Coding for Random Projections

Ping Li PINGLI@STAT.RUTGERS.EDU
Dept. of Statistics and Biostatistics, Dept. of Computer Science, Rutgers University, Piscataway, NJ 08854, USA

Michael Mitzenmacher MICHAELM@EECS.HARVARD.EDU
School of Engineering and Applied Sciences, Harvard University, Cambridge, MA 02138, USA

Anshumali Shrivastava ANSHU@CS.CORNELL.EDU
Dept. of Computer Science, Computing and Information Science, Cornell University, Ithaca, NY 14853, USA

Abstract

The method of random projections has become popular for large-scale applications in statistical learning, information retrieval, bio-informatics and other applications. Using a well-designed **coding** scheme for the projected data, which determines the number of bits needed for each projected value and how to allocate these bits, can significantly improve the effectiveness of the algorithm, in storage cost as well as computational speed. In this paper, we study a number of simple coding schemes, focusing on the task of similarity estimation and on an application to training linear classifiers. We demonstrate that **uniform quantization** outperforms the standard and influential method (Datar et al., 2004), which used a *window-and-random offset* scheme. Indeed, we argue that in many cases coding with just a small number of bits suffices. Furthermore, we also develop a **non-uniform 2-bit** coding scheme that generally performs well in practice, as confirmed by our experiments on training linear support vector machines (SVM). Proofs and additional experiments are available at *arXiv:1308.2218*.

In the context of using coded random projections for **approximate near neighbor search** by building hash tables (*arXiv:1403.8144*) (Li et al., 2014), we show that the step of random offset in (Datar et al., 2004) is again not needed and may hurt the performance. Furthermore, we show that, unless the target similarity level is high, it usually suffices to use only 1 or 2 bits to code each hashed value for this task. Section 7 presents some experimental results for LSH.

1. Introduction

The method of random projections has become popular for large-scale machine learning applications such as classification, regression, matrix factorization, singular value decomposition, near neighbor search, bio-informatics, and more (Papadimitriou et al., 1998; Dasgupta, 1999; Bingham & Mannila, 2001; Buhler & Tompa, 2002; Fradkin & Madigan, 2003; Freund et al., 2008; Weinberger et al., 2009; Vempala, 2004; Dasgupta, 2000; Johnson & Lindenstrauss, 1984; Wang & Li, 2010). In this paper, we study a number of simple and effective schemes for **coding** the projected data, with the focus on similarity estimation and training linear classifiers (Joachims, 2006; Shalev-Shwartz et al., 2007; Fan et al., 2008; Bottou). We will compare our method with the influential prior coding scheme in (Datar et al., 2004) (which is a part of the standard LSH package).

Consider two high-dimensional data vectors, $u, v \in \mathbb{R}^D$. The idea of random projections is to multiply them with a random normal projection matrix $\mathbf{R} \in \mathbb{R}^{D \times k}$ (where $k \ll D$), to generate two (much) shorter vectors x, y:

$$x = u \times \mathbf{R} \in \mathbb{R}^k, \qquad y = v \times \mathbf{R} \in \mathbb{R}^k,$$
$$\mathbf{R} = \{r_{ij}\}_{i=1}^{D}{}_{j=1}^{k}, \qquad r_{ij} \sim N(0,1) \text{ i.i.d.}$$

In real applications, a dataset will consist of a large number of vectors (not just two). Without loss of generality, we use one pair of data vectors (u, v) to demonstrate our results.

In this study, for convenience, we assume that the marginal Euclidian norms of the original data vectors, i.e., $\|u\|, \|v\|$, are known. This assumption is reasonable in practice (Li et al., 2006). For example, the input data fed to a support vector machine (SVM) are usually normalized, i.e., $\|u\| = \|v\| = 1$. Computing the marginal norms for the entire dataset only requires one linear scan of the data, which is anyway needed during data collection/processing.

Without loss of generality, we assume $\|u\| = \|v\| = 1$. The joint distribution of (x_j, y_j) is hence a bi-variant normal:

$$\begin{bmatrix} x_j \\ y_j \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right), \text{ i.i.d. } j = 1, 2, ..., k.$$

where $\rho = \sum_{i=1}^{D} u_i v_i$ (as $\|u\| = \|v\| = 1$). In this paper, we adopt the conventional notation for the standard normal pdf $\phi(x) = \frac{1}{\sqrt{2\pi}} e^{-\frac{x^2}{2}}$ and cdf $\Phi(x) = \int_{-\infty}^{x} \phi(x) dx$.

Proceedings of the 31^{st} International Conference on Machine Learning, Beijing, China, 2014. JMLR: W&CP volume 32.

Note that this paper focuses on dense projections. It will be interesting future work to study coding for very sparse random projections (Li, 2007), count sketch (Charikar et al., 2004), or count-min sketch (Cormode & Muthukrishnan, 2005). Another useful line of research would be on the impact of pseudo-random numbers on coding (Carter & Wegman, 1977; Nisan, 1990; Mitzenmacher & Vadhan, 2008).

1.1. Uniform Quantization

Our first proposal is perhaps the most intuitive scheme, based on a simple uniform quantization:

$$h_w^{(j)}(u) = \lfloor x_j/w \rfloor, \qquad h_w^{(j)}(v) = \lfloor y_j/w \rfloor \tag{1}$$

where $w > 0$ is the bin width and $\lfloor . \rfloor$ is the standard floor operation, i.e., $\lfloor z \rfloor$ is the largest integer which is smaller than or equal to z. Later we will also use the standard ceiling operation $\lceil . \rceil$. We show that the collision probability $P_w = \mathbf{Pr}\left(h_w^{(j)}(u) = h_w^{(j)}(v)\right)$ is monotonically increasing in the similarity ρ, making (1) a suitable coding scheme for similarity estimation and near neighbor search.

The potential benefits of coding with a small number of bits arise because the (uncoded) projected data, $x_j = \sum_{i=1}^D u_i r_{ij}$ and $y_j = \sum_{i=1}^D v_i r_{ij}$, being real-valued numbers, are neither convenient/economical for storage and transmission, nor well-suited for indexing.

Since the original data are assumed to be normalized, i.e., $\|u\| = \|v\| = 1$, the marginal distribution of x_j (and y_j) is the standard normal, which decays rapidly at the tail, e.g., $1 - \Phi(3) = 10^{-3}$, $1 - \Phi(6) = 9.9 \times 10^{-10}$. If we use 6 as cutoff, i.e., values with absolute value greater than 6 are just treated as -6 and 6, then the number of bits needed would be $1 + \log_2 \lceil \frac{6}{w} \rceil$. In particular, if we choose the bin width $w \geq 6$, we can just record the sign of the outcome (i.e., a one-bit scheme). In general, the optimum choice of w depends on the similarity ρ and the task. In this paper we focus on the task of similarity estimation (of ρ) and we will provide the optimum w values for all similarity levels. Interestingly, using our uniform quantization scheme, we find in a certain range the optimum w values are indeed large (and larger than 6) which suggests in some cases a 1-bit scheme could be advantageous.

