A Sketch-based Sampling Algorithm on Sparse Data

Ping Li PINGLI@STAT.STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Kenneth W. Church CHURCH@MICROSOFT.COM
Microsoft Research
Microsoft Corporation
Redmond, WA 98052, USA

Trevor J. Hastie HASTIE@STANFORD.EDU
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Editor: March 20, 2006

Abstract

We propose a sketch-based sampling algorithm, which effectively exploits the data sparsity. Sampling methods have become popular in large-scale data mining and information retrieval, where high data sparsity is a norm. A distinct feature of our algorithm is that it combines the advantages of both conventional random sampling and more modern randomized algorithms such as local sensitive hashing (LSH). While most sketch-based algorithms are designed for specific summary statistics, our proposed algorithm is a general purpose technique, useful for estimating any summary statistics including two-way and multi-way distances and joint histograms.

Keywords: Random Sampling, Sketches, Data Sparsity

1. Introduction

In databases, information retrieval, and machine learning, there has been considerable interest in sampling techniques (Vempala, 1997; Indyk and Motwani, 1998; Lee et al., 1998; Surajit Chaudhuri, 1998; Manku et al., 1999; Srinivasan, 1999; Achlioptas et al., 2001; Achlioptas and McSherry, 2001; Domingo et al., 2002; Charikar, 2002; Gilbert et al., 2003; Drineas and Mahoney, 2005) for efficiently computing summary statistics, useful for numerous applications including association rules (Brin et al., 1997b,a; Sarawagi et al., 2000; Ravichandran et al., 2005), clustering (Sudipto Guha, 1998; Broder, 1998; Aggarwal et al., 1999; Haveliwala et al., 2000, 2002; Rocke and Dai, 2003), histograms (Gilbert et al., 2002), query optimizations (Matias et al., 1998; Chaudhuri et al., 1999; Dasu et al., 2002; Wu et al., 2003), duplicate detections (Brin et al., 1995; Broder, 1997), and more.

We consider a data matrix $\mathbf{A}$ of n rows and D columns. For example, $\mathbf{A}$ can be the *term-by-document* matrix with n word types and D documents. In modern search engines, $n \approx 10^6 \sim 10^7$ and $D \approx 10^{10} \sim 10^{11}$. In general, n is the number of data points and D is the number of "features."

Three are at least three reasons why sampling can be useful.

- *Sampling can speed up computations*. For example, the cost of computing $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ can be reduced from $O(n^2D)$ to $O(n^2D_s)$ by sampling D_s columns from $\mathbf{A}$. $\mathbf{A}\mathbf{A}^{\mathrm{T}}$ is often called "Gram matrix" in machine learning (especially kernels). Several methods for approximating Gram matrix have been proposed, e.g., (Achlioptas et al., 2001; Drineas and Mahoney, 2005).

- *Sampling can save memory space*. The original data are usually so large that they have to be stored on disks. Disk operations are often the bottleneck in databases and search engines, e.g., (Brin and Page, 1998). A sample may be small enough to reside in the main memory.

- *Sampling can generate stable fingerprint*. Various hashing or sketching algorithms, e.g., (Rabin, 1981), can produce a "sketch" of the data, which is relatively insensitive to changes in the original data. In a broad sense, these sketching algorithms (including Latent Semantic Indexing (Deerwester et al., 1999)) can be considered as sampling methods.

There are two basic strategies of sampling. The *conventional* approach is to draw *random samples* from the data. This approach is simple but often suffers from inaccuracy (i.e., large variances).

A different strategy is *sketching*, which may be regarded as special-purpose lossy data compressions. Sketching involves scanning the data at least once. For example, random projections (Vempala, 2004) multiply $\mathbf{A}$ with a random matrix $\mathbf{R} \in \mathbb{R}^{D\times k}$, whose entries are (typically) i.i.d. samples of standard normals. $\mathbf{AR}$ preserves pairwise distances in the expectation at the cost of $O(nDk+n^2k)$, a significant reduction when $k \ll \min(n, D)$. Sketching algorithms are often more accurate than random sampling because each "sample" of sketches contains more information than a mere random sample. See (Indyk and Motwani, 1998; Indyk, 2000, 2001; Charikar, 2002) for more examples of sketching , local sensitive hashing (LSH), and geometric embedding.

The disadvantage of sketching methods is that they are designed for specific summary statistics. For example, random projections may not be used to estimate 1-norm distance, which is often more robust than 2-norm. Database query optimization requires estimating multi-way joins while many distance-preserving techniques including random projections are restricted to pairwise distances.

There has been interest in combining random sampling with sketching, for example, *data squashing*, (DuMouchel et al., 1999; Madigan et al., 2002; DuMouchel and Agarwal, 2003; Owen, 2003) which generates pseudo data points with weights to approximate the original data distribution.

We propose a new sketching-based sampling algorithm that effectively exploits the data sparsity.

1.1 Data Sparsity

Large-scale datasets are often highly sparse, for example, the term-by-document matrix. While functional words such as "THE" and "A" occur in almost every English document, most words only appear in a very small fraction of documents (Dhillon and Modha, 2001). It is often the case that these infrequent, words such as names, are interesting (e.g., in search engines). Another example is the market basket data, which are also very sparse because typically a customer only purchases a very small fraction of products.

For sparse data, convectional random sampling may not work well because most of the samples are zeros. Sampling fixed D_s columns from the data matrix is also inflexible because different rows may have very different *sparsity factors*, defined as the percentages of non-zero elements.

1.2 Our Method, A Brief Introduction

Our sketch-based algorithm only samples the non-zero elements with flexible sample sizes for different data points. To better explain our method, we start with constructing random samples from a data matrix as shown in Figure 1. Then we show how to generate equivalent random samples using sketches in Figure 2.

In Figure 1, assuming that the column IDs are uniform at random (we will soon discuss how to achieve this), we can simply take the first D_s columns from the data matrix of D columns ($D_s \ll D$ in real applications).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_1	0	1	0	2	0	1	0	0	1	2	1	0	1	0	2
u_2	1	3	0	0	1	2	0	1	0	0	3	0	0	2	1
u_3	0	0	1	4	2	0	1	0	3	0	0	2	0	1	0

Figure 1: A data matrix with $D = 15$. If the column IDs are random, the first $D_s = 10$ columns constitute a random sample. u_i denotes the ith row in the data matrix.

.

For sparse data, we only need to store the non-zero elements in the form of a tuple "ID (Value)," where "ID" is the column ID of the entry in the original data matrix and "Value" is the value of that entry. This structure is often called "postings" (or inverted index). We denote the postings by P_i for each row u_i. Figure 2(a) shows the postings for the same data matrix in Figure 1. The tuples are sorted ascending by the IDs.

P_1:	2 (1)	4 (2)	6 (1)	9 (1)	10 (2)	11 (1)	13 (1)	15 (2)
P_2:	1 (1)	2 (3)	5 (1)	6 (2)	8 (1)	11 (3)	14 (2)	15 (1)
P_2:	3 (1)	4 (4)	5 (2)	7 (1)	9 (3)	12 (2)	14 (1)	

(a) Postings

K_1:	2 (1)	4 (2)	6 (1)	9 (1)	10 (2)	
K_2:	1 (1)	2 (3)	5 (1)	6 (2)	8 (1)	11 (3)
K_3:	3 (1)	4 (4)	5 (2)	7 (1)	9 (3)	12 (2)

(b) Sketches

Figure 2: (a) Postings consist of tuples in the form "ID (Value)," where "ID" is the column ID of the entry in the original data matrix and "Value" is the value of that entry. (b) Sketches are simply the first few entries of postings. In this example, K_1, K_2, and K_3, are the first $k_1 = 5$, $k_2 = 6$, and $k_3 = 6$ elements of P_1, P_2, and P_3, respectively. Let $D_s = \min(\max(\text{ID}(K_1)), \max(\text{ID}(K_2)), \max(\text{ID}(K_3)))$ = $\min(10, 11, 12) = 10$. We should then exclude the entries 11(3) in K_2 and 12(2) in K_3 from the samples.

We sample directly from beginning of the postings as shown in Figure 2(b). We call the samples "sketches." A sketch, K_i, of postings P_i, is the first k_i entries (i.e., the smallest k_i IDs) of P_i. The central observation is that if we exclude all elements of sketches whose IDs are larger than

$$D_s = \min\left(\max(\text{ID}(K_1)), \max(\text{ID}(K_2)), \max(\text{ID}(K_3))\right),$$

we can get exactly the same samples as if we directly sampled the first D_s columns from the data matrix in Figure 1. This way, we can convert sketches into random samples by conditioning on D_s,

which we do not know in advance. For example, when estimating pairwise distances for all n data points, we will have $\frac{n(n-1)}{2}$ different values of D_s.

Our algorithm consists of the following steps:

- Construct sketches for all data points.
- Construct equivalent random samples from sketches online. Depending on the goal, we can construct different random samples from the same sketches.
- Estimate the original space. This step can be very simple, by scaling up (by a factor of $\frac{D}{D_s}$) any summary statistics computed from the samples. In this study, we will show that we can often do better if we take advantage of the marginal information. The estimation task will be slightly more involving but still follows simple statistical principles.

Readers may have noticed that our sketch construction is similar to Broder's approach(Broder, 1997) with some important distinctions. We will compare with Broder's sketches in Section 3.4.

1.3 Paper Organization

2. Theoretical Framework

This section studies in more details why our sketch-based sampling algorithm works.

Compared with *conventional random sampling*, which randomly selects D_s columns from the data matrix $\mathbf{A}$ of D columns, our algorithm only samples the non-zero elements and offers the flexibility of varying the sample (sketch) sizes according to the sparsity of each row of data.

Compared with other sketching algorithms, our method has the distinct advantage that we construct random samples online. Thus, our algorithm can estimate any summary statistics, not restricted to, say, pairwise distances. Statistical tools for random sampling are abundant.

As indicated in Figures 1 and 2, in order for our algorithm to work, we have to make sure that the columns are random. This can be achieved in various ways, e.g., hashing (Rabin, 1981; Broder, 1997). For simplicity, we apply a random permutation[1], denoted by π, on the column IDs, i.e.,

$$\pi : \Omega \to \Omega, \qquad \Omega = \{1, 2, 3, ..., D\}. \tag{1}$$

Let $\pi(\mathrm{P}_i)$ denote the postings P_i after permutation. Recall a sketch K_i is the k_i smallest elements in $\pi(\mathrm{P}_i)$. Thus, we have to scan $\pi(\mathrm{P}_i)$ to find the k_i smallest. This takes time $O(D)$ assuming $k_i \ll D$. Therefore, generating sketches for $\mathbf{A} \in \mathbb{R}^{n \times D}$ costs $O(nD)$, or $O(\sum_{i=1}^{n} f_i)$, where f_i is the number of non-zero elements in the ith row, i.e., $f_i = |\mathrm{P}_i|$.

Apparently it is reasonable to assume that f_i's are known. In general, we can assume that all marginal information (e.g., marginal norms, marginal histograms) are known.

2.1 Properties of D_s

The *effective sample size* D_s is computed online. Suppose we are interested in some summary statistics (e.g., multi-way associations) involving data u_1, u_2, ..., u_m, then

$$D_s = \min\left(\max(\mathrm{ID}(\mathrm{K}_1)), \max(\mathrm{ID}(\mathrm{K}_2)), ..., \max(\mathrm{ID}(\mathrm{K}_m))\right). \tag{2}$$

1. Generating a uniform sample of random permutation on $\Omega = \{1, 2, 3, ..., D\}$ is similar to card shuffling (Aldous and Diaconis, 1986). A well-known algorithm in (Knuth, 1997, Algorithm 3.4.2.P) takes $O(D)$ time.

We have two important approximations, justified in Appendix A.

$$\mathrm{E}\left(\frac{D_s}{D}\right) \approx \min\left(\frac{k_1}{f_1}, \frac{k_2}{f_2}, ..., \frac{k_m}{f_m}\right), \tag{3}$$

$$\mathrm{E}\left(\frac{D}{D_s}\right) \approx \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}, ..., \frac{f_m}{k_m}\right), \tag{4}$$

which are quite intuitive. Since the column IDs are assumed to be uniform in $\Omega = \{1, 2, ..., D\}$ at random, it is expected that $\mathrm{E}\left(\frac{\max(\mathrm{ID}(\mathrm{K}_i))}{D}\right) \approx \frac{k_i}{f_i}$. From (2), we can expect that (3) holds with high accuracy. (4) is just the reciprocal. In fact, in our experiments, we observe that (3) and (4) are very accurate when $k_i \geq 10 \sim 20$. We will use (3) and (4) as if they were exact.

We define $\frac{f_i}{D}$ to be the *sparsity factor* of row u_i. The more sparse, the more efficient our algorithm. From (3) and (4), we can infer that $D_s \approx k\frac{D}{f}$ (suppose all $f_i = f$ and $k_i = k$). If $\frac{f}{D} = 10^{-3}$, then $D_s \approx 10^3 k$, i.e., 10 sketch samples can be equivalent to 10^4 regular random samples!

