CONDITIONAL RANDOM SAMPLING: A SKETCH-BASED SAMPLING TECHNIQUE FOR SPARSE DATA

by

Ping Li
Kenneth W. Church
Trevor J. Hastie

Technical Report No. 2006-8
June 2006

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065
CONDITIONAL RANDOM SAMPLING: A SKETCH-BASED SAMPLING TECHNIQUE FOR SPARSE DATA

by

Ping Li
Department of Statistics
Stanford University

Kenneth W. Church
Microsoft Corporation
Redmond, WA 98052

Trevor J. Hastie
Department of Statistics
Stanford University

Technical Reports No. 2006-8
June 2006

This research was supported in part by National Science Foundation grants DMS-0306612 and DMS-0505676 and National Institutes of Health grant 2R01 CA 72028-07.

Department of Statistics
STANFORD UNIVERSITY
Stanford, California 94305-4065

http://www-stat.stanford.edu
Conditional Random Sampling: A Sketch-based Sampling Technique for Sparse Data

Ping Li
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Kenneth W. Church
Microsoft Research
Microsoft Corporation
Redmond, WA 98052, USA

Trevor J. Hastie
Department of Statistics
Stanford University
Stanford, CA 94305, USA

Editor:

Abstract

We develop a sketch-based sampling algorithm, called Conditional Random Sampling (CRS), particularly suitable for sparse data. In many important applications such as information retrieval, the datasets are often very large and highly sparse. Our technique is a combination of sketching and sampling in that it converts sketches of the data into conditional random samples online in the estimation stage, from which we estimate the original space using well-understood statistical methods. In addition, we can take advantage of the marginal information to (often considerably) enhance the estimation accuracy at little incremental cost.

Our method can efficiently approximate pairwise distances (inner product, $l_1$, $l_2$, or $l_p$ distances) and multi-way associations, as well as many other summary statistics. This study focuses on pairwise $l_2$ and $l_1$ distances (and inner products) for which random projections are popular. Our method is provably better than random projections in boolean (0/1) data. We show using real-valued text and image data that our algorithm often outperforms random projections in approximating inner product and $l_2$ distance. In many learning and data mining tasks including association rules, estimating joins, distance-based clustering, nearest neighbor searching, and kernels for (e.g.,) support vector machines (SVM), computing pairwise (or multi-way) distances is usually the vital step. Therefore, Conditional Random Sampling will be useful for these applications.

1. Introduction

We develop a sketch-based sampling algorithm for sparse data, called Conditional Random Sampling (CRS). In many important learning and data mining applications, the datasets are often very large and highly sparse. For example, a term-by-document matrix often contains less than 1% non-zero entries (Dhillon and Modha, 2001) (also see Table 1). In market-basket analysis, the customer-by-product matrix is also highly sparse (Aggarwal and Wolf, 1999; Strehl and Ghosh, 2000). The conventional random sampling (i.e., randomly picking a small fraction of the data) often performs poorly in sparse data because most of the samples are zeros. Also, the estimation errors (variances) could be large in heavy-tailed data, ubiquitous in large datasets (Newman, 2005).
As alternatives to sampling, various sketching algorithms have become popular, for example, random projections (Vempala, 2004) (for general real-valued data) and Broder’s min-wise sketches (Broder, 1997; Broder et al., 1998; Charikar, 2002) (for boolean data). Sketching algorithms are designed for approximating specific summary statistics, in particular, various distance measures.

Table 1: Page hits (Google, June 19, 2006) for a few words. The table suggests that the total number of (English) Web pages should be at least $10^{10}$. The term-by-document matrix at Web scale is highly sparse. However, even for the rare words, the page hits are still large. The sparsity varies drastically from word to word (and of course from document to document also).

<table>
<thead>
<tr>
<th>Query</th>
<th>Hits (Google)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>22,340,000,000</td>
</tr>
<tr>
<td>The</td>
<td>20,980,000,000</td>
</tr>
<tr>
<td>This</td>
<td>15,520,000,000</td>
</tr>
<tr>
<td>University</td>
<td>3,980,000,000</td>
</tr>
<tr>
<td>Country</td>
<td>2,290,000,000</td>
</tr>
<tr>
<td>Conference</td>
<td>1,640,000,000</td>
</tr>
<tr>
<td>Knuth</td>
<td>5,530,000</td>
</tr>
<tr>
<td>“John Nash”</td>
<td>1,090,000</td>
</tr>
<tr>
<td>Diaconis</td>
<td>164,000</td>
</tr>
<tr>
<td>Kalevala</td>
<td>1,330,000</td>
</tr>
<tr>
<td>Saccade</td>
<td>526,000</td>
</tr>
<tr>
<td>Griceofulvin</td>
<td>423,000</td>
</tr>
</tbody>
</table>

We abstract the data to be a matrix $A$ of $n$ rows (data points) and $D$ columns (features, or attributes). For example, $A$ can be the term-by-document matrix with $n$ as the total number of word types and $D$ as the total number of documents. In modern search engines, roughly $n \approx 10^6 \sim 10^7$ and $D \approx 10^{10} \sim 10^{11}$. Table 1 lists the page hits (i.e., the total number of Web pages containing the word) for a few words, including function words, frequent words, names, and rare words. This table, to an extent, verifies that the term-by-document matrix at Web scale would be highly sparse (especially for these infrequent and interesting words such as names). On the other hand, even for these rare words, the page hits are still large (e.g., half a million); hence dimension reduction could be still desirable even if we were only interested in the so-called rare items.

There is another important consideration. Since the sparsity is very different for different data points, we would like to have a flexible sampling algorithm that could adjust the sampling rate according to the sparsity. For example, for some extremely sparse data points, we may be able to consider all non-zero entries (i.e., sampling rate = 100%). Conditional Random Sampling provides such a desired flexibility.

Many important applications concern only the data pairwise (or multi-way) distances including association rules, distance-based clustering, multi-dimensional scaling, kernels for (e.g.,) support vector machines (SVM). For these applications, computing distances is often the first and vital step. For a large training set, however, computing distances exactly is often too time-consuming or even infeasible. For example, given a data matrix $A$ with large $n$ and $D$, evaluating the Gram matrix (or the inner product kernel) $AA^T$ costs $O(n^2D)$, which could be daunting. Sampling and sketching methods have been proposed for approximating the Gram matrix and kernels (Achlioptas et al., 2001; Drineas and Mahoney, 2005). For example, random projections multiply $A$ with a small
random matrix \( R \in \mathbb{R}^{D \times k} \) and compute pairwise distances using the projected data, at a total cost of \( O(nDk + n^2k) \). The cost reduction would be enormous if \( k \ll \min(n, D) \).

We develop Conditional Random Sampling (CRS), which could be an alternative to random projections. It is also capable of estimating \( l_1 \) distances (in fact in any \( l_p \) norms). Using the \( l_1 \) norm has become popular because of its well-known robustness. Success stories include Lasso (Tibshirani, 1996), 1-norm SVM (Zhu et al., 2003), and 1-norm kernels (e.g., Laplacian radial basis kernel (Chapelle et al., 1999) and total variation kernel (Hein and Bousquet, 2005)).

Conditional Random Sampling can also approximate multi-way distances (e.g., multi-way set intersections) and multi-way histograms while random projections are limited to the pairwise (two-way) case. We will mainly focus on the pairwise case since it is more convenient to compare our algorithm with others in the pairwise case.

The rest of the paper is organized as follows. Section 2 describes the procedure of Conditional Random Sampling. Since we will compare our method with random projections, we provide a brief introduction in Section 3. Section 4 describes the obvious estimators (and the estimation variances) of the inner product, \( l_1 \) distance, and \( l_2 \) distance, using our algorithm. In Section 5, we show that the estimation errors can often be much reduced by taking advantage of the marginal information. Section 6 theoretically illustrates that our method outperforms random projections in the boolean (0/1) data case. Section 7 empirically demonstrates that our algorithm often outperforms random projections in approximating inner products and \( l_2 \) distances (if we take advantage of the margins), using a variety of text and image datasets. Finally, Section 8 concludes the paper.