We can build **linear classifier** (e.g., linear SVM) using coded random projections. For example, assume the projected values are within $(-6, 6)$. If $w = 2$, then the code values output by h_w will be within the set $\{-3, -2, -1, 0, 1, 2\}$ and hence we can represent a projected value using a vector of length 6 (with exactly one 1) and the total length of the new feature vector will be $6 \times k$. This is inspired by the work on linear learning with *b-bit minwise hashing* (Li et al., 2012; Shrivastava & Li, 2014).

Near neighbor search is a basic problem studied since the early days of modern computing (Friedman et al., 1975). The use of coded projection data for near neighbor search is related to *locality sensitive hashing (LSH)* (Indyk & Motwani, 1998). In the context of LSH, the purpose of coding is for determining which buckets the projected data should be placed in. After the hash tables are built, there might be no need to store the coded data. Therefore, this task is different from coding for similarity estimation. A separate technical report (Li et al., 2014) elaborates on coding for LSH; also see Section 7. Note that, even in the context of LSH, we often still need similarity estimation because we need to determine, among all retrieved data points, the truly similar data points; a step often called "re-ranking". This will require evaluating similarities on the fly because in general we can not store all pairwise similarities.

1.2. Advantages over the Window-and-Offset Scheme

(Datar et al., 2004) proposed the following well-known coding scheme, which uses windows and a random offset:

$$h_{w,q}^{(j)}(u) = \left\lfloor \frac{x_j + q_j}{w} \right\rfloor, \qquad h_{w,q}^{(j)}(v) = \left\lfloor \frac{y_j + q_j}{w} \right\rfloor \tag{2}$$

where $q_j \sim uniform(0, w)$. (Datar et al., 2004) showed that the collision probability can be written as

$$\begin{aligned} P_{w,q} =& \mathbf{Pr}\left(h_{w,q}^{(j)}(u) = h_{w,q}^{(j)}(v)\right) \\ =& \int_0^w \frac{1}{\sqrt{d}} 2\phi\left(\frac{t}{\sqrt{d}}\right)\left(1 - \frac{t}{w}\right) dt \end{aligned} \tag{3}$$

where $d = ||u - v||^2 = 2(1 - \rho)$ is the Euclidean distance between u and v. The difference between (2) and our proposal (1) is that we do not use the additional randomization with $q \sim uniform(0, w)$ (i.e., the offset). We will demonstrate the following advantages of our scheme:

1. Operationally, our scheme h_w is simpler than $h_{w,q}$.
2. With a fixed w, our scheme h_w is always more accurate than $h_{w,q}$, often significantly so. For each coding scheme, we can separately find the optimum bin width w. We will show that the optimized h_w is also more accurate than optimized $h_{w,q}$, often significantly so.
3. For a wide range of ρ values (e.g., $\rho < 0.56$), the optimum w values for our scheme h_w are relatively large (e.g., > 6), while for the existing scheme $h_{w,q}$, the optimum w values are small (e.g., about 1). This means h_w requires a smaller number of bits than $h_{w,q}$.

In summary, uniform quantization is simpler, more accurate, and uses fewer bits than the influential prior work.

1.3. Organization

In Section 2, we analyze the collision probability for the uniform quantization scheme and then compare it with the collision probability of the well-known prior work (Datar et al., 2004) which uses an additional random offset. Because the collision probabilities are monotone functions of

the similarity ρ, we can always estimate ρ from the observed (empirical) collision probabilities. In Section 3, Our comparisons of the estimation variances concludes that the random offset step in (Datar et al., 2004) is not needed.

In Section 4, we develop a 2-bit non-unform coding scheme and demonstrate that its performance largely matches the performance of the uniform quantization scheme (which requires storing more bits). Interestingly, for certain range of the similarity ρ, we observe that only one bit is needed. Thus, Section 5 is devoted to comparing the 1-bit scheme, which shows that the 1-bit scheme does not perform as well when the similarity ρ is high. In Section 6, we provide a set of experiments on training linear SVM using all the coding schemes we have studied, to confirm the variance analysis. Sections 7 provides additional experiments for LSH.

2. The Collision Probability of Uniform Quantization h_w

To use our coding scheme h_w (1), we need to evaluate $P_w = \mathbf{Pr}\left(h_w^{(j)}(u) = h_w^{(j)}(v)\right)$, the collision probability. From practitioners' perspective, as long as P_w is a monotonically increasing function of the similarity ρ, it is a suitable coding scheme. In other words, it does not matter whether P_w has a closed-form expression, as long as we can demonstrate its advantage over the alternative (Datar et al., 2004), whose collision probability is denoted by $P_{w,q}$. Note that $P_{w,q}$ can be expressed in a closed-form in terms of the standard ϕ and Φ functions:

$$\begin{aligned} P_{w,q} &= \mathbf{Pr}\left(h_{w,q}^{(j)}(u) = h_{w,q}^{(j)}(v)\right) \\ &= 2\Phi\left(\frac{w}{\sqrt{d}}\right) - 1 - \frac{2}{\sqrt{2\pi}w/\sqrt{d}} + \frac{2}{w/\sqrt{d}}\phi\left(\frac{w}{\sqrt{d}}\right) \end{aligned} \quad (4)$$

Clearly, $P_{w,q} \to 1$ as $w \to \infty$. Recall $d = 2(1-\rho) = \|u - v\|^2$ is the Euclidean distance.

The following Lemma 1 will help derive the collision probability P_w (in Theorem 1). Note that, due to the space constraint, all the proofs can be found at *arXiv:1308.2218*.

Lemma 1 *Let* $\begin{bmatrix} x \\ y \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix}\right)$. *Then*

$$\begin{aligned} Q_{s,t}(\rho) &= \mathbf{Pr}\left(x \in [s,t],\ y \in [s,t]\right) \quad (5) \\ &= \int_s^t \phi(z)\left[\Phi\left(\frac{t-\rho z}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{s-\rho z}{\sqrt{1-\rho^2}}\right)\right] dz \\ \frac{\partial Q_{s,t}(\rho)}{\partial \rho} &= \frac{1}{2\pi}\frac{1}{(1-\rho^2)^{1/2}} \quad (6) \\ &\quad \times \left(e^{-\frac{t^2}{(1+\rho)}} + e^{-\frac{s^2}{(1+\rho)}} - 2e^{-\frac{t^2+s^2-2st\rho}{2(1-\rho^2)}}\right) \end{aligned}$$

As nonnegative data are common in practice, we will focus on $\rho \geq 0$, for the rest of the paper.

