Computationally Efficient Estimators for Dimension Reductions Using Stable Random Projections

Ping Li
Department of Statistical Science
Faculty of Computing and Information Science
Cornell University
Ithaca, NY 14850, USA
pingli@cornell.edu

Abstract

The method of ***stable random projections*** *is an efficient tool for computing the l_α distances using low memory, where $0 < \alpha \leq 2$ may be viewed as a tuning parameter. This method boils down to a statistical estimation task and various estimators have been proposed, based on the geometric mean, harmonic mean, and fractional power etc.*

This study proposes the ***optimal quantile*** *estimator, whose main operation is* ***selecting****, which is considerably less expensive than taking fractional power, the main operation in previous estimators. Our experiments report that this estimator is nearly one order of magnitude more computationally efficient than previous estimators. For large-scale tasks in which storing and computing pairwise distances is a serious bottleneck, this estimator should be desirable.*

In addition to its computational advantage, the optimal quantile estimator exhibits nice theoretical properties. It is more accurate than previous estimators when $\alpha > 1$. We derive its theoretical error bound and establish the explicit (i.e., no hidden constants) sample complexity bound.

1 Introduction

The method of *stable random projections*[36, 16, 21, 30], as an efficient tool for computing pairwise distances in massive, high-dimensional, and possibly dynamic data, provides a powerful mechanism to tackle some of the challenges in modern data mining and machine learning. In this paper, we provide an easy-to-implement algorithm for *stable random projections*. Our algorithm is both statistically accurate and computationally efficient.

1.1 Massive High-dimensional Data

We denote a data matrix by $\mathbf{A} \in \mathbb{R}^{n\times D}$, i.e., n data points in D dimensions. Data sets in modern applications exhibit important characteristics which impose tremendous challenges in data mining and machine learning [5]:

- Modern data sets with $n = 10^5$ or even $n = 10^6$ points are not uncommon in supervised learning, e.g., in image/text classification, ranking algorithms for search engines (e.g., [24]), etc. In the unsupervised domain (e.g., Web clustering, ads clickthroughs, word/term associations), n can be even much larger.
- Modern data sets are often of ultra high dimensions (D), sometimes in the order of millions or higher, e.g., image and text. In image analysis, D may be $10^3 \times 10^3 = 10^6$ if using pixels as features, or $D = 256^3 \approx 16$ million if using color histograms as features.
- Modern data sets are sometimes collected in a dynamic fashion, e.g., data streams[32].
- Large-scale data are often heavy-tailed, e.g., image, text, and Internet data.

1.2 Dynamic Streaming Data

"Scaling up for high dimensional data and high speed data streams" has been identified to be among the "ten challenging problems in data mining research"[37]. The method of *stable random projections* is often regarded as the standard algorithm for stream computations, provided that the data are generated from the following *Turnstile* model[32].

The input stream $s_t = (i_t, I_t)$, $i_t \in [1,\ D]$ arriving sequentially describes the underlying signal S_t, meaning

$$S_t[i_t] = S_{t-1}[i_t] + I_t. \tag{1}$$

The increment I_t can be either positive (insertion) or negative (deletion). For example, in an online bookstore, $S_{t-1}[i]$ may represent the number of books that the user i has ordered up to time $t-1$ and I_t is the additional orders (or

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.95

cancels of orders) at the time t. If a user is identified by his/her IP address, then $D = 2^{64}$ is possible.

This study mainly concerns computing pairwise distances. We can view the data matrix $\mathbf{A} \in \mathbb{R}^{n\times D}$ as n data streams, whose entries are subject to updating. In reality, the data may not be stored (even on disks)[32]. Thus, a one-pass algorithm is needed to compute and update distances for training. Learning with dynamic (or incremental) data has become an active topic of research, e.g., [11, 2].

1.3 Pairwise Distances and Kernels

Many mining and learning algorithms require a similarity matrix computed from pairwise distances of the data matrix $\mathbf{A} \in \mathbb{R}^{n\times D}$. Examples include clustering, nearest neighbors, multidimensional scaling, and kernel SVM (support vector machines). The similarity matrix requires $O(n^2)$ storage space and $O(n^2D)$ computing time.

This study focuses on the l_α distance ($0 < \alpha \leq 2$). Consider two vectors u_1, $u_2 \in \mathbb{R}^D$ (e.g., the leading two rows in $\mathbf{A}$), the l_α distance between u_1 and u_2 is

$$d_{(\alpha)} = \sum_{i=1}^{D} |u_{1,i} - u_{2,i}|^\alpha. \tag{2}$$

Note that, strictly speaking, the l_α distance should be defined as $d_{(\alpha)}^{1/\alpha}$. However, since the power operation $(.)^{1/\alpha}$ is the same for all pairs, it often makes no difference whether we use $d_{(\alpha)}^{1/\alpha}$ or just $d_{(\alpha)}$; and hence we focus on $d_{(\alpha)}$.

The radial basis kernel (e.g., for SVM) is constructed from $d_{(\alpha)}$ [7, 35], i.e., for $0 < \alpha \leq 2$,

$$\mathbf{K}(u_1, u_2) = \exp\left(-\gamma \sum_{i=1}^{D} |u_{1,i} - u_{1,i}|^\alpha\right). \tag{3}$$

When $\alpha = 2$, this is the Gaussian radial basis kernel. Here α can be viewed as a *tuning* parameter. For example, in their histogram-based image classification project using SVM, [7] reported that $\alpha = 0$ and $\alpha = 0.5$ achieved good performance. For heavy-tailed data, tuning α has the similar effect as term-weighting the original data, often a critical step in a lot of machine learning applications [19, 34].

For popular kernel SVM solvers including the *Sequential Minimal Optimization (SMO)* algorithm[33], storing and computing kernels is the major bottleneck. Three computational challenges were summarized in [5, page 12]:

- ***Computing kernels is expensive***.
- ***Computing full kernel matrix is wasteful***.
  Efficient SVM solvers often do not need to evaluate all pairwise kernels.
- ***Kernel matrix does not fit in memory.***
  Storing the kernel matrix at the memory cost $O(n^2)$ is challenging when $n > 10^5$, and is currently not realistic for $n > 10^6$, because $O\left(10^{12}\right)$ consumes at least 1000 GBs memory.

A popular strategy in large-scale learning is to evaluate distances **on the fly**[5]. That is, instead of loading the similarity matrix in memory at the cost $O(n^2)$, one can load the original data matrix at the cost $O(nD)$ and recompute pairwise distances on-demand. Apparently this strategy is problematic when D is not too small. For high-dimensional data, either loading the data matrix in memory is unrealistic or computing distances on-demand becomes too expensive.

Those challenges are general issues in distanced-based algorithms, not unique to kernel SVM. The method of *stable random projections* provides a promising scheme to reduce the dimension D to a small k (e.g., $k \leq 100$), facilitating compact data storage and efficient distance computations.

1.4 Stable Random Projections

The basic procedure of *stable random projections* is to multiply $\mathbf{A} \in \mathbb{R}^{n\times D}$ by a random matrix $\mathbf{R} \in \mathbb{R}^{D\times k}$ (k is small), which is generated by sampling each entry r_{ij} i.i.d. from a symmetric stable distribution $S(\alpha, 1)$. The resultant matrix $\mathbf{B} = \mathbf{A} \times \mathbf{R} \in \mathbb{R}^{n\times k}$ is much smaller than $\mathbf{A}$ and hence it may fit in memory.

In general, a stable random variable $x \sim S(\alpha, d)$, where d is the scale parameter, does not have a closed-form density. However, its characteristic function (Fourier transform of the density function) has a closed-form:

$$\mathrm{E}\left(\exp\left(\sqrt{-1}x\theta\right)\right) = \exp\left(-d|\theta|^\alpha\right), \qquad \text{for any } \theta, \tag{4}$$

which does not have a closed-form inverse (i.e., density) except for $\alpha = 2$ (normal) or $\alpha = 1$ (Cauchy). Note that when $\alpha = 2$, d corresponds to "σ^2" (not "σ") in a normal. The fact that stable distributions in general do not have closed-form density makes the estimation task more difficult.