2.2 Estimation Methods

The estimation task can be very simple. Since our algorithm generates equivalent random samples, we can estimate the original space from samples by a simple scaling.

An important part of our work is to develop estimators taking advantage of the marginal information, which in many situations, can improve the accuracy substantially. For example, estimating two-way contingency tables may benefit considerably from knowing the margins.

We will focus on the following scenarios:

- Two-way and Multi-way associations in boolean data.[2]
- Histograms in binned data (including integer-valued data).
- Inner products in general (real-valued) data.

2.3 Evaluations

Although some MSN Web crawl data are tested to verify the theoretical results, most of our evaluations will be based on comparisons with well-known algorithms in terms of the estimation variances.[3] We will show that

- In boolean data, our algorithm is roughly twice as accurate as Broder's well-known (minwise) sketch method in estimating two-way associations or resemblance.
- In boolean data, our algorithm is (almost) always more accurate than random projections.
- Our algorithm is about the same as random projections in normal-like data. Random projections can be more accurate in heavy-tailed data, while our algorithm can be more accurate in nearly independent data or highly sparse data.

2. Some of the results on two-way and multi-way associations in boolean data were reported in a technical report (Li and Church, 2005). We include these results to give a complete description of our algorithm.
3. The variances serve two main purposes. First, we can compare our method with other algorithms by the variances. Second, we can choose sample sizes by controlling variances. Because all estimators we study are single-modal and either unbiased or asymptotically unbiased, variances often suffice for analyzing the estimation errors in practice.

3. Two-way and Multi-way Associations in Boolean Data

Boolean data (i.e., taking values in $\{0, 1\}$) are important in many applications. For example, the market basket data are often boolean. The term-by-document matrix is sometimes quantized to be boolean. In databases, multi-way set intersections often do not consider duplicates (if any) hence also involve only boolean data. With boolean data, the representation of sketches, i.e., the tuple "ID (Value)" can be simplified because we only need the column IDs.

Suppose we are interested in the associations (intersections) among m rows of boolean data, which in terms of postings, are denoted by P_1, P_2, ..., P_m. There are $N = 2^m$ different combinations of intersections, denoted by $x_1, x_2, ..., x_N$:

$$
\begin{aligned}
&a = x_1 = |\mathrm{P}_1 \cap \mathrm{P}_2 \cap ... \cap \mathrm{P}_{m-1} \cap \mathrm{P}_m|,\\
&x_2 = |\mathrm{P}_1 \cap \mathrm{P}_2 \cap ... \cap \mathrm{P}_{m-1} \cap \neg \mathrm{P}_m|,\\
&...,\\
&x_{N-1} = |\neg \mathrm{P}_1 \cap \neg \mathrm{P}_2 \cap ... \cap \neg \mathrm{P}_{m-1} \cap \mathrm{P}_m|,\\
&x_N = |\neg \mathrm{P}_1 \cap \neg \mathrm{P}_2 \cap ... \cap \neg \mathrm{P}_{m-1} \cap \neg \mathrm{P}_m|,
\end{aligned}
$$

which can be directly related to the binary representation of integers (see Table 1). Here we also denote $x_1 = a$, which is our symbol for intersections and inner products.

Table 1: We number $x_1, x_2, ..., x_N$ according to binary numbers. A "0" indicates that P_j is included in the intersection, a "1" indicates the complement. This way, the binary representation of the subscript of x_i minus 1 (i.e, $i - 1$) corresponds to the set intersections. For example, if $m = 3$, the binary representation of 4 is "1 0 0," indicating $x_5 = |\neg \mathrm{P}_1 \cap \mathrm{P}_2 \cap \mathrm{P}_3|$.

	P_1	P_2
x_1	0	0
x_2	0	1
x_3	1	0
x_4	1	1

(a) $m = 2$

	P_1	P_2	P_3
x_1	0	0	0
x_2	0	0	1
x_3	0	1	0
x_4	0	1	1
x_5	1	0	0
x_6	1	0	1
x_7	1	1	0
x_8	1	1	1

(b) $m = 3$

For each P_i, we take the smallest k_i elements to form a sketch, K_i. Let

$$
D_s = \min\{\max(\mathrm{K}_1), \max(\mathrm{K}_2), ..., \max(\mathrm{K}_m)\}.
$$

After excluding the elements in all K_i's that are larger than D_s, we intersect the trimmed sketches to generate the sample counts. The samples are denoted by $\mathbf{S} = [s_1, s_2, ..., s_N]^{\mathrm{T}}$. Similarly, we let the original space $\mathbf{X} = [x_1, x_2, ..., x_N]^{\mathrm{T}}$. Conditioning on D_s and assuming "sample-with-

replacement[4]," the samples follow a multinomial distribution (conditional on D_s, which is random)

$$\mathbf{Pr}(\mathbf{S}|D_s;\mathbf{X}) \propto \prod_{i=1}^{N}\left(\frac{x_i}{D}\right)^{s_i} \propto \prod_{i=1}^{N} x_i^{s_i}. \tag{5}$$

The most straightforward (unbiased) estimator would be

$$\hat{x}_{i,MF} = \frac{D}{D_s}s_i, \qquad 1 \le i \le N \tag{6}$$

where we use the subscript "MF" to indicate "Margin-free," i.e., not using any marginal information.

From the property of a multinomial distribution, we can compute the variance of $\hat{x}_{i,MF}$

$$\begin{aligned}
&\text{Var}\left(\hat{x}_{i,MF}\right) = \text{E}\left(\text{Var}\left(\hat{x}_{i,MF}|D_s\right)\right)\\
=&\text{E}\left(\frac{D^2}{D_s^2}D_s\left(\frac{x_i}{D}\right)\left(1-\frac{x_i}{D}\right)\right) = \text{E}\left(\frac{D}{D_s}\right)\frac{1}{\frac{1}{x_i}+\frac{1}{D-x_i}}\\
\approx&\max\left(\frac{f_1}{k_1},...,\frac{f_m}{k_m}\right)\frac{1}{\frac{1}{x_i}+\frac{1}{D-x_i}}.
\end{aligned} \tag{7}$$

3.1 A Margin-constrained MLE Estimator

We can improve the estimates using the margins, denoted by $\mathbf{F} = [f_1, f_2, ..., f_m, D]^{\text{T}}$, where $f_i = |\text{P}_i|$. The margin constraints can be represented in a linear matrix equation $\mathbf{CX} = \mathbf{F}$, where $\mathbf{C}$ is the constraint matrix, e.g.,

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad (m=2) \qquad \mathbf{C} = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \end{bmatrix}, \quad (m=3) \tag{8}$$

which basically revert the bit values in Table 1. Note that $\mathbf{C}$ is generated automatically.

The margin-constrained maximum likelihood estimator (MLE) amounts to a standard convex optimization problem,

$$\begin{aligned}
&\text{minimize} \quad -Q = -\sum_{i=1}^{N} s_i \log x_i,\\
&\text{subject to} \;\; \mathbf{CX} = \mathbf{F}, \;\; \text{and} \;\; \mathbf{X} \succeq \mathbf{S},
\end{aligned} \tag{9}$$

where $\mathbf{X} \succeq \mathbf{S}$ is a compact representation for $x_i \ge s_i, 1 \le i \le N$. This program can be solved by standard methods such as the Newton's method (Boyd and Vandenberghe, 2004, Chapter 10.2).

Note that the total number of constraints is $m+1$ and the total number of variables (cells) is $N = 2^m$, i.e., the number of degrees of freedom would be $2^m - (m+1)$, increasing exponentially

4. Since $D_s \ll D$, it is reasonable to assume "sample-with-replacement" for simplicity. However, this assumption is not necessary if we do not intend to take advantage of the margins, as the margin-free estimator $\hat{x}_{i,MF}$ is still unbiased by the property of a multivariate hypergeometric distribution. Assuming "sample-with-replacement" will slightly over-estimate the variance. See (Rosen, 1972a,b) for rigorous analysis of "sample-without-replacement."

fast. Therefore, we expect that margins will not help much when (e.g.,) $m > 4$. Margins help the most when $m = 2$, i.e., only one degree of freedom. When $m \leq 4$, this small optimization problem can be solved very easily.

Historical note Estimating contingency tables under marginal constraints dated back to 1940's, in studying the census of population data (Deming and Stephan, 1940), where the marginal information was available. Deming and Stephan (1940) developed a straightforward iterative estimation method called *iterative proportional scaling*. They first scaled the contingency table row-wise to satisfy the row marginal constraints then scaled the table column-wise to satisfied the column marginal constraints and repeated the procedure iteratively till convergence. They hoped that this procedure could minimize a chi-square statistic (in the same form of a weighted least square problem), which is different from the maximum likelihood approach. Later Stephan (1942) proved the convergence of the *iterative proportional scaling* algorithm in the case of two-way contingency tables and also showed that this algorithm only gave an approximate solution to the least square problem. Fienberg (1970) further proved the convergence of *iterative proportional scaling* in the general case.

We experiment with the *iterative proportional scaling* algorithm and find out that its solutions are often close to the solutions given by (9).

It turns out that for the important special case of $m = 2$, there is a closed-form solution. The estimator for $x_1 = a = |\mathrm{P}_1 \cap \mathrm{P}_2|$ is the solution to a cubic equation

$$\frac{s_1}{x_1} - \frac{s_2}{f_1 - x_1} - \frac{s_3}{f_2 - x_1} + \frac{s_4}{D - f_1 - f_2 + x_1} = 0. \tag{10}$$

3.2 Covariance Estimation

In Appendix B, we provide the (asymptotic) covariance matrix of $\hat{\mathbf{X}}$ estimated by MLE. In particular, for $m = 2$, we can write down the variance of $\hat{x}_{1,MLE}$ explicitly as

$$\mathrm{Var}(\hat{x}_{1,MLE}) = \mathrm{E}\left(\frac{D}{D_s}\right) \frac{1}{\frac{1}{x_1} + \frac{1}{f_1 - x_1} + \frac{1}{f_2 - x_2} + \frac{1}{D - f_1 - f_2 + x_1}}. \tag{11}$$

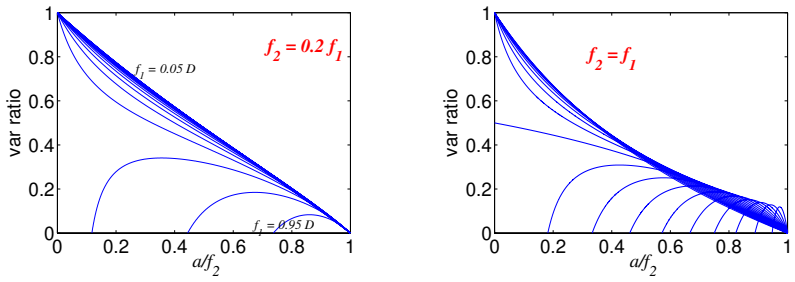
Figure 3 plots the ratio $\frac{\mathrm{Var}(\hat{x}_{1,MLE})}{\mathrm{Var}(\hat{x}_{1,MF})}$ (for $m = 2$), indicating that considering the margins may significantly improve the estimates in certain region of the data.

3.3 Some Experimental Results

We randomly picked words from some MSN Web crawl data (quantized to be binary). We computed all two-way, three-way, and four-way associations and averaged the results in Figure 4. As expected, margins help the most for the two-way case.

3.4 Comparisons with Broder's Sketches

Our sketch construction is the same as Broder's sketches, when estimating two-way associations in boolean data. Broder's sketches (Broder et al., 1998, 2000; Charikar, 2002; Broder et al., 2003) were originally introduced to remove duplicate documents from the Alta Vista Web crawl (Broder, 1997; Broder et al., 1997), though they have been applied subsequently to a variety of applications (Broder,



(a) (b)

Figure 3: The ratio $\frac{\text{Var}(\hat{x}_{1,MLE})}{\text{Var}(\hat{x}_{1,MF})}$ indicates that $\text{Var}(\hat{x}_{1,MLE}) \leq \text{Var}(\hat{x}_{1,MF})$ and margins may improve estimates considerably in certain region of the data. Here we consider $f_2 = 0.2 f_1$ and $f_2 = f_1$ in panels (a) and (b), respectively, which are quite similar. We do not see much change in other cases (e.g., $f_2 = 0.5 f_1$). In each panel, different curves are for different f_1's, ranging from $0.05D$ to $0.95D$ spaced at $0.05D$. Recall $a = x_1$.