2. The Sampling Procedure

![Fig 1](image1.png)
(a) Original sparse data matrix
(b) Column-permuted data matrix
(c) Postings: non-zero entries
(d) sketches: front of postings

Figure 1: A global view of the sketching stage. We randomly permute the columns of a sparse data matrix and store only the non-zero entries called postings. Sketches are simply the front of postings. Note that in the implementation, we only need to maintain a permutation mapping of the column IDs. Sketches are then the first few non-zero entries with the smallest (permuted) column IDs.

Conditional Random Sampling is two-stage procedure. In the sketching stage, we scan the data matrix once and store a fraction of the non-zero elements in each data point, as “sketches.” In the estimation stage, we generate conditional random samples online pairwise (for two-way) or group-wise (for multi-way); hence we name it Conditional Random Sampling (CRS). This procedure effectively exploits data sparsity while benefiting from random sampling, for which matured statistical estimation techniques are available.

Figure 1 provides a global view of the sketching stage. The columns of a sparse data matrix (a) are first randomly permuted (b). Then only the non-zero entries are stored, called postings (c). Sketches are simply the front of postings (d). This figure, of course, only provides an over-simplified
description of the sketching procedure. The actual implementation may be more optimized for efficiency purposes.

Obviously sketches are not random samples, which may make the estimation task difficult. We show that sketches are almost random samples pairwise (or group-wise). The following two figures illustrate this discovery. Figure 2 constructs conventional random samples from a data matrix. Figure 3 generates the same (conditional) random samples from sketches.

In Figure 2, assuming that the column IDs are random (achieved by a random permutation), we can construct conventional random samples by simply taking the first $D_s$ columns from the data matrix of $D$ columns ($D_s \ll D$ in real applications).

$$
\begin{array}{ccccccccc}
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\
10 & 11 & 12 & 13 & 14 & 15 & 16 \\
\hline
u_1 & 0 & 3 & 0 & 2 & 0 & 1 & 0 & 0 & 1 & 2 & 1 & 0 & 1 & 0 & 2 & 0 \\
\hline
u_2 & 1 & 4 & 0 & 0 & 1 & 2 & 0 & 1 & 0 & 0 & 1 & 3 & 0 & 0 & 2 & 1 & 1 \\
\end{array}
$$

Figure 2: A data matrix with $D = 16$. If the column IDs are random, the first $D_s = 10$ columns constitute a random sample. $u_i$ denotes the $i$th row of the data matrix.

For sparse data, we only need to store the non-zero elements in the form of tuples “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 called “postings.” We denote the postings by $P_i$ for the row $u_i$. Figure 3(a) shows the postings for the same data matrix in Figure 2. The tuples are sorted ascending by the IDs.

$$
P_1: 2 (3) 4 (2) 6 (1) 9 (1) 10 (2) 11 (1) 13 (1) 15 (2) \\
P_2: 1 (1) 2 (4) 5 (1) 6 (2) 8 (1) 11 (3) 14 (2) 15 (1) 16 (1)
$$

Figure 3: (a): Postings consist of tuples “ID (Value),” (b): Sketches are the first few ($k_i$) entries of postings sorted ascending by IDs. We choose $k_1 = 5, k_2 = 6$. In this case, $D_s = \min(10, 11) = 10$. If excluding 11(3) in $K_2$, we obtain the same samples as if we sampled the first $D_s = 10$ columns in the data matrix.

We sample from beginning of the postings as shown in Figure 3(b). 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

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

we obtain exactly the same samples as if we directly sampled the first $D_s$ columns from the data matrix in Figure 2. This way, we can convert sketches into random samples by conditioning on $D_s$, which differs pairwise and we do not know beforehand. For example, when estimating pairwise distances for all $n$ data points, we will have $\frac{n(n-1)}{2}$ different values of $D_s$.

The estimation task can be extremely simple. After we have constructed the conditional random samples for sketches $K_1$ and $K_2$ with the effective sample size $D_s$, we can compute any distances ($l_2, l_1$, or inner products) from the samples and multiply them by $\frac{D_s}{D}$ to estimate the original space.

Of course, we can also estimate other summary statistics or multi-way distances using sketches.
2.1 How to Achieve Randomness?

In Figure 2, we assume that the matrix column IDs are random, which can be achieved in various ways (e.g., hashing (Rabin, 1981; Broder, 1997)). We apply a random permutation, denoted by \( \pi \), on the column IDs, i.e.,

\[
\pi : \Omega \rightarrow \Omega, \quad \Omega = \{1, 2, 3, \ldots, D\}. \tag{2}
\]

Let \( \pi(P_i) \) denote the postings \( P_i \) after permutation. Recall a sketch \( K_i \) is the \( k_i \) smallest elements in \( \pi(P_i) \). Thus, we have to scan \( \pi(P_i) \) to find the \( k_i \) smallest. Therefore, generating sketches for \( A \in \mathbb{R}^{n \times D} \) costs \( O(nD) \), or \( O(\sum_{i=1}^{n} f_i) \), where \( f_i \) is the number of non-zero elements in the \( i \)th row, i.e., \( f_i = |P_i| \). The cost \( O(nD) \) is less than the cost of random projections \( O(nDk) \).\(^2\)

2.2 Marginal Information Helps

Because the sketching algorithms have to scan the data matrix, it is reasonable to assume that the marginal information is known (e.g., \( f_i \) and \( ||u_i||^2 \)). This often leads to (much) sharper estimates. Of course, the estimation task will be slightly more involving. For example, the authors' recent paper on normal random projections (Li et al., 2006a) improves the estimates by solving a cubic maximum likelihood equation. In section 5, we will discuss how to utilize the marginal information to improve estimates of the histograms and inner products. In the empirical evaluations (i.e., Section 7), we show that, using marginal norms, Conditional Random Sampling usually outperforms random projections in estimating \( l_2 \) distances.

2.3 The Effective Sample Size: \( D_s \)

The "effective sample size" \( D_s \) is computed online and is different pairwise or group-wise. Using the first two rows \( u_1 \) and \( u_2 \) for the illustration, we have two important approximations (for the pairwise case), useful for later theoretical analysis:

\[
E \left( \frac{D_s}{D} \right) \approx \min \left( \frac{k_1}{f_1 + 1}, \frac{k_2}{f_2 + 1} \right), \quad E \left( \frac{D}{D_s} \right) \approx \max \left( \frac{f_1 + 1}{k_1}, \frac{f_2 + 1}{k_2} \right). \tag{3}
\]

Since the column IDs are assumed to be uniform in \( \Omega = \{1, 2, \ldots, D\} \) at random, it is intuitive that \( E \left( \frac{\max(\|K_i\|)}{D} \right) \approx \frac{k}{f_i} \). In Appendix A, we show that (3) is asymptotically exact. The errors are usually within 5% as illustrated in Figure 4. For simplicity, we replace \( f_i + 1 \) by \( f_i \).

From (3), we can infer that \( D_s \approx \frac{kD}{f} \) (suppose \( f_1 = f_2 = f \) and \( k_1 = k_2 = k \)). If \( f = 10^{-2} \), then \( D_s \approx 10^2k \), i.e., 10 sketch samples will be equivalent to \( 10^3 \) regular random samples! Because the estimation variance is inverse proportional to \( D_s \), our algorithm can reduce the variance proportionally to the data sparsity.