Corresponding to the leading two rows in $\mathbf{A}$, u_1, $u_2 \in \mathbb{R}^D$, the leading two rows in $\mathbf{B}$ are $v_1 = \mathbf{R}^\mathrm{T}u_1$, $v_2 = \mathbf{R}^\mathrm{T}u_2$. The entries of the difference, for $j = 1$ to k,

$$\begin{aligned} x_j = v_{1,j} - v_{2,j} &= \sum_{i=1}^{D} r_{ij}\left(u_{1,i} - u_{2,i}\right) \\ &\sim S\left(\alpha, d_{(\alpha)} = \sum_{i=1}^{D} |u_{1,i} - u_{2,i}|^\alpha\right), \end{aligned}$$

are i.i.d. samples of a stable distribution whose scale parameter is the l_α distance $d_{(\alpha)}$, due to properties of Fourier transforms. For example, a weighted sum of i.i.d. standard normals ($\alpha = 2$) is also normal with the scale parameter (i.e., variance) being the sum of squares of all weights.

After obtaining the stable samples, one can discard the original matrix $\mathbf{A}$ and the remaining task is to estimate $d_{(\alpha)}$.

Sampling from stable distributions is based on the Chambers-Mallows-Stuck method[6]. Recently, [20] suggested a much simpler (but approximate) procedure.

1.5 Summary of Applications

- ***Computing all pairwise l_α distances*** The cost of computing all pairwise distances of $\mathbf{A} \in \mathbb{R}^{n\times D}$, $O(n^2D)$, is significantly reduced to $O(nDk + n^2k)$.

- ***Estimating l_α distances online*** For $n > 10^5$, it is challenging or unrealistic to materialize all pairwise distances in $\mathbf{A}$. In applications such as online learning, databases, search engines, and online recommendation systems, it may be more efficient if we store $\mathbf{B} \in \mathbb{R}^{n\times k}$ in memory and estimate distances *on-demand*.

- ***Learning with (Turnstile) dynamic streaming data*** In reality, the data matrix may be updated over time. In fact, with streaming data arriving at high-rate[16, 3], the "data matrix" may be never stored and hence all operations (such as clustering and classification) must be conducted on the fly. Because the *Turnstile* model (1) is linear and the matrix multiplication $\mathbf{B} = \mathbf{A} \times \mathbf{R}$ for random projection is also linear, we can conduct the $\mathbf{A} \times \mathbf{R}$ incrementally, assuming the data in $\mathbf{A}$ are updated according to the *Turnstile* model.

- ***Estimating entropy*** There is a recent trend in entropy computations using *stable random projections* and the αth frequency moments with α close to 1 [38, 15, 14, 22, 23]. We will not delve into this new topic.

2 The Statistical Estimation Problem

Recall that the method of *stable random projections* boils down to estimating the scale parameter $d_{(\alpha)}$ from k i.i.d. samples $x_j \sim S(\alpha, d_{(\alpha)})$, $j = 1$ to k. We consider a good estimator $\hat{d}_{(\alpha)}$ should have the following properties:

- (Asymptotically) unbiased and small variance.
- Computationally efficient.
- Exponential decrease of error (tail) probabilities.

The *arithmetic mean* estimator $\frac{1}{k}\sum_{j=1}^k |x_j|^2$ is good for $\alpha = 2$. When $\alpha < 2$, the task is less straightforward because (1) no explicit density of x_j exists unless $\alpha = 1$ or $0+$; and (2) $\mathrm{E}(|x_j|^t) < \infty$ only when $-1 < t < \alpha$.

2.1 Several Previous Estimators

[21] proposed the *geometric mean* estimator

$$\hat{d}_{(\alpha),gm} = \frac{\prod_{j=1}^k |x_j|^{\alpha/k}}{\left[\frac{2}{\pi}\Gamma\left(\frac{\alpha}{k}\right)\Gamma\left(1-\frac{1}{k}\right)\sin\left(\frac{\pi}{2}\frac{\alpha}{k}\right)\right]^k},$$

where $\Gamma(.)$ is the Gamma function, and the *harmonic mean* estimator

$$\hat{d}_{(\alpha),hm} = \frac{-\frac{2}{\pi}\Gamma(-\alpha)\sin\left(\frac{\pi}{2}\alpha\right)}{\sum_{j=1}^k |x_j|^{-\alpha}}\left(k + \frac{\pi\Gamma(-2\alpha)\sin\left(\pi\alpha\right)}{\left[\Gamma(-\alpha)\sin\left(\frac{\pi}{2}\alpha\right)\right]^2} + 1\right).$$

More recently, [28] proposed the *fractional power* estimator

$$\hat{d}_{(\alpha),fp} = \left(\frac{1}{k}\frac{\sum_{j=1}^k |x_j|^{\lambda^*\alpha}}{\frac{2}{\pi}\Gamma(1-\lambda^*)\Gamma(\lambda^*\alpha)\sin\left(\frac{\pi}{2}\lambda^*\alpha\right)}\right)^{1/\lambda^*} \times$$

$$\left(1 - \frac{1}{k}\frac{1}{2\lambda^*}\left(\frac{1}{\lambda^*}-1\right)\left(\frac{\frac{2}{\pi}\Gamma(1-2\lambda^*)\Gamma(2\lambda^*\alpha)\sin\left(\pi\lambda^*\alpha\right)}{\left[\frac{2}{\pi}\Gamma(1-\lambda^*)\Gamma(\lambda^*\alpha)\sin\left(\frac{\pi}{2}\lambda^*\alpha\right)\right]^2}-1\right)\right),$$

where

$$\lambda^* = \underset{-\frac{1}{2\alpha}\lambda<\frac{1}{2}}{\text{argmin}} \; \frac{1}{\lambda^2}\left(\frac{\frac{2}{\pi}\Gamma(1-2\lambda)\Gamma(2\lambda\alpha)\sin\left(\pi\lambda\alpha\right)}{\left[\frac{2}{\pi}\Gamma(1-\lambda)\Gamma(\lambda\alpha)\sin\left(\frac{\pi}{2}\lambda\alpha\right)\right]^2}-1\right).$$

All three estimators are unbiased or asymptotically (as $k \to \infty$) unbiased. Figure 1 compares their asymptotic variances in terms of the Cramér-Rao efficiency, which is the ratio of the smallest possible asymptotic variance over the asymptotic variance of the estimator, as $k \to \infty$.

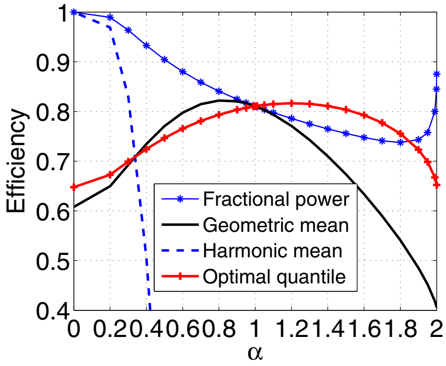


Figure 1. The Cramér-Rao efficiencies (the higher the better, max = 1.00) of various estimators, including the *optimal quantile* estimator proposed in this study.

The *geometric mean* estimator $\hat{d}_{(\alpha),gm}$ exhibits exponential tail bounds, i.e., the errors decrease exponentially fast:

$$\mathbf{Pr}\left(|\hat{d}_{(\alpha),gm} - d_{(\alpha)}| \geq \epsilon d_{(\alpha)}\right) \leq 2\exp\left(-k\frac{\epsilon^2}{G_{gm}}\right).$$

where the constant G_{gm} was explicitly provided in [21].