Theorem 1 *The collision probability of the coding scheme h_w defined in (1) is*

$$P_w = 2\sum_{i=0}^{\infty}\int_{iw}^{(i+1)w} \phi(z) \times \left[\Phi\left(\frac{(i+1)w-\rho z}{\sqrt{1-\rho^2}}\right) - \Phi\left(\frac{iw-\rho z}{\sqrt{1-\rho^2}}\right)\right] dz \quad (7)$$

which is a monotonically increasing function of $\rho \geq 0$.

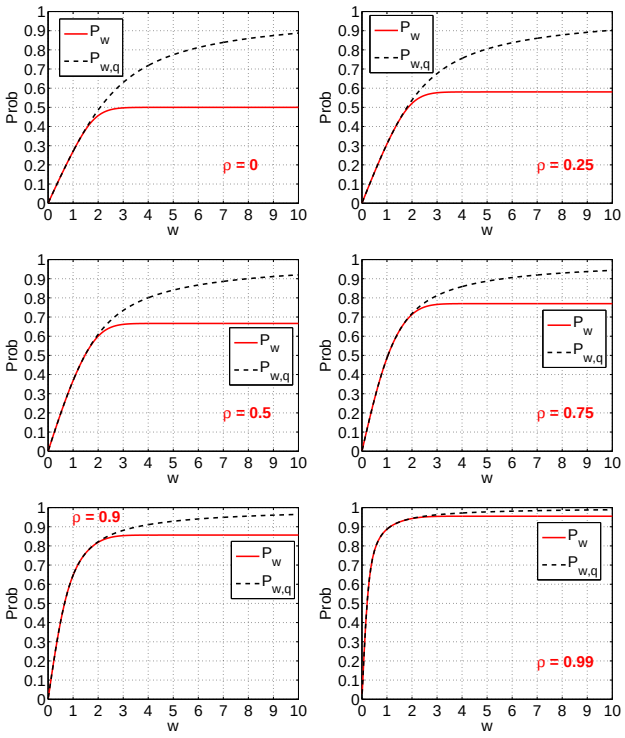

Figure 1. Collision probabilities, P_w and $P_{w,q}$, for $\rho = 0, 0.25, 0.5, 0.75, 0.9$, and 0.99. Our proposed scheme (h_w) has smaller collision probabilities than the existing scheme (Datar et al., 2004) ($h_{w,q}$), especially when $w > 2$.

Figure 1 plots both P_w and $P_{w,q}$ for selected ρ values. The difference between P_w and $P_{w,q}$ becomes apparent after about $w > 2$. When $\rho = 0$, P_w quickly approaches the limit 0.5 while $P_{w,q}$ keeps increasing (to 1) as w increases. Intuitively, the fact that $P_{w,q} \to 1$ when $\rho = 0$, is undesirable because it means two orthogonal vectors will have the same coded value. Thus, it is not surprising that our proposed scheme h_w will have better performance than $h_{w,q}$.

3. Analysis of Two Coding Schemes (h_w and $h_{w,q}$) for Similarity Estimation

In both schemes (corresponding to h_w and $h_{w,q}$), the collision probabilities P_w and $P_{w,q}$ are monotonically increas-

ing functions of the similarity ρ. Since there is a one-to-one mapping between ρ and P_w, we can tabulate P_w for each ρ (e.g., at a precision of 10^{-3}). From k independent projections, we can compute the empirical $\hat{P}_w$ and $\hat{P}_{w,q}$ and find the estimates, denoted by $\hat{\rho}_w$ and $\hat{\rho}_{w,q}$, respectively, from the tables. Theorem 2 provides the variance of h_w.

Theorem 2 *Recall* $d = 2(1-\rho)$.

$$Var\left(\hat{\rho}_{w,q}\right) = \frac{V_{w,q}}{k} + O\left(\frac{1}{k^2}\right), \quad \textit{where} \tag{8}$$

$$V_{w,q} = d^2/4 \left(\frac{w/\sqrt{d}}{\phi\left(w/\sqrt{d}\right) - 1/\sqrt{2\pi}}\right)^2 P_{w,q}(1-P_{w,q}) \tag{9}$$

Figure 2 plots the variance factor $V_{w,q}$ defined in (9) without the $\frac{d^2}{4}$ term. (Recall $d = 2(1-\rho)$.) The minimum is 7.6797 (keeping four digits), attained at $w/\sqrt{d} = 1.6476$. The plot also suggests that the performance of this popular scheme can be sensitive to the choice of the bin width w. This is a practical disadvantage. Since we do not know ρ (or d) in advance and we must specify w in advance, the performance of this scheme might be unsatisfactory, as one can not really find one "optimum" w for all pairs.

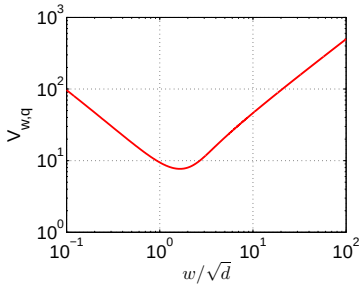

Figure 2. The variance factor $V_{w,q}$ (9) without the $\frac{d^2}{4}$ term.

In comparison, our proposed scheme has smaller variance and is not as sensitive to the choice of w.

Theorem 3

$$Var\left(\hat{\rho}_w\right) = \frac{V_w}{k} + O\left(\frac{1}{k^2}\right), \quad \textit{where} \tag{10}$$

$$V_w = \frac{\pi^2(1-\rho^2)P_w(1-P_w)}{\left[\sum_{i=0}^{\infty}\left(e^{-\frac{(i+1)^2w^2}{(1+\rho)}} + e^{-\frac{i^2w^2}{(1+\rho)}} - 2e^{-\frac{w^2}{2(1-\rho^2)}}e^{-\frac{i(i+1)w^2}{1+\rho}}\right)\right]^2} \tag{11}$$

In particular, when $\rho = 0$, *we have*

$$V_w|_{\rho=0} = \left[\frac{\sum_{i=0}^{\infty}\left(\Phi((i+1)w) - \Phi(iw)\right)^2}{\sum_{i=0}^{\infty}\left(\phi((i+1)w) - \phi(iw)\right)^2}\right] \times \left[\frac{1/2 - \sum_{i=0}^{\infty}\left(\Phi((i+1)w) - \Phi(iw)\right)^2}{\sum_{i=0}^{\infty}\left(\phi((i+1)w) - \phi(iw)\right)^2}\right] \tag{12}$$

Remark: At $\rho = 0$, the minimum is $V_w = \frac{\pi^2}{4}$ attained at $w \to \infty$, as shown in Figure 3. Note that when $w \to \infty$, we have $\sum_{i=0}^{\infty}\left(\Phi((i+1)w) - \Phi(iw)\right)^2 \to 1/4$ and $\sum_{i=0}^{\infty}\left(\phi((i+1)w) - \phi(iw)\right)^2 \to 1/(2\pi)$, and hence $V_w|_{\rho=0} \to \left[\frac{1/4}{1/(2\pi)}\right]\left[\frac{1/2-1/4}{1/(2\pi)}\right] = \frac{\pi^2}{4}$, which is substantially smaller than 7.6797, the smallest $V_{w,q}$ when $\rho = 0$.