It is possible to improve (3). From Figure 4, we can see that \( E(D_s) \) is not affected strongly by the correlations (i.e., the \( \gamma \) values in Figure 4). Therefore, we can assume that \( ID(K_1) \) and \( ID(K_2) \) are independent for the purpose of estimating \( E(D_s) \). With this assumption, we can in principle express \( E(D_s) \) in some summation form, which can be evaluated numerically. However, we still

---

1. We ignore the possible (depending on the implementation) \( \log(k) \) factor in finding the \( k \) smallest.
2. The authors' recent work on random projections (Li et al., 2006c) suggests that we can reduce the cost from \( O(nDk) \) down to \( O(nDk^{1/2}) \) in most reasonable practical settings.
Figure 4: The ratios $E\left(\frac{D}{D_k}\right) / \max\{f_1+1, f_2+1\}k$ show that the errors from using (3) are usually within 5% when $k \geq 20$. In the simulations, $D = 10^6$, $f_1 = \alpha D$, and $f_2 = \beta f_1$. In each panel, the five curves correspond to different values of $\#\{j : u_{1,j} > 0 \& u_{2,j} > 0, 1 \leq j \leq D\}$, which measure the correlations and are set to be $\gamma f_2$, with $\gamma = 0.01, 0.1, 0.5, 0.8, 1.0$, respectively. It is clear that $\gamma$ does not affect $D_s$ strongly. In each panel, we only simulated $10^3$ times (permutations) for each $k$. Therefore the curves are not very smooth.

prefer the approximations in (3) because they are intuitive, simple, and accurate enough. Note that, we only need $E\left(\frac{D}{D_k}\right)$ for theoretically evaluating the estimation variances; it is not needed in our sketching procedure nor in the estimation stage.

2.4 Sample without Replacement

Our scheme is sample without replacement. When a data point $u_i$ is very sparse (e.g., $f_i < 100$), it is probably affordable to take the whole posting as a sketch; thus the absolute errors could be very small. For simplicity, however, we will assume sample with replacement in the analysis, which is slightly conservative. That is, the actual errors may be slightly smaller than what are theoretically predicted. This, to an extent, may compensate the errors due to the approximations in (3).

3. Brief Introduction to Random Projections

We give a brief introduction to random projections, with which we compare our method. Random projections (Vempala, 2004; Achatzios, 2003) are widely used in learning, data mining, and bioinformatics (Kaski, 1998; Arriaga and Vempala, 1999; Bingham and Mannila, 2001; Achatzios et al., 2001; Fradkin and Madigan, 2003; Buhler and Tompa, 2002; Balcan et al., 2004). The AMS sketching algorithm (Alon et al., 1996) is also one form of random projections.
Random projections multiply the data matrix $A \in \mathbb{R}^{n \times D}$ with a random matrix $R \in \mathbb{R}^{D \times k}$ to generate a compact representation $B = AR \in \mathbb{R}^{n \times k}$, from which we can estimate the original distances. For estimating $l_2$ distances, $R$ typically consists of entries of i.i.d. $N(0, 1)$ entries; hence we call it normal random projections. For $l_1$, $R$ consists of i.i.d. Cauchy $C(0, 1)$, as suggested by Indyk (2000), which is the only known method for dimension reduction in $l_1$ (Brinkman and Charikar, 2005), besides sampling. Later, impossibility results (Lee and Naor, 2004; Brinkman and Charikar, 2005) were proved for dimension reduction in $l_1$ using linear estimators.

For many applications, we only need the distances; and hence we can consider nonlinear estimators. For example, Indyk (2000) suggested using the sample median. The authors have developed various (improved) nonlinear estimators for dimension reduction in $l_1$ (Li et al., 2006b).

### 3.1 Normal Random Projections

By convention, we write $B = \frac{1}{\sqrt{k}} AR$, where $R$ consists of i.i.d. $N(0, 1)$. We focus on the first two rows of $A$ (i.e., $u_1$ and $u_2$). The first two rows in $B$ are denoted by $v_1, v_2 \in \mathbb{R}^k$. We denote

$$a = u_1^T u_2, \quad d^{(2)} = \|u_1 - u_2\|^2, \quad m_1 = \|u_1\|^2, \quad m_2 = \|u_2\|^2$$

(4)

It is easy to show that the following linear estimators of $a$ and $d^{(2)}$ are unbiased

$$\hat{a}_{NRP,MF} = v_1^T v_2, \quad \hat{d}^{(2)}_{NRP,MF} = \|v_1 - v_2\|^2,$$

(5)

with variances (Vempala, 2004; Li et al., 2006a)

$$\text{Var}(\hat{a}_{NRP,MF}) = \frac{1}{k} \left(m_1 m_2 + a^2\right), \quad \text{Var}(\hat{d}^{(2)}_{NRP,MF}) = \frac{2[d^{(2)}]^2}{k}$$

(6)

Assuming that the margins $m_1 = \|u_1\|^2$ and $m_2 = \|u_2\|^2$ are known, Li et al. (2006a) provides a maximum likelihood estimator (MLE), denoted by $\hat{a}_{NRP,MLE}$, as the solution to a cubic equation:

$$a^3 - a^2 (v_1^T v_2) + a (-m_1 m_2 + m_1 \|v_2\|^2 + m_2 \|v_1\|^2) - m_1 m_2 v_1^T v_2 = 0.$$  

(7)

The variance of $\hat{a}_{NRP,MLE}$, up to $O(k^{-2})$ terms, is

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

(8)

### 3.2 Cauchy Random Projections for Dimension Reduction in $l_1$

We let $B = AR$ with $R$ consisting of i.i.d. entries in Cauchy $C(0, 1)$. The first two rows $(u_1$ and $u_2)$ in $A$ correspond to the first two rows $(v_1$ and $v_2)$ in $B$. It is not difficult to show that $v_{1,i} - v_{2,i} \sim C(0, d^{(1)}), i = 1, 2, \ldots, k$, where $d^{(1)}$ is the $l_1$ distance, i.e., $d^{(1)} = |u_1 - u_2| = \sum_{j=1}^{D} |u_{1,j} - u_{2,j}|$. Therefore, the problem amounts to estimating the Cauchy scale parameter from $k$ i.i.d. samples. The proposed estimator is (see the technical report (Li et al., 2006b) for more details)

$$\hat{d}^{(1)}_{CRP,MLE} = \hat{d} \left(1 - \frac{1}{k}\right),$$

(9)
where the term \((1 - \frac{1}{k})\) is for bias correction and \(\hat{d}\) solves a nonlinear MLE equation:

\[
-\frac{k}{d} + \frac{2d}{\sum_{i=1}^{k} (v_{1,i} - v_{2,i})^2 + d^2} = 0. \tag{10}
\]

The leading term of the variance would be

\[
\text{Var} (d_{CRP,MLE}^{(1)}) = \frac{2[d^{(1)}]^2}{k}. \tag{11}
\]

### 3.3 Limitations of Random Projections

Random projections are designed for approximating specific pairwise distance in a specific norm. In some applications, we might want to consider both \(l_1\) and \(l_2\) norms, e.g., the Elastic Net (Zou and Hastie, 2005). For SVM, we might be interested in using other types of kernels such as the chi-squared statistic kernel (Chapelle et al., 1999). Multi-way distances (e.g., multiple set intersections) are also important for many applications in databases and information retrieval.