The *harmonic mean* estimator, $\hat{d}_{(\alpha),hm}$, works well for small α, and has exponential tail bounds when $\alpha = 0+$.

The *fractional power* estimator, $\hat{d}_{(\alpha),fp}$, has smaller asymptotic variance than both the *geometric mean* and *harmonic mean* estimators. However, it does not have exponential tail bounds, due to the restriction $-1 < \lambda^*\alpha < \alpha$ in its definition. As shown in [28], it only has finite moments slightly higher than the $2nd$ order, when $\alpha \to 2$ (because $\lambda^* \to 0.5$), meaning that large errors may have a good chance to occur. We will demonstrate this by simulations.

2.2 The Issue of Computational Efficiency

All three estimators, $\hat{d}_{(\alpha),gm}$, $\hat{d}_{(\alpha),hm}$ and $\hat{d}_{(\alpha),fp}$, require evaluating fractional powers, e.g., $|x_j|^{\alpha/k}$. This operation is expensive, especially if we need to conduct this tens of billions of times (e.g., $n^2 = 10^{10}$). For example, [7, 17] reported that, although the radial basis kernel (3) with $\alpha = 0.5$ achieved good performance, it was not preferred because evaluating the square root was expensive.

2.3 Our Proposed Estimator

We propose the *optimal quantile* estimator, by selecting the $(q^* \times k)$th smallest $|x_j|$ (i.e., $0 \leq q^* \leq 1$):

$$\hat{d}_{(\alpha),oq} \propto \left(q^*\text{-quantile}\{|x_j|, j = 1, 2, ..., k\}\right)^{\alpha}, \quad (5)$$

where $q^* = q^*(\alpha)$ is chosen to minimize the asymptotic variance. This estimator is computationally attractive because **selecting** should be much less expensive than evaluating fractional powers. If we are interested in $d_{(\alpha)}^{1/\alpha}$ instead, then we do not even need to evaluate any fractional powers.

As mentioned previously, in many cases using either $d_{(\alpha)}$ or $d_{(\alpha)}^{1/\alpha}$ makes no difference. The radial basis kernel (3) requires $d_{(\alpha)}$ and hence this study focuses on $d_{(\alpha)}$. On the other hand, if applications only need $d_{(\alpha)}^{1/\alpha}$, we can simply use (5) without the αth power.

In addition to the computational advantages, this estimator also has good theoretical properties, in terms of both the variances and tail probabilities:

1. Figure 1 illustrates that, compared with the *geometric mean* estimator, the asymptotic variance of the *optimal quantile* estimator is about the same when $\alpha < 1$, and is considerably smaller when $\alpha > 1$. Compared with the *fractional power* estimator, it has smaller asymptotic variance when $1 < \alpha \leq 1.8$. In fact, as will be shown by simulations, when the sample size k is not too large, the *optimal quantile* estimator actually has considerably smaller mean square errors than the *fractional power* estimator, for all $1 < \alpha \leq 2$.
2. The *optimal quantile* estimator exhibits tail bounds in exponential form. This theoretical result is practically important, for selecting the sample size k. While it is well-known that the generalization bounds in machine learning theory are often loose, our bounds are tight and practical because the distribution is specified.

The next section will be devoted to analyzing the *optimal quantile* estimator.

3 The Optimal Quantile Estimator

Recall the goal is to estimate $d_{(\alpha)}$ from $\{x_j\}_{j=1}^k$, where $x_j \sim S(\alpha, d_{(\alpha)})$, i.i.d. Since the distribution belongs to the scale family, one can estimate the scale parameter $d_{(\alpha)}$ from quantiles. Due to symmetry, it is natural to consider the absolute values:

$$\hat{d}_{(\alpha),q} = \left(\frac{q\text{-Quantile}\{|x_j|, j = 1, 2, ..., k\}}{q\text{-Quantile}\{|S(\alpha, 1)|\}}\right)^{\alpha}, \quad (6)$$

which can be understood by the fact that if $x \sim S(\alpha, 1)$, then $d^{1/\alpha}x \sim S(\alpha, d)$, or more obviously, if $x \sim N(0, 1)$, then $\left(\sigma^2\right)^{1/2} x \sim N\left(0, \sigma^2\right)$. By properties of order statistics[10], $\hat{d}_{(\alpha),q}$ provides an asymptotically unbiased estimator.

Lemma 1 provides the asymptotic variance of $\hat{d}_{(\alpha),q}$.

Lemma 1 *Denote $f_X\left(x; \alpha, d_{(\alpha)}\right)$ and $F_X\left(x; \alpha, d_{(\alpha)}\right)$ the probability density function and the cumulative density function of $X \sim S(\alpha, d_{(\alpha)})$, respectively.*

The asymptotic variance of $\hat{d}_{(\alpha),q}$ defined in (6) is

$$\mathit{Var}\left(\hat{d}_{(\alpha),q}\right) = \frac{1}{k}\frac{(q-q^2)\alpha^2/4}{f_X^2\left(W; \alpha, 1\right)W^2} d_{(\alpha)}^2 + O\left(\frac{1}{k^2}\right) \quad (7)$$

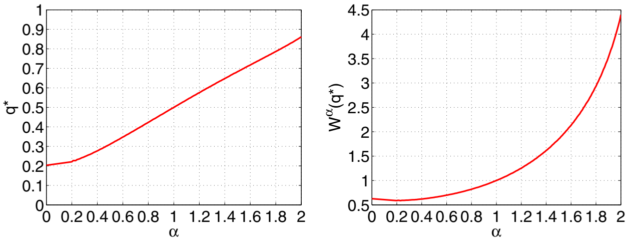
where $W = F_X^{-1}\left((q+1)/2; \alpha, 1\right) = q$-Quantile$\{|S(\alpha, 1)|\}$.

Proof: *See Appendix A.* □.

3.1 The Optimal Quantile $q^*(\alpha)$

We choose $q = q^*(\alpha)$ so that the asymptotic variance (7) is minimized, i.e.,

$$q^*(\alpha) = \underset{q}{\text{argmin}}\; g(q; \alpha), \qquad g(q; \alpha) = \frac{q - q^2}{f_X^2\left(W; \alpha, 1\right)W^2}. \quad (8)$$



(a) q^*

(b) $W^\alpha(q^*)$

Figure 2. (a) The optimal values for $q^*(\alpha)$, which minimizes asymptotic variance of $\hat{d}_{(\alpha),q}$, i.e., the solution to (8). (b) The constant $W^\alpha(q^*) = \{q^*\text{-quantile}\{|S(\alpha, 1)|\}\}^\alpha$.

The convexity of $g(q; \alpha)$ ensures a unique minimum. Graphically, $g(q; \alpha)$ is a convex function of q. An algebraic proof, however, is difficult. Nevertheless, we can obtain analytical solutions when $\alpha = 1$ and $\alpha = 0+$.

Lemma 2 *When $\alpha = 1$ or $\alpha = 0+$, the function $g(q; \alpha)$ defined in (8) is a convex function of q. When $\alpha = 1$, the optimal $q^*(1) = 0.5$. When $\alpha = 0+$, $q^*(0+) = 0.203$ is the solution to $-\log q^* + 2q^* - 2 = 0$.*

Proof: *See Appendix B.* □.

It is also easy to show that when $\alpha = 2$, $q^*(2) = 0.862$.

We denote the *optimal quantile* estimator by $\hat{d}_{(\alpha),oq}$, which is same as $\hat{d}_{(\alpha),q^*}$. For general α, we resort to numerical solutions, as presented in Figure 2.

