. These implementations may have utility in computer graphics and statistics.

1 Introduction

This paper explores consequences of work on data reduction of multi-dimensional data via sampling. In [3, 5, 7–9], we have used ideas originally developed in the field of computational geometry to develop practical deterministic sampling algorithms for multi-dimensional data. The main product of the previous research has consisted of deterministic algorithms (EASE [7,8], Biased-EA and Biased-L2 [3,5]) that find a sample S which optimizes the error of the frequency vector of items over the sample (when compared to the original frequency vector of items). Those algorithms are a clear improvement over simple random sampling (SRS) and other more specialized sampling algorithms such as FAST [11]. Sampling has been widely used in computational geometry, and derandomization methods based on the geometric discrepancy error, ε-approximations and finite VC-dimension, have seen remarkable development in the last two decades [10, 15]. Samples are used to speed up geometric algorithms for divide-and-conquer, and lead naturally to derandomization by using ε-approximations instead of random samples. In this context, one would like to sample objects in some set X (e.g., points with Cartesian coordinates in $\mathbb{R}^d$) and each object belongs to one or more class $\mathcal{R}$ of subsets (also called *ranges*, e.g., halfspaces, disks, simplices, or axis-aligned boxes). The pair $(X, \mathcal{R})$ is called a *set system*, or sometimes also a *range space*.

We illustrate the practical usefulness of our sample with applications. A natural application of samples with geometric count data is range counting. A range (or counting) query gives a range $R \in \mathcal{R}$ and asks for the subset (or number) of objects contained in $\mathcal{R}$. The approach consists of approximating range counting queries by performing them on a sample. The accuracy of the answer depends on the discrepancy of the sample. Another natural application of samples in geometry is in estimating the statistical depth of a point in a point cloud. Many notions of statistical depths [17] exist, all expensive to compute. Among competitors, halfspace depth is one of the most desirable. Outliers consist of points of small depth, and statisticians are interested in computing the deepest point, or depth contours (the region of points at a given depth). All these computations are expensive and are well approximated by performing them on a sample. Again, the accuracy of the answer depends on the discrepancy of the sample. Other applications could include simplification of Point Cloud Data popular in computer graphics for either model reconstruction or rendering (with items representing various quality criteria).

Our contribution. In our previous work, we have considered data arising from transactional databases. In this paper, however, we return to the original geometric setting. The main difference here is that the items / ranges are not given explicitly, as was the case with transactional data sets. Instead, the data implicitly encodes the items as the trace of an infinite set system (halfspaces, disks, simplices, or axis-aligned boxes) over the finite point set. We present two of our algorithms Biased-EA and Biased-L2 in a geometric setting to deterministically sample points from a set of points and we give simulation results to compare our algorithms with random sampling algorithms.

Related work. A seminal result of Vapnik and Chervonenkis shows that a random sample of size $\mathcal{O}(\frac{d}{\varepsilon^2} \log \frac{1}{\varepsilon})$ is an ε-approximation, where d is the VC dimension of the pointset defined as the maximum size of a set of points such that every subset corresponds to the trace of some 1-itemset. This result establishes the link between random samples and frequency estimations over several items simultaneously. Many geometric range spaces have finite VC-dimension d. For literature on ε-approximations and geometric discrepancy, see e.g. the books by Alon and Spencer [4], Chazelle [10, Ch.4], [16]. Algorithms for computing the bichromatic discrepancy of a point set were given by Dobkin *et al.* [12]. Recently, Bagchi *et al.* [6] presented an extension of the ε-approximation technique for geometric data streams, with applications to range counting.

2 Deterministic sampling algorithms

We present (from [3]) two deterministic sampling algorithms, Biased-EA and Biased-L2. Both assume that the set system has been computed explicitly (i.e., for every point in the data set D, an explicit list of ranges (items) that contain

[*] Research of H. Brönnimann and H. Akcan supported by NSF CAREER Grant CCR-0133599. All authors: CIS Dept, Polytechnic University, Six Metrotech, Brooklyn, NY 11201, USA.

[†] hakcan01@cis.poly.edu

[‡] hbr@poly.edu

[§] Othmer Institute Honors College. rmarin01@utopia.poly.edu

the point, is given). All maintain a sample S from the data set D, and work by considering adding the points one by one to the sample. Each point is accepted with a small probability $\alpha < 1$, to yield a sample of size $n = \alpha N$. For each algorithm, we give the code and a short summary of the results when available.