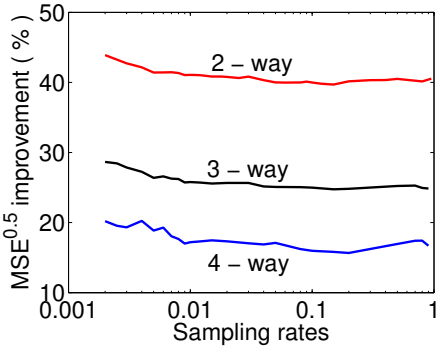


Figure 4: We combine the results of estimating two-way, four-way, and four-way associations, using some MSN Web crawl data. The average relative improvement of the mean square error (MSE) of $\hat{x}_1$, suggests that the MLE is consistently better but the improvement decreases monotonically as the order of associations increases. The sampling rate $= \frac{k_i}{f_i}$, ranging from 0.2% to 100%.

1998; Chen et al., 2000; Haveliwala et al., 2000; Mitzenmacher and Owen, 2001; Haveliwala et al., 2002; Ramaswamy et al., 2003; Poon and Chang, 2003) in data mining and information retrieval.

Border's sketches was designed for estimating the *resemblance* between sets P_1 and P_2, defined as $\frac{|P_1 \cap P_2|}{|P_1 \cup P_2|}$, which can be written as $\frac{a}{f_1+f_2-a}$. Although it is not impossible to extend the concept of *resemblance* to multiple sets, no prior literature has done that and it appears not straightforward to convert *multi-way resemblance* to multi-way associations. Extending Broder's sketches to real-valued data does not seem straightforward either, though (Charikar, 2002) pointed out such an extension may be possible as shown in (Kleinberg and Tardos, 1999).

We shall point out that even in the case of two-way associations (for boolean data), our estimation method is always more accurate than Broder's sketches, by roughly halving the estimation variances. Appendix C derives the variance formula for Broder's sketches

$$\mathrm{Var}(\hat{x}_{1,B}) = \frac{1}{k}\frac{a(f_1+f_2-2a)(f_1+f_2-a)^2}{(f_1+f_2)^2}, \tag{12}$$

where k is the sketch size, which has to be fixed in Broder's construction, while our method offers more flexibility. For example we can let $k_1 = |\mathrm{K}_1|$ and $k_2 = |\mathrm{K}_2|$, with $\frac{k_1}{f_1} = \frac{k_2}{f_2}$, i.e., so-called "proportional sampling."

We plot $\frac{\mathrm{Var}(\hat{x}_{1,MLE})}{\mathrm{Var}(\hat{x}_{1,B})}$ in Figure 5 (with $k_1 = k_2 = k$) and Figure 6 (with "proportional sampling"). These figures indicate that our algorithm always has smaller variances, as can be shown algebraically. The ratios are roughly $\frac{1}{2}$.

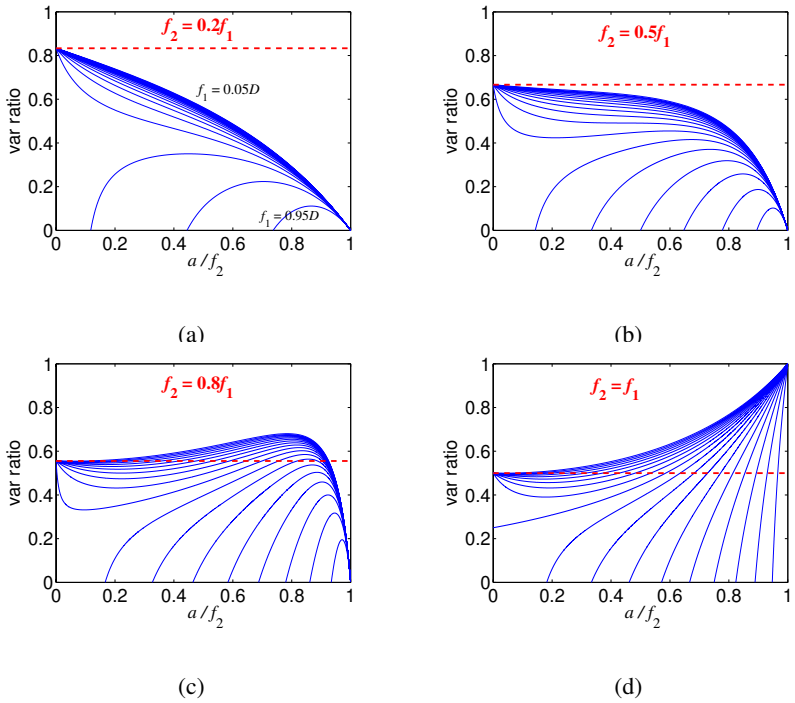
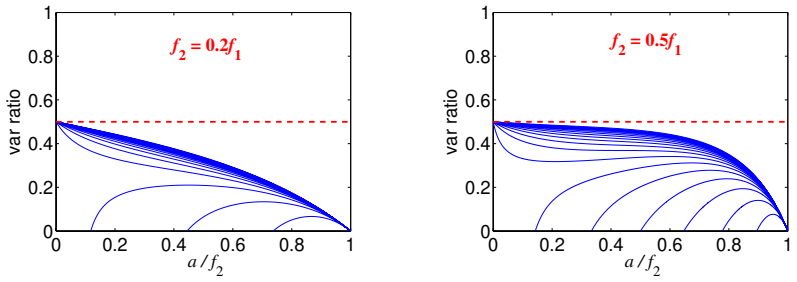


Figure 5: The ratios $\frac{\mathrm{Var}(\hat{x}_{1,MLE})}{\mathrm{Var}(\hat{x}_{1,B})}$ indicate that our algorithm always has smaller variances than Broder's sketches (when $k_1 = k_2 = k$). The panels (a), (b), (c) and (d) correspond to $f_2 = 0.2f_1$, $f_2 = 0.5f_1$, $f_2 = 0.8f_1$ and $f_2 = f_1$, respectively. Different curves are for different f_1's, from $0.05D$ to $0.95D$, spaced at $0.05D$.

4. Histograms In Binned Data

In this section, we generalize the concept of *associations* to *histograms*. Histograms are useful for answering queries like $\mathbf{Pr}(1 < u_1 < 2 \;\&\; u_2 > 2)$. Histograms contain more information than



(a) (b)

Figure 6: Compared with Figure 5, proportional sampling can reduce $\frac{\text{Var}(\hat{x}_{1,MLE})}{\text{Var}(\hat{x}_{1,B})}$.

the *inner product* $a = u_1^T u_2$, which measures the similarity between data points. While univariate histograms are easy to compute and store, joint histograms are much more difficult especially for high-order joins.

Our sketch algorithm provides a simple solution for sparse data. In this case, the sketches store "ID (Binned data value)." Without loss of generality, we number each bin $\{0, 1, ...\}$ as shown in Figure 7(a). We can also consider the data are the generalization of boolean data. For example, the data may take values in $\{0, 1, 2\}$ instead of only in $\{0, 1\}$.

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
u_1	0	1	0	2	0	1	0	0	1	2	1	0	1	0	2
u_2	1	3	0	0	1	2	0	1	0	0	3	0	0	2	1

(a) A data matrix (integers)

u_1 \ u_2	0	1	2	3	
0	3	3	1	0	7
1	2	0	1	2	5
2	2	1	0	0	3
	7	4	2	2	

(b) Contingency table

u_1 \ u_2	0	1	2	3	
0	2	3	0	0	5
1	1	0	1	1	3
2	2	0	0	0	2
	5	3	1	1	

(c) Sample table

Figure 7: (a): A data matrix of binned (integers) data, $D = 15$. The entries of $u_1 \in \{0, 1, 2\}$ and $u_2 \in \{0, 1, 2, 3\}$. We can construct a 3×4 contingency table for u_1 and u_2 in (b). For example, in three columns ($j = 3$, $j = 7$, and $j = 12$), we have $u_{1,j} = u_{2,j} = 0$, hence the (0,0) entry in the table is 3. Suppose the column IDs of the data matrix are random, we can construct a random sample by taking the first $D_s = 10$ columns of the data matrix. A corresponding sample contingency table is then constructed in (c).

Histograms can be conveniently represented by contingency tables, e.g., Figure 7(b). Here, we only consider two-way histograms for the simplicity of presentation. The notation in this section is slightly different from that in Section 3. We denote the original contingency table by $\mathbf{X} = \{x_{i,j}\}_{i=0}^{I} {}_{j=0}^{J}$. Similarly, we denote the sample contingency table by $\mathbf{S} = \{s_{i,j}\}_{i=0}^{I} {}_{j=0}^{J}$. An example of sample contingency table is shown in Figure 7(c) by taking the first $D_s = 10$ columns from the binned data matrix. Of course, we generate the equivalent sample table online using sketches.

4.1 Estimations

Conditioning on D_s and assuming "sample-with-replacement," the sample contingency table $\mathbf{S} = \{s_{i,j}\}_{i=0}^{I}{}_{j=0}^{J}$ follows a multinomial distribution. Therefore we can estimate the original table in a straightforward fashion:

$$\hat{x}_{i,j,MF} = \frac{D}{D_s} s_{i,j}, \tag{13}$$

$$\text{Var}\left(\hat{x}_{i,j,MF}\right) = \text{E}\left(\frac{D}{D_s}\right) \frac{1}{\frac{1}{x_{i,j}} + \frac{1}{D - x_{i,j}}}. \tag{14}$$

Next, we would like to take advantage of marginal histograms, i.e., the row and column sums of the contingency table. There are $I+1$ row sums and $J+1$ column sums. The total number of degrees of freedom would be $(I+1) \times (J+1) - (I+1) - (J+1) + 1 = I \times J$.[5]

When all margins are known, we expect to estimate the table more accurately, especially when the number of degrees of freedom $I \times J$ is not too large. Denote the row sums by $\{x_{i+}\}_{i=0}^{I}$ and the column sums by $\{x_{+j}\}_{j=0}^{J}$. We use a maximum likelihood estimator (MLE) to estimate $x_{i,j}$ under marginal constraints, which amounts to a convex program:

$$\begin{aligned} &\text{Minimize } -\sum_{i=0}^{I}\sum_{j=0}^{J} s_{i,j} \log x_{i,j} \\ &\text{such that } \sum_{j=0}^{J} x_{i,j} = x_{i+},\ \sum_{i=0}^{I} x_{i,j} = x_{+j},\ x_{i,j} \geq s_{i,j}, \end{aligned} \tag{15}$$

which can be solved easily using any standard algorithms such as Newton's method. We can also use the more straightforward *iterative proportional scaling* algorithm for approximate solutions. The estimated table cells are denoted by $\hat{x}_{i,j,MLE}$.

One can also estimate the inner product $a = u_1^{\text{T}} u_2$ from the estimated contingency table because

$$a = u_1^{\text{T}} u_2 = \sum_{i=1}^{I}\sum_{j=1}^{J} (ij) x_{i,j}. \tag{16}$$

Therefore, we can estimate a by

$$\hat{a}_{MLE,c} = \sum_{i=1}^{I}\sum_{j=1}^{J} (ij) \hat{x}_{i,j,MLE}, \tag{17}$$

where the subscript "c" indicates that a is computed from contingency tables. Similarly, we can have $\hat{a}_{MF,c}$.

Appendix D derives the variances of $\hat{x}_{i,j,MLE}$ and $\hat{a}_{MLE,c}$.

4.2 Numerical Examples

Two words "THIS" and "HAVE" are taken from a chunk of MSN Web crawl data ($D = 2^{16}$). The data are quantized into a few histogram bins. Two experiments are conducted, with 5 bins and 3 bins, respectively, as shown in Table 2.

5. Note that the sum of the row sums has to be equal to the sum of the column sums, which is equal to D (sum of all cells). Therefore, the effective number of constraints is $I + J + 1$, instead of $I + J + 2$.

Table 2: The two word vectors "THIS" and "HAVE" are quantized. (a) Exp #1: 5 bins numbered from 0 to 4. (b) Exp #2: 3 bins from 0 to 2.

Bin ID	Data
0	0
1	$1 \sim 2$
2	$3 \sim 4$
3	$5 \sim 10$
4	> 10

(a) Exp.#1

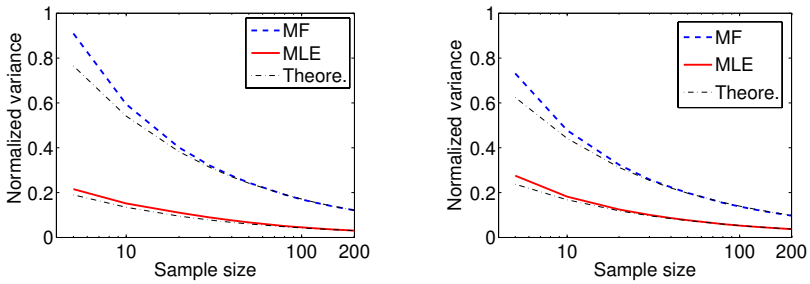
Bin ID	Data
0	0
1	$1 \sim 2$
2	> 3

(b) Exp.#2

The two (quantized) word vectors are sampled by sketches with sketch sizes ranging from 5 to 200. Sample contingency tables are then constructed (online) from sketches and the original contingency tables are estimated using both margin-free (MF) and MLE estimators.