3.2 Bias Correction

Although $\hat{d}_{(\alpha),oq}$ (i.e., $\hat{d}_{(\alpha),q^*}$) is asymptotically (as $k \rightarrow \infty$) unbiased, it is seriously biased for small k. Thus, it is practically important to remove the bias. The unbiased version of the *optimal quantile* estimator is

$$\hat{d}_{(\alpha),oq,c} = \hat{d}_{(\alpha),oq}/B_{\alpha,k}, \tag{9}$$

where $B_{\alpha,k}$ is the expectation of $\hat{d}_{(\alpha),oq}$ at $d_{(\alpha)} = 1$. For $\alpha = 1, 0+$, or 2, we can evaluate the expectations (i.e., integrals) analytically or by numerical integrations. For general α, because the probability density is not available, the task is difficult and prone to numerical instability. On the other hand, since the Monte-Carlo simulation is a popular alternative for evaluating difficult integrals, a practical solution is to simulate the expectations, as presented in Figure 3.

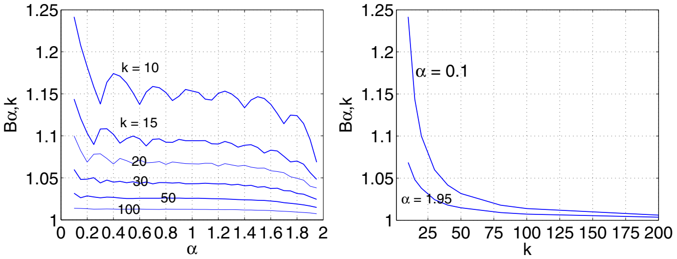


Figure 3. The bias correction factor $B_{\alpha,k}$ in (9), obtained from 10^8 simulations for every combination of α (spaced at 0.05) and k. Note that $B_{\alpha,k} = \text{E}\left(\hat{d}_{(\alpha),oq}; d_{(\alpha)} = 1\right)$.

Figure 3 illustrates that $B_{\alpha,k} > 1$, meaning that this correction also reduces variance while removing bias (because Var(x/c) = Var$(x)/c^2$). For example, when $\alpha = 0.1$ and $k = 10$, $B_{\alpha,k} \approx 1.24$, which is significant, because $1.24^2 = 1.54$ implies a 54% difference in terms of variance, and even more considerable in terms of the mean square errors MSE = variance + bias^2.

$B_{\alpha,k}$ can be tabulated for small k, and absorbed into other coefficients, i.e., it does not increase the computational cost. We fix $B_{\alpha,k}$ as reported in Figure 3. The simulations in Section 4 directly used those fixed $B_{\alpha,k}$ values.

3.3 Computational Efficiency

Figure 4 compares the computational costs of the *geometric mean*, the *fractional power*, and the *optimal quantile* estimators. The *harmonic mean* estimator was not included as it costs very similarly to the *fractional power* estimator.

We used the build-in function *pow* in **gcc** for evaluating the fractional powers. We implemented a "quick select" algorithm, which is similar to quick sort and requires on average linear time. For simplicity, our implementation used recursions and the middle element as pivot. Also, to ensure fairness, for all estimators, coefficients which are functions of α and/or k were pre-computed.

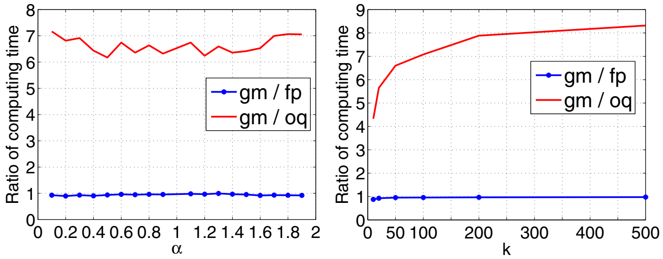


Figure 4. Relative computational cost ($\hat{d}_{(\alpha),gm}$ over $\hat{d}_{(\alpha),oq,c}$ and $\hat{d}_{(\alpha),gm}$ over $\hat{d}_{(\alpha),fp}$), from 10^6 simulations at each combination of α and k. The left panel averages over all k and the right panel averages over all α. Note that the cost of $\hat{d}_{(\alpha),oq,c}$ includes evaluating the αth fractional power once.

Normalized by the computing time of $\hat{d}_{(\alpha),gm}$, we observe that relative computational efficiency does not strongly depend on α. We do observe that the ratio of computing time of $\hat{d}_{(\alpha),gm}$ over that of $\hat{d}_{(\alpha),oq,c}$ increases consistently with increasing k. This is because, in the definition of $\hat{d}_{(\alpha),oq}$ (and hence also $\hat{d}_{(\alpha),oq,c}$), it is required to evaluate the fractional power once, which contributes to the total computing time more significantly at smaller k.

Figure 4 illustrates that, (A) the *geometric mean* estimator and the *fractional power* estimator are similar in terms of computational efficiency; (B) the *optimal quantile* estimator is nearly one order of magnitude more computationally efficient than the *geometric mean* and *fractional power* estimators. Because we implemented a naíve "quick select" using recursions and simple pivoting, the actual improvement may be more significant. Also, if applications require only $d_{(\alpha)}^{1/\alpha}$, then no fractional power operations are needed and hence the improvement will be even more considerable.

3.4 Error (Tail) Bounds

Error (tail) bounds are crucial for determining k; the variance in general is not sufficient for this purpose. If an estimator of d, say $\hat{d}$, is normally distributed, $\hat{d} \sim N\left(d, \frac{1}{k}V\right)$, then the variance factor V suffices for choosing k because its error (tail) probability $\mathbf{Pr}\left(|\hat{d} - d| \geq \epsilon d\right) \leq 2\exp\left(-k\frac{\epsilon^2}{2V}\right)$ is determined by V. Usually, a reasonable estimator will be asymptotically normal, for small enough

ϵ and large enough k. For a finite k and a fixed ϵ, however, the normal approximation may be (very) poor.

Lemma 3 provides the error (tail) probability bounds of $\hat{d}_{(\alpha),q}$ for any q, not just for the optimal quantile q^*.

Lemma 3 *Denote $X \sim S(\alpha, d_{(\alpha)})$ and its probability density function by $f_X(x;\alpha,d_{(\alpha)})$ and cumulative function by $F_X(x;\alpha,d_{(\alpha)})$. Given $x_j \sim S(\alpha, d_{(\alpha)})$, i.i.d., $j = 1$ to k. Using $\hat{d}_{(\alpha),q}$ in (6), then*

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \geq (1+\epsilon)d_{(\alpha)}\right) \leq \exp\left(-k\frac{\epsilon^2}{G_{R,q}}\right), \epsilon > 0, \quad (10)$$

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \leq (1-\epsilon)d_{(\alpha)}\right) \leq \exp\left(-k\frac{\epsilon^2}{G_{L,q}}\right), 0 < \epsilon < 1, \quad (11)$$

$$\frac{\epsilon^2}{G_{R,q}} = -(1-q)\log\left(2-2F_R\right) - q\log(2F_R-1) + (1-q)\log(1-q) + q\log q, \quad (12)$$

$$\frac{\epsilon^2}{G_{L,q}} = -(1-q)\log\left(2-2F_L\right) - q\log(2F_L-1) + (1-q)\log(1-q) + q\log q, \quad (13)$$

$$W = F_X^{-1}((q+1)/2;\alpha,1) = q\text{-quantile}\{|S(\alpha,1)|\},$$

$$F_R = F_X\left((1+\epsilon)^{1/\alpha}W;\alpha,1\right), \quad F_L = F_X\left((1-\epsilon)^{1/\alpha}W;\alpha,1\right).$$

As $\epsilon \rightarrow 0+$

$$\lim_{\epsilon\rightarrow 0+} G_{R,q} = \lim_{\epsilon\rightarrow 0+} G_{L,q} = \frac{q(1-q)\alpha^2/2}{f_X^2\left(W;\alpha,1\right)W^2}. \quad (14)$$

Proof: *See Appendix C.* □

The limit in (14) as $\epsilon \rightarrow 0$ is precisely twice the asymptotic variance factor of $\hat{d}_{(\alpha),q}$ in (7), consistent with the normality approximation mentioned previously. This explains why we express the constants as ϵ^2/G. (14) also indicates that the tail bounds achieve the "optimal rate" for this estimator, in the language of large deviation theory.

