Optimal hash functions for approximate closest pairs on the n-cube

Daniel M. Gordon, Victor Miller and Peter Ostapenko

***Abstract*—One way to find closest pairs in large datasets is to use hash functions [6], [12]. In recent years locality-sensitive hash functions for various metrics have been given: projecting an n-cube onto k bits is simple hash function that performs well.**

In this paper we investigate alternatives to projection. For various parameters hash functions given by complete decoding algorithms for codes work better, and asymptotically random codes perform better than projection.

I. Introduction

Given a set of M n-bit vectors, the closest pair problem is to find the two with smallest Hamming distance. This problem has applications in numerous areas, such as information retrieval and DNA sequence comparison. One approach ([6], [9], [12]) is to apply a hash function to the vectors, choosing the hash to be locality-sensitive, so that the probability of two vectors colliding is large if they are close, and small otherwise.

The standard hash to use is projection onto k of the n coordinates. This hash is the best known for general n and k [9]. An alternative family of hashes is based on minimum-weight decoding with error-correcting codes [4], [16]. A $[n, k]$ code $\mathcal{C}$ with a complete decoding algorithm defines a hash $h^{\mathcal{C}}$, where each $\mathbf{v} \in \mathcal{V} := \mathbb{F}_2^n$ is mapped to the codeword $\mathbf{c} \in \mathcal{C} \subset \mathcal{V}$ that $\mathbf{v}$ decodes to. Using linear codes for hashing schemes has been independently suggested many times; see [4], [7], and the patents [3] and [16].

In [4] the binary Golay code is suggested to find approximate matches in bit-vectors. Data is given that suggests it is effective, but it is still not clear when the Golay or other codes work better than projection. In this paper we attempt to quantify this, using tools from coding theory.

Let $P^{\mathcal{C}}(p)$ be the probability that $h^{\mathcal{C}}(\mathbf{x}) = h^{\mathcal{C}}(\mathbf{x}+\mathbf{e})$, where $\mathbf{x}$ is a random element of $\mathcal{V}$ and where each bit of the error vector $\mathbf{e}$ is nonzero with probability p. For a linear code with a complete translation invariant decoding algorithm (so that $h(\mathbf{x}) = \mathbf{c}$ implies that $h(\mathbf{x} + \mathbf{c}') = \mathbf{c} + \mathbf{c}'$), studying $P^{\mathcal{C}}$ is equivalent to studying the properties of the set $\mathcal{S}$ of all points in $\mathcal{V}$ that decode to 0.

Suppose that we pick a random $\mathbf{x} \in \mathcal{S}$. Then the probability that $\mathbf{y} = \mathbf{x} + \mathbf{e}$ is in $\mathcal{S}$ is

$$P_{\mathcal{S}}(p) = \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{S}} p^{d(\mathbf{x},\mathbf{y})}(1-p)^{n-d(\mathbf{x},\mathbf{y})}. \tag{1}$$

This function has been studied extensively in the setting of error-detecting codes [13]. In the case where $\mathcal{S}$ is a code, $P_{\mathcal{S}}(p)$ is the probability of an undetected error, and the goal is to minimize this probability. Here, on the other hand, we will call a region *optimal* for p if no region in $\mathcal{V}$ of size $|\mathcal{S}|$ has greater probability.

As the error rate p approaches $1/2$, this coincides with the definition of *distance-sum optimal sets*, which were first studied by Ahlswede and Katona [1].

Define the error exponent of a code P to be

$$E^{\mathcal{C}}(p) = -\frac{1}{n} \lg P^{\mathcal{C}}(p).$$

In this paper lg denotes log to base 2. We are interested in properties of the error exponent over codes of rate $R = k/n$ as $n \to \infty$. In Section IV we will show that hash functions from random (nonlinear) codes have a better error exponent than projection.