Notation. Let D denote the point set of interest, $n = |D|$ the number of points, S a deterministic sample drawn from D, and $r = |S|$ the number of points in it . The *sampling rate* is $\alpha = r/n$. We denote by $\mathcal{I}$ the set of all ranges (items) that are in D, by m the total number of such ranges, and by $\mathrm{size}(t)$ the number of ranges a point $t \in D$ belongs to. We let T_{avg} denote the average number of ranges a point belongs to, so that nT_{avg} denotes the total description size of D (as counted by a complete range per point enumeration). For a set T of points and a range (item) $A \in \mathcal{I}$, we let $n(A;T)$ be the number of points in T that belong to A and $|T|$ the total number of points in T. Then the support of A in T is given by $f(A;T) = n(A;T)/|T|$. In particular, $f(A;D) = n(A;D)/|D|$ and $f(A;S) = n(A;S)/|S|$. The *discrepancy* of any subset S of a superset $D \subseteq X$ w.r.t. items $\mathcal{I}$ (that is, the distance between S and D with respect to the item frequencies) is computed by using the $Dist_\infty$ metric:

$$Dist_\infty(S, D; \mathcal{I}) = \max_{A \in \mathcal{I}} \Big| f(A;S) - f(A;D) \Big| \qquad (1)$$

A sample S such that $Dist_\infty(S, D) \leq \varepsilon$ is called an *ε-approximation*. Our deterministic algorithms use a different metric, the L2-norm (also called 'root-mean-square' - RMS):

$$Dist_2(S, D; \mathcal{I}) = \sqrt{\sum_{A \in \mathcal{I}} (f(A,S) - f(A,D))^2}. \qquad (2)$$

2.1 Biased-EA

EASE is a deterministic sampling algorithm originally developed in [7, 8]. The EASE algorithm tries to find a small subset having item supports as close as possible to those in the entire data set. The algorithm maintains counts over the set of ranges (items) and prunes the point set by continuously halving the number of points until the sample size r is reached. The decision to keep or discard a point in the halving is based on a penalty function Q_i per item i, which increases exponentially when the item support of the sample deviates from the support in the entire data set.

The Biased-EA algorithm improves the performance of EASE [8] by alleviating the need to loop over the penalties for each halving. To accomplish this, Biased-EA samples the points at a predetermined rate ($\alpha \ll 0.5$), and maintains a biased penalty funtion.

We can show that the following guarantees hold:

Theorem 1 *[3] The Biased-EA algorithm with sampling ratio α produces a sample of discrepancy ε and size $\alpha n(1 \pm \varepsilon)$, where $\varepsilon = \mathcal{O}\big(\sqrt{\log(2m)/(\alpha n)}\big)$. The running time is $\mathcal{O}(nT_{\mathrm{avg}})$ and the space complexity $\mathcal{O}(m + \alpha n T_{\mathrm{avg}})$.*

```
BIASED-L2 $(D, \alpha)$
$S_{Biased-L2} \leftarrow \emptyset$
for each item $i$ in $D$ do
    $n_i \leftarrow r_i \leftarrow 0$
for each point $j$ in $D$ do
    $sum_r \leftarrow sum_n \leftarrow 0$
    for each range $i$ that contains $j$ do
        $n_i \leftarrow n_i + 1$
        $sum_r \leftarrow sum_r + r_i$; $sum_n \leftarrow sum_n + n_i$
    if size$j/2 + sum_r - \alpha \cdot sum_n \leq 0$ then
        ◁ Keep it
        Insert $j$ into $S_{Biased-L2}$
        for each range $i$ in $j$ do
            $r_i \leftarrow r_i + 1$
return $S_{Biased-L2}$
```

Figure 1: The Biased-L2 algorithm.

The algorithm and the proofs of the theorems are omitted here for the sake of brevity. The complete proofs can be found in [3].

2.2 Biased-L2

Biased-L2 (see Figure 1) uses a different penalty function which increases polynomially instead of exponentially as a function of the deviation. As a result, the theoretical guarantee is worse, but in practice, it performs better for RMS as well as discrepancy errors. This may be explained by the fact that when penalties are small, both penalty functions are equivalent to the second order. Biased-L2 also has the advantage of simplicity (the penalty function can be reduced to integer computations) and maintains both the range (item) counts in the whole point set (n_i) and in the sample (r_i) as a byproduct. These make it the algorithm of choice in practice.