How to evaluate the results? A chi-squared statistic is probably appropriate, but we prefer not to deal with the case in which some of cells are zeros. For simplicity, we evaluate the results in terms of a, the inner product.

Figure 8 compares the empirical variances with the theoretical predictions for $\hat{a}_{MF,c}$ and $\hat{a}_{MLE,c}$. The figure verifies that our theoretical variances are accurate at reasonable sketch sizes (e.g., $\geq 10 \sim 20$). The errors are mostly due to the approximation $\mathrm{E}\left(\frac{D}{D_s}\right) = \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}\right)$. Also, notice that in n this case, the marginal histograms help considerably.



(a) Exp. #1 (b) Exp. #2

Figure 8: The inner product a (after quantization) between "THIS" and "HAVE" is estimated by both $\hat{a}_{MF,c}$ and $\hat{a}_{MLE,c}$. Results are reported in $\frac{\sqrt{\mathrm{Var}(\hat{a})}}{a}$. The two thin dashed lines both labeled "theore." are theoretical variances, which match the empirical values well especially after sketch sizes $\geq 10 \sim 20$. In this case, marginal histograms help considerably.

5. Inner Products In General Data

This section concerns general (real-valued) data, in particular, estimating pairwise inner products. We assume that the data are also highly sparse (mostly zeros) hence our sketch-based sampling algorithm can be useful.

Again, we construct sketches for all data points $\{u_i\}_{i=1}^n \in \mathbb{R}^D$. We then construct equivalent random samples (online) when we need to estimate $a = u_1^{\mathrm{T}} u_2$. Suppose the computed effective sample size is D_s. We use $\tilde{u}_{i,j}$, $j = 1$ to D_s, to denote these random samples in u_i.

The obvious estimator of $a = u_1^{\mathrm{T}} u_2$ is

$$\hat{a}_{MF} = \frac{D}{D_s} \sum_{j=1}^{D_s} \tilde{u}_{1,j} \tilde{u}_{2,j}, \tag{18}$$

$$\mathrm{Var}\left(\hat{a}_{MF}\right) = \mathrm{E}\left(\frac{D}{D_s}\right)\left(\sum_{j=1}^{D} u_{1,j}^2 u_{2,j}^2 - \frac{a^2}{D}\right). \tag{19}$$

Basically, $\hat{a}_{MF}$ estimates the population correlation from the sample correlation.

We would also like to consider the marginal information such as $m_1 = \|u_1\|^2$, $m_2 = \|u_2\|^2$. However, without making further assumptions on the data distribution, we can not conduct conventional maximum likelihood estimations. We have many options. We could quantize the data so that we can use the contingency table estimation technique described in Section 4. We could also use a non-parametric maximum likelihood such as the "Empirical Likelihood," (Owen, 2001) which amounts to solving a convex optimization problem. A Bayesian approach is also reasonable. This is a general statistical model selection/inference problem.

A practical solution is to assume some parametric form of the data distribution based on prior knowledge; and then solve an MLE considering various of constraints. For example, when the data are not "too far" from normal, we could assume normality on the data. This is often the case in many important applications. Take the term-by-document matrix as an example. When the popular logarithmic weighting is applied, the data become approximately normal.[6] Therefore, we consider the following estimator based on the normality is practically useful.

5.1 An MLE Assuming Normality

Suppose the samples $(\tilde{u}_{1,j},\ \tilde{u}_{2,j})$ are i.i.d. normal with moments determined by the population moments, i.e.,

$$\begin{bmatrix} \tilde{v}_{1,j} \\ \tilde{v}_{2,j} \end{bmatrix} = \begin{bmatrix} \tilde{u}_{1,j} - \bar{u}_1 \\ \tilde{u}_{2,j} - \bar{u}_2 \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \tilde{\Sigma}\right), \tag{20}$$

$$\tilde{\Sigma} = \frac{1}{D_s}\frac{D_s}{D}\begin{bmatrix} \|u_1\|^2 - D\bar{u}_1^2 & u_1^{\mathrm{T}} u_2 - D\bar{u}_1\bar{u}_2 \\ u_1^{\mathrm{T}} u_2 - D\bar{u}_1\bar{u}_2 & \|u_2\|^2 - D\bar{u}_2^2 \end{bmatrix} = \frac{1}{D_s}\begin{bmatrix} \ddot{m}_1 & \ddot{a} \\ \ddot{a} & \ddot{m}_2 \end{bmatrix},$$

where $\bar{u}_1 = \sum_{j=1}^D u_{1,j}/D$, $\bar{u}_2 = \sum_{j=1}^D u_{2,j}/D$ are the population means. $\ddot{m}_1 = \frac{D_s}{D}\left(\|u_1\|^2 - D\bar{u}_1^2\right)$, $\ddot{m}_2 = \frac{D_s}{D}\left(\|u_2\|^2 - D\bar{u}2^2\right)$, $\ddot{a} = \frac{D_s}{D}\left(u_1^{\mathrm{T}} u_2 - D\bar{u}_1\bar{u}_2\right)$. Suppose that $\bar{u}_1$, $\bar{u}_2$, $m_1 = \|u_1\|^2$ and $m_2 = \|u_2\|^2$ are known, an MLE for $a = u_1^{\mathrm{T}} u_2$, denoted by $\hat{a}_{MLE,N}$ (the subscript "N" for "normal"), is then

$$\hat{a}_{MLE,N} = \frac{D}{D_s}\hat{\ddot{a}} + D\bar{u}_1\bar{u}_2, \tag{21}$$

6. Although heavy-tailed data are ubiquitous (Leland et al., 1994; Faloutsos et al., 1999; Newman, 2005), it is a common practice to carefully weight the data. Various term weighting schemes have been proposed e.g., (Yu et al., 1982; Salton and Buckley, 1988; Dumais, 1991; Greiff, 2003; Liu et al., 2001). It is well-known (e.g., (Leopold and Kindermann, 2002; Rennie et al., 2003; Lan et al., 2005)) that choosing an appropriate term weighting method is vital.

where $\hat{\ddot{a}}$ is the solution to a cubic equation:

$$\ddot{a}^3 - \ddot{a}^2 \left(\tilde{v}_1^{\mathrm{T}} \tilde{v}_2\right) + \ddot{a} \left(-\ddot{m}_1 \ddot{m}_2 + \ddot{m}_1 \|\tilde{v}_2\|^2 + \ddot{m}_2 \|\tilde{v}_1\|^2\right) - \ddot{m}_1 \ddot{m}_2 \tilde{v}_1^{\mathrm{T}} \tilde{v}_2 = 0. \tag{22}$$

The proof is not difficult though a little tedious. In a similar idea and with detailed proofs, the authors' recent work on random projections (Li et al., 2006a,b) describes using the marginal information to improve random projections.

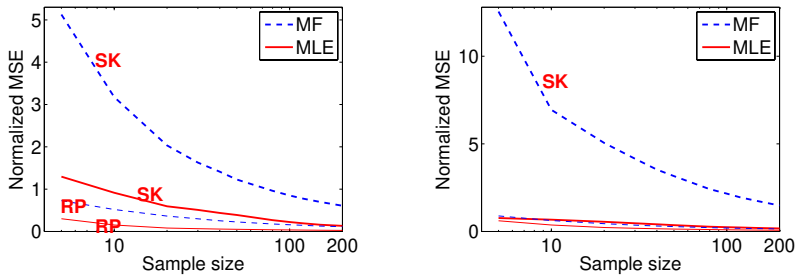
$\hat{a}_{MLE,N}$ is fairly robust unless the data are very far from normal (e.g., heavy-tailed). Evaluating $\mathrm{Var}\left(\hat{a}_{MLE,N}\right)$, however, is difficult, because theoretical variances are very sensitive to model mis-specification(White, 1982). In our experiments, we observe that $\hat{a}_{MLE,N}$ actually works well even in heavy-tailed data. Our concern is that $\hat{a}_{MLE,N}$ may be highly biased in certain heavy-tailed data, although we have not observe this phenomena. We only recommend this estimator when the data are known to be approximately normal (e.g., after careful term weighting).

5.2 Numerical Experiments

Two pairs of words "THIS - HAVE" and "MONDAY - SATURDAY" are selected from the MSN Web crawl data. We estimate the inner products by both sketches and random projections. We will soon give a brief introduction to random projections in Section 6.

We test both the original unweighted data (heavy-tailed), and the weighted data with "logarithmic weighting," i.e., any non-zero counts is replaced by 1 + log(original count).

With no term weighting (Figure 9), sketches exhibit large errors, although considering marginal information (i.e., using $\hat{a}_{MLE,N}$) can significantly reduce the errors. With logarithmic weighting (Figure 10), sketches are about as accurate as random projections.

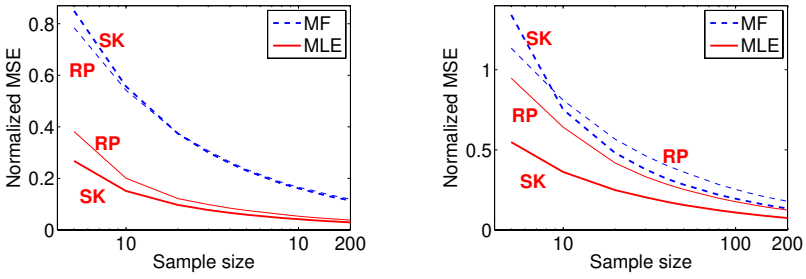


(a) THIS-HAVE (b) MONDAY-SATURDAY

Figure 9: No term weighting. Random projections (RP) are more accurate (smaller MSE) than sketches (SK). Without using marginal constraints, sketches have large errors. In (b), the solid curve for "RP" is lower than the solid curve for "SK." Results are presented in terms of $\frac{\sqrt{\mathrm{MSE}(\hat{a})}}{a}$.

6. Theoretical Comparisons With Random Projections

Random projections (Vempala, 2004; Achlioptas, 2003) have been widely used in Machine Learning, data mining, and bio-informatics (Papadimitriou et al., 1998; Arriaga and Vempala, 1999; Bing-



(a) THIS-HAVE (b) MONDAY-SATURDAY

Figure 10: Logarithmic weighting. Sketches are about as accurate as random projections.

ham and Mannila, 2001; Achlioptas et al., 2001; Fradkin and Madigan, 2003; Fern and Brodley, 2003; Liu et al., 2006; Charikar, 2002; Buhler and Tompa, 2002; Leung et al., 2005).

Random projections multiply the original data matrix $\mathbf{A} \in \mathbb{R}^{n \times D}$ with a random matrix $\mathbf{R} \in \mathbb{R}^{D \times k}$, whose entries are typically i.i.d. normals $N(0,1)$.[7] $\frac{1}{\sqrt{k}}\mathbf{AB}$ preserves all pairwise distances of $\mathbf{A}$ in expectations.

(Li et al., 2006a) provides two estimators of $a = u_1^T u_2$, a margin-free (MF) estimator, denoted by $\hat{a}_{RP,MF}$, and a maximum likelihood estimator (MLE), denoted by $\hat{a}_{RP,MLE}$, assuming that $m_1 = \|u_1\|^2$ and $m_2 = \|u_2\|^2$ are known. Their variances are

$$\text{Var}\left(\hat{a}_{RP,MF}\right) = \frac{1}{k}\left(m_1 m_2 + a^2\right), \tag{23}$$

$$\text{Var}\left(\hat{a}_{RP,MLE}\right) = \frac{1}{k}\frac{(m_1 m_2 - a^2)^2}{m_1 m_2 + a^2}. \tag{24}$$

The cost of random projections is $O(nDk)$ for processing and $O(n^2 k)$ for computing all pairwise distances. Recall that the processing time for our sketch algorithm is $O(nD)$. We are particularly interested in comparing their variances because estimators with smaller variances require less samples to achieve the specified level of accuracy.

Our comparisons show that

- In boolean data, sketches are almost always more accurate than random projections.
- In normal-like data, both sketches and random projections are roughly the same.
- In heavy-tailed data, sketches have larger errors.
- In nearly independent data, random projections have larger errors.
- In highly sparse data, sketches tend to be more accurate than random projections, depending on how heavy-tailed the data are.