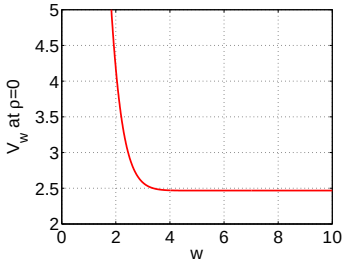

Figure 3. The minimum of $V_w|_{\rho=0} \to \pi^2/4$, as $w \to \infty$.

To compare the variances of the two estimators, $Var\left(\hat{\rho}_w\right)$ and $Var\left(\hat{\rho}_{w,q}\right)$, we compare their leading constants, V_w and $V_{w,q}$. Figure 4 plots the V_w and $V_{w,q}$ at selected ρ values, confirming that (i) the variance of the proposed scheme (1) can be significantly lower than the existing scheme (2); and (ii) the performance of the proposed scheme is not as sensitive to the choice of w (e.g., when $w > 2$).

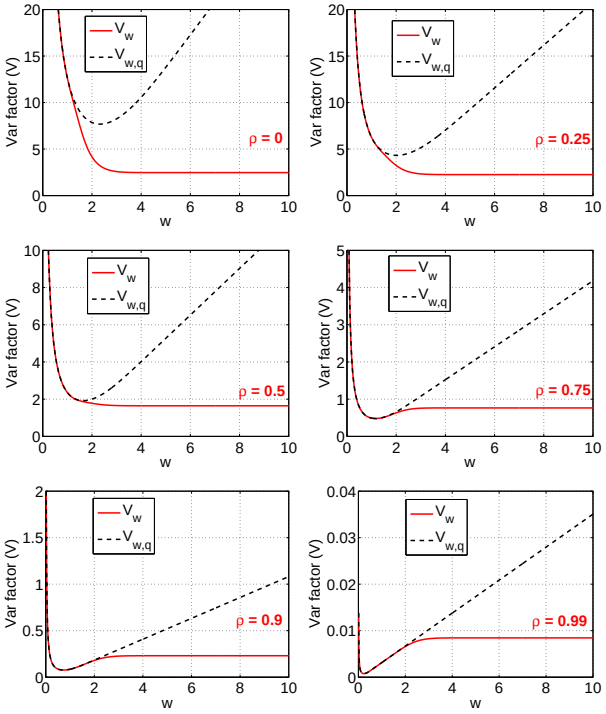

Figure 4. Comparisons of two coding schemes at fixed bin width w, i.e., V_w (11) vs $V_{w,q}$ (9). V_w is smaller than $V_{w,q}$ especially when $w > 2$ (or even when $w > 1$ and ρ is small). For both schemes, at a fixed ρ, we can find the optimum w value which minimizes V_w (or $V_{w,q}$). In general, once $w > 1 \sim 2$, V_w is not sensitive to w (unless ρ is very close to 1). This is one significant advantage of the proposed scheme h_w.

It is also informative to compare V_w and $V_{w,q}$ at their "optimum" w values (for fixed ρ). Note that V_w is not sensitive to w once $w > 1 \sim 2$. The left panel of Figure 5 plots the best values for V_w and $V_{w,q}$, confirming that V_w is significantly lower than $V_{w,q}$ at smaller ρ values (e.g., $\rho < 0.56$).

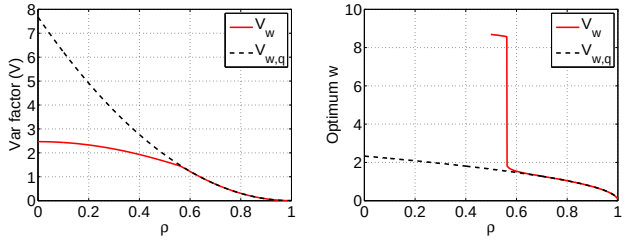

Figure 5. Comparisons of two coding schemes, h_w and $h_{w,q}$, at their optimum bin width (w) values.

The right panel of Figure 5 plots the optimum w values (for fixed ρ). Around $\rho = 0.56$, the optimum w for V_w becomes significantly larger than 6 and may not be reliably evaluated. From the remark for Theorem 3, we know that at $\rho = 0$ the optimum w grows to ∞. Thus, we can conclude that if $\rho < 0.56$, it suffices to implement our coding scheme using just 1 bit (i.e., signs of the projected data). In comparison, for the existing scheme $h_{w,q}$, the optimum w varies much slower. Even at $\rho = 0$, the optimum w is around 2. This means $h_{w,q}$ will always need to use more bits than h_w, to code the projected data.

In practice, we do not know ρ in advance and we often care about high similarities. When $\rho > 0.56$, Figure 4 and Figure 5 illustrate that we might want to choose small w values (e.g., $w < 1$). However, using a small w value will hurt the performance in the pairs of low similarities. This motivates us to develop non-uniform coding schemes.

4. A 2-Bit Non-Uniform Coding Scheme

If we quantize the projected data according to four regions $(-\infty, -w), [-w, 0), [0, w), [w, \infty)$, we obtain a 2-bit non-uniform scheme. At the risk of abusing notation, we name this scheme "$h_{w,2}$", not to be confused with the name of the existing scheme $h_{w,q}$.

According to Lemma 1, $h_{w,2}$ is also a valid coding scheme. We can theoretically compute the collision probability, denoted by $P_{w,2}$, which is again a monotonically increasing function of the similarity ρ. With k projections, we can estimate ρ from the empirical observation of $P_{w,2}$ and we denote this estimator by $\hat{\rho}_{w,2}$.

Theorem 4 provides expressions for $P_{w,2}$ and $Var\left(\hat{\rho}_{w,2}\right)$.