II. Hash Functions From Codes

For a set $\mathcal{S} \subset \mathcal{V}$, let

$$A_i = \#\{(\mathbf{x}, \mathbf{y}) : \mathbf{x}, \mathbf{y} \in \mathcal{S} \text{ and } d(\mathbf{x}, \mathbf{y}) = i\}$$

count the number of pairs of words in $\mathcal{S}$ at distance i. The *distance distribution function* is

$$A(\mathcal{S}, \zeta) := \sum_{i=0}^{n} A_i \zeta^i. \tag{2}$$

This function is directly connected to $P_{\mathcal{S}}(p)$ [13]. If $\mathbf{x}$ is a random element of $\mathcal{S}$, and $\mathbf{y} = \mathbf{x} + e$, where e is an error vector where each bit is nonzero with probability p, then the probability that $\mathbf{y} \in \mathcal{S}$ is

$$\begin{aligned} P_{\mathcal{S}}(p) &:= \frac{1}{|\mathcal{S}|} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{S}} p^{d(\mathbf{x},\mathbf{y})}(1-p)^{n-d(\mathbf{x},\mathbf{y})} \\ &= \frac{1}{|\mathcal{S}|} \sum_{i=0}^{n} A_i p^i (1-p)^{n-i} \\ &= \frac{(1-p)^n}{|\mathcal{S}|} A\left(\mathcal{S}, \frac{p}{1-p}\right). \end{aligned} \tag{3}$$

In this section we will evaluate (3) for projection and for perfect codes, and then consider other linear codes.

A. Projection

The simplest hash is to project vectors in $\mathcal{V}$ onto k coordinates. Let *k-projection* denote the $[n, k]$ code $\mathcal{P}_{n,k}$ corresponding to this hash. The associated $\mathcal{S}$ of vectors mapped to $\mathbf{0}$ is an 2^{n-k}-subcube of $\mathcal{V}$. The distance distribution function is

$$A(\mathcal{S}, \zeta) = (2(1+\zeta))^{n-k}, \tag{4}$$

so the probability of collision is

$$\mathsf{P}^{\mathcal{P}_{n,k}}(p) = \frac{(1-p)^n}{2^{n-k}}\left(\frac{2}{1-p}\right)^{n-k} = (1-p)^k. \quad (5)$$

$\mathcal{P}_{n,k}$ is not a good error-correcting code, but for sufficiently small error rates its hash function is optimal.

Theorem 1: Let $\mathcal{S}$ be the 2^{n-k}-subcube of $\mathcal{V}$. For any error rate $p \in (0, 2^{-2(n-k)})$, $\mathcal{S}$ is an optimal region, and so k-projection is an optimal hash.

Proof: The distance distribution function for $\mathcal{S}$ is

$$A(\mathcal{S}, \zeta) = 2^{n-k}(1+\zeta)^{n-k}.$$

The edge isoperimetric inequality for an n-cube [10] states that

Lemma 2: Any subset S of the vertices of the n-dimensional cube Q_n has at most

$$\frac{1}{2}|S| \lg |S|$$

edges between vertices in S, with equality if and only if S is a subcube.

Any set $\mathcal{S}'$ with 2^{n-k} points has distance distribution function

$$A(\mathcal{S}', \zeta) = \sum_{i=0}^{k} c_i \zeta^i,$$

where $c_0 = 2^{n-k}$, $c_1 < (n-k)2^{n-k}$ by Lemma 2, and the sum of the c_i's is $2^{2(n-k)}$. By (5) the probability of collision is $(1-p)^n 2^{n-k} A(\mathcal{S}', p/(1-p))$.

$$\begin{aligned} A(\mathcal{S}', \zeta) \le\ & 2^{n-k} + \zeta((n-k)2^{n-k} - 1) \\ & + \zeta^2\left(2^{2(n-k)} - (n-k+1)2^{n-k} + 1\right), \end{aligned}$$

and

$$\begin{aligned} & A(\mathcal{S}, \zeta) - A(\mathcal{S}', \zeta) \\ & \quad \ge \zeta - \zeta^2\left(2^{2(n-k)} + 2^{n-k-1}\left(n - k^2 + n - k + 2\right) + 1\right) \\ & \quad > \zeta - \zeta^2(2^{2(n-k)} - 1). \end{aligned}$$

This is positive if $p < 1/2$ and $(1-p)/p > 2^{2(n-k)} - 1$, i.e., for $p < 2^{-2(n-k)}$. ∎