7. See (Achlioptas, 2003; Li et al., 2006b) for different variations of random projections.

6.1 The Margin-Free Case ($\hat{a}_{RP,MF}$ v.s. $\hat{a}_{MF}$)

We compare Var $(\hat{a}_{RP,MF})$ in (23) with Var $(\hat{a}_{MF})$ in (19) since both have closed-form expressions. Here we assume equal samples, i.e., $k_1 = k_2 = k$. Rewrite

$$\text{Var}\left(\hat{a}_{RP,MF}\right) = \frac{1}{k}\left(m_1 m_2 + a^2\right) = \frac{D^2}{k}\left(\text{E}\left(\tilde{u}_{1,j}^2\right)\text{E}\left(\tilde{u}_{2,j}^2\right) + \left(\text{E}\left(\tilde{u}_{1,j}\tilde{u}_{2,j}\right)\right)^2\right), \tag{25}$$

$$\begin{aligned}\text{Var}\left(\hat{a}_{MF}\right) &= \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}\right)\left(\sum_{j=1}^{D} u_{1,j}^2 u_{1,j}^2 - \frac{a^2}{D}\right) \\ &= \frac{\max(f_1, f_2)}{D}\frac{D^2}{k}\left(\text{E}\left(\tilde{u}_{1,j}^2\tilde{u}_{2,j}^2\right) - \left(\text{E}\left(\tilde{u}_{1,j}\tilde{u}_{2,j}\right)\right)^2\right).\end{aligned} \tag{26}$$

Recall that $\tilde{u}_{1,j}$ and $\tilde{u}_{2,j}$ denote random samples from u_1 and u_2, respectively; f_1 and f_2 are the numbers of non-zero elements in u_1 and u_2, respectively.

Comparing (25) with (26), we can see immediately that if the data are very sparse, i.e., $\frac{\max(f_1,f_2)}{D}$ is small, then Var $(\hat{a}_{MF})$ tends to be smaller than Var $(\hat{a}_{RP,MF})$.

When the data are exactly normal, then $\text{E}\left(\tilde{u}_{1,j}^2\tilde{u}_{2,j}^2\right) - \left(\text{E}\left(\tilde{u}_{1,j}\tilde{u}_{2,j}\right)\right)^2 = \text{E}\left(\tilde{u}_{1,j}^2\right)\text{E}\left(\tilde{u}_{2,j}^2\right) + \left(\text{E}\left(\tilde{u}_{1,j}\tilde{u}_{2,j}\right)\right)^2$, and the sparsity factors $\frac{f_1}{D} = 1$ and $\frac{f_2}{D} = 1$, almost surely, hence Var $(\hat{a}_{MF})$ = Var $(\hat{a}_{RP,MF})$.

We can take a look at two extreme cases.

First, when $\tilde{u}_{1,j}$ and $\tilde{u}_{2,j}$ are independent, then $\text{E}\left(\tilde{u}_{1,j}^2\tilde{u}_{2,j}^2\right) = \text{E}\left(\tilde{u}_{1,j}^2\right)\text{E}\left(\tilde{u}_{2,j}^2\right)$, which implies that Var $(\hat{a}_{MF}) \leq$ Var $(\hat{a}_{RP,MF})$, even ignoring the sparsity factors.

Next, we can consider when $u_1 = u_2$. In this case, neglecting the sparsity factors, we have

$$\begin{aligned}\text{Var}\left(\hat{a}_{MF}\right) - \text{Var}\left(\hat{a}_{RP,MF}\right) \leq & \frac{D^2}{k}\left(\text{E}\left(\tilde{u}_{1,j}^4\right) - 3\left(\text{E}\left(\tilde{u}_{1,j}^2\right)\right)^2\right) \\ = & \frac{D^2}{k}\left(\left(\text{E}\left(\tilde{u}_{1,j}^2\right)\right)^2\left(\frac{\text{E}\left(\tilde{u}_{1,j}^4\right)}{\left(\text{E}\left(\tilde{u}_{1,j}^2\right)\right)^2} - 3\right)\right).\end{aligned} \tag{27}$$

If $u_{1,j}$ has zero mean, the term $\left(\frac{\text{E}\left(\tilde{u}_{1,j}^4\right)}{\left(\text{E}\left(\tilde{u}_{1,j}^2\right)\right)^2} - 3\right)$ is the "kurtosis," which measures the tail thickness. In general, this term also contains information about the "skewness." Therefore, when the data are heavy-tailed (or highly-skewed), random projections can be more accurate, if the sparsity factors are not small enough to compensate.

6.2 The Boolean Data Case

This comparison only considers boolean data. In this case, the marginal norms are the same as the numbers of non-zero elements, i.e., $m_i = \|u_i\|^2 = f_i$.

We have derived the variance formula for sketches with and without using margins, in (7) and (11), respectively.

Figure 11 plots the ratio $\frac{\text{Var}(\hat{a}_{MF})}{\text{Var}(\hat{a}_{RP,MF})}$, verifying that sketches are (considerably) more accurate:

$$\frac{\text{Var}\left(\hat{a}_{MF}\right)}{\text{Var}\left(\hat{a}_{RP,MF}\right)} = \frac{\max(f_1, f_2)}{f_1 f_2 + a^2} \frac{1}{\frac{1}{a} + \frac{1}{D-a}} \leq \frac{\max(f_1, f_2)a}{f_1 f_2 + a^2} \leq 1.$$

Now we consider the margins. Figure 12 plots $\frac{\text{Var}(\hat{a}_{MLE})}{\text{Var}(\hat{a}_{RP,MLE})}$. In most range of the data possible, this ratio is less than 1, especially when $f_1 \neq f_2$. When u_1 and u_2 are very close (e.g., $a \approx f_2 \approx f_1$), random projections appear more accurate than sketches. However, when this does occur, the absolute variances are so small (even zero) that the variance ratio does not matter.

Note that here we assume equal sampling: $k_1 = k_2 = k$. Sketches can be improved by proportional sampling: $\frac{k_1}{f_1} = \frac{k_2}{f_2}$.

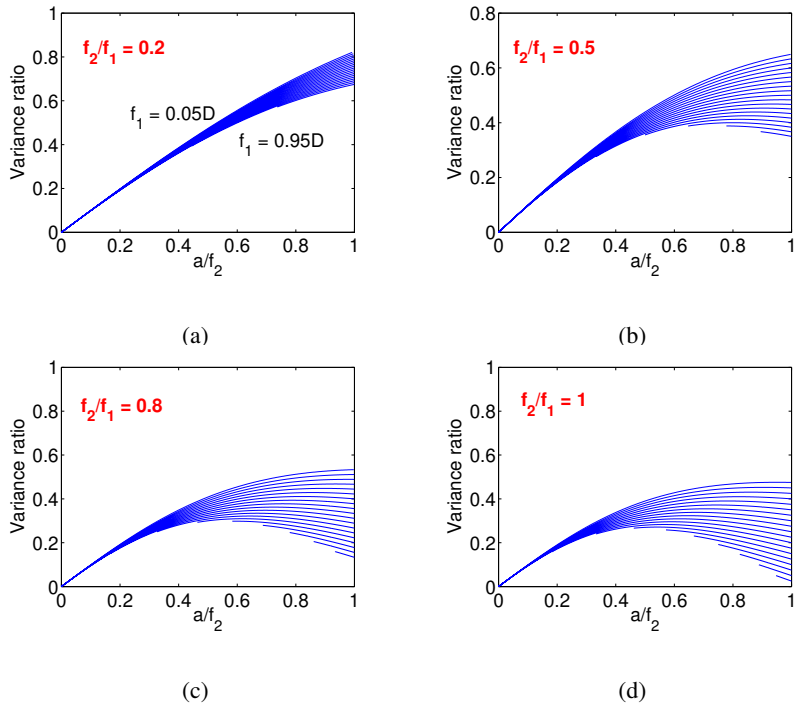


Figure 11: The variance ratios, $\frac{\text{Var}(\hat{a}_{MF})}{\text{Var}(\hat{a}_{RP,MF})}$ show that our algorithm has smaller variances than random projections, when no marginal information is used in both methods. Here we assume $f_1 \geq f_2$ and consider $f_2 = \alpha f_1$ with $\alpha = 0.2, 0.5, 0.8, 1.0$ in (a), (b), (c), and (d), respectively. For each α, we plot from $f_1 = 0.05D$ to $f_1 = 0.95D$ spaced at $0.05D$.

7. Conclusion

We propose a sketch-based sampling algorithm, which only samples the non-zero data with the sample sizes flexibly adjusted according to data sparsity. Our method differs from many sketching algorithms in that we convert sketches into random samples online. Therefore, we can conduct

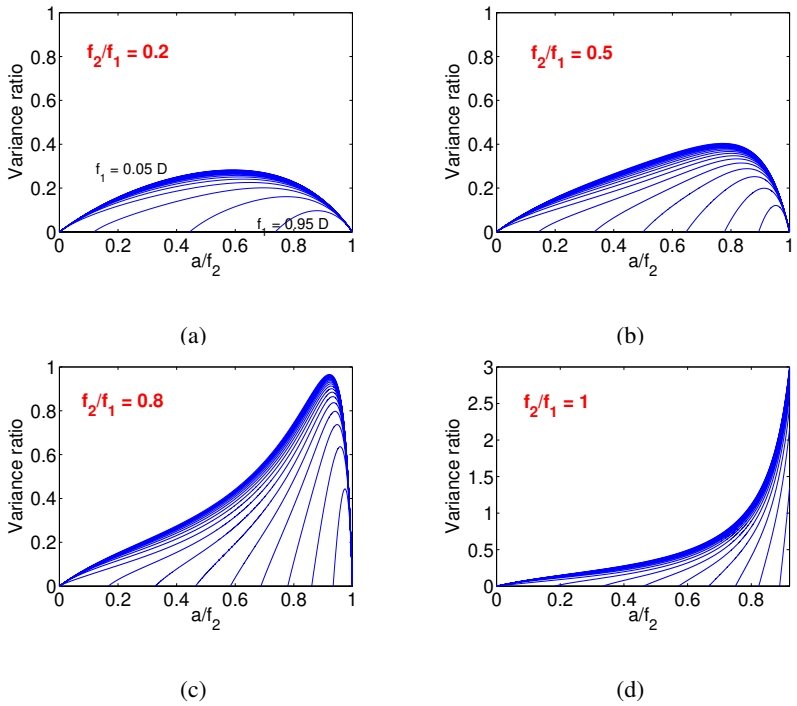


Figure 12: The ratios, $\frac{\mathrm{Var}(\hat{a}_{MLE})}{\mathrm{Var}(\hat{a}_{RP,MLE})}$ show that our sketch algorithm usually has smaller variances than random projections, except when $f_1 \approx f_2 \approx a$.

estimations just like conventional random sampling. Based on well-understood statistical principles, we have developed various estimators taking advantages of the marginal information, which can often improve the estimates considerably.

8. Acknowledgments

Special thanks to Chris Meek for reading various drafts on this work and providing very constructive comments. We thank the feedbacks from David Heckerman, Pat Langley, Mark Manasse, Andew Ng, Amin Saberi and Robert Tibshirani. Ping Li also thanks Persi Diaconis, Bradley Efron, Jonathan Goldstein, Tze Leung Lai, Richard Olshen, Art Owen, David Siegmund and Yiyuan She for helpful conversations (or email communications).

References

Dimitris Achlioptas. Database-friendly random projections: Johnson-Lindenstrauss with binary coins. *Journal of Computer and System Sciences*, 66(4):671–687, 2003.

Dimitris Achlioptas and Frank McSherry. Fast computation of low rank matrix. In *Proc. of STOC*, pages 611–618, Heraklion, Crete, Greece, 2001.

Dimitris Achlioptas, Frank McSherry, and Bernhard Schölkopf. Sampling techniques for kernel methods. In *Proc. of NIPS*, pages 335–342, Vancouver, BC, Canada, 2001.

Charu C. Aggarwal, Cecilia Magdalena Procopiuc, Joel L. Wolf, Philip S. Yu, and Jong Soo Park. Fast algorithms for projected clustering. In *Proc. of SIGMOD*, pages 61–72, Philadelphia, PA, 1999.

David Aldous and Persi Diaconis. Shuffling cards and stopping times. *The American Mathematical Monthly*, 93(5):333–348, 1986.

Rosa Arriaga and Santosh Vempala. An algorithmic theory of learning: Robust concepts and random projection. In *Proc. of FOCS (Also to appear in Machine Learning)*, pages 616–623, New York, 1999.

Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *Proc. of KDD*, pages 245–250, San Francisco, CA, 2001.

Stephen Boyd and Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, Cambridge, UK. Also online www.stanford.edu/ boyd/bv_cvxbook.pdf, 2004.

Sergey Brin, James Davis, and Hector Garcia-Molina. Copy detection mechanisms for digital documents. In *Proc. of SIGMOD*, pages 398–409, San Jose, CA, 1995.

Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proc. of WWW*, pages 107–117, Brisbane, Australia, 1998.

Sergy Brin, Rajeev Motwani, and Craig Silverstein. Beyond market baskets: Generalizing association rules to correlations. In *Proc. of SIGMOD*, pages 265–276, Tucson, AZ, 1997a.

Sergy Brin, Rajeev Motwani, Jeffrey D. Ullman, and Shalom Tsur. Dynamic itemset counting and implication rules for market basket data. In *Proc. of SIGMOD*, pages 265–276, Tucson, AZ, 1997b.