Theorem 4

$$P_{w,2} = \mathbf{Pr}\left(h^{(j)}_{w,2}(u) = h^{(j)}_{w,2}(v)\right) \tag{13}$$

$$= \left\{1 - \frac{1}{\pi}\cos^{-1}\rho\right\} - 4\int_0^w \phi(z)\Phi\left(\frac{-w+\rho z}{\sqrt{1-\rho^2}}\right)dz$$

$$Var\left(\hat{\rho}_{w,2}\right) = \frac{V_{w,2}}{k} + O\left(\frac{1}{k^2}\right), \tag{14}$$

$$V_{w,2} = \frac{\pi^2(1-\rho^2)P_{w,2}(1-P_{w,2})}{\left[1 - 2e^{-\frac{w^2}{2(1-\rho^2)}} + 2e^{-\frac{w^2}{1+\rho}}\right]^2} \tag{15}$$

Figure 6 plots $P_{w,2}$ (and P_w) for selected ρ values. Note that $P_{w,2}$ has the same value at $w = 0$ and $w = \infty$, and in fact, when $w = 0$ or $w = \infty$, we just need one bit (i.e., the signs). When $w > 1$, $P_{w,2}$ and P_w largely overlap. For small w, these two behave very differently, as expected. Will this be beneficial? The answer again depends on ρ.

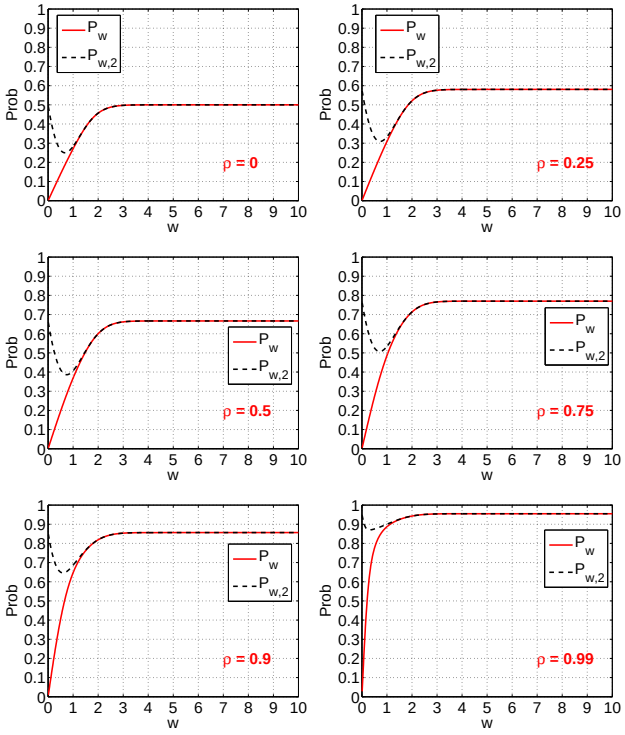

Figure 6. Collision probabilities for the two proposed coding scheme h_w and $h_{w,2}$. Note that, while $P_{w,2}$ is a monotonically increasing function in ρ, it is no longer monotone in w.

Figure 7 plots $V_{w,2}$ and V_w at selected ρ values, to compare their variances. For $\rho \leq 0.5$, the variance of the estimator using the 2-bit scheme $h_{w,2}$ is significantly lower than that of h_w. However, when ρ is high, $V_{w,2}$ might be higher than V_w. This means that, in general, we expect the performance of $h_{w,2}$ will be similar to h_w. When applications care about highly similar data pairs, we expect h_w will have (slightly) better performance at the cost of more bits.

Figure 8 presents the smallest $V_{w,2}$ values and the optimum w values at which the smallest $V_{w,2}$ are attained. It verifies h_w and $h_{w,2}$ should perform similarly, although h_w will have better performance at high ρ. Also, for a wide range, e.g., $\rho \in [0.2\ 0.62]$, it is preferable to implement $h_{w,2}$ using just 1 bit because the optimum w values are large.

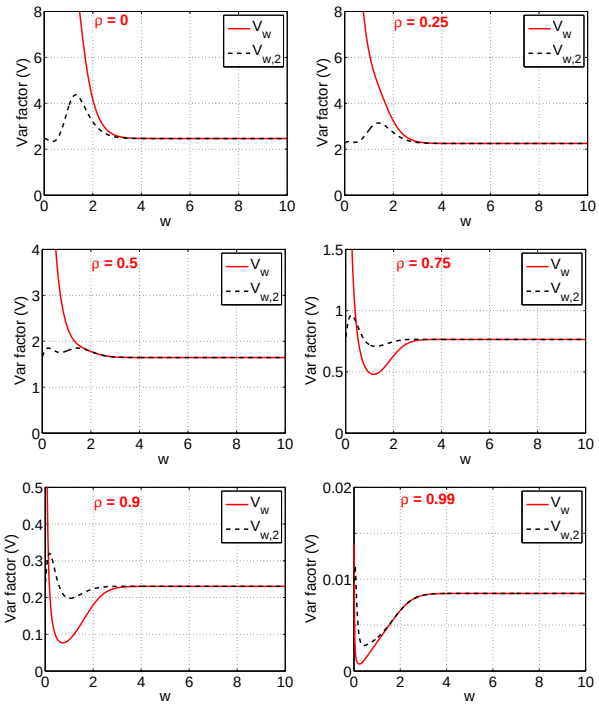

Figure 7. Comparisons of the estimation variances of two proposed schemes: h_w and $h_{w,2}$ in terms of V_w and $V_{w,2}$. When $\rho \leq 0.5$, $V_{w,2}$ is significantly lower than V_w at small w. However, when ρ is high, $V_{w,2}$ will be somewhat higher than V_w. Note that $V_{w,2}$ is not as sensitive to w, unlike V_w.

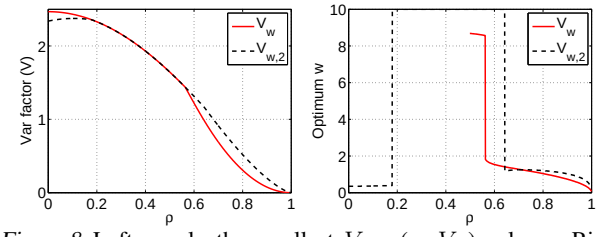

Figure 8. Left panel: the smallest $V_{w,2}$ (or V_w) values. Right panel: the optimum w values at which the smallest $V_{w,2}$ (or V_w) is attained at a fixed ρ.

5. The 1-Bit Scheme h_1 and Comparisons with $h_{w,2}$ and h_w

When $w > 6$, it is sufficient to implement h_w or $h_{w,2}$ using just one bit, because the normal probability density decays rapidly: $1 - \Phi(6) = 9.9 \times 10^{-10}$. With the 1-bit scheme, we simply code the projected data by recording their signs. We denote this scheme by h_1, the corresponding collision probability by P_1, and the corresponding estimator by $\hat{\rho}_1$.