The Bonferroni bound can determine the sample size k

$$\mathbf{Pr}\left(|\hat{d}_{(\alpha),q} - d_{(\alpha)}| \geq \epsilon d_{(\alpha)}\right) \leq 2\exp\left(-k\frac{\epsilon^2}{G}\right) \leq \delta/(n^2/2)$$

$$\Longrightarrow k \geq \frac{G}{\epsilon^2}\left(2\log n - \log\delta\right).$$

Lemma 4 *Using $\hat{d}_{(\alpha),q}$ with $k \geq \frac{G}{\epsilon^2}(2\log n - \log\delta)$, any pairwise l_α distance among n points can be approximated within a $1 \pm \epsilon$ factor with probability $\geq 1-\delta$. It suffices to let $G = \max\{G_{R,q}, G_{L,q}\}$, where $G_{R,q}$, $G_{L,q}$ are given in Lemma 3.*

The Bonferroni bound can be too conservative. It is often reasonable to replace $\delta/(n^2/2)$ by δ/T, meaning that except for a $1/T$ fraction of pairs, any distance can be approximated within a $1 \pm \epsilon$ factor with probability $1-\delta$.

Figure 5 plots the error bound constants for $\epsilon < 1$, for both the recommended *optimal quantile* estimator $\hat{d}_{(\alpha),oq}$ and the baseline *sample median* estimator $\hat{d}_{(\alpha),q=0.5}$. Although we choose $\hat{d}_{(\alpha),oq}$ based on the asymptotic variance, it turns out $\hat{d}_{(\alpha),oq}$ also exhibits (much) better tail behaviors (i.e., smaller constants) than $\hat{d}_{(\alpha),q=0.5}$, at least for $\epsilon < 1$.

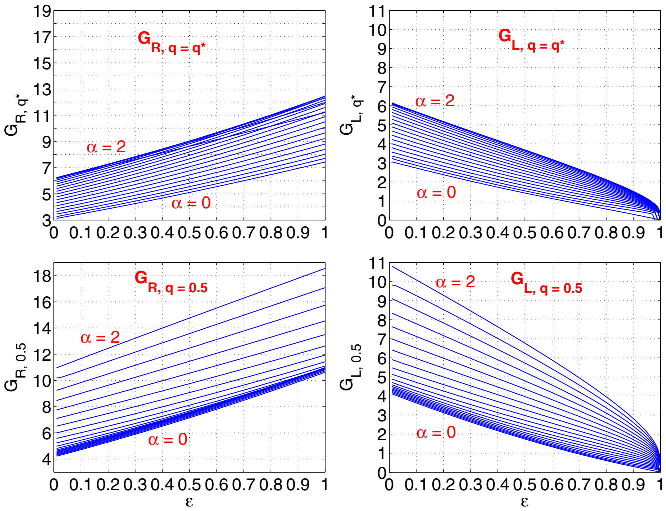


Figure 5. Tail bound constants for quantile estimators; the lower the better. Upper panels: optimal quantile estimators $\hat{d}_{(\alpha),q^*}$. Lower panels: median estimators $\hat{d}_{(\alpha),q=0.5}$.

Consider $k = \frac{G}{\epsilon^2}(\log 2T - \log\delta)$ (recall we suggest replacing $n^2/2$ by T), with $\delta = 0.05$, $\epsilon = 0.5$, and $T = 10$. Because $G_{R,q^*} \approx 5 \sim 9$ around $\epsilon = 0.5$, we obtain $k \approx 120 \sim 215$.

It is possible $k = 120 \sim 215$ might be still conservative, for three reasons: (A) the tail bounds, although "sharp," are still upper bounds; (B) using $G = \max\{G_{R,q^*}, G_{L,q^*}\}$ is conservative because G_{L,q^*} is usually much smaller than G_{R,q^*}; (C) this type of tail bounds is based on relative error, which may be stringent for small (≈ 0) distances.

In fact, some earlier studies on *normal random projections* (i.e., $\alpha = 2$) [4, 13] empirically demonstrated that $k \geq 50$ appeared sufficient.

4 Experiments

One advantage of *stable random projections* is that we know the (manually generated) distributions and the only source of errors is from random number generations. After stable projections, the projected data follow exactly the stable distribution, regardless of the original real data distribution. Therefore, for the purpose of evaluating the proposed estimator, it suffices to simply rely on simulations.

Without loss of generality, we simulate samples from $S(\alpha, 1)$ and estimate the scale parameter (i.e., 1) from the samples. Repeating the procedure 10^7 times, we can evaluate the mean square errors (MSE) and tail probabilities.

4.1 Mean Square Errors (MSE)

As illustrated in Figure 6, in terms of the MSE, the *optimal quantile* estimator $\hat{d}_{(\alpha),oq,c}$ outperforms both the *geometric mean* and *fractional power* estimators when $\alpha > 1$ and $k \geq 20$. The *fractional power* estimator does not appear to be very suitable for $\alpha > 1$, especially for α close to 2, even when k is not too small (e.g., $k = 50$). For $\alpha < 1$, however, the *fractional power* estimator has good performance in terms of MSE, even for small k.

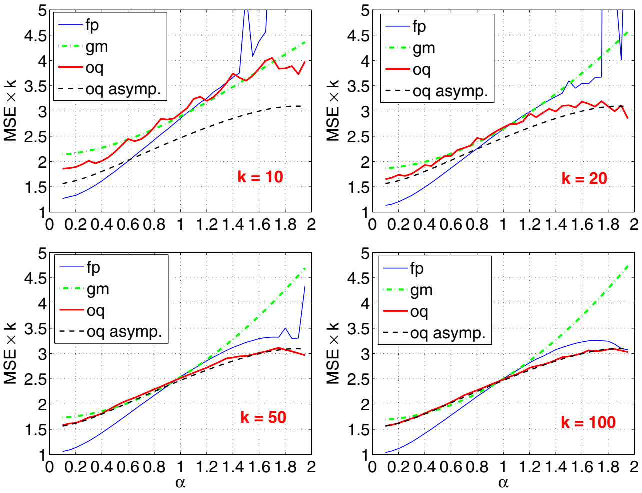


Figure 6. Empirical mean square errors (MSE, the lower the better), from 10^7 simulations at every combination of α and k. The values are multiplied by k so that four plots can be at about the same scale. The MSE for the *geometric mean* (gm) estimator is computed exactly since its closed-form expression exists. The lower dashed curves are the asymptotic variances of the *optimal quantile* (oq) estimator.

4.2 Error(Tail) Probabilities

Figure 7 presents the simulated right tail probabilities, $\mathbf{Pr}\left(\hat{d}_{(\alpha)} \geq (1+\epsilon)d_{(\alpha)}\right)$, illustrating that, when $\alpha > 1$, the *optimal quantile* estimator consistently outperforms the *fractional power* and the *geometric mean* estimators. In fact, when $\alpha > 1$, the *fractional power* estimator exhibits very bad tail behaviors. However, for $\alpha < 1$, the *fractional power* estimator demonstrates good performance at least in the simulated probability range.

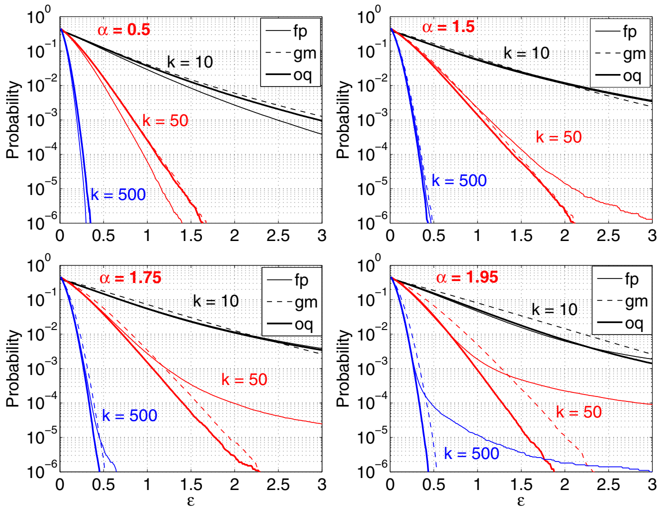


Figure 7. The right tail probabilities (the lower the better), from 10^7 simulations at each combination of α and k.