### 4. Simple Estimators and Variances Using Conditional Random Sampling

Recall that we compute the effective random sample size \(D_s\) and construct conditional random samples (online) from sketches \(K_1\) and \(K_2\), whenever we need to estimate \(a = u_1^T u_2, d^{(2)} = \|u_1 - u_2\|^2\), or \(d^{(1)} = |u_1 - u_2|\). We use \(\bar{u}_{1,j}\) and \(\bar{u}_{2,j}, j = 1\) to \(D_s\), to denote these random samples in \(u_1\) and \(u_2\), respectively. Assuming sample with replacement, the obvious unbiased estimators are

\[
\hat{\alpha}_{MF} = \frac{D}{D_s} \sum_{j=1}^{D_s} \bar{u}_{1,j} \bar{u}_{2,j}, \tag{12}
\]

\[
\hat{d}_{MF}^{(2)} = \frac{D}{D_s} \sum_{j=1}^{D_s} (\bar{u}_{1,j} - \bar{u}_{2,j})^2, \tag{13}
\]

\[
\hat{d}_{MF}^{(1)} = \frac{D}{D_s} \sum_{j=1}^{D_s} |\bar{u}_{1,j} - \bar{u}_{2,j}|. \tag{14}
\]

By the conditional variance formula: \(\text{Var}(\hat{X}) = E \left( \text{Var}(\hat{X} | D_s) \right) + \text{Var} \left( E(\hat{X} | D_s) \right)\)

if \(\hat{X}\) is conditionally unbiased,\(^3\) we can obtain (denote \(d^{(4)} = \sum_{j=1}^{D} (u_{1,j} - u_{2,j})^4\))

\[
\text{Var}(\hat{\alpha}_{MF}) = E \left( \frac{D}{D_s} \right) \left( \frac{1}{D_s} \sum_{j=1}^{D_s} u_{1,j}^2 u_{2,j}^2 - \frac{a^2}{D} \right) \approx \frac{\max(f_1, f_2)}{k} \frac{1}{D} \left( \frac{D}{D_s} \sum_{j=1}^{D_s} u_{1,j}^2 u_{2,j}^2 - a^2 \right), \tag{15}
\]

\[
\text{Var}(\hat{d}_{MF}^{(2)}) = E \left( \frac{D}{D_s} \right) \left( d^{(4)} - \frac{[d^{(2)}]^2}{D} \right) \approx \frac{\max(f_1, f_2)}{k} \frac{1}{D} \left( D d^{(4)} - [d^{(2)}]^2 \right), \tag{16}
\]

\[
\text{Var}(\hat{d}_{MF}^{(1)}) = E \left( \frac{D}{D_s} \right) \left( d^{(2)} - \frac{[d^{(2)}]^2}{D} \right) \approx \frac{\max(f_1, f_2)}{k} \frac{1}{D} \left( D d^{(2)} - [d^{(1)}]^2 \right). \tag{17}
\]

\(^3\) If \(\hat{X}\) is only asymptotically conditionally unbiased (e.g., MLE), \(\text{Var}(E(\hat{X} | D_s))\) is asymptotically negligible.
The sparsity $\max(f_1, f_2)$ reduces the variances. If $\max(f_1, f_2) = 0.01$, the variances can be reduced by a factor of 100. On the other hand, the variances are strongly affected by higher moments. That is, the variances of $d^{(1)}_{MF}$ and $d^{(2)}_{MF}$ are affected by the second and fourth moments, respectively. In comparison, the variances of random projections are not affected by higher moments.

Therefore, data sparsity and data heavy-tailedness are two competing factors controlling whether our method has smaller variance than random projections. There are at least three reasons why our algorithm can still work well in heavy-tailed data.

1. The data are often transformed (weighted) to ensure that the distances (especially $l_2$) are meaningful (Manning and Schutze, 1999, Chapter 15.2; Yu et al., 1982; Salton and Buckley, 1988; Dumais, 1991). Appropriate term weighting is often vital. For example, Leopold and Kindermann (2002); Lan et al. (2005) showed that in text categorization using SVM, choosing an appropriate term weighting scheme is far more important than tuning kernel functions. See similar comments in (Rennie et al., 2003) for the work on Naive Bayes text classifier.

2. Also, we can utilize the marginal information in $l_2$, as $d^{(2)} = \|u_1 - u_2\|^2 = m_1 + m_2 - 2a$. Knowing the marginal information effectively reduces the impact of data heavy-tailedness.

3. Random projections may perform poorly in estimating inner products. In (6), $\text{Var}(\hat{a}_{NRP,MF})$ is dominated by the margins. This is undesirable because even when $u_1$ and $u_2$ are only weakly correlated, the estimation variance (using random projections) is still very large. Let's take a look at the extreme case. Suppose $u_1$ and $u_2$ are independent, then $\sum_{j=1}^D u_{1,j}^2 u_{2,j}^2$ and $m_1 m_2 = \sum_{j=1}^D u_{1,j}^2 \sum_{j=1}^D u_{2,j}^2$ are asymptotically equivalent by the law of large numbers. In this case, $\text{Var}(\hat{a}_{MF}) < \text{Var}(\hat{a}_{NRP,MF})$ even ignoring the data sparsity.

Therefore, as soon to be verified empirically, we expect that our algorithm usually outperforms random projections in estimating inner products. If using the marginal norms, our algorithm is also likely to outperform random projections in estimating $l_2$ distances, even in heavy-tailed data.

5. Improving Estimates Using Marginal Information

As mentioned previously, it is reasonable to assume that we know the marginal information such as marginal norms, numbers of non-zero elements, or even marginal histograms. We utilize the marginal information by maximizing the joint likelihood under marginal constraints.

In some cases, we know the explicit likelihood. For example, when the data are integer valued (which is also the case in histograms), the joint likelihood is equivalent to that of a two-way or multi-way contingency table. In the 0/1 case, we can even express the MLE solution explicitly and derive the closed-form (asymptotic) variance.

In general (i.e., real-valued data), the joint likelihood is not available. One option is to use non-parametric maximum likelihood such as the "Empirical Likelihood," (Owen, 2001). Since our goal is to develop simple tools that could be practically useful, we make some assumptions on the data and propose an approximate MLE, which is conceptually reasonable and works well in real data.

We first describe a procedure for estimating two-way histograms (i.e., integer-valued data), which amounts to estimating contingency tables under margin constraints. Once we have estimated the contingency table, we can compute the inner product (and other summary statistics) easily.
5.1 Integer-valued Data (Histograms)

Histograms are useful for answering queries like \( \Pr(1 < u_1 < 2 \& u_2 > 2) \). Histograms contain more information than the mere inner product \( a = u_1^T u_2 \). While univariate histograms are easy to compute and store, joint histograms are much more difficult especially for high-order joins. We will focus only on two-way histograms, as the notation gets messy in the multi-way case.

Without loss of generality, we number each histogram bin \( \{0, 1, 2, \ldots\} \) as shown in Figure 5(a). Apparently, we can also think these are the original data, which happen to be integer-valued. Histograms can be conveniently represented by contingency tables, e.g., Figure 5(b)

```
<table>
<thead>
<tr>
<th>u_1</th>
<th>1 2 3 4 5 6 7 8 9 10 11 12 13 14 15</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_1</td>
<td>0 1 0 2 0 1 0 0 1 2 1 0 1 0 2</td>
</tr>
<tr>
<td>u_2</td>
<td>1 3 0 0 1 2 0 1 0 0 3 0 0 2 1</td>
</tr>
</tbody>
</table>

(a) Data matrix
```

```
<table>
<thead>
<tr>
<th>u_1</th>
<th>0 1 2 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_2</td>
<td>0 2 3 5</td>
</tr>
</tbody>
</table>

(b) Contingency table
```

```
<table>
<thead>
<tr>
<th>u_1</th>
<th>0 1 2 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>u_2</td>
<td>1 1 0 1 1</td>
</tr>
</tbody>
</table>

(c) Sample table
```

Figure 5: (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 \( 3 \times 4 \) 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).

We denote the original contingency table by \( X = \{x_{i,j}\}_{i=0}^I_{j=0} \). Similarly, we denote the sample contingency table by \( S = \{s_{i,j}\}_{i=0}^I_{j=0} \). An example of sample contingency table is shown in Figure 5(c) by taking the first \( D_s = 10 \) columns from the binned data matrix. Of course, we generate the equivalent sample table online by Conditional Random Sampling.

5.1.1 Estimations

Conditioning on \( D_s \), we can estimate the original table in a straightforward fashion:

\[
\hat{x}_{i,j,MF} = \frac{D}{D_s} s_{i,j},
\]

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

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 \((I+1) \times (J+1) - (I+1) - (J+1) + 1 = I \times J \).

Denote the row sums by \( \{x_{i+}\}_{i=0}^I \) and the column sums by \( \{x_{+j}\}_{j=0}^J \).