B. Concatenated Hashes

Here we show that if h and h' are good hashes, then the concatenation is as well. First we identify $\mathcal{C}$ with $\mathbb{F}_2^k$ and treat $h^{\mathcal{C}}$ as a hash h from $\mathbb{F}_2^n \to \mathbb{F}_2^k$. We denote $\mathsf{P}^{\mathcal{C}}$ by $\mathcal{P}^h$. From $h : \mathbb{F}_2^n \to \mathbb{F}_2^k$ and $h' : \mathbb{F}_2^{n'} \to \mathbb{F}_2^{k'}$, we get a concatenated hash $(h, h') : \mathbb{F}_2^{n+n'} \to \mathbb{F}_2^{k+k'}$.

Lemma 3: Fix $p \in (0, 1/2)$. Let h and h' be hashes. Then

$$\min\{\mathsf{E}^h(p), \mathsf{E}^{h'}(p)\} \le \mathsf{E}^{(h,h')(p)} \le \max\{\mathsf{E}^h(p), \mathsf{E}^{h'}(p)\},$$

with strict inequalities if $\mathsf{E}^h(p) \ne \mathsf{E}^{h'}(p)$.

Proof: Since p is fixed, we drop it from the notation. Suppose $\mathsf{E}^h \le \mathsf{E}^{h'}$. Then

$$\frac{\lg \mathsf{P}^h}{n} \le \frac{\lg \mathsf{P}^h + \lg \mathsf{P}^{h'}}{n+n'} \le \frac{\lg \mathsf{P}^{h'}}{n'}.$$

Since $\mathsf{P}^{(h,h')} = \mathsf{P}^h\,\mathsf{P}^{h'}$, we have $\mathsf{E}^h \le \mathsf{E}^{(h,h')} \le \mathsf{E}^{h'}$. ∎

TABLE I
CROSSOVER ERROR RATES p FOR HAMMING CODES $\mathcal{H}_m$.

m	k	p
4	11	0.2826
5	26	0.1518
6	57	0.0838
7	120	0.0468

C. Perfect Codes

An e-*sphere* around a vector $\mathbf{x}$ is the set of all vectors $\mathbf{y}$ with $d(\mathbf{x}, \mathbf{y}) \le e$. An $[n, k, 2e+1]$ code Π is *perfect* if the e-spheres around codewords cover $\mathcal{V}$. Minimum weight decoding with perfect codes is a reasonable starting point for hashing schemes, since all vectors are closest to a unique codeword. The only perfect binary codes are trivial repetition codes, the Hamming codes, and the binary Golay code. Repetition codes do badly, but the other perfect codes give good hash functions.

1) Binary Golay Code: The $[23, 12, 7]$ binary Golay code $\mathcal{G}$ is an important perfect code. The 3-spheres around each code codeword cover $\mathbb{F}_2^{23}$. The 3-sphere around $\mathbf{0}$ in the 23-cube has distance distribution function

$$\begin{aligned} & 2048 + 11684\zeta + 128524\zeta^2 + 226688\zeta^3 \\ & \qquad + 1133440\zeta^4 + 672980\zeta^5 + 2018940\zeta^6. \end{aligned}$$

From this we find $\mathsf{E}^{\mathcal{G}}(p) > \mathsf{E}^{\mathcal{P}_{23,12}}(p)$ for $p \in (0.2555, 1/2)$.

2) Hamming Codes: Aside from the repetition codes and the Golay code, the only perfect binary codes are the Hamming codes. The $[2^m - 1, 2^m - m - 1, 3]$ Hamming code $\mathcal{H}_m$ corrects one error.

The distance distribution function for a 1-sphere is

$$2^m + 2(2^m - 1)\zeta + (2^m - 1)(2^m - 2)\zeta^2, \quad (6)$$

so the probability of collision $\mathsf{P}^{\mathcal{H}_m}(p)$ is

$$\begin{aligned} & \frac{(1-p)^{2^m-1}}{2^m}\Big(2^m + 2(2^m-1)\frac{p}{1-p} \\ & \qquad + (2^m-1)(2^m-2)\frac{p^2}{(1-p)^2}\Big) \end{aligned} \quad (7)$$

Table I gives the crossover error rates where the first few Hamming codes become better than projection.