5 The Related Work

5.1 Normal Random Projections

For $\alpha = 2$, there have been many studies of *normal random projections* in machine learning, for dimension reduction in the l_2 norm, e.g., [36, 13], highlighted by the Johnson-Lindenstrauss (JL) Lemma [18], which says $k = O\left(\log n/\epsilon^2\right)$ suffices when using normal (or normal-like, e.g., [1, 29]) projection methods.

This paper studies $0 < \alpha \leq 2$, not just $\alpha = 2$. The tail bounds and sample complexity bounds are provided for all $0 < \alpha \leq 2$. We should mention that our bounds at $\alpha = 2$ do not precisely recover the (optimal) bounds for *normal random projections*, because the *optimal quantile* estimator is not statistically optimal at $\alpha = 2$, as shown in Figure 1.

5.2 Previous Quantile-Based Estimators

Quantile-based estimators for stable distributions were studied in statistics literature[12, 31]. [12] focused on $1 \leq \alpha \leq 2$ and recommended using $q = 0.44$ quantiles (mainly for the sake of smaller bias). [31] focused on $0.6 \leq \alpha \leq 2$ and recommended $q = 0.5$ quantiles.

This study considers all $0 < \alpha \leq 2$ and recommends q based on the minimum asymptotic variance. Because the bias can be easily removed (at least in the practical sense), it appears not necessary to use other quantiles only for the sake of smaller bias. Tail bounds, which are useful for choosing q and k, were not provided in [12, 31].

For $\alpha = 1$, the classical work[16] suggested the *median* (i.e., $q = 0.5$ quantile) estimator for $\alpha = 1$ and argued that the sample complexity bound should be $O\left(1/\epsilon^2\right)$ ($n =$

1 in their study), although their bound did not specify the constant and required an "ϵ small enough" argument.

For $\alpha = 1$, [9] used a linear combination of quantiles with carefully chosen coefficients to obtain an asymptotically optimal estimator of the scale parameter. While it is possible to extend their result to general $0 < \alpha < 2$ (requiring some non-trivial work), whether it will be practically better than the *optimal quantile* estimator is unclear because the extreme quantiles severely affect the performance. Discarding (truncating) extreme quantiles reduces the sample size. Also, exponential tail bounds of the linear combination of quantiles for stable distributions may not exist or may not be feasible to derive. In addition, the *optimal quantile* estimator is computationally more efficient.

5.3 Compressed Counting (CC)

This paper and all previous work on stable random projections used ***symmetric*** *stable distributions*, i.e., the distribution specified by the Fourier transform (4). Recently, [23] proposed ***Compressed Counting (CC)***, which dramatically improves the performance of *(symmetric) stable random projections*, especially as $\alpha \rightarrow 1$. One application of CC is for estimating entropy of data stream[15, 14, 22].

CC used ***skewed*** *stable random projections* and is only applicable to the *strict Turnstile* model, which restricts $S_t[i] \geq 0$ in the *Turnstile* model (1) (but the increment I_t can be either negative or positive). In most data stream computations, the *strict Turnstile* model suffices. For example, one can only cancel an order if he/she did place the order.

A limitation of CC is that it is not applicable to estimating pairwise distances (e.g., comparing two streams).

5.4 Conditional Random Sampling (CRS)

One competitor of *stable random projections* is the technique called *Conditional Random Sampling (CRS)*[25, 26, 27]. CRS only works well in sparse data such as text and histogram-based image data. A distinct feature of CRS is **One-Sketch-for-All**, meaning that the same set of *sketches* (samples) can be utilized for approximating many different types of distances including the l_α distance and χ^2 distance.

6 Conclusion

Many data mining and machine learning algorithms operate on the training data only through pairwise distances. Computing, storing, updating and retrieving the "matrix" of pairwise distances is challenging in applications involving massive, high-dimensional, and possibly streaming, data. For example, the pairwise distance matrix can not fit in memory when the number of observations exceeds 10^6.

The method of *stable random projections* provides an efficient mechanism for computing pairwise distances using low memory, by transforming the original high-dimensional data into *sketches*, i.e., a small number of samples from α-stable distributions, which are much easier to store and retrieve. This method provides a uniform scheme for computing the l_α pairwise distances for all $0 < \alpha \leq 2$.

To recover the original distances, we face an estimation task. Compared with previous estimators based on the *geometric mean*, *harmonic mean*, or *fractional power*, the proposed *optimal quantile* estimator exhibits two advantages. Firstly, the *optimal quantile* estimator is nearly one order of magnitude more efficient (e.g., reducing the training time from one week to one day). Secondly, the *optimal quantile* estimator is considerably more accurate when $\alpha > 1$, in terms of both the variances and error (tail) probabilities.

One theoretical contribution is the explicit tail bounds for general quantile estimators and consequently the sample complexity bound $k = O\left(\log n/\epsilon^2\right)$, which may guide practitioners in choosing k, the number of projections.

Acknowledgement

The author wishes to thank the anonymous reviewers for their constructive comments. The author thanks John Linderman from AT&T Labs Research for some interesting discussions about sorting algorithms. This work is supported by NSF grant DMS-0808864 and a gift from Google.

A Proof of Lemma 1

Denote $f_X\left(x;\alpha,d_{(\alpha)}\right)$ and $F_X\left(x;\alpha,d_{(\alpha)}\right)$ the probability density function and the cumulative density function of $X \sim S(\alpha, d_{(\alpha)})$, respectively. Similarly we use $f_Z\left(z;\alpha,d_{(\alpha)}\right)$ and $F_Z\left(z;\alpha,d_{(\alpha)}\right)$ for $Z = |X|$. Due to symmetry, the following relations hold

$$f_Z\left(z;\alpha,d_{(\alpha)}\right) = 2f_X\left(z;\alpha,d_{(\alpha)}\right) = 2/d_{(\alpha)}^{1/\alpha} f_X\left(z/d_{(\alpha)}^{1/\alpha};\alpha,1\right),$$

$$F_Z\left(z;\alpha,d_{(\alpha)}\right) = 2F_X\left(z;\alpha,d_{(\alpha)}\right) - 1 = 2F_X\left(z/d_{(\alpha)}^{1/\alpha};\alpha,1\right) - 1,$$

$$\begin{aligned}F_Z^{-1}\left(q;\alpha,d_{(\alpha)}\right) &= F_X^{-1}\left((q+1)/2;\alpha,d_{(\alpha)}\right)\\ &= d_{(\alpha)}^{1/\alpha} F_X^{-1}\left((q+1)/2;\alpha,1\right).\end{aligned}$$

Let $W = q\text{-Quantile}\{|S(\alpha,1)|\} = F_X^{-1}\left((q+1)/2;\alpha,1\right)$ and $W_d = F_Z^{-1}\left(q;\alpha,d_{(\alpha)}\right) = d_{(\alpha)}^{1/\alpha}W$. Then, following known statistical results, e.g., [10, Theorem 9.2], the asymptotic variance of $\hat{d}_{\alpha,q}^{1/\alpha}$ should be

$$\begin{aligned}\text{Var}\left(\hat{d}_{\alpha,q}^{1/\alpha}\right) =&\frac{1}{k}\frac{q-q^2}{f_Z^2\left(W_d;\alpha,d_{(\alpha)}\right)W^2} + O\left(\frac{1}{k^2}\right)\\ =&\frac{1}{k}\frac{q-q^2}{d_{(\alpha)}^{-2/\alpha}f_Z^2\left(W;\alpha,1\right)W^2} + O\left(\frac{1}{k^2}\right)\\ =&\frac{1}{k}\frac{q-q^2}{4d_{(\alpha)}^{-2/\alpha}f_X^2\left(W;\alpha,1\right)W^2} + O\left(\frac{1}{k^2}\right).\end{aligned}$$