Andrei Z. Broder. On the resemblance and containment of documents. In *Proc. of the Compression and Complexity of Sequences*, pages 21–29, Positano, Italy, 1997.

Andrei Z. Broder. Filtering near-duplicate documents. In *Proc. of FUN*, Isola d'Elba, Italy, 1998.

Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations (extended abstract). In *Proc. of STOC*, pages 327–336, Dallas, TX, 1998.

Andrei Z. Broder, Moses Charikar, Alan M. Frieze, and Michael Mitzenmacher. Min-wise independent permutations. *Journal of Computer Systems and Sciences*, 60(3):630–659, 2000.

Andrei Z. Broder, Moses Charikar, and Michael Mitzenmacher. A derandomization using min-wise independent permutations. *Journal of Discrete Algorithms*, 1(1):11–20, 2003.

Andrei Z. Broder, Steven C. Glassman, Mark S. Manasse, and Geoffrey Zweig. Syntactic clustering of the web. In *Proc. of WWW*, pages 1157 – 1166, Santa Clara, CA, 1997.

Jeremy Buhler and Martin Tompa. Finding motifs using random projections. *Journal of Computational Biology*, 9(2):225–242, 2002.

Moses S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc. of STOC*, pages 380–388, Montreal, Quebec, Canada, 2002.

Surajit Chaudhuri, Rajeev Motwani, and Vivek R. Narasayya. On random sampling over joins. In *Proc. of SIGMOD*, pages 263–274, Philadelphia, PA, 1999.

Zhiyuan Chen, Flip Korn, Nick Koudas, and S. Muthukrishnan. Selectivity estimation for boolean queries. In *Proc. of PODS*, pages 216–225, Dollas, TX, 2000.

T. Dasu, T. Johnson, S. Muthukrishnan, and V. Shkapenyuk. Mining database structure; or, how to build a data quality browser. In *Proc. of SIGMOD*, pages 240–251, Madison, WI, 2002.

Scott Deerwester, Susan T. Dumais, George W. Furnas, and Thomas K. Landauer. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 1999.

W. Edwards Deming and Frederick F. Stephan. On a least squares adjustment of a sampled frequency table when the expected marginal totals are known. *The Annals of Mathematical Statistics*, 11(4):427–444, 1940.

Inderjit S. Dhillon and Dharmendra S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, 42(1-2):143–175, 2001.

Carlos Domingo, Ricard Gavald, and Osamu Watanabe. Adaptive sampling methods for scaling up knowledge discovery algorithms. *Data Mining and Knowledge Discovery*, 6(2):131–152, 2002.

Petros Drineas and Michael W. Mahoney. On the nystrom method for approximating a gram matrix for improved kernel-based learning. *Journal of Machine Learning Research*, 6(Dec):2153–2175, 2005.

Susan T. Dumais. Improving the retrieval of information from external sources. *Behavior Research Methods, Instruments and Computers*, 23(2):229–236, 1991.

William DuMouchel and Deepak K. Agarwal. Applications of sampling and fractional factorial designs to model-free data squashing. In *Proc. of KDD*, pages 511–516, Washington, DC, 2003.

William DuMouchel, Chris Volinsky, Theodore Johnson, Corinna Cortes, and Daryl Pregibon. Squashing flat files flatter. In *Proc. of KDD*, pages 6–15, San Diego, CA, 1999.

Michalis Faloutsos, Petros Faloutsos, and Christos Faloutsos. On power-law relationships of the Internet topology. In *Proc. of SIGMOD*, pages 251–262, Cambridge,MA, 1999.

Xiaoli Zhang Fern and Carla E. Brodley. Random projection for high dimensional data clustering: A cluster ensemble approach. In *Proc. of ICML*, pages 186–193, Washington, DC, 2003.

Stephen E. Fienberg. An iterative procedure for estimation in contingency tables. *The Annals of Mathematical Statistics*, 41(3):907–917, 1970.

Dmitriy Fradkin and David Madigan. Experiments with random projections for machine learning. In *Proc. of KDD*, pages 517–522, Washington, DC, 2003.

Anna C. Gilbert, Yannis Kotidis, Sudipto Guha, S. Muthukrishnan, and Strauss Piotr Indyk. Fast, small-space algorithms for approximate histogram maintenance. In *Proc. of STOC*, pages 389–398, Montreal, Quebec, Canada, 2002.

Anna C. Gilbert, Yannis Kotidis, S. Muthukrishnan, and Martin J. Strauss. One-pass wavelet decompositions of data streams. *IEEE Transactions on Knowledge and Data Engineering*, 15(3):541–554, 2003.

Warren R. Greiff. A theory of term weighting based on exploratory data analysis. In *Proc. of SIGIR*, pages 11–19, Melbourne, Australia, 2003.

Taher H. Haveliwala, Aristides Gionis, and Piotr Indyk. Scalable techniques for clustering the web. In *Proc. of WebDB*, pages 129–134, 2000.

Taher H. Haveliwala, Aristides Gionis, Dan Klein, and Piotr Indyk. Evaluating strategies for similarity search on the web. In *Proc. of WWW*, pages 432–442, Honolulu, HI, 2002.

Piotr Indyk. Stable distributions, pseudorandom generators, embeddings and data stream computation. In *FOCS*, pages 189–197, Redondo Beach,CA, 2000.

Piotr Indyk. Algorithmic applications of low-distortion geometric embeddings. In *Proc. of FOCS*, pages 10–33, Las Vegas, NV, 2001.

Piotr Indyk and Rajeev Motwani. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proc. of STOC*, pages 604–613, Dallas, TX, 1998.

Jon Kleinberg and Eva Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and Markov random fields. In *Proc. of FOCS*, pages 14–23, New York, 1999.

Donald E. Knuth. *The Art of Computer Programming (V. 2): Seminumerical Algorithms*. Addison-Wesley, New York, NY, third edition, 1997.

Man Lan, Chew Lim Tan, Hwee-Boon Low, and Sam Yuan Sung. A comprehensive comparative study on term weighting schemes for text categorization with support vector machines. In *Proc. of WWW*, pages 1032–1033, Chiba, Japan, 2005.

S. D. Lee, David W. Cheung, and Ben Kao. Is sampling useful in data mining? a case in the maintenance of discovered association rules. *Data Mining and Knowledge Discovery*, 2(3):233–262, 1998.

Erich L. Lehmann and George Casella. *Theory of Point Estimation*. Springer, New York, NY, second edition, 1998.

Will E. Leland, Murad S. Taqqu, Walter Willinger, and Daniel V. Wilson. On the self-similar nature of Ethernet traffic. *IEEE/ACM Trans. Networking*, 2(1):1–15, 1994.

Edda Leopold and Jorg Kindermann. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, 46(1-3):423–444, 2002.

Henry C.M. Leung, Francis Y.L. Chin, S.M. Yiu, Roni Rosenfeld, and W.W. Tsang. Finding motifs with insufficient number of strong binding sites. *Journal of Computational Biology*, 12(6):686–701, 2005.

Ping Li and Kenneth W. Church. Using sketches to estimate two-way and multi-way associations. Technical Report TR-2005-115, Microsoft Research, Microsoft Corporation, WA, September 2005.

Ping Li, Trevor J. Hastie, and Kenneth W. Church. Improving random projections using marginal information (accepted to colt 2006). Technical report, www.stanford.edu/ ~pingli98/report/COLT_rp.pdf, 2006a.

Ping Li, Trevor J. Hastie, and Kenneth W. Church. Very sparse random projections. Technical report, www.stanford.edu/~pingli98/report/srp.pdf, 2006b.

Bing Liu, Yiming Ma, and Philip S. Yu. Discovering unexpected information from your competitors' web sites. In *Proc. of KDD*, pages 144–153, San Francisco, CA, 2001.

Kun Liu, Hillol Kargupta, and Jessica Ryan. Random projection-based multiplicative data perturbation for privacy preserving distributed data mining. *IEEE Transactions on Knowledge and Data Engineering*, 18 (1):92–106, 2006.

David Madigan, Nandini Raghavan, Martha Nason, and Greg Ridgeway. Likelihood-based data squashing: A modeling approach to instance construction. *Data Mining and Knowledge Discovery*, 6(2):173–190, 2002.

Gurmeet Singh Manku, Sridhar Rajagopalan, and Bruce G. Lindsay. Random sampling techniques for space efficient online computation of order statistics of large datasets. In *Proc. of SIGCOMM*, pages 251–262, Philadelphia, PA, 1999.

Yossi Matias, Jeffrey Scott Vitter, and Min Wang. Wavelet-based histograms for selectivity estimation. In *Proc. of SIGMOD*, pages 448–459, Seattle, WA, 1998.

Michael Mitzenmacher and Sean Owen. Estimating resemblance of midi documents. In *Proc. of ALENEX*, pages 78–90, Washington, DC, 2001.

M. E. J. Newman. Power laws, pareto distributions and zipf's law. *Contemporary Physics*, 46(5):232–351, 2005.

Art Owen. Data squashing by empirical likelihood. *Data Mining and Knowledge Discovery*, 7(1):101–113, 2003.

Art B. Owen. *Empirical Likelihood*. Chapman & Hall/CRC, New York, NY, 2001.

Christos H. Papadimitriou, Prabhakar Raghavan, Hisao Tamaki, and Santosh Vempala. Latent semantic indexing: A probabilistic analysis. In *Proc. of PODS*, pages 159–168, Seattle,WA, 1998.

Chung Keung Poon and Matthew Chang. An email classifier based on resemblance. In *Proc. of ISMIS*, pages 344–348, Maebashi City, Japan, 2003.

Michael O. Rabin. Fingerprinting by random polynomials. Technical Report TR-15-81, Center for Research in Computing Technology, Cambridge, MA, 1981.

Lakshmish Ramaswamy, Arun Iyengar, Ling Liu, and Fred Douglis. Techniques for efficient fragment detection in web pages. In *Proc. of CIKM*, pages 516–519, New Orleans, LA, 2003.

Deepak Ravichandran, Patrick Pantel, and Eduard Hovy. Randomized algorithms and NLP: Using locality sensitive hash function for high speed noun clustering. In *Proc. of ACL*, pages 622–629, Ann Arbor, MI, 2005.

Jason D. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. In *Proc. of ICML*, pages 616–623, Washington, DC, 2003.

David M. Rocke and Jian Dai. Sampling and subsampling for cluster analysis in data mining: With applications to sky survey data. *Data Mining and Knowledge Discovery*, 7(2):215–232, 2003.

Bengt Rosen. Asymptotic theory for successive sampling with varying probabilities without replacement, I. *The Annals of Mathematical Statistics*, 43(2):373–397, 1972a.

Bengt Rosen. Asymptotic theory for successive sampling with varying probabilities without replacement, II. *The Annals of Mathematical Statistics*, 43(3):748–776, 1972b.

Gerard Salton and Chris Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, 1988.

Sunita Sarawagi, Shiby Thomas, and Rakesh Agrawa. Integrating association rule mining with relational database systems: Alternatives and implications. *Data Mining and Knowledge Discovery*, 4(2-3):89–125, 2000.

Kyle Siegrist. *Finite Sampling Models, www.ds.unifi.it/VL/VL_EN/urn/index.html*. Virtual Laboratories in Probability and Statistics, Huntsville, AL, 1997.

Ashwin Srinivasan. A study of two sampling methods for analyzing large datasets with ILP. *Data Mining and Knowledge Discovery*, 3(1):95–123, 1999.

Frederick F. Stephan. An iterative method of adjusting sample frequency tables when expected marginal totals are known. *The Annals of Mathematical Statistics*, 13(2):166–178, 1942.

Kyuseok Shim Sudipto Guha, Rajeev Rastogi. Cure: An efficient clustering algorithm for large databases. In *Proc. of SIGMOD*, pages 73–84, Seattle, WA, 1998.

Vivek R. Narasayya Surajit Chaudhuri, Rajeev Motwani. Random sampling for histogram construction: How much is enough? In *Proc. of SIGMOD*, pages 436–447, Seattle, WA, 1998.

Santosh Vempala. A random sampling based algorithm for learning the intersection of half-spaces. In *Proc. of FOCS*, pages 508–513, Miami Beach, FL, 1997.

Santosh S. Vempala. *The Random Projection Method*. American Mathematical Society, Providence, RI, 2004.

Halbert White. Maximum likelihood estimation of misspecified models. *Econometrica*, 50(1):1–26, 1982.

Yuqing Wu, Jignesh Patel, and H. V. Jagadish. Using histograms to estimate answer size for XML queries. *Information System*, 28(1-2):33–59, 2003.

Clement T. Yu, K. Lam, and Gerard Salton. Term weighting in information retrieval using the term precision model. *Journal of ACM*, 29(1):152–170, 1982.