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. The sample contingency table \( S =

4. 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 \( I + J + 2 \).
\( \{s_{i,j}\}_{i=0}^{I} \}_{j=0}^{J} \) follows a multinomial distribution. Therefore, the maximum likelihood estimator (MLE) of \( x_{i,j} \) under marginal constraints amounts to a convex program:

\[
\text{Minimize} \quad - \sum_{i=0}^{I} \sum_{j=0}^{J} s_{i,j} \log x_{i,j}
\]

such that \( \sum_{j=0}^{J} x_{i,j} = x_{i+}, \quad \sum_{i=0}^{I} x_{i,j} = x_{+j}, \quad x_{i,j} \geq s_{i,j}, \quad (20) \)

which can be solved easily using any standard convex optimization algorithm such as Newton’s method.\(^5\) We denote the estimated table cells as \( \hat{x}_{i,j,MLE} \).

One can also estimate the inner product \( a = u_{1}^{T} u_{2} \) from the estimated contingency table because

\[
a = u_{1}^{T} u_{2} = \sum_{i=1}^{I} \sum_{j=1}^{J} (ij) x_{i,j}. \quad (21) \]

Therefore, we can estimate \( a \) by one of the following:

\[
\hat{a}_{MF,c} = \sum_{i=1}^{I} \sum_{j=1}^{J} (ij) \hat{x}_{i,j,MF} \quad (22) \\
\hat{a}_{MLE,c} = \sum_{i=1}^{I} \sum_{j=1}^{J} (ij) \hat{x}_{i,j,MLE} \quad (23) 
\]

where the subscript “c” indicates that we compute \( a \) from contingency tables. We can also compute many other summary statistics such as the chi-square statistic from the estimated contingency table.

Appendix B presents the derivations and results for evaluating the variances of \( \hat{x}_{i,j,MLE} \) and \( \hat{a}_{MLE,c} \).

5.1.2 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.

The two (quantized) vectors are sampled by Conditional Random Sampling 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. Then we estimate the inner product \( a \) from the estimated contingency table.

Figure 6 compares the empirical variances with the theoretical predicts 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 \( E \left( \frac{D}{D_s} \right) = \max \left( \frac{k_1, f_a}{k_2, k_a} \right) \) (we let \( k_1 = k_2 = k \)). In this case, the marginal histograms help considerably.

---

\(^5\) In the implementation, the linear dependency in the constraints can be removed by discarding the two constraints on \( x_{i+} \) and \( x_{+j} \), and adding a new constraint: \( \sum_{i=0}^{I} \sum_{j=0}^{J} x_{i,j} = D. \)
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.

<table>
<thead>
<tr>
<th>Bin ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>2 ~ 3</td>
</tr>
<tr>
<td>3</td>
<td>4 ~ 9</td>
</tr>
<tr>
<td>4</td>
<td>≥ 10</td>
</tr>
</tbody>
</table>

(a) Exp.#1

<table>
<thead>
<tr>
<th>Bin ID</th>
<th>Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>≥ 2</td>
</tr>
</tbody>
</table>

(b) Exp.#2

(a) Exp. #1

(b) Exp. #2

Figure 6: 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 $\sqrt{\frac{\text{Var}(a)}{a}}$. The two thin dashed lines both labeled “theor.” are theoretical variances, which match the empirical values well especially when sketch sizes $\geq 10 \sim 20$. In this case, marginal histograms help considerably.

5.1.3 The Important Special Case: 0/1 Data

For the boolean (0/1) data, the contingency table has only four cells $(x_{0,0}, x_{0,1}, x_{1,0}, x_{1,1}) = (D - f_1 - f_2 + a, f_2 - a, f_1 - a, a)$ and the inner product $a = u_1^T u_2$ is the same as $x_{1,1}$. The MLE solution of (20), denoted by $\hat{a}_{0/1,MLE}$, is the solution to a cubic equation

$$\frac{s_{1,1}}{a} - \frac{s_{1,0}}{f_1 - a} - \frac{s_{0,1}}{f_2 - a} + \frac{s_{0,0}}{D - f_1 - f_2 + a} = 0,$$

(24)

where $s_{1,1} = \# \{ j : \bar{u}_{1,j} = \bar{u}_{2,j} = 1 \}$, $s_{1,0} = \# \{ j : \bar{u}_{1,j} = 1, \bar{u}_{2,j} = 0 \}$, $s_{0,1} = \# \{ j : \bar{u}_{1,j} = 0, \bar{u}_{2,j} = 1 \}$, $s_{0,0} = \# \{ j : \bar{u}_{1,j} = 0, \bar{u}_{2,j} = 0 \}$, $j = 1, 2, \ldots, D_g$.

The (asymptotic) variance of $\hat{a}_{0/1,MLE}$, as derived in Appendix B, would be

$$\text{Var}(\hat{a}_{0/1,MLE}) = E\left( \frac{D}{D_g} \right) \frac{1}{\frac{1}{a} + \frac{1}{f_1 - a} + \frac{1}{f_2 - a} + \frac{1}{D - f_1 - f_2 + a}}.$$

(25)

The 0/1 case is important in many applications including information retrieval, databases, and market-basket analysis, because the original data are often quantized to be binary (absent/present).
5.2 Real-valued Data

In real-valued data, we do not know the data distribution. A practical solution is to assume some parametric form of the (bivariate) data distribution based on prior knowledge; and then solve an MLE considering various constraints. The simplest bivariate distribution is the bivariate normal. 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.,

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

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

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

\[
\hat{a}_{MLE,N} = \frac{D}{D_s} \hat{a} + D\bar{u}_1 \bar{u}_2,
\]

where \(\hat{a}\) is the solution to a cubic equation:

\[
\bar{a}^3 - \bar{a}^2 (\bar{v}_1^T \bar{v}_2) + \bar{a} (-\tilde{m}_1 \tilde{m}_2 + \tilde{m}_1 \|\bar{v}_2\|^2 + \tilde{m}_2 \|\bar{v}_1\|^2) - \tilde{m}_1 \tilde{m}_2 \bar{v}_1^T \bar{v}_2 = 0,
\]

which can be derived similarly to Lemma 2 of (Li et al., 2006a), the authors’ recent work on random projections taking advantage of the marginal norms. \(\hat{a}_{MLE,N}\) is fairly robust, although sometimes we observe the biases are quite noticeable. In general, this is a good bias-variance trade-off.

5.2.1 Estimating l2 Distance

Once we have estimated the inner product \(a\), we can estimate the \(l_2\) distance by \(m_1 + m_2 - 2\hat{a}_{MLE,N}\), which is usually significantly better than the obvious estimator \(d_{MF}^{(2)}\) in (13), as to be shown empirically. We denote this estimator by \(d_{MLE,N}^{(2)}\).

Alternatively, one can estimate \(a\) by \(\hat{a}_{MF}\) in (12) and \(d_{MF}^{(2)}\) by \(m_1 + m_2 - 2\hat{a}_{MF}\). When the data are only weakly dependent, this estimator may work even better, because \(\hat{a}_{MLE,N}\) has the intrinsic bias as the data are not normal.

In the empirical evaluations in Section 7, we only present the results for \(d_{MLE,N}^{(2)}\).

6. Theoretical Comparisons With Random Projections in 0/1 Data

We show that in boolean (0/1) data, Conditional Random Sampling is (almost) always more accurate than random projections by comparing their variances.