By "delta method," i.e., $\text{Var}\left(h(x)\right) \approx \text{Var}\left(x\right)\left(h'(E(x))\right)^2$,

$$\text{Var}\left(\hat{d}_{\alpha,q}\right) = \text{Var}\left(\hat{d}_{\alpha,q}\right)\left(\alpha d_{(\alpha)}^{(\alpha-1)/\alpha}\right)^2 + O\left(\frac{1}{k^2}\right)$$
$$= \frac{1}{k}\frac{(q-q^2)\alpha^2/4}{f_X^2\left(W;\alpha,1\right)W^2}d_{(\alpha)}^2 + O\left(\frac{1}{k^2}\right).$$

B Proof of Lemma 2

First, consider $\alpha = 1$. In this case,

$$f_X(x;1,1) = \frac{1}{\pi}\frac{1}{x^2+1},\ W = F_X^{-1}\left((q+1)/2;1,1\right) = \tan\left(\frac{\pi}{2}q\right),$$
$$g(q;1) = \frac{q-q^2}{\left(\frac{2}{\pi}\frac{1}{\tan^2\left(\frac{\pi}{2}q\right)+1}\right)^2\tan^2\left(\frac{\pi}{2}q\right)} = \frac{q-q^2}{\sin^2(\pi q)}\pi^2.$$

It suffices to study $L(q) = \log g(q;1)$.

$$L'(q) = \frac{1}{q} - \frac{1}{1-q} - \frac{2\pi\cos(\pi q)}{\sin(\pi q)},$$
$$L''(q) = -\frac{1}{q^2} - \frac{1}{(1-q)^2} + \frac{2\pi^2}{\sin^2(\pi q)}.$$

Because $\sin(x) \leq x$ for $x \geq 0$, it is easy to see that $\frac{\pi}{\sin(\pi q)} - \frac{1}{q} \geq 0$, and $\frac{\pi}{\sin(\pi q)} - \frac{1}{1-q} = \frac{\pi}{\sin(\pi(1-q))} - \frac{1}{1-q} \geq 0$. Thus, $L'' \geq 0$, i.e., $L(q)$ is convex and so is $g(q;1) = e^{L(q)}$. Since $L'(1/2) = 0$, we know $q^*(1) = 0.5$.

Next we consider $\alpha = 0+$, using the fact [21] that as $\alpha \to 0+$, $|S(\alpha,1)|^\alpha$ converges to $1/E_1$, where E_1 stands for an exponential distribution with mean 1.

Denote $h = d_{(0+)}$ and $z_j \sim h/E_1$. The sample quantile estimator becomes

$$\hat{d}_{(0+),q} = \frac{q\text{-Quantile}\{|z_j|, j = 1,2,...,k\}}{q\text{-Quantile}\{1/E_1\}}.$$

In this case,

$$f_Z(z;h) = e^{-h/z}\frac{h}{z^2},\quad F_Z^{-1}(q;h) = -\frac{h}{\log q},$$
$$\text{Var}\left(\hat{d}_{(0+),q}\right) = \frac{1}{k}\frac{1-q}{q\log^2 q}h^2 + O\left(\frac{1}{k^2}\right).$$

It is straightforward to show that $\frac{1-q}{q\log^2 q}$ is a convex function of q and the minimum is attained by solving $-\log q^* + 2q^* - 2 = 0$, i.e., $q^* = 0.203$.

C Proof of Lemma 3

Given k i.i.d. samples, $x_j \sim S(\alpha, d_{(\alpha)})$, $j = 1$ to k. Let $z_j = |x_j|$, $j = 1$ to k. Denote by $F_Z(t;\alpha,d_{(\alpha)})$ the cumulative density of z_j, and by $F_{Z,k}(t;\alpha,d_{(\alpha)})$ the empirical cumulative density of z_j, $j = 1$ to k.

The basic result of order statistics says $kF_{Z,k}(t;\alpha,d_{(\alpha)})$ follows a binomial distribution[10], i.e., $kF_{Z,k}(t;\alpha,d_{(\alpha)}) \sim Bin(k, F_Z(t;\alpha,d_{(\alpha)}))$. For simplicity, we replace $F_Z(t;\alpha,d_{(\alpha)})$ by $F(t,d)$, $F_{Z,k}(t;\alpha,d_{(\alpha)})$ by $F_k(t,d)$, and $d_{(\alpha)}$ by d, only in this proof.

Using the ***original*** binomial Chernoff bounds [8], we obtain, for $\epsilon' > 0$,

$$\mathbf{Pr}\left(kF_k(t;d) \geq (1+\epsilon')kF(t;d)\right) \leq$$
$$\left(\frac{k-kF(t;d)}{k-(1+\epsilon')kF(t;d)}\right)^{k-k(1+\epsilon')F(t;d)}\left(\frac{kF(t;d)}{(1+\epsilon')kF(t;d)}\right)^{(1+\epsilon')kF(t;d)}$$
$$=\left[\left(\frac{1-F(t;d)}{1-(1+\epsilon')F(t;d)}\right)^{1-(1+\epsilon')F(t;d)}\left(\frac{1}{1+\epsilon'}\right)^{(1+\epsilon')F(t;d)}\right]^k,$$

and for $0 < \epsilon' < 1$,

$$\mathbf{Pr}\left(kF_k(t;d) \leq (1-\epsilon')kF(t;d)\right)$$
$$\leq\left[\left(\frac{1-F(t;d)}{1-(1-\epsilon')F(t;d)}\right)^{1-(1-\epsilon')F(t;d)}\left(\frac{1}{1-\epsilon'}\right)^{(1-\epsilon')F(t;d)}\right]^k.$$

Consider the general quantile estimator $\hat{d}_{(\alpha),q}$ defined in (6). For $\epsilon > 0$, (again, denote $W = q$-quantile$\{|S(\alpha,1)|\}$),

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \geq (1+\epsilon)d\right) = \mathbf{Pr}\left(q\text{-quantile}\{|x_j|\}\right) \geq \left((1+\epsilon)d\right)^{1/\alpha}W)$$
$$=\mathbf{Pr}\left(kF_k\left((1+\epsilon)^{1/\alpha}W;1\right) \leq qk\right) = \mathbf{Pr}\left(kF_k(t;1) \leq (1-\epsilon')kF(t;1)\right),$$

where $t = (1+\epsilon)^{1/\alpha}W$ and $q = (1-\epsilon')F(t;1)$. Thus

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \geq (1+\epsilon)d\right)$$
$$\leq\left[\left(\frac{1-F\left(((1+\epsilon))^{1/\alpha}W;1\right)}{1-q}\right)^{1-q}\left(\frac{F\left(((1+\epsilon))^{1/\alpha}W;1\right)}{q}\right)^q\right]^k$$
$$=\exp\left(-k\frac{\epsilon^2}{G_{R,q}}\right),$$

where

$$\frac{\epsilon^2}{G_{R,q}} = -(1-q)\log\left(1-F\left((1+\epsilon)^{1/\alpha}W;1\right)\right)$$
$$-q\log\left(F\left((1+\epsilon)^{1/\alpha}W;1\right)\right) + (1-q)\log(1-q) + q\log(q).$$