From Theorem 4, by setting $w = \infty$, we can directly infer

$$P_1 = \mathbf{Pr}\left(h_1^{(j)}(u) = h_1^{(j)}(v)\right) = 1 - \frac{1}{\pi}\cos^{-1}\rho \quad (16)$$

$$Var\left(\hat{\rho}_1\right) = \frac{V_1}{k} + O\left(\frac{1}{k^2}\right) \quad (17)$$

$$V_1 = \pi^2(1-\rho^2)P_1(1-P_1) \quad (18)$$

This collision probability is known (Goemans & Williamson, 1995; Charikar, 2002).

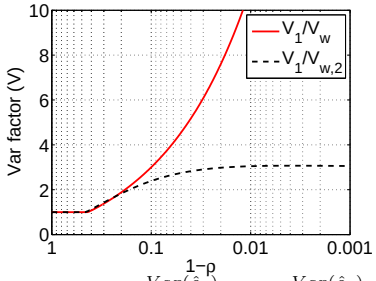

Figure 9. Variance ratios: $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_w)}$ and $\frac{Var(\hat{\rho}_1)}{Var\left(\hat{\rho}_{w,2}\right)}$, to illustrate the reduction of estimation accuracies if the 1-bit coding scheme is used. We plot the maximum ratios (over all w) at each ρ. To visualize the high similarity region, we plot $1 - \rho$ in log-scale.

Figure 9 and Figure 10 plot the ratios of the variances: $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_w)}$ and $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_{w,2})}$, to illustrate how much we lose in accuracy by using only one bit. Note that $\hat{\rho}_1$ is not related to the bin width w while $Var\left(\hat{\rho}_w\right)$ and $Var\left(\hat{\rho}_{w,2}\right)$ are functions of w. In Figure 9, we plot the maximum values of the ratios, i.e., we use the smallest $Var\left(\hat{\rho}_w\right)$ and $Var\left(\hat{\rho}_{w,2}\right)$ at each ρ. These ratios demonstrate that potentially both h_w and $h_{w,2}$ could substantially outperform h_1, the 1-bit scheme. Note that we plot $1 - \rho$ in the horizontal axis with log-scale, so that the high similarity region can be visualized better. In practice, applications (e.g., duplicate detections) are often interested in high similarity regions.

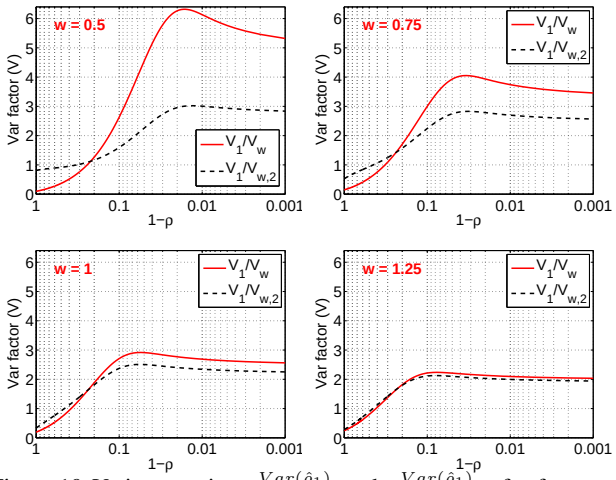

Figure 10. Variance ratios: $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_w)}$ and $\frac{Var(\hat{\rho}_1)}{Var\left(\hat{\rho}_{w,2}\right)}$, for four selected w values. In the high similarity region (e.g., $\rho \geq 0.9$), the 2-bit scheme $h_{w,2}$ significantly outperforms the 1-bit scheme h_1. In the low similarity region, $h_{w,2}$ still works reasonably well while the performance of h_w can be poor (e.g., when $w \leq 0.75$). This justifies the use of the 2-bit scheme.

In practice, we must pre-specify the quantization bin width w in advance. Thus, the improvement of h_w and $h_{w,2}$ over the 1-bit scheme h_1 will not be as drastic as shown in Figure 9. For more realistic comparisons, Figure 10 plots

$\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_w)}$ and $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_{w,2})}$, for fixed w values. This figure advocates the use of the 2-bit coding scheme $h_{w,2}$:

1. In high similarity region, $h_{w,2}$ significantly outperforms h_1. The improvement drops as w becomes larger (e.g., $w > 1$). h_w also works well, in fact better than $h_{w,2}$ when w is small.
2. In low similarity region, $h_{w,2}$ still outperforms h_1 unless ρ is very low and w is not small. The performance of h_w is much worse than $h_{w,2}$ and h_1 at low ρ.

Thus, we believe the 2-bit scheme $h_{w,2}$ with w around 0.75 provides an overall good compromise. In fact, this is consistent with our SVM experiments in Section 6.

Can we simply use the 1-bit scheme? When $w = 0.75$, in the high similarity region, the variance ratio $\frac{Var(\hat{\rho}_1)}{Var(\hat{\rho}_{w,2})}$ is between 2 and 3. Note that, per projected data value, the 1-bit scheme requires 1 bit but the 2-bit scheme needs 2 bits. In a sense, the performances of $h_{w,2}$ and h_1 are actually similar in terms of the total number bits needed, according the analysis in this paper.

For similarity estimation, we believe it is preferable to use the 2-bit scheme, for the following reasons:

- The processing cost of the 2-bit scheme would be lower. If we use k projections for the 1-bit scheme and $k/2$ projections for the 2-bit scheme, although they have the same storage, the processing cost of $h_{w,2}$ for generating the projections would be only 1/2 of h_1. For high-dimensional data, processing can be costly.
- In this study, we restrict our attention to linear estimators (which can be written as inner products) by using the (overall) collision probability, e.g., $P_{w,2} = \mathbf{Pr}\left(h^{(j)}_{w,2}(u) = h^{(j)}_{w,2}(v)\right)$. There is considerable room for improvement by using nonlinear maximum likelihood estimators, which can be useful (e.g.,) in the re-ranking stage of near neighbor search.
- There is an interesting phenomenon. The training and testing speed of linear SVM largely depends on the sparsity of the input data. In other words, using k projections with 2-bit coding will likely lead to faster learning than using $2k$ projections with 1-bit coding.
- Quantization is a non-reversible process. Once we quantize the data by the 1-bit scheme, there is no hope of recovering any information other than the signs.

In practice, which coding scheme to use will depend on the data and the tasks. Our work provides the necessary theoretical justifications for making practical choices.