Theorem 4: For any $m > 4$ and $p > m/(2^m - m)$, the Hamming code $\mathcal{H}_m$ beats $(2^m - m - 1)$-projection.

Proof: The difference between the distribution functions of the cube and the 1-sphere in dimension $2^m - 1$ is

$$\begin{aligned} f_m(\zeta) &:= A(\mathcal{S}, \zeta) - A(\mathcal{H}_m, \zeta) \\ &= 2^m(1+\zeta)^m \\ &\quad -(2^m + 2(2^m-1)\zeta + (2^m-1)(2^m-2)\zeta^2). \end{aligned} \quad (8)$$

We will show that, for $m \ge 4$, $f_m(\zeta)$ has exactly one root in $(0, 1)$, denoted by α_m, and that $\alpha_m \in ((m-2)/2^m, m/2^m)$.

We calculate

$$\begin{aligned} f_m(\zeta) =\ & ((m-2)2^m + 1)\zeta + 2^m \sum_{i=3}^{m} \binom{m}{i}\zeta^i \\ & - \left(2^{2m} - \left(3 + \binom{m}{2}\right)2^m + 2\right)\zeta^2. \end{aligned}$$

All the coefficients of $f_m(\zeta)$ are non-negative with the exception of the coefficient of ζ^2, which is negative for $m \geq 2$. Thus, by Descartes' rule of signs $f(\zeta)$ has 0 or 2 positive roots. However, it has a root at $\zeta = 1$. Call the other positive root α_m. We have $f_m(0) = f_m(1) = 0$, and since $f'(0) = (m-2)2^m + 2 > 0$ and $f'(1) = 2^{2m-1}(m-4) + 2^{m+2} - 2 > 0$ for $m \geq 4$, we must have $\alpha_m < 1$ for $m \geq 4$.

For $p > \alpha_m$ the Hamming code $\mathcal{H}_m$ beats projection.

Using (8) and Bernoulli's inequality, it is easy to show that $f_m(\zeta) > 0$ for $\zeta < c(m-2)/2^m$ for any $c < 1$ and $m \geq 4$. For the other direction, we may use Taylor's theorem to show

$$2^m \left(1 + \frac{m}{2^m}\right)^m < 2^m + m^2 + \frac{m^4}{2^{m+1}} \left(1 + \frac{m}{2^m}\right)^{m-2}.$$

Plugging this into (8), we have that $f_m(m/2^m) < 0$ for $m > 6$.

■

D. Other Linear Codes

The above codes give hashing strategies for a few values of n and k, but we would like hashes for a wider range. For a hashing strategy using error-correcting codes, we need a code with an efficient *complete decoding* algorithm; that is a way to map every vector to a codeword. Given a translation invariant decoder, we may determine $\mathcal{S}$, the set of vectors that map to $\mathbf{0}$, in order to compare strategies as the error rate changes.

`Magma` [5] has a built-in database of linear codes over $\mathbb{F}_2$ of length up to 256. Most of these do not come with efficient complete decoding algorithms, but `magma` does provide syndrome decoding. Using this database new hashing schemes were found. For each dimension k and minimum distance d, an $[n, k, d]$ binary linear code with minimum length n was chosen for testing.[1] (This criterion excludes any codes formed by concatenating with a projection code.) Figure 1 shows the results. Not surprisingly, the $[23, 12, 7]$ Golay code $\mathcal{G}$ and Hamming codes $\mathcal{H}_4$ and $\mathcal{H}_5$ all do well. The facts that concatenating the Golay code with projection beats the chosen code for $13 \leq k \leq 17$ and concatenating $\mathcal{H}_m$ with projection beats the chosen codes for $27 \leq k \leq 30$ show that factors other than minimum length are important in determining an optimal hashing code.

III. Optimal Regions

An $[n, k]$ code with a complete decoding algorithm gives a hashing region of size 2^{n-k}. In the previous section we looked at the performances of regions associated with various good error-correcting codes. In this section we consider general regions $\mathcal{S} \subset \mathbb{F}_2^n$.