For $0 < \epsilon < 1$,

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \leq (1-\epsilon)d\right) = \mathbf{Pr}\left(kF_k\left((1-\epsilon)^{1/\alpha}W;1\right) \geq qk\right)$$
$$= \mathbf{Pr}\left(kF_k(t;1) \geq (1+\epsilon')kF(t;1)\right),$$

where $t = (1-\epsilon)^{1/\alpha}W$ and $q = (1+\epsilon')F(t;1)$. Thus,

$$\mathbf{Pr}\left(\hat{d}_{(\alpha),q} \leq (1-\epsilon)d\right)$$
$$\leq\left[\left(\frac{1-F\left((1-\epsilon)^{1/\alpha}W;1\right)}{1-q}\right)^{1-q}\left(\frac{F\left((1-\epsilon)^{1/\alpha}W;1\right)}{q}\right)^q\right]^k$$
$$=\exp\left(-k\frac{\epsilon^2}{G_{L,q}}\right),$$

where

$$\frac{\epsilon^2}{G_{L,q}} = -(1-q)\log\left(1-F\left((1-\epsilon)^{1/\alpha}W;1\right)\right)$$
$$-q\log\left(F\left((1-\epsilon)^{1/\alpha}W;1\right)\right) + (1-q)\log(1-q) + q\log(q).$$

Denote $f(t;d) = F'(t;d)$. Using L'Hospital's rule

$$\begin{aligned}
&\lim_{\epsilon\to 0+}\frac{1}{G_{R,q}} = \lim_{\epsilon\to 0+}\frac{-(1-q)\log\left(1-F\left((1+\epsilon)^{1/\alpha}W;1\right)\right)}{\epsilon^2} \\
&\quad+\frac{-q\log\left(F\left((1+\epsilon)^{1/\alpha}W;1\right)\right)+(1-q)\log(1-q)+q\log(q)}{\epsilon^2} \\
&=\lim_{\epsilon\to 0+}\frac{f\left((1+\epsilon)^{1/\alpha}W;1\right)\frac{W}{\alpha}(1+\epsilon)^{1/\alpha-1}}{F\left((1+\epsilon)^{1/\alpha}W;1\right)\left(1-F\left((1+\epsilon)^{1/\alpha}W;1\right)\right)}\times \\
&\frac{F\left((1+\epsilon)^{1/\alpha}W;1\right)-q}{2\epsilon} \\
&=\lim_{\epsilon\to 0+}\frac{\left(f\left((1+\epsilon)^{1/\alpha}W;1\right)\frac{W}{\alpha}(1+\epsilon)^{1/\alpha-1}\right)^2}{2F\left((1+\epsilon)^{1/\alpha}W;1\right)\left(1-F\left((1+\epsilon)^{1/\alpha}W;1\right)\right)} \\
&=\frac{f^2(W;1)W^2}{2q(1-q)\alpha^2}, \qquad\qquad (q=F(W,1)).
\end{aligned}$$

Similarly

$$\lim_{\epsilon\to 0+} G_{L,q} = \frac{2q(1-q)\alpha^2}{f^2(W;1)W^2}.$$

To complete the proof, apply the relations about $Z = |X|$.

References

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

[2] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu. On demand classification of data streams. In *KDD*, pages 503–508, 2004.

[3] B. Babcock, S. Babu, M. Datar, R. Motwani, and J. Widom. Models and issues in data stream systems. In *PODS*, pages 1–16, 2002.

[4] E. Bingham and H. Mannila. Random projection in dimensionality reduction: Applications to image and text data. In *KDD*, pages 245–250, 2001.

[5] L. Bottou, O. Chapelle, D. DeCoste, and J. Weston, editors. *Large-Scale Kernel Machines*. The MIT Press, 2007.

[6] J. M. Chambers, C. L. Mallows, and B. W. Stuck. A method for simulating stable random variables. *Journal of the American Statistical Association*, 71(354):340–344, 1976.

[7] O. Chapelle, P. Haffner, and V. N. Vapnik. Support vector machines for histogram-based image classification. *IEEE Trans. Neural Networks*, 10(5):1055–1064, 1999.

[8] H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23(4):493–507, 1952.

[9] H. Chernoff, J. L. Gastwirth, and M. V. Johns. Asymptotic distribution of linear combinations of functions of order statistics with applications to estimation. *The Annals of Mathematical Statistics*, 38(1):52–72, 1967.

[10] H. A. David. *Order Statistics*. John Wiley & Sons, Inc., New York, NY, second edition, 1981.

[11] C. Domeniconi and D. Gunopulos. Incremental support vector machine construction. In *ICDM*, pages 589–592, 2001.

[12] E. F. Fama and R. Roll. Parameter estimates for symmetric stable distributions. *Journal of the American Statistical Association*, 66(334):331–338, 1971.

[13] D. Fradkin and D. Madigan. Experiments with random projections for machine learning. In *KDD*, 2003.

[14] N. J. A. Harvey, J. Nelson, and K. Onak. Sketching and streaming entropy via approximation theory. In *FOCS*, 2008.

[15] N. J. A. Harvey, J. Nelson, and K. Onak. Streaming algorithms for estimating entropy. In *ITW*, 2008.

[16] P. Indyk. Stable distributions, pseudorandom generators, embeddings, and data stream computation. *Journal of ACM*, 53(3):307–323, 2006.

[17] Y. Jiang, C. Ngo, and J. Yang. Towards optimal bag-of-features for object categorization and semantic video retrieval. In *CIVR*, pages 494–501, 2007.

[18] W. B. Johnson and J. Lindenstrauss. Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

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

[20] P. Li. Very sparse stable random projections for dimension reduction in l_α $(0 < \alpha \leq 2)$ norm. In *KDD*, 2007.

[21] P. Li. Estimators and tail bounds for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *SODA*, pages 10 – 19, 2008.

[22] P. Li. A very efficient scheme for estimating entropy of data streams using compressed counting. Technical report, 2008.

[23] P. Li. Compressed counting. In *SODA*, 2009.

[24] P. Li, C. J. Burges, and Q. Wu. Mcrank: Learning to rank using classification and gradient boosting. In *NIPS*, 2008.

[25] P. Li and K. W. Church. Using sketches to estimate associations. In *HLT/EMNLP*, pages 708–715, 2005.

[26] P. Li and K. W. Church. A sketch algorithm for estimating two-way and multi-way associations. *Computational Linguistics*, 33(3):305–354, 2007.

[27] P. Li, K. W. Church, and T. J. Hastie. One sketch for all: Theory and applications of conditional random sampling. In *NIPS*, 2009.

[28] P. Li and T. J. Hastie. A unified near-optimal estimator for dimension reduction in l_α $(0 < \alpha \leq 2)$ using stable random projections. In *NIPS*, 2008.

[29] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *KDD*, pages 287–296, 2006.

[30] P. Li, T. J. Hastie, and K. W. Church. Nonlinear estimators and tail bounds for dimensional reduction in l_1 using cauchy random projections. *Journal of Machine Learning Research*, 8:2497–2532, 2007.

[31] J. H. McCulloch. Simple consistent estimators of stable distribution parameters. *Communications on Statistics-Simulation*, 15(4):1109–1136, 1986.

[32] S. Muthukrishnan. Data streams: Algorithms and applications. *Foundations and Trends in Theoretical Computer Science*, 1:117–236, 2 2005.

[33] J. C. Platt. Using analytic QP and sparseness to speed training of support vector machines. In *NIPS*, 1998.

[34] J. D. Rennie, L. Shih, J. Teevan, and D. R. Karger. Tackling the poor assumptions of naive Bayes text classifiers. In *ICML*, pages 616–623, 2003.

[35] B. Schölkopf and A. J. Smola. *Learning with Kernels*. The MIT Press, 2002.

[36] S. Vempala. *The Random Projection Method*. American Mathematical Society, 2004.

[37] Q. Yang and X. Wu. 10 challeng problems in data mining research. *International Journal of Information Technology and Decision Making*, 5(4):597–604, 2006.

[38] H. Zhao, A. Lall, M. Ogihara, O. Spatscheck, J. Wang, and J. Xu. A data streaming algorithm for estimating entropies of od flows. In *IMC*, 2007.