Theorem 2 *[3] The Biased-L2 algorithm with sampling ratio α produces a sample of discrepancy ε and size $\alpha n(1 \pm \varepsilon)$, where $\varepsilon = \mathcal{O}\big(\sqrt{(1-\alpha)m/(\alpha n)}\big)$. The running time is $\mathcal{O}(nT_{\mathrm{avg}})$ and the space complexity $\mathcal{O}(m + \alpha n T_{\mathrm{avg}})$.*

3 Implementations

In this section, we describe the pre-processing steps for our algorithms to sample large amounts of geometric data. In order to measure the quality of the samples obtained, we also give algorithms that compute the discrepancy of a sample S with respect to D.

3.1 Halfspace range counting

We consider the range space $(\mathbb{R}^d, \mathcal{H}_d)$ which consists of points in $\mathbb{R}^d$ with ranges as the set $\mathcal{H}_d$ of all halfspaces (for a given $h \in \mathcal{H}_d$, we let h^+ and h^- denote the halfspace above and below h). It is known [16] that

$$D(n, \mathcal{H}_d) := \min_{|S|=n} Dist_\infty(S, \mathbb{R}^d; \mathcal{H}_d) = \mathcal{O}(n^{1/2 - 1/2d}).$$

Given a set D of n points, we wish to compute a sample S of $r \ll n$ points that has a small discrepancy w.r.t. halfplanes $\mathcal{H}_2$. It is known that a random sample of the lines of size r achieves an expected discrepancy of $\mathcal{O}(\frac{\log r}{\sqrt{r}})$.The basic approach will be to use the Biased-L2 algorithm on $(D, \mathcal{H}_{|D})$, where $\mathcal{H}_{|D}$ consists of the $\mathcal{O}(n^2)$ halfspaces defined by points of D. On average, each point belongs to a constant fraction of these halfplanes, and so the complexity of Biased-L2 will be $\mathcal{O}(n^3)$.

In order to alleviate this burden, we are going to compute a random sample of halfplanes. The question arises as to how many random halfplanes we should choose for our items. We need to choose at least $\Omega(r^2/\log r)$ halfplanes, or else the discrepancy can be worse than that of a random sample. In fact, we need to sample the slopes uniformly. The procedure is thus to take a random sample of Kr points, and construct all the $\mathcal{O}(r^2)$ lines joining two of these points (this is similar to what is done for computing efficient partition trees [14]). The halfplanes bounded above by those lines will be the sample of halfplanes defining the items. The sampling algorithm now takes time $\mathcal{O}(nr^2)$.

Computing the halfplane discrepancy of the points can be done by sweeping the dual arrangement of D in time $\mathcal{O}(n^2 \log n)$, keeping track of the counts of S and D above each vertices.

3.2 Traditional range counting (boxes)

We now consider the range space $(\mathbb{R}^d, \mathcal{B}_d)$, where $\mathcal{B}_d$ is the set of all axis-aligned boxes of the form $\prod_{i=1}^{d}(a_i, b_i]$. This range space is closely related to $(\mathbb{R}^d, \mathcal{Q}_d)$ where $\mathcal{Q}_d$ consists of generalized quadrants of the form $\hat{p} = \prod_{i=1}^{d}(-\infty, x_d(p_i)]$, since a count $|S \cap B|$, $B \in \mathcal{B}_d$, is an inclusion-exclusion of 2^d counts $|S \cap Q_j|$ for some quadrants Q_j defined by the 2^d corners of B. It is known that

$$D(n, \mathcal{Q}_d) := \min_{|S|=n} Dist_\infty(S, \mathbb{R}^d; \mathcal{Q}_d) = \mathcal{O}(\log^{d-1} n).$$

From now on, we focus on the quadrant discrepancy in $\mathbb{R}^2$.

The objective as in the halfspace range counting is to find a representative sample of points. Similar to halfspace range counting, a trivial selection of all the ranges $(D, \mathcal{B}_d \cap D)$ forces us to deal with $\Theta(n^2)$ quadrants and Biased-L2 algorithm runtime becomes $\mathcal{O}(n^3)$. One of our options is to compute a sample based on $(D, \mathcal{I}_1)$ vertical and horizontal halfplanes only. In theory, this setting may not work for certain point distributions, but on average, a sample with small discrepancy for those halfplanes is expected to be good for quadrants (and therefore boxes) as well. Since there are $\mathcal{O}(r)$ such slabs, the complexity of Biased-L2 will be $\mathcal{O}(n^2)$. Using more advanced data structures, ordering the items linearly and introducing hierarchy to increment and sum a set of $\mathcal{O}(n)$ counters of interest in $\mathcal{O}(\log n)$ time, we can achieve $\mathcal{O}(n \log n)$ runtime.