The general question of finding an optimal region of size 2^t in $\mathcal{V}$ for an error rate p is quite hard. In this section we will find the answer for $t \leq 6$, and look at what happens when p is near $1/2$.

[1]The `magma` call `BLLC(GF(2),k,d)` was used to choose a code.

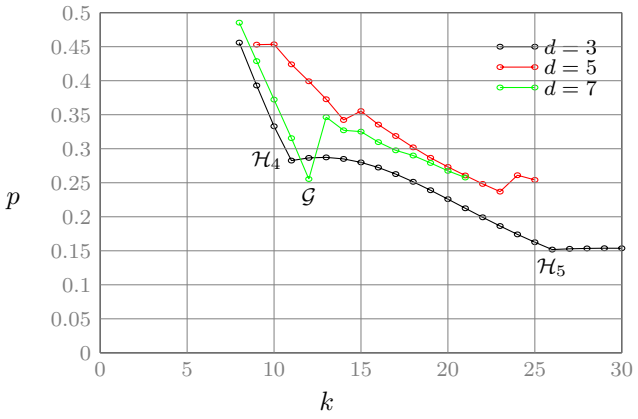

Fig. 1. Crossover error rates for minimum length linear codes.

A. Optimal Regions of Small Size

For a vector $\mathbf{x} = (x_1, \ldots, x_n) \in \mathcal{V}$, let

$$r_i(\mathbf{x}) := (x_1, x_2, \ldots, x_{i-1}, 1 - x_i, x_{i+1}, \ldots x_n)$$

be $\mathbf{x}$ with the i-th coordinate complemented, and let

$$s_{ij}(\mathbf{x}) := (x_1, \ldots, x_{i-1}, x_j, x_{i+1}, \ldots, x_{j-1}, x_i, x_{j+1}, \ldots x_n)$$

be $\mathbf{x}$ with the i-th and j-th coordinates switched.

Definition 5: Two sets are isomorphic if one can be gotten from the other by a series of r_i and s_{ij} transformations.

The corresponding non-invertible transformation are:

$$\rho_i(\mathbf{x}) := (x_1, x_2, \ldots, x_{i-1}, 0, x_{i+1}, \ldots x_n),$$
$$\sigma_{ij}(\mathbf{x}) := \begin{cases} \mathbf{x}, & x_{\min(i,j)} = 0, \\ s_{ij}(\mathbf{x}), & x_{\min(i,j)} = 1. \end{cases}$$

Definition 6: A set $\mathcal{S} \subset \mathcal{V}$ is a *down-set* if $\rho_i(\mathcal{S}) \subset \mathcal{S}$ for all $i \leq n$.

Definition 7: A set $\mathcal{S} \subset \mathcal{V}$ is *right-shifted* if $\sigma_{ij}(\mathcal{S}) \subset \mathcal{S}$ for all $i, j \leq n$.

Theorem 8: If a set $\mathcal{S}$ is optimal, then it is isomorphic to a right-shifted down-set.

Proof: We will show that any optimal region is isomorphic to a right-shifted set. The proof that it must be isomorphic to a down-set as well is similar. A similar proof for distance-sum optimal regions (see Section III-B) was given by Kündgen in [14]).

Recall that

$$P_{\mathcal{S}}(p) = \frac{(1-p)^n}{|\mathcal{S}|} \sum_{\mathbf{x},\mathbf{y} \in \mathcal{S}} \zeta^{d(\mathbf{x},\mathbf{y})},$$

where $\zeta = p/(1-p) \in (0, 1)$. If $\mathcal{S}$ is not right-shifted, there is some $\mathbf{x} \in \mathcal{S}$ with $x_i = 1$, $x_j = 0$, and $i < j$. Let $\varphi_{ij}(\mathcal{S})$ replace all such sets $\mathbf{x}$ with $r_{ij}(\mathbf{x})$. We only need to show that this will not decrease $P_{\mathcal{S}}(p)$.