Appendix A. Analysis of $\frac{D_s}{D}$ and $\frac{D}{D_s}$

Recall we compute the effective sample size D_s from sketches:

$$D_s = \min\left(\max(\text{ID}(\text{K}_1)), \max(\text{ID}(\text{K}_2)), ..., \max(\text{ID}(\text{K}_m))\right).$$

We will show that the following two approximations hold with high accuracy.

$$\text{E}\left(\frac{D_s}{D}\right) \approx \min\left(\frac{k_1}{f_1}, \frac{k_2}{f_2}, ..., \frac{k_m}{f_m}\right),$$
$$\text{E}\left(\frac{D}{D_s}\right) \approx \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}, ..., \frac{f_m}{k_m}\right),$$

where f_i are the number of non-zero elements in u_i and k_i the number of sketches taken from postings P_i.

Substituting $Z_i = \max(\text{ID}(\text{K}_i))$, $D_s = \min(Z_1, Z_2, ..., Z_m)$. It is clear that Z_i is the (k_i)th order statistics of the set $\text{ID}(\text{K}_i) \in \Omega = \{1, 2, ..., D\}$, with the probability mass function (PMF)

and moments (Siegrist, 1997):

$$
\begin{aligned}
P(Z_i = t) &= \frac{\binom{t-1}{k_i-1}\binom{D-t}{f_i-k_i}}{\binom{D}{f_i}}, \quad \mathrm{E}(Z_i) = \frac{k_i(D+1)}{f_i+1} \approx \frac{k_i}{f_i}D, \\
\mathrm{Var}(Z_i) &= \frac{(D+1)(D-f_i)k_i(f_i+1-k_i)}{(f_i+1)^2(f_i+2)} \\
&\approx \frac{D(D-f_i)k_i(f_i-k_i)}{f_i^3} \leq \frac{1}{k_i}\frac{f_i-k_i}{f_i}\left(\mathrm{E}(Z_i)\right)^2. \\
\Longrightarrow \frac{\sqrt{\mathrm{Var}(Z_i)}}{\mathrm{E}(Z_i)} &\leq \sqrt{\frac{1}{k_i}\frac{f_i-k_i}{f_i}} \to 0 \quad \text{very quickly}
\end{aligned}
$$

Therefore Z_i is sharply concentrated about its mean.

By Jensen's inequality, we know that

$$
\begin{aligned}
&\mathrm{E}(D_s) = \mathrm{E}(\min(Z_1, Z_2, ..., Z_m)) \\
\leq &\min(\mathrm{E}(Z_1), \mathrm{E}(Z_2), ..., \mathrm{E}(Z_m)) = \min\left(\frac{k_1}{f_1}D, ..., \frac{k_m}{f_m}D\right).
\end{aligned}
$$

Assuming $\mathrm{E}(Z_1)$ is the smallest among all $\mathrm{E}(Z_i)$,

$$
\begin{aligned}
&\min(\mathrm{E}(Z_1), \mathrm{E}(Z_2), ..., \mathrm{E}(Z_m)) - \mathrm{E}(D_s) \\
=&\mathrm{E}(\max(\mathrm{E}(Z_1) - Z_1, \mathrm{E}(Z_1) - Z_2, ..., \mathrm{E}(Z_1) - Z_m) \\
\leq&\mathrm{E}(\max(\mathrm{E}(Z_1) - Z_1, \mathrm{E}(Z_2) - Z_2, ..., \mathrm{E}(Z_m) - Z_m) \\
\leq&\sum_{i=1}^{m}\mathrm{E}(\mathrm{E}(Z_i) - Z_i) \leq \sum_{i=1}^{m}\sqrt{\mathrm{Var}(Z_i)} \leq \sum_{i=1}^{m}\sqrt{\frac{1}{k_i}\frac{f_i-k_i}{f_i}}\mathrm{E}(Z_i),
\end{aligned}
$$

which is very crude but nevertheless shows that our approximation of $\mathrm{E}(D_s)$ is asymptotically exact. For convenience, we write

$$
\mathrm{E}\left(\frac{D_s}{D}\right) \approx \min\left(\frac{k_1}{f_1}, \frac{k_2}{f_2}, ..., \frac{k_m}{f_m}\right),
$$

Again by Jensen's inequality, we have

$$
\mathrm{E}\left(\frac{D}{D_s}\right) \geq \frac{1}{\mathrm{E}\left(\frac{D_s}{D}\right)} \geq \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}, ..., \frac{f_m}{k_m}\right)
$$

From statistical results, we know that we can approximate $\mathrm{E}\left(\frac{1}{X}\right)$ by $\frac{1}{\mathrm{E}(X)}$, with errors determined by $\mathrm{Var}(X)$, which vanishes very quickly in our case. Thus, we can approximate

$$
\mathrm{E}\left(\frac{D}{D_s}\right) \approx \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}, ..., \frac{f_m}{k_m}\right).
$$

Appendix B. Covariance Matrix of Margin-constrained Multi-way Tables

This section derives the covariance matrix for the margin-constrained maximum likelihood estimator in Section 3.

Recall the log likelihood function is

$$\log \mathbf{Pr}(\mathbf{S}|D_s;\mathbf{X}) \propto Q = s_i \sum_{i=1}^{N} \log x_i,$$

whose Hessian ($\bigtriangledown^2 Q$) is :

$$\bigtriangledown^2 Q = \left[\frac{\partial^2 Q}{\partial x_i x_j}\right] = -\text{diag}\left[\frac{s_1}{x_1^2}, \frac{s_2}{x_2^2}, ..., \frac{s_N}{x_N^2}\right].$$

Normally, the (asymptotic) covariance matrix of a maximum likelihood estimator (MLE), $\hat{\mathbf{X}}$ is $\text{Cov}\left(\hat{\mathbf{X}}\right) = (\text{I}(\mathbf{X}))^{-1}$, where $\text{I}(\mathbf{X})$, the expected Fisher Information, is $\text{I}(\mathbf{X}) = \text{E}\big(-\bigtriangledown^2 Q\big)$ (Lehmann and Casella, 1998, Theorem 6.3.10). Our situation is more special because the log likelihood is conditional on D_s and we need to consider the margin constraints.

We seek a partition of the constraint matrix $\mathbf{C} = [\mathbf{C_1}, \mathbf{C_2}]$, such that $\mathbf{C_2}$ is invertible. In our construction, the jth column of $\mathbf{C}_2$ is the column of $\mathbf{C}$ such that last entry of the jth row of $\mathbf{C}$ is 1. An example for $m = 3$ would be

$$\mathbf{C_1} = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \quad \mathbf{C_2} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},$$

where $\mathbf{C_1}$ is the [1 2 3 5] columns of $\mathbf{C}$ and $\mathbf{C_2}$ is the [4 6 7 8] columns of $\mathbf{C}$. $\mathbf{C}_2$ constructed this way is always invertible because its determinant is always one. Corresponding to the partition of $\mathbf{C}$, we partition $\mathbf{X} = [\mathbf{X}_1, \mathbf{X}_2]^{\text{T}}$. For example, when $m = 3$, $\mathbf{X}_1 = [x_1, x_2, x_3, x_5]^{\text{T}}$, $\mathbf{X}_2 = [x_4, x_6, x_7, x_8]^{\text{T}}$. Note that all these partitions are done systematically. We can then express $\mathbf{X_2}$ to be

$$\mathbf{X}_2 = \mathbf{C}_2^{-1}\left(\mathbf{F} - \mathbf{C}_1\mathbf{X_1}\right) = \mathbf{C}_2^{-1}\mathbf{F} - \mathbf{C}_2^{-1}\mathbf{C}_1\mathbf{X_1}.$$

The log likelihood function Q, which is separable, can then be expressed as $Q(\mathbf{X}) = Q_1(\mathbf{X}_1) + Q_2(\mathbf{X}_2)$.

By the matrix derivative chain rule, the Hessian of Q with respect to $\mathbf{X}_1$ would be

$$\bigtriangledown_1^2 Q = \bigtriangledown_1^2 Q_1 + \bigtriangledown_1^2 Q_2 = \bigtriangledown_1^2 Q_1 + \left(\mathbf{C}_2^{-1}\mathbf{C}_1\right)^{\text{T}} \bigtriangledown_2^2 Q_2 \left(\mathbf{C}_2^{-1}\mathbf{C}_1\right),$$

where we use $\bigtriangledown_1^2$ and $\bigtriangledown_2^2$ to indicate the Hessians are with respect to $\mathbf{X}_1$ and $\mathbf{X}_2$, respectively.

Conditional on D_s,

$$\begin{aligned} \text{I}(\mathbf{X}_1) =& \text{E}\left(-\bigtriangledown_1^2 Q|D_s\right) \\ =& -\text{E}(\bigtriangledown_1^2 Q_1|D_s) - \left(\mathbf{C}_2^{-1}\mathbf{C}_1\right)^{\text{T}} \text{E}(\bigtriangledown_2^2 Q_2|D_s)\left(\mathbf{C}_2^{-1}\mathbf{C}_1\right). \end{aligned}$$

Therefore, the (asymptotic) unconditional covariance matrix would be

$$\text{Cov}(\hat{\mathbf{X}}_1) = \text{E}\left(\text{I}(\mathbf{X}_1)^{-1}\right) = \text{E}\left(\frac{D}{D_s}\right) \times$$
$$\left(\text{diag}\left[\frac{1}{x_i}, x_i \in \mathbf{X}_1\right] + \left(\mathbf{C}_2^{-1}\mathbf{C}_1\right)^{\text{T}} \text{diag}\left[\frac{1}{x_i}, x_i \in \mathbf{X}_2\right] \left(\mathbf{C}_2^{-1}\mathbf{C}_1\right)\right)^{-1}.$$

$\text{E}\left(\frac{D}{D_s}\right)$ is approximated by $\text{E}\left(\frac{D}{D_s}\right) \approx \max\left(\frac{f_1}{k_1}, \frac{f_2}{k_2}, ..., \frac{f_m}{k_m}\right)$.

B.1 A Special Case for $m = 2$

When $m = 2$, we have

$$\bigtriangledown^2 Q = -\text{diag}\left[\frac{s_1}{x_1^2}, \frac{s_2}{x_2^2}, \frac{s_3}{x_3^2}, \frac{s_4}{x_4^2}\right],$$
$$\bigtriangledown_1^2 Q_1 = -\frac{s_1}{x_1^2},\ \bigtriangledown_2^2 Q_2 = -\text{diag}\left[\frac{s_1}{x_1^2}, \frac{s_2}{x_2^2}, \frac{s_3}{x_3^2}\right],$$

$$\mathbf{C} = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix},\ \mathbf{C_1} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},\ \mathbf{C_2} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix}$$

$$\left(\mathbf{C}_2^{-1}\mathbf{C}_1\right)^{\text{T}} \bigtriangledown_2^2 Q_2 \mathbf{C}_2^{-1}\mathbf{C}_1$$
$$= -\begin{bmatrix} 1 & 1 & -1 \end{bmatrix} \begin{bmatrix} \frac{s_2}{x_2^2} & 0 & 0 \\ 0 & \frac{s_3}{x_3^2} & 0 \\ 0 & 0 & \frac{s_4}{x_4^2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ -1 \end{bmatrix} = -\frac{s_2}{x_2^2} - \frac{s_3}{x_3^2} - \frac{s_4}{x_4^2}$$

Hence,

$$-\bigtriangledown_1^2 Q = \frac{s_1}{x_1^2} + \frac{s_2}{x_2^2} + \frac{s_3}{x_3^2} + \frac{s_4}{x_4^2}$$
$$\text{Cov}(\hat{\mathbf{X}}_1) = \text{Var}(\hat{x}_1) = \text{E}\left(\frac{D}{D_s}\right) \frac{1}{\frac{1}{x_1} + \frac{1}{f_1 - x_1} + \frac{1}{f_2 - x_2} + \frac{1}{D - f_1 - f_2 + x_1}}$$

Appendix C. Broder's Sketches

Broder's sketches can be conveniently explained by a set intersection problem. Suppose there are two sets of integers (e.g., postings), P_1 and P_2, ranging from 1 to D, i.e., $\text{P}_1, \text{P}_2 \subseteq \Omega = \{1, 2, ..., D\}$. Broder's min-wise sketch algorithm applies k random permutations (π_1, π_2, ..., π_k) on Ω. Upon each permutation π_i, the probability that the minimums in $\pi_i(\text{P}_1)$ and $\pi_i(\text{P}_2)$ are equal is

$$\mathbf{Pr}\left(\min(\pi_i(\text{P}_1)) = \min(\pi_i(\text{P}_2))\right) = \frac{|\text{P}_1 \cap \text{P}_2|}{|\text{P}_1 \cup \text{P}_2|} = R, \tag{28}$$

where R is referred as "resemblance" or "Jaccard coefficient." With k min-wise independent random permutations, the resemblance R can be estimated without bias.