Another option is to sample the quadrants to be used as items in Biased-L2: Most of the $\Theta(n^2)$ quadrants will intersect the sample in the same pattern, so we choose all the $\Theta(r^2)$ quadrants defined by the vertices of a grid whose x- and y-projections are the $(n/(Kr))$-quantiles of the x- and y-coordinates of the points, for a sufficiently large constant K. The complexity of Biased-L2 becomes $\mathcal{O}(nr^2)$.[1] We will refer to this approach as the *gridded quadrants algorithm.*

Finally, we can further benefit from combining both previous approaches. Instead of measuring all the horizontal and vertical halfplanes, we choose the $\mathcal{O}(r)$ ones defined by the quantiles in x- and y-directions, so the runtime will be $\mathcal{O}(nr)$.[2] We will refer to this approach as the *quantile halfplanes algorithm.*

The quadrant discrepancy of the sample can be easily computed in $\mathcal{O}(nr(r + \log n))$ time by enumerating all $\mathcal{O}(r^2)$ quadrants defined by S and keeping track of the number of points of S and D inside. (In particular, it is not necessary to consider all $\Theta(n^2)$ quadrants since the discrepancy is necessarily maximized for an open or closed quadrant defined by points in S.)

3.3 Divide-and-conquer schemes

In order to speed up the process for very large data sets, we consider implementing the divide-and-conquer algorithm, by computing a sample of size r for every block of Kr points in D (there are $\lceil n/(Kr) \rceil$ such blocks, and the last block may have fewer than Kr points). For reasons of time, we will not implement divide-and-conquer in this version.

4 Experimental results

In this section we compare Biased-L2 algorithm to simple random sample, both in halfplane and traditional box range space settings. For both range spaces, we created a uniform random pointset, and sampled this pointset using different sampling rates. Each experiment is run 50 times with different random pointsets, and the results are the average, min-max, and standard deviation of the maximum discrepancy values.

4.1 Average and minimum discrepancy

In a full version, we would evaluate experimentally $D(n, \mathcal{H}_2)$ defined above, by taking random samples of size r of a random point set of size n. Note that this is unlikely to yield a tight estimate, and we would also take many explicit constructions into account (van der Corput, Halton-Hammersely, and Faure's constructions, b-ary nets, scrambled versions, lattice sets, etc.; read [16, Chap. 2]).

4.2 Evaluation of Biased-L2

We show how Biased-L2 performs compared to a random sample in detail in Figure 2 with respect to halfplane discrepancy, and in Figure 3 for quadrants discrepancy. For these experiments, we took $n = 10,000$ points uniformly at random in a disk, and computed a sample of size $r = \alpha n$, for various values of α. The plots are given in log-log scale. The boxes, vertical lines and the plot lines represent respectively the standard deviation, min-max values and mean values.

[1] By a combination of prefix sums and range trees, the algorithm can very likely be implemented in $\mathcal{O}(n \log r)$ time, although we yet have to work out all the details; the version we implemented below is the simple $\mathcal{O}(nr^2)$-time algorithm.

[2] Also with appropriate data structures, Biased-L2 can run in $\mathcal{O}(n \log r)$ time. Again, we have implemented only the $\mathcal{O}(nr)$-time algorithm.

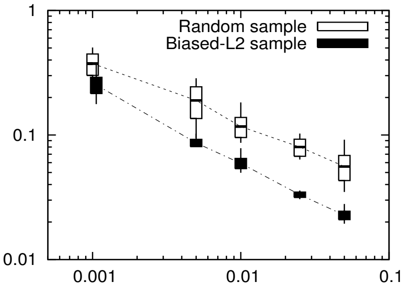

Figure 2: Log-log scale comparison of $Dist_\infty(S_{Biased-L2}, D; \mathcal{H}_2)$ vs. $Dist_\infty(S_{SRS}, D; \mathcal{H}_2)$ for random D of size $n = 10000$, and various values of $r = \alpha n$ (as a function of α).

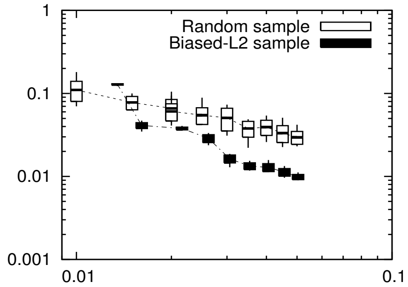

Figure 3: Log-log scale comparison of $Dist_\infty(S_{Biased-L2}, D; \mathcal{Q}_2)$ vs. $Dist_\infty(S_{SRS}, D; \mathcal{Q}_2)$ for random D of size $n = 10000$, and various values of $r = \alpha n$ (as a function of α).