Consider such an $\mathbf{x}$ and any $\mathbf{y} \in \mathcal{S}$. If $y_i = y_j$, then $d(\mathbf{x}, \mathbf{y}) = d(r_{ij}(\mathbf{x}), \mathbf{y})$, and $P_{\mathcal{S}}(p)$ will not change. If $y_i = 0$ and $y_j = 1$, then $d(\mathbf{x}, \mathbf{y}) = d(r_{ij}(\mathbf{x}), \mathbf{y}) - 2$, and since $\zeta^{l-2} \geq \zeta^l$, that term's contribution to $P_{\mathcal{S}}(p)$ increases.

Suppose $y_i = 1$ and $y_j = 0$. If $r_{ij}(\mathbf{y}) \in \mathcal{S}$, then $d(\mathbf{x},\mathbf{y}) + d(\mathbf{x}, r_{ij}(\mathbf{y})) = d(r_{ij}(\mathbf{x}),\mathbf{y}) + d(r_{ij}(\mathbf{x}), r_{ij}(\mathbf{y}))$, and $P_{\mathcal{S}}(p)$ is unchanged. Otherwise, $\varphi_{ij}(\mathcal{S})$ will replace $\mathbf{y}$ by $r_{ij}(\mathbf{y})$, and $d(\mathbf{x},\mathbf{y}) = d(r_{ij}(\mathbf{x}), r_{ij}(\mathbf{y}))$ means that $P_{\mathcal{S}}(p)$ will again be unchanged. ∎

Let $R_{s,n}$ denote an optimal region of size s in $\mathbb{F}_2^n$. By computing all right-shifted down-sets of size 2^t, for $t \leq 6$, we have the following result:

Theorem 9: The optimal regions $R_{2^t,n}$ for $t \in \{1,\ldots,6\}$ correspond to Tables III [pg. 6] and IV [pg. 7].

These figures, and details of the computations, are given the Appendix. Some of the optimal regions for $t = 6$ do better than the regions corresponding to the codes in Figure 1, although it is not known whether they tile $\mathcal{V}$.

B. Optimal Regions for Large Error Rates

Theorem 1 states that for any n and k, for a sufficiently small error rate p, a 2^{n-k}-subcube is an optimal region. One may also ask what an optimal region is at the other extreme, a large error rate. In this section we use existing results about minimum average distance subsets to list additional regions that are optimal as $p \to 1/2^-$.

We have

$$P_{\mathcal{S}}(p) := \frac{(1-p)^n}{|\mathcal{S}|} A\left(\mathcal{S}, \frac{p}{1-p}\right) = \frac{1}{|\mathcal{S}|}\sum\nolimits_i A_i p^i (1-p)^{n-i}.$$

Letting $p = 1/2 - \varepsilon$ and $s = |\mathcal{S}|$, $P_{\mathcal{S}}(\gamma)$ becomes

$$\begin{aligned} s^{-1}&\sum\nolimits_i A_i \left(1/2-\varepsilon\right)^i \left(1/2+\varepsilon\right)^{n-i} \\ &= \frac{1}{s\,2^n}\left(\sum\nolimits_i A_i + \varepsilon\left(\sum\nolimits_i 2(n-2i)A_i\right) + O(\varepsilon^2)\right) \\ &= \frac{s}{2^n}(1+2n\varepsilon) - \frac{4\varepsilon}{s\,2^n}\sum\nolimits_i iA_i + O(\varepsilon^2)\,. \end{aligned}$$

Therefore, an optimal region for $p \to 1/2^-$ must minimize the *distance-sum* of $\mathcal{S}$

$$d(\mathcal{S}) := \frac{1}{2}\sum_{\mathbf{x},\mathbf{y}\in\mathcal{S}} d(\mathbf{x},\mathbf{y}) = \frac{1}{2}\sum\nolimits_i iA_i\,. \tag{9}$$

Denote the minimal distance sum by

$$f(s,n) := \min\left\{d(\mathcal{S}) : \mathcal{S} \subset \mathbb{F}_2^n,\, |\mathcal{S}| = s\right\}.$$

If $d(\mathcal{S}) = f(s,n)$ for a set $\mathcal{S}$ of size s, we say that $\mathcal{S}$ is *distance-sum optimal*. The question of which sets are distance-sum optimal was proposed by Ahlswede and Katona in 1977; see Kündgen [14] for references and recent results.