Broder's original sketches described in (Broder, 1997), however, required only one permutation π hence more efficient although the estimation is slightly more sophisticated. After a permutation π on Ω, a sketch K_i for P_i is the k smallest elements from $\pi(\mathrm{P}_i)$. (Broder, 1997) provided an unbiased estimator of R:

$$\frac{1}{k}\left(|\{k \text{ smallest in } \mathrm{K}_1 \cup \mathrm{K}_2\} \cap \{\mathrm{K}_1 \cap \mathrm{K}_2\}|\right), \tag{29}$$

Why (29)? Our explanation is slightly different from the one in (Broder, 1997). Within the set $\{k \text{ smallest in } \mathrm{K}_1 \cup \mathrm{K}_2\}$, the elements that belong to $\mathrm{P}_1 \cap \mathrm{P}_2$ are: $\{k \text{ smallest in } \mathrm{K}_1 \cup \mathrm{K}_2\} \cap \{\mathrm{K}_1 \cap \mathrm{K}_2\}$, whose size is denoted by a_s^{B}. This produces a hypergeometric sample. That is, we sample k elements from $\mathrm{P}_1 \cup \mathrm{P}_2$ randomly without replacement, obtaining a_s^{B} elements that belong to $\mathrm{P}_1 \cap \mathrm{P}_2$. Then

$$\mathrm{E}\left(a_s^{\mathrm{B}}\right) = k\frac{|\mathrm{P}_1 \cap \mathrm{P}_2|}{|\mathrm{P}_1 \cup \mathrm{P}_2|} = kR, \;\Rightarrow\; \mathrm{E}\left(\frac{a_s^{\mathrm{B}}}{k}\right) = \mathrm{E}(\hat{R}_B) = R.$$

The variance of $\hat{R}_B$ is then:

$$\mathrm{Var}\left(\hat{R}_B\right) = \frac{1}{k}R(1-R)\frac{|\mathrm{P}_1 \cup \mathrm{P}_2| - k}{|\mathrm{P}_1 \cup \mathrm{P}_2| - 1} \approx \frac{1}{k}R(1-R).$$

We can estimate the association ($a = |\mathrm{P}_1 \cap \mathrm{P}_2|$) from the resemblance ($R$), by the definition: $R = \frac{|\mathrm{P}_1 \cap \mathrm{P}_2|}{|\mathrm{P}_1 \cup \mathrm{P}_2|} = \frac{a}{f_1+f_2-a}$, i.e., one can get an estimator of a: [8]

$$\hat{a}_B = (f_1+f_2)\frac{\hat{R}_B}{1+\hat{R}_B},$$
$$\mathrm{Var}\left(\hat{a}_B\right) = \frac{1}{k}\frac{a(f_1+f_2-2a)(f_1+f_2-a)^2}{(f_1+f_2)^2}.$$

C.1 Our improvement

Broder's sketches used only half (k out of $2k$) of the samples, as can be easily seen from (29). The other half discarded still contain useful information. Throwing out half of the samples and using a fixed k make the analysis easier, of course.

In contrast, our method always uses more samples because only the samples larger than $D_s = \min\left(\max(\mathrm{K}_1), \max(\mathrm{K}_2)\right)$ are discarded. If we sample the postings proportional to their lengths, we expect that almost all sketch samples can be utilized.

Therefore, it is intuitive why our method can halve the variance of Broder's sketches.

Less obviously, another reason why our estimator improves Broder's is that we are working with a four-cell contingency table while Broder worked with a two-cell model. Considering more refined structure of the data allows for more effective use of the higher order interactions of the data, hence further improves the results.

8. If $\hat{a}$ is an unbiased estimator of a, then $\mathrm{Var}\left(h(\hat{a})\right) = \mathrm{Var}\left(\hat{a}\right)\left(h'(a)\right)^2$, asymptotically, for any continuous function $h(a)$, provided the first derivative $h'(a)$ exists and non-zero. This is the so-called "Delta method."

Appendix D. Variances For Histogram Estimation

This section derives the variances for estimating histograms under marginal constraints as described in Section 4.

We represent joint histograms as contingency tables. Recall that $\mathrm{S} = \{s_{i,j}\}_{i=0}^{I}{}_{j=0}^{J}$ denotes the sample contingency table and $\mathbf{X} = \{x_{i,j}\}_{i=0}^{I}{}_{j=0}^{J}$ denotes the original contingency table to be estimated. We vectorize the tables row-wise, i.e., $\mathbf{Z} = \mathrm{vec}\{\mathbf{X}\} = \{z_m\}_{m=1}^{(I+1)(J+1)}$ for the original table and $\mathbf{H} = \mathrm{vec}\{\mathbf{S}\} = \{h_m\}_{m=1}^{(I+1)(J+1)}$ for the observed sample table. We will give a simple example for $I = J = 2$ to help visualize the procedure at the end of this section.

There are $I + J + 1$ constraints, i.e., row sums $\{x_{i+}\}_{i=1}^{I}$, column sums $\{x_{+j}\}_{j=1}^{J}$, and the total sum $\sum_{m=1}^{(I+1)(J+1)} z_m = D$. Since the effective number of degrees of freedom is $I \times J$, we will partition the table into two parts: $\mathbf{Z}_1$ and $\mathbf{Z}_2$. $\mathbf{Z}_1$ corresponds to $\mathbf{X}_1 = \{x_{i,j}\}_{i=1}^{I}{}_{j=1}^{J}$ and $\mathbf{Z}_2$ corresponds to the rest of the table. The trick is to represent $\mathbf{Z}_2$ in terms of $\mathbf{Z}_1$ so that we can apply the multivariate large sample theory for the asymptotic covariance matrix of $\mathbf{Z}_1$. It is not hard to show that

$$\mathbf{Z}_2 = \mathbf{C}_1 - \mathbf{C}_2\mathbf{Z}_1,$$

where

$$\mathbf{C}_1 = \begin{bmatrix} x_{0+} + x_{+0} - D \\ \{x_{+j}\}_{j=1}^{J} \\ \{x_{i+}\}_{i=1}^{I} \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} & -1_{IJ}^{\mathrm{T}} & \\ \mathrm{I}_J & \mathrm{I}_J & \ldots & \mathrm{I}_J \\ 1_J^{\mathrm{T}} & 0_J^{\mathrm{T}} & \ldots & 0_J^{\mathrm{T}} \\ 0_J^{\mathrm{T}} & 1_J^{\mathrm{T}} & \ldots & 0_J^{\mathrm{T}} \\ & \ldots & & \\ 0_J^{\mathrm{T}} & 0_J^{\mathrm{T}} & \ldots & 1_J^{\mathrm{T}} \end{bmatrix},$$

where I_J denotes the identity matrix of size $J \times J$, 1_J denotes a vector of ones of length J and 0_J denotes a vector of zeros of length J.

Assuming "sample-with-replacement," $\mathbf{Z}$ follows a multinomial distribution, with a log likelihood function (let $N = (I + 1)(J + 1)$):

$$Q(\mathbf{Z}) \propto \sum_{m=1}^{N} h_m \log z_m, \qquad \bigtriangledown^2 Q = -\mathrm{diag}\left[\frac{h_1}{z_1^2}, \frac{h_2}{z_2^2}, ..., \frac{h_N}{z_N^2}\right].$$

The log likelihood function Q, which is separable, can then be expressed as

$$Q(\mathbf{Z}) = Q_1(\mathbf{Z}_1) + Q_2(\mathbf{Z}_2).$$

By the matrix derivative chain rule, the Hessian of Q with respect to $\mathbf{Z}_1$ would be

$$\bigtriangledown_1^2 Q = \bigtriangledown_1^2 Q_1 + \bigtriangledown_1^2 Q_2 = \bigtriangledown_1^2 Q_1 + \mathbf{C}_2^{\mathrm{T}} \bigtriangledown_2^2 Q_2 \mathbf{C}_2,$$

where we use $\bigtriangledown_1^2$ and $\bigtriangledown_2^2$ to indicate that the Hessians are with respect to $\mathbf{Z}_1$ and $\mathbf{Z}_2$, respectively.

Since we estimate $\mathbf{Z}$ by MLE, the expected Fisher Information of $\mathbf{Z}_1$ is

$$\mathrm{I}(\hat{\mathbf{Z}}_1) = \mathrm{E}\left(-\bigtriangledown_1^2 Q|D_s\right) = -\mathrm{E}(\bigtriangledown_1^2 Q_1|D_s) - \mathbf{C}_2^{\mathrm{T}}\mathrm{E}(\bigtriangledown_2^2 Q_2|D_s)\mathbf{C}_2.$$

Because $\mathrm{E}\left(h_m|D_s\right) = \frac{D_s}{D} z_m$, we can evaluate the above expectations, i.e.,

$$\begin{aligned}
\mathrm{E}(-\bigtriangledown_1^2 Q_1|D_s) &= \mathrm{diag}\left[\mathrm{E}\left(\frac{h_m}{z_m^2}|D_s\right), z_m \in \mathbf{Z}_1\right] \\
&= \frac{D_s}{D}\mathrm{diag}\left[\frac{1}{z_m}, z_m \in \mathbf{Z}_1\right], \\
\mathrm{E}(-\bigtriangledown_2^2 Q_2|D_s) &= \frac{D_s}{D}\mathrm{diag}\left[\frac{1}{z_m}, z_m \in \mathbf{Z}_2\right].
\end{aligned}$$

By the large sample theory, the asymptotic covariance matrix of $\hat{\mathbf{Z}}_1$ would be

$$\begin{aligned}
&\mathrm{Cov}(\hat{\mathbf{Z}}_1) = \mathrm{E}\left(\mathrm{I}(\hat{\mathbf{Z}}_1)^{-1}\right) \\
=&\mathrm{E}\left(\frac{D}{D_s}\right)\left(\mathrm{diag}\left[\frac{1}{z_m}, z_m \in \mathbf{Z}_1\right] + \mathbf{C}_2^{\mathrm{T}}\mathrm{diag}\left[\frac{1}{z_m}, z_m \in \mathbf{Z}_2\right]\mathbf{C}_2\right)^{-1}.
\end{aligned}$$

The following example for $I = J = 2$ may help visualize the above formulations.

D.1 An Example with $I = 2, J = 2$

$$\begin{aligned}
&\mathbf{Z} = [z_1\ z_2\ z_3 \quad z_4\ z_5\ z_6 \quad z_7\ z_8\ z_9]^{\mathrm{T}} \\
=&[x_{0,0}\ x_{0,1}\ x_{0,2} \quad x_{1,0}\ x_{1,1}\ x_{1,2} \quad x_{2,0}\ x_{2,1}\ x_{2,2}]^{\mathrm{T}} \\
&\mathbf{Z}_1 = [z_5\ z_6\ z_8\ z_9]^{\mathrm{T}} = [x_{1,1}\ x_{1,2}\ x_{2,1}\ x_{2,2}]^{\mathrm{T}}, \\
&\mathbf{Z}_2 = [z_1\ z_2\ z_3\ z_4\ z_7]^{\mathrm{T}} = [x_{0,0}\ x_{0,1}\ x_{0,2}\ x_{1,0}\ x_{2,0}]^{\mathrm{T}}.
\end{aligned}$$

$$\mathbf{C}_1 = \begin{bmatrix} x_{0+} + x_{+0} - D \\ x_{+1} \\ x_{+2} \\ x_{1+} \\ x_{2+} \end{bmatrix}, \quad \mathbf{C}_2 = \begin{bmatrix} -1 & -1 & -1 & -1 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix},$$

$$\mathrm{Cov}(\hat{\mathbf{Z}}_1) = \mathrm{E}\left(\frac{D}{D_s}\right) \times \left(\begin{array}{c} \mathrm{diag}\left[\frac{1}{x_{1,1}}, \frac{1}{x_{1,2}}, \frac{1}{x_{2,1}}, \frac{1}{x_{2,2}}\right] + \\ \mathbf{C}_2^{\mathrm{T}}\mathrm{diag}\left[\frac{1}{x_{0,0}}, \frac{1}{x_{0,1}}, \frac{1}{x_{0,2}}, \frac{1}{x_{1,0}}, \frac{1}{x_{2,0}}\right]\mathbf{C}_2 \end{array} \right)^{-1}.$$

D.2 The Variance of $\hat{a}_{MLE,c}$

Recall we can estimate the inner product from the contingency table:

$$\hat{a}_{MLE,c} = \sum_{i=1}^{I}\sum_{j=1}^{J}(ij)\hat{x}_{i,j,MLE},$$

whose variance would be

$$\text{Var}\left(\hat{a}_{MLE,c}\right) = \mathbf{e}^{\text{T}}\text{Cov}\left(\hat{\mathbf{Z}}_1\right)\mathbf{e},$$

where $\mathbf{e}$ is a vector:

$$\mathbf{e}^{\text{T}} = [(ij)]_{i=1\ \ j=1}^{I\ \ J} = [1, 2, ..., J, \quad 2, 4, ..., 2J, \quad ..., \quad I, ..., IJ]^{\text{T}}.$$