In Figure 2, one can see that in both cases Biased-L2 is clearly superior to a simple random sample, sometimes with a factor of 2 (in normal scale). Moreover the slope of Biased-L2 is slightly more pronounced, meaning that its accuracy gets proportionately better as the sampling ratio gets higher. We clearly see the asymptotic nature of the discrepancy, with a slope of about $-\frac{1}{2}$ in log-log scale corresponding to a discrepancy in $\mathcal{O}(\frac{1}{\sqrt{r}})$. Our algorithm achieves a discrepancy of 0.02 with only 4% of the data, compared with more than 0.06 for a simple random sample of the same size.

In Figure 3, we also see the excellent behavior of the discrepancy for quadrants, for a sample obtained with the quantile halfplanes algorithm.

5 Conclusion

This paper presents preliminary results on deterministically sampling from a geometric data set. Although not different in spirit from the algorithms described in "classical" geometric discrepancy theory by Chazelle, Matoušek, and coll. [10,15,16], our emphasis is on practical sampling methods. The experiments clearly show the superiority of deterministic sampling as opposed to random sampling. In the full version, we wish to expand the experiments as well as work on variants that do provide theoretical guarantees in all cases (by a more judicious choice of the items chosen for the Biased-EA/L2 algorithm) and retain good behavior in practice (both in quality and runtime). We also wish to reinforce our algorithms to produce a reasonably tight upper-bound on the discrepancy as part of the sampling process, to avoid the expensive overcost of the (exact) discrepancy computation afterwards. Note that such upper-bounds are desirable in many applications, such as statistics or metrology.

References

[1] R. Agrawal, T. Imielinski and A. Swami. Mining association rules between sets of items in large databases. *Proc. ACM SIGMOD'93*, pp. 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. VLDB'94*, pp. 487–499, 1994.

[3] H. Akcan, A. Astashyn, H. Brönnimann, and L. Bukhman. Sampling multi-dimensional data. *Technical Report TR-CIS-2006-01*, CIS Department, Polytechnic University, February 2006.

[4] N. Alon and J. H. Spencer. *The probabilistic method.* Wiley Interscience, New York, 1992.

[5] A. Astashyn. *Deterministic data reduction methods for transactional data sets.* Master thesis, Polytechnic Univ., June 2004.

[6] A. Bagchi, A. Chaudhury, D. Eppstein and M.T. Goodrich. Deterministic sampling and range counting in geometric data streams. *Proc. ACM SCG'04*, pp. 144–151, 2004.

[7] H. Brönnimann, B. Chen, M. Dash, P.J. Haas, Y. Qiao and P. Scheuermann. Efficient data-reduction methods for online association rule discovery. Chapter 4 of *Selected papers from the NSF Workshop on Next-Generation Data Mining (NGDM'02)*, pp. 190–208, MIT Press, 2004.

[8] H. Brönnimann, B. Chen, M. Dash, P.J. Haas and P. Scheuermann. Efficient data reduction with EASE. *Proc. ACM KDD'03*, pp. 59–68, 2003.

[9] L. Bukhman. *Approximation of iceberg cubes using data reduction techniques.* Master thesis, Polytechnic Univ., June 2005. `http://www.lbsharp.com/thesis.php`

[10] B. Chazelle. *The discrepancy method.* Cambridge University Press, Cambridge, United Kingdom, 2000.

[11] B. Chen, P.J. Haas and P. Scheuermann. A new two-phase sampling based algorithm for discovering association rules. *Proc. ACM KDD'02*, pp. 462–468, 2002.

[12] D. Dobkin, D. Gunopolos, and W. Maass. Computing the maximum bichromatic discrepancy with applications in computer graphics and machine learning. *J. Comput. Syst. Sci.*, 52(3):453–470, 1996.

[13] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer-Verlag, Heidelberg, Germany, 1987.

[14] J. Matoušek. Efficient partition trees. *Discrete Comput. Geom.* 8:315– 334, 1992.

[15] J. Matoušek. Derandomization in computational geometry. *Journal of Algorithms* 20(3):545–580, 1996.

[16] J. Matoušek. *Geometric Discrepancy.* Springer, 1999.

[17] R. Serfling and Y. Zuo. General notions of statistical depth function *Ann. Statist.* 28(2):461482, 2000.

[18] J. S. Vitter. Random sampling with a reservoir. *ACM Trans. Math. Software* 11(1):37–57, March 1985.