In this case, the marginal norms are the same as the numbers of non-zero elements, i.e., \(m_i = \|u_i\|^2 = \tilde{f}_i\). In (15), \(\text{Var} (\hat{a}_{MF}) \) becomes \(\frac{\max(\tilde{f}_1, \tilde{f}_2)}{k} (a - \bar{a}^2)\), assuming \(E \left( \frac{D}{D_s} \right) = \frac{\max(\tilde{f}_1, \tilde{f}_2)}{k}\) is exact. \(\text{Var} (\hat{a}_{NRP, MF}) = \frac{\tilde{f}_1 + a^2}{k}\) in 0/1 data.
We first consider that both estimators do not utilize the marginal information. Figure 7 plots the ratio of variance $\frac{\text{Var}(\hat{\alpha}_{MF})}{\text{Var}(\hat{\alpha}_{NRP,MF})}$, verifying that our algorithm is (considerably) more accurate:

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

Next, suppose both estimators take advantage of the marginal information. Figure 8 plots $\frac{\text{Var}(\hat{\alpha}_{1/MLE})}{\text{Var}(\hat{\alpha}_{NRP,MLE})}$. In most possible range of the data, this ratio is less than 1. When $u_1$ and $u_2$ are very close (e.g., $a \approx f_2 \approx f_1$), random projections appear more accurate. However, when this does occur, the absolute variances are so small (even zero) that their ratio does not matter.

Figure 7: The variance ratios, $\frac{\text{Var}(\hat{\alpha}_{MF})}{\text{Var}(\hat{\alpha}_{NRP,MF})}$ show that Conditional Random Sampling has smaller variances than random projections, when no marginal information is used. $f_2 = \alpha f_1$ with $\alpha = 0.2, 0.5, 0.8, 1.0$. For each $\alpha$, we plot from $f_1 = 0.05D$ to $f_1 = 0.95D$ spaced at 0.05D.

7. Empirical Evaluations

We evaluate Conditional Random Sampling and compare it with random projections using various datasets as presented in Table 3.

The NSF dataset (Dhillon and Modha, 2001) contains 13297 documents (in 5298 dimensions). We randomly sampled 100 documents (i.e., 4950 pairs).

The NEWSGROUP dataset (Bingham and Mannila, 2001) were used for experimenting with random projections. It contains 2262 documents (in 5000 dimensions). Again, we randomly sampled 100 documents.

The COREL dataset has been used in quite a few SVM image classification papers (e.g., (Chapelle et al., 1999)). We selected one class (out of 14), which contains 80 images in 4096 dimensions.

---

Figure 8: When both estimators utilize the marginal information, the ratios, \( \frac{\text{Var}(\hat{a}_{1/1}^{\text{MLE}})}{\text{Var}(\hat{a}_{NRP,MLR})} \), show that Conditional Random Sampling usually has smaller variances than random projections, except when \( f_1 \approx f_2 \approx a \).

The DEXTER dataset is taken from NIPS 2003 workshop on feature extraction.\(^7\) We take the first 100 data points (out of 300) from the Dexter training dataset.

For the above four datasets, we use the original data values without any other processing.

Finally, we randomly selected 100 words, each in 65536 (document) dimensions, provided by MSN. The original data are extremely heavy-tailed and highly skewed as can be seen from Table 3. We process data by both square root weighting (i.e., replacing any entries with its square root), and logarithmic weighting (i.e., replacing any non-zero entry with \( 1 + \log(\text{original value}) \)). We include results for the original data as well as the weighted data.

<table>
<thead>
<tr>
<th></th>
<th>n</th>
<th>( D )</th>
<th>Sparsity</th>
<th>Kurtosis</th>
<th>Skewness</th>
</tr>
</thead>
<tbody>
<tr>
<td>NSF</td>
<td>100</td>
<td>5298</td>
<td>1.09%</td>
<td>349.8</td>
<td>16.3</td>
</tr>
<tr>
<td>NEWSGROUP</td>
<td>100</td>
<td>5000</td>
<td>1.01%</td>
<td>352.9</td>
<td>16.5</td>
</tr>
<tr>
<td>COREL</td>
<td>80</td>
<td>4096</td>
<td>4.82%</td>
<td>765.9</td>
<td>24.7</td>
</tr>
<tr>
<td>DEXTER</td>
<td>100</td>
<td>20000</td>
<td>0.45%</td>
<td>1729.0</td>
<td>35.8</td>
</tr>
<tr>
<td>MSN (original)</td>
<td>100</td>
<td>65536</td>
<td>3.65%</td>
<td>4161.5</td>
<td>49.6</td>
</tr>
<tr>
<td>MSN (square root)</td>
<td>100</td>
<td>65536</td>
<td>3.65%</td>
<td>175.3</td>
<td>10.7</td>
</tr>
<tr>
<td>MSN (logarithmic)</td>
<td>100</td>
<td>65536</td>
<td>3.65%</td>
<td>111.8</td>
<td>9.5</td>
</tr>
</tbody>
</table>

Table 3: For each dataset, we compute the overall sparsity, sample median kurtosis, and sample median skewness. Our data are highly heavy-tailed (and severely skewed). Recall a normal random variable has zero kurtosis and zero skewness.

We estimate the pairwise inner product, \( l_1 \) distance, and \( l_2 \) distance for every pair, using our Conditional Random Sampling as well as random projections. For each pair, we conduct 50 runs.

\(^7\) http://www.nipsfsc.ecs.soton.ac.uk/datasets/
and average the absolute errors. Since in each dataset, we have \( \frac{n(n-1)}{2} \) pairs, we compare the median errors and the percentage of pairs in which our algorithm outperforms random projections.

The results are presented in Figures 10, 9, 11, 12, 13, 14, and 15. In each figure, the upper four panels are the ratio (Conditional Random Sampling over random projections) of the median (among \( n(n-1)/2 \) pairs) average (over 50 runs) absolute errors. The bottom four panels are the percentage of pairs for which our algorithm has smaller absolute errors than random projections.

In each panel, the dashed curve indicates that we sample each data point with equal sample size \( (k) \). For Conditional Random Sampling, we can adjust the sample size according to the sparsity, reflected by the solid curves. We adjust sample sizes only crudely (not optimizing anything). The data points are divided into 3 (5 for the MSN data) groups according to sparsity. Data in different groups are assigned different sketch sizes. For random projections, we use the "average" sample size. For example, in Figure 9, when the "sample size k" is 15, we assign sketch sizes 10, 15, and 20 to the data points whose sparsities fall in \([0/13],[1/3,2/3],\) and \([2/3,1]\) quantiles, respectively; and we let the sample size of random projections to be 15.

The results agree well with what we would expect:

- **Conditional Random Sampling** works particularly well in estimating inner products, even in extremely heavy-tailed datasets (e.g., Figure 13).

- **Conditional Random Sampling** is usually comparable to (Cauchy) random projections in estimating \( l_1 \) distances.

- Without using marginal information, **Conditional Random Sampling** often performs poorly in estimating \( l_2 \) distances in extremely heavy-tailed data (e.g., Figures 11 and 13).

- Using marginal information (i.e., use the estimator derived in Section 5.2 to estimate inner products, from which we can estimate \( l_2 \) distances), **Conditional Random Sampling** can be improved dramatically and, in our experiments, always outperforms random projections even in extremely heavy-tailed data, in estimating \( l_2 \) distances.

- Adjusting the sketch size according to data sparsity in general improves the overall performance.

Note that we have assumed that at the same sample size (i.e., same \( k \)), the storage cost of random projections is the same as that of sketches. This assumption, of course, is only for convenience, not necessarily true in the implementations. Each sketch sample contains an "ID" and a "Value," while each projection sample contains a projected value. (Note that, in 0/1 data, we only need to store IDs for sketches.) The IDs are usually stored by the increments (gaps). That is, if the data are 99% sparse, then we need on average 7 bits (i.e., \( 2^7 = 128 \)) to store each ID. The data values can be integers or real numbers (e.g., after term weighting). In random projections, the projected data are usually real numbers but could also be integers if we use the sparse random projections suggested in (Achlioptas, 2003), or the very sparse random projections proposed in the authors' recent work (Li et al., 2006c). Another complication is that we often have to store the random projection matrix \( R \) (i.e., \( O(Dk) \) storage) in case new data points come. For **Conditional Random Sampling**, we only need to store the random permutation mapping (i.e., \( O(D) \) storage).