6. Experiments for Training Linear SVM

We conduct experiments with random projections for training ($L2$-regularized) linear SVM (e.g., LIBLINEAR (Fan et al., 2008)) on three high-dimensional datasets: *ARCENE, FARM, URL*, which are available from the UCI repository. The original *URL* dataset has about 2.4 million examples (collected in 120 days) in 3231961 dimensions. We only used the data from the first day, with 10000 examples for training and 10000 for testing. The *ARCENE* dataset contains 100 training and 100 testing examples in 10000 dimensions. The *FARM* dataset has 2059 training and 2084 testing examples in 54877 dimensions. See detailed experimental results in *arXiv:1308.2218*.

We implement the four coding schemes studied in this paper: $h_{w,q}$, h_w, $h_{w,2}$, and h_1. Recall $h_{w,q}$ (Datar et al., 2004) was based on uniform quantization plus a random offset, with bin width w. Here, we first illustrate exactly how we utilize the coded data for training linear SVM. Suppose we use $h_{w,2}$ and $w = 0.75$. We can code an original projected value x into a vector of length 4 (i.e., 2-bit):

$$x \in (-\infty\ -0.75) \Rightarrow [1\ 0\ 0\ 0], x \in [-0.75\ 0) \Rightarrow [0\ 1\ 0\ 0],$$
$$x \in [0\ 0.75) \Rightarrow [0\ 0\ 1\ 0], \qquad x \in [0.75\ \infty) \Rightarrow [0\ 0\ 0\ 1]$$

This way, with k projections, for each feature vector, we obtain a new vector of length $4k$ with exactly k 1's. This new vector is then fed to a solver such as LIBLINEAR.

Figure 11 reports the test accuracies on the *URL* data, for comparing $h_{w,q}$ with h_w. The results basically confirm our analysis of the estimation variances. For small bin width w, the two schemes perform very similarly. When using a relatively large w, the scheme $h_{w,q}$ suffers from noticeable reduction of classification accuracies. The experimental results on the other two datasets demonstrate the same phenomenon. This experiment confirms that the step of random offset in $h_{w,q}$ is not needed in this context.

Figure 11 reports the accuracies for a wide range of SVM tuning parameter C values, from 10^{-3} to 10^3. Before feeding data to LIBLINEAR, we normalize them to unit norm.

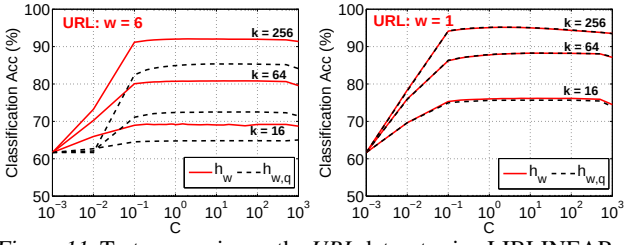

Figure 11. Test accuracies on the *URL* dataset using LIBLINEAR, for comparing two coding schemes: h_w vs. $h_{w,q}$. We report the classification results for $k = 16$, $k = 64$, and $k = 256$. In each panel, there are 3 solid curves for h_w and 3 dashed curves for $h_{w,q}$. We report the results for a wide range of L_2-regularization parameter C. When w is large, h_w noticeably outperforms $h_{w,q}$. When w is small, the two schemes perform similarly.

Figure 12 reports the test classification accuracies (averaged over 20 repetitions) for the *URL* dataset. When $w = 0.5 \sim 1$, both h_w and $h_{w,2}$ produce similar results

as using the original projected data. The 1-bit scheme h_1 is obviously less competitive.

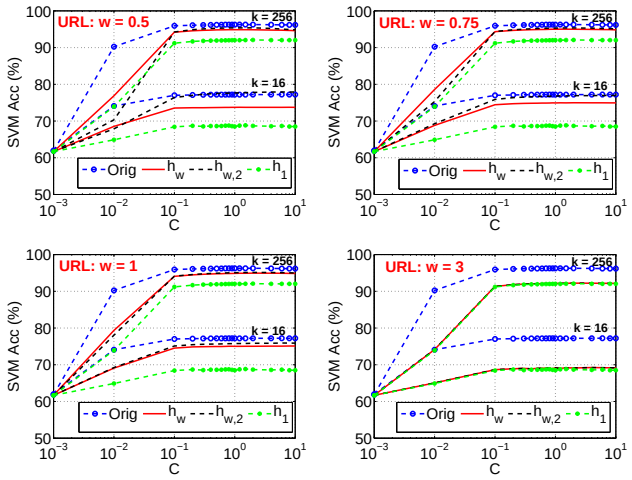

Figure 12. Test accuracies on the *URL* dataset using LIBLINEAR, for comparing four coding schemes: uncoded ("Orig"), h_w, $h_{w,2}$, and h_1. We report the results for $k = 16$ and $k = 256$. Thus, in each panel, there are 2 groups of 4 curves. We report the results for a wide range of L_2-regularization parameter C (i.e., the horizontal axis). When $w = 0.5 \sim 1$, both h_w and $h_{w,2}$ produce similar results as using the original projected data, while the 1-bit scheme h_1 is less competitive.

We summarize the experiments in Figure 13. The upper panels report, for each k, the best (highest) test classification accuracies among all C values and w values (for $h_{w,2}$ and h_w). The results show a clear trend: (i) the 1-bit (h_1) scheme produces noticeably lower accuracies compared to others; (ii) the performances of $h_{w,2}$ and h_w are quite similar. The bottom panels of Figure 13 report the w values at which the best accuracies were attained. For $h_{w,2}$, the optimum w values are close to 0.75.

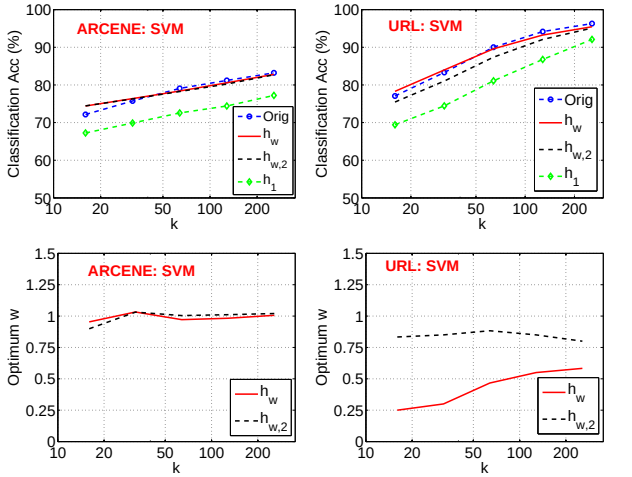

Figure 13. Summary of the linear SVM results on two datasets. The upper panels report, for each k, the best (highest) classification accuracies among all C values and w values (for $h_{w,2}$ and h_w). The bottom panels report the best w values.