Therefore, to avoid these complications, we assume that at the same \( k \), **Conditional Random Sampling** and random projections cost the same in storage.
8. Conclusion

In many data mining and learning tasks such as association rules, clustering, nearest neighboring searching, and support vector machines, computing the pairwise or multi-way distances (inner products) is the vital step but often too time-consuming or even practically infeasible in high dimensions. Various sampling and sketching techniques have been proposed for generating compact representation of the data and speeding up the computations. Our proposed algorithm, called Conditional Random Sampling (CRS), is also a sketching technique but differs from the existing algorithms quite significantly. Like existing sketching methods, we scan the data matrix and generate sketches, which are simply a fraction of the non-zero entries in our case. Unlike other sketching algorithms, in the estimation stage, we generate conditional random samples online pairwise or group-wise. Once we have the random samples, we can then apply matured statistical techniques. Another feature of Conditional Random Sampling is the flexibility in dealing with sparse data. When we use the same sketch size for all data points, the more infrequent (sparse) data points, which are often more interesting, are effectively sampled at higher sampling rates than those more frequent ones. We can also adjust the sketch size according to data sparsity. To the best of our knowledge, existing sketching algorithms only use one fixed sample (sketch) size for all data points.

Conditional Random Sampling is not bounded to any specific summary statistics. This could be advantageous in some applications, which may want to re-use sketches for approximating different summary statistics, for example, both $l_1$ and $l_2$ distances (or inner products).
Figure 10: NEWSGROUP data. The results are quite similar to those in Figure 9 for the NSF data. *Conditional Random sampling* is overwhelmingly better than random projections in approximating inner products and $l_2$ distances (using margins). *Conditional Random Sampling* is also significantly better in approximating $l_1$ distances. In this case, it is more obvious that adjusting sketch sizes helps.

Figure 11: COREL image data. This dataset is more heavy-tailed than NSF and NEWSGROUP datasets but not as sparse. *Conditional Random Sampling* is still much better than random projections in approximating inner products as well as $l_2$ distances (using margins). We observe that at large sample sizes, using margins (i.e., assuming normality) may cause quite noticeable biases. However, this is good bias-variance trade-off, because without using margins, the errors would be much larger than random projections in approximating $l_2$ distances.

We ought to take advantage of the marginal information, which can be easily obtained with little incremental cost during the sketching stage. The estimation task is only slightly more involving (e.g., solving a cubic equation) but the improvement in terms of the estimation accuracy can be often quite considerable.
Figure 12: DEXTER data. This dataset is significantly more heavy-tailed than NSF, NEWSGROUP, and COREL datasets. Sketches are better than random projections in approximating inner products, $l_1$, and $l_2$ distances (using margins).

Figure 13: MSN data (original). The data are extremely heavy-tailed. Even so, Conditional Random Sampling is still significantly better than random projections in approximating inner products, and are also better in approximating $l_2$ distances using margins.

Much of the work is devoted to comparing Conditional Random Sampling with (normal and Cauchy) random projections in approximating $l_1$ and $l_2$ distances (and inner products). Normal random projections have become popular recently. The authors’ concurrent work has developed nonlinear estimators for Cauchy random projections in recovering $l_1$ distances. We show theoretically that Conditional Random Sampling outperforms random projections in boolean (0/1) data. Using text and image datasets, we show empirically that Conditional Random Sampling considerably outperforms random projections in approximating inner products. Taking advantage of the marginal norms, Conditional Random Sampling also considerably outperforms random projections in approximating $l_2$ distances, even in extremely heavy-tailed datasets.
Figure 14: MSN data (square root weighting). After the square root transformation, Conditional Random Sampling is significantly better than random projections in estimating inner products, $l_1$ distances, and $l_2$ distances (using margins). Without using margins, Conditional Random Sampling is about the same as random projections in approximating $l_2$ distances, especially when the sketch sizes are adjusted.

Figure 15: MSN data (logarithmic weighting). The results are even better, compared to the results in Figure 14. In particular, when estimating $l_2$ distances without using margins, Conditional Random Sampling is strictly better than random projections.

Although not fully spinned in this paper, Conditional Random Sampling is also well suited for approximating multi-way associations and multi-way histograms in high dimensions; hence the algorithm will be useful for applications (e.g., databases, market-basket analysis, association rules) concerning multi-way distances (or associations).
Acknowledgment

We thank Ella Bingham, Inderjit Dhillon, and Matthias Hein for the datasets. Trevor Hastie was partially supported by grant DMS-0505676 from the National Science Foundation, and grant 2R01 CA 72028-07 from the National Institutes of Health.

References


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

Recall that we compute the effective sample size $D_s$ from sketches:

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

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

$$E \left( \frac{D_s}{D} \right) \approx \min \left( \frac{k_1}{f_1 + 1}, \frac{k_2}{f_2 + 1} \right), \quad E \left( \frac{D}{D_s} \right) \approx \max \left( \frac{f_1 + 1}{k_1}, \frac{f_2 + 1}{k_2} \right), \quad (30)$$

where $f_i$ is the number of non-zero elements in data vector $u_i$ and $k_i$ is the sketch size.

Substitute $Z_i = \max(\text{ID}(K_i))$, then $D_s = \min(Z_1, Z_2)$. It is clear that $Z_i$ is the $(k_i)$th order statistic of the set $\text{ID}(K_i) \in \Omega = \{1, 2, ..., D\}$, with the probability mass function (PMF) and moments:

$$P(Z_i = t) = \frac{(t-1)!}{k_i-1!} \frac{D-t}{D} \frac{(D-t)!}{(D-t-k_i)} \frac{(D-t-k_i)!}{(f_i^3)(f_i + 1)^2}, \quad E(Z_i) = \frac{k_i(D + 1)}{f_i + 1} \approx \frac{k_i}{f_i + 1} \cdot D,$n

$$\text{Var}(Z_i) = \frac{(D+1)(D-f_i)k_i(f_i+1-k_i)}{f_i^2(f_i+2)} \approx \frac{D(D-f_i)k_i(f_i-k_i)}{f_i^3} \leq \frac{1}{k_i} \cdot \frac{f_i-k_i}{f_i} \cdot (E(Z_i))^2.$$

$$\Rightarrow \frac{\sqrt{\text{Var}(Z_i)}}{E(Z_i)} \leq \sqrt{\frac{1}{k_i} \cdot \frac{f_i-k_i}{f_i}} \to 0 \quad \text{very quickly}$$

By Jensen’s inequality, we know that (Treat $D + 1$ as $D$)

$$E(D_s) = E(\min(Z_1, Z_2)) \leq \min(E(Z_1), E(Z_2)) = \min \left( \frac{k_1}{f_1 + 1}, \frac{k_2}{f_2 + 1} \right).$$

Without loss of generality, assume $E(Z_1) \leq E(Z_2)$. Then

$$\text{min}(E(Z_1), E(Z_2)) = E(D_s) = E(Z_1) - E(\min(Z_1, Z_2)) = E(\max(E(Z_1) - Z_1, E(Z_1) - Z_2)) \leq E(\max(E(Z_1) - Z_1, E(Z_2) - Z_2)) \leq \sum_{i=1}^{m} E(E(Z_i) - Z_i) \leq \sum_{i=1}^{2} \sqrt{\text{Var}(Z_i)} \leq \sum_{i=1}^{2} \sqrt{\frac{1}{k_i} \cdot \frac{f_i-k_i}{f_i}} \cdot E(Z_i),$$

which is crude but nevertheless shows that our approximation of $E(D_s)$ is asymptotically exact.

Statistical results let us approximate $E \left( \frac{1}{D_s} \right)$ by $\frac{1}{E(D_s)}$, with errors determined by $\text{Var}(D_s)$, which vanishes quickly. Thus, the approximations in (3) are reasonably accurate and asymptotically exact.

8. http://www.ds.unifi.it/VL/VL_EN/urn/urn5.html
Appendix B. Covariance Matrix in Estimating Contingency Tables with Marginal Constraints

Recall that $S = \{s_{i,j}\}_{i=0}^I_{j=0}$ denotes the sample contingency table and $X = \{x_{i,j}\}_{i=0}^I_{j=0}$ denotes the original contingency table to be estimated. We vectorize the tables row-wise, i.e., $Z = \text{vec}(X) = \{z_m\}_{m=1}^{(I+1)(J+1)}$ for the original contingency table and $H = \text{vec}(S) = \{h_m\}_{m=1}^{(I+1)(J+1)}$ for the observed sample contingency table. We will give simple examples for $I = J = 2$ and $I = J = 1$ to help visualize the procedure.

There are in total $I + J + 1$ 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: $Z_1$ and $Z_2$. $Z_1$ corresponds to $X_1 = \{x_{i,j}\}_{i=1}^I_{j=1}$ and $Z_2$ corresponds to the rest of the table. The trick is to represent $Z_2$ in terms of $Z_1$ so that we can apply the multivariate large sample theory for the asymptotic covariance matrix of $Z_1$. It is not hard to show that

$$Z_2 = C_1 - C_2 Z_1,$$

where

\[
C_1 = \begin{bmatrix}
x_{0+} + x_{+0} - D \\
\{x_{i,j}\}_{j=1}^J \\
\{x_{i+}\}_{i=1}^I
\end{bmatrix}, \quad C_2 = \begin{bmatrix}
-1^T_J \\
I_J & I_J & \ldots & I_J \\
1^T_J & 0^T_J & \ldots & 0^T_J \\
0^T_J & 1^T_J & \ldots & 0^T_J \\
0^T_J & 0^T_J & \ldots & 1^T_J
\end{bmatrix},
\]

(32)

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, $Z$ follows a multinomial distribution, with a log likelihood function (let $N = (I + 1)(J + 1)$):

$$Q(Z) \propto \sum_{m=1}^N h_m \log z_m$$

(33)

In order to apply the large sample theory for the covariance matrix, we will need to compute the Hessian $(\nabla^2 Q)$, which is a matrix whose $(i, j)^{th}$ entry is the partial derivative $\frac{\partial^2 Q}{\partial z_i \partial z_j}$, i.e.,

$$\nabla^2 Q = -\begin{bmatrix}
\frac{h_1}{z_1} & 0 & \cdots & 0 \\
0 & \frac{h_2}{z_2} & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & \frac{h_N}{z_N}
\end{bmatrix} = -\text{diag} \left[ \frac{h_1}{z_1^2}, \frac{h_2}{z_2^2}, \ldots, \frac{h_N}{z_N^2} \right].$$

(34)

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

$$Q(Z) = Q_1(Z_1) + Q_2(Z_2).$$

(35)
By the matrix derivative chain rule, the Hessian of $Q$ with respect to $Z_1$ would be

$$
\nabla_1^2 Q = \nabla_1^2 Q_1 + \nabla_1^2 Q_2 = \nabla_1^2 Q_1 + C_2^T \nabla_2^2 Q_2 C_2, \tag{36}
$$

where we use $\nabla_1^2$ and $\nabla_2^2$ to indicate that the Hessians are with respect to $Z_1$ and $Z_2$, respectively. The expected Fisher Information of $Z_1$ is

$$
I(Z_1) = E(-\nabla_1^2 Q) = -E(\nabla_1^2 Q_1) - C_2^T E(\nabla_2^2 Q_2) C_2. \tag{37}
$$

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

$$
E(-\nabla_1^2 Q_1) = \text{diag} \left[ E \left( \frac{h_m}{z_m} \right), z_m \in Z_1 \right] = \frac{D_s}{D_s} \text{diag} \left[ \frac{1}{z_m}, z_m \in Z_1 \right], \tag{38}
$$

$$
E(-\nabla_2^2 Q_2) = \frac{D_s}{D_s} \text{diag} \left[ \frac{1}{z_m}, z_m \in Z_2 \right]. \tag{39}
$$

By the large sample theory, the asymptotic covariance matrix of $X_1$ would be

$$
\text{Cov}(Z_1) = I(Z_1)^{-1} = \frac{D_s}{D_s} \left( \text{diag} \left[ \frac{1}{z_m}, z_m \in Z_1 \right] + C_2^T \text{diag} \left[ \frac{1}{z_m}, z_m \in Z_2 \right] C_2 \right)^{-1}. \tag{40}
$$

Of course, since we generate random samples conditionally (on $D_s$), we need to replace $\frac{D_s}{D_s}$ by $E \left( \frac{D_s}{D_s} \right)$.

The following two examples may help visualize the above formulations.

**B.1 An Example with $I = 2, J = 2$**

$$
Z = [z_1, z_2, z_3, z_4, z_5, z_6, z_7, z_8, z_9]^T = [x_{0,0}, x_{0,1}, x_{0,2}, x_{1,0}, x_{1,1}, x_{1,2}, x_{2,0}, x_{2,1}, x_{2,2}]^T \tag{41}
$$

$$
Z_1 = [z_5, z_6, z_8, z_9]^T = [x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}]^T, \tag{42}
$$

$$
Z_2 = [z_1, z_2, z_3, z_4, z_7]^T = [x_{0,0}, x_{0,1}, x_{0,2}, x_{1,0}, x_{2,0}]^T. \tag{43}
$$

$$
C_1 = \begin{bmatrix}
x_{0,+} + x_{+0} - D \\
x_{+1} \\
x_{+2} \\
x_{1,+} \\
x_{2,+}
\end{bmatrix}, \quad C_2 = \begin{bmatrix}
-1 & -1 & -1 & -1 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 1 & 0 & 0 \\
0 & 0 & 1 & 1
\end{bmatrix}, \tag{44}
$$

$$
\text{Cov}(Z_1) = \frac{D_s}{D_s} \left( \text{diag} \left[ \frac{1}{x_{1,1}}, \frac{1}{x_{1,2}}, \frac{1}{x_{2,1}}, \frac{1}{x_{2,2}} \right] + C_2^T \text{diag} \left[ \frac{1}{x_{0,0}}, \frac{1}{x_{0,1}}, \frac{1}{x_{0,2}}, \frac{1}{x_{1,0}}, \frac{1}{x_{2,0}} \right] C_2 \right)^{-1}. \tag{45}
$$
B.2 An Example with $I = 1, J = 1$ (0/1 Data)

\[
Z = [z_1 \ z_2 \ z_3 \ z_4]^T = [x_{0,0} \ x_{0,1} \ x_{1,0} \ x_{1,1}]^T \\
Z_1 = [z_4]^T = [x_{1,1}]^T, \\
Z_2 = [z_1 \ z_2 \ z_3]^T = [x_{0,0} \ x_{0,1} \ x_{1,0}]^T.
\]

(46) (47) (48)

\[
C_2 = \begin{bmatrix} -1 \\ 1 \\ 1 \end{bmatrix},
\]

(49)

\[
\text{Cov}(Z_1) = \frac{D}{Ds} \left( \text{diag} \left[ \frac{1}{x_{1,1}} \right] + C_2^T \text{diag} \left[ \frac{1}{x_{0,0}}, \frac{1}{x_{0,1}}, \frac{1}{x_{1,0}} \right] C_2 \right)^{-1}
\]

(50)

\[
= \frac{D}{Ds} \frac{1}{\frac{1}{x_{1,1}} + \frac{1}{x_{0,0}} + \frac{1}{x_{0,1}} + \frac{1}{x_{1,0}}}
\]

B.3 The Variance of $\hat{a}_{MLE,c}$

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

\[
\hat{a}_{MLE,c} = \sum_{i=1}^{I} \sum_{j=1}^{J} (ij) \tilde{z}_{i,j,MLE},
\]

whose variance would be

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

where $e$ is a vector:

\[
e^T = \left[ (ij)_{i=1}^{I} \right]_{j=1}^{J} \\
= [1, 2, ..., J, 2, 4, ..., 2J, ..., I, ..., IJ]^T.
\]