7. Coding for Building Hash Tables for LSH

Coding for building hash tables for LSH (Indyk & Motwani, 1998) is different from coding for similarity estimation. For the former task, we need to use the coded values to determine which buckets the corresponding data points should be placed in. The report (*arXiv:1403.8144*) (Li et al., 2014) contains the details. In summary, comparing the two coding scheme: h_w and $h_{w,q}$, it is always preferable to use h_w (i.e., no random offset) and often only a small number of bits are needed. See Figure 14.

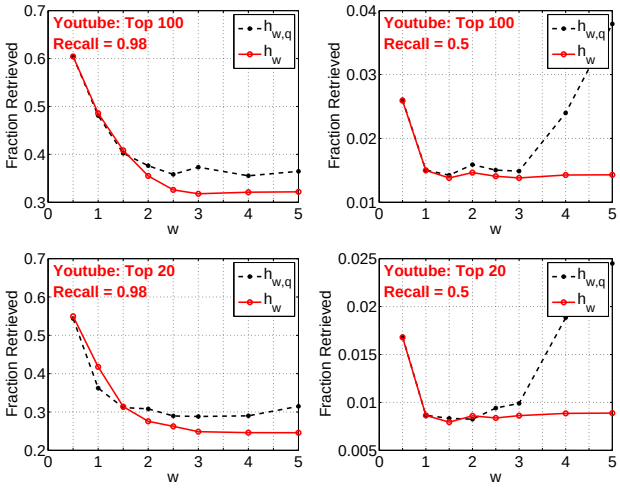

Figure 14. Standard LSH retrieval experiments on *Youtube* dataset to compare two schemes: h_w (solid) and $h_{w,q}$ (dashed). Here we use the exact similarities for evaluating the recalls and fraction of retrieve data points (relative to the original training set), for two target recall values. This set of experiments demonstrate that advantages of h_w. (A lower curve is better in this experiment.) See (*arXiv:1403.8144*) (Li et al., 2014) for more details and explanations. We thank the referees for suggesting us to add this section.

8. Conclusion

The method of random projections has become a standard algorithmic approach for machine learning and data management. A compact representation (coding) of the projected data is crucial for efficient transmission, retrieval, and energy consumption. We have compared a simple scheme based on uniform quantization with the influential coding scheme using windows with a random offset (Datar et al., 2004); our scheme appears operationally simpler, more accurate, not as sensitive to parameters (e.g., the widow/bin width w), and uses fewer bits. We furthermore develop a 2-bit non-uniform coding scheme which performs similarly to uniform quantization.

Acknowledgements: P. Li is supported by AFOSR (FA9550-13-1-0137), ONR (N00014-13-1-0764), and NSF (III1360971, BIGDATA1419210). M. Mitzenmacher is supported by NSF (CCF0915922, IIS0964473, CNS1228598, CCF1320231). A. Shrivastava is supported by NSF (III1249316) and ONR (N00014-13-1-0764).

References

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

Bottou, Leon. http://leon.bottou.org/projects/sgd.

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

Carter, J. Lawrence and Wegman, Mark N. Universal classes of hash functions. In *STOC*, pp. 106–112, 1977.

Charikar, Moses, Chen, Kevin, and Farach-Colton, Martin. Finding frequent items in data streams. *Theor. Comput. Sci.*, 312(1):3–15, 2004.

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

Cormode, Graham and Muthukrishnan, S. An improved data stream summary: the count-min sketch and its applications. *Journal of Algorithm*, 55(1):58–75, 2005.

Dasgupta, Sanjoy. Learning mixtures of gaussians. In *FOCS*, pp. 634–644, New York, 1999.

Dasgupta, Sanjoy. Experiments with random projection. In *UAI*, pp. 143–151, Stanford, CA, 2000.

Datar, Mayur, Immorlica, Nicole, Indyk, Piotr, and Mirrokn, Vahab S. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, pp. 253 – 262, Brooklyn, NY, 2004.

Fan, Rong-En, Chang, Kai-Wei, Hsieh, Cho-Jui, Wang, Xiang-Rui, and Lin, Chih-Jen. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

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

Freund, Yoav, Dasgupta, Sanjoy, Kabra, Mayank, and Verma, Nakul. Learning the structure of manifolds using random projections. In *NIPS*, Vancouver, BC, Canada, 2008.

Friedman, Jerome H., Baskett, F., and Shustek, L. An algorithm for finding nearest neighbors. *IEEE Transactions on Computers*, 24:1000–1006, 1975.

Goemans, Michel X. and Williamson, David P. Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of ACM*, 42(6):1115–1145, 1995.

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

Joachims, Thorsten. Training linear svms in linear time. In *KDD*, pp. 217–226, Pittsburgh, PA, 2006.

Johnson, William B. and Lindenstrauss, Joram. Extensions of Lipschitz mapping into Hilbert space. *Contemporary Mathematics*, 26:189–206, 1984.

Li, Ping. Very sparse stable random projections for dimension reduction in l_α $(0 < \alpha \leq 2)$ norm. In *KDD*, San Jose, CA, 2007.

Li, Ping, Hastie, Trevor J., and Church, Kenneth W. Improving random projections using marginal information. In *COLT*, pp. 635–649, Pittsburgh, PA, 2006.

Li, Ping, Owen, Art B, and Zhang, Cun-Hui. One permutation hashing. In *NIPS*, Lake Tahoe, NV, 2012.

Li, Ping, Mitzenmacher, Michael, and Shrivastava, Anshumali. Coding for random projections and approximate near neighbor search. Technical report, arXiv:1403.8144, 2014.

Mitzenmacher, Michael and Vadhan, Salil. Why simple hash functions work: exploiting the entropy in a data stream. In *SODA*, 2008.

Nisan, Noam. Pseudorandom generators for space-bounded computations. In *Proceedings of the twenty-second annual ACM symposium on Theory of computing*, STOC, pp. 204–212, 1990.

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

Shalev-Shwartz, Shai, Singer, Yoram, and Srebro, Nathan. Pegasos: Primal estimated sub-gradient solver for svm. In *ICML*, pp. 807–814, Corvalis, Oregon, 2007.

Shrivastava, Anshumali and Li, Ping. Densifying one permutation hashing via rotation for fast near neighbor search. In *ICML*, Beijing, China, 2014.

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

Wang, Fei and Li, Ping. Efficient nonnegative matrix factorization with random projections. In *SDM*, 2010.

Weinberger, Kilian, Dasgupta, Anirban, Langford, John, Smola, Alex, and Attenberg, Josh. Feature hashing for large scale multitask learning. In *ICML*, pp. 1113–1120, 2009.