This question is also difficult. Kündgen presents distance-sum optimal regions for small s and n, which include the ones of size 16 from Table III. Jaeger et al. [11] found the distance-sum optimal region for n large.

Theorem 10: (Jaeger, et al. [11], cf. [14, pg. 151]) For $n \geq s-1$, a generalized 1-sphere (with s points) is distance-sum optimal unless $s \in \{4, 8\}$ (in which case the subcube is optimal).

From this we have:

Corollary 11: For $n \geq 2^t - 1$, with $t \geq 4$ and p sufficiently close to $1/2$, a $(2^t - 1)$-dimensional 1-sphere is hashing optimal.

IV. Hashes from Random Codes

In this section we will show that hashes from random linear codes under minimum weight decoding[2] perform better than projection. Let $\mathcal{R}$ be a random linear code of rate $R = k/n$. The error exponent for k-projection is

$$-\frac{1}{n}\lg(1-p)^k = -R\lg(1-p).$$

Theorem 4 shows that for any $p > 0$ there are codes with rate $R \approx 1$ which beat projection. In this section we will show that this is true for random codes with any R.

Let H be the binary entropy

$$H(\delta) := -\delta\lg\delta - (1-\delta)\lg(1-\delta)\,. \tag{10}$$

Fix $\delta \in [0, 1/2)$. Let $d := \lfloor \delta n \rfloor$, let $\mathcal{S}_d(\mathbf{x})$ denote the sphere of radius d around $\mathbf{x}$, and let $V(d) := |\mathcal{S}_d(\mathbf{x})|$. From [8], Theorem 2.2, we have

Lemma 12: Let $\mathcal{R}$ be a random linear code of rate R. For $\mathbf{c} \in \mathcal{R}$, the probability that there is another codeword in $\mathcal{S}_d(\mathbf{c})$ is at most

$$\frac{1}{1-2\delta}\sqrt{\frac{1-\delta}{2\pi n\delta}}\; e^{n(H(\delta)-1+R)}.$$

Lemma 12 implies that, with high probability, everything in $\mathcal{S}_d(\mathbf{c})$ will be decoded to $\mathbf{c}$, including any vector $\mathbf{x}$ of distance exactly d from $\mathbf{c}$. Let $P_{\mathcal{R}}(p)$ be the probability that a random point $\mathbf{x}$ and $\mathbf{x}+\mathbf{e}$ both hash to $\mathbf{c}$. This is greater than the probability that $\mathbf{x}+\mathbf{e}$ has weight exactly d, so

$$P_{\mathcal{R}}(p) > \sum_{i=0}^{d}\binom{d}{i}\binom{n-d}{i} p^{2i}(1-p)^{n-2i}.$$

Theorem 4 of [2] gives a bound for this:

Theorem 13:

$$\begin{aligned}\limsup_{n\to\infty}\left(-\frac{1}{n}\lg P_{\mathcal{R}}(p)\right) \geq\; &\varepsilon\lg p + (1-\varepsilon)\lg(1-p) \\ &+ \delta H\left(\frac{\varepsilon}{2\delta}\right) + (1-\delta)H\left(\frac{\varepsilon}{2(1-\delta)}\right)\end{aligned}$$

for any $\varepsilon \leq 1/2$. The right hand side is maximized at $\varepsilon_{\max}$ satisfying

$$\frac{(2\delta-\varepsilon_{\max})(2(1-\delta)-\varepsilon_{\max})}{{\varepsilon_{\max}}^2} = \frac{(1-p)^2}{p^2}.$$

Define

$$\begin{aligned}D(p,\delta,\varepsilon) := \;&\varepsilon\lg p + (1-\varepsilon)\lg(1-p) + \delta H\left(\frac{\varepsilon}{2\delta}\right) \\ &+(1-\delta)H\left(\frac{\varepsilon}{2(1-\delta)}\right) - (1-H(\delta))\lg(1-p)\,.\end{aligned}$$

This function bounds the difference between the expected log probability of collisions for random codes and for projection. The following theorem shows that for any error probability and code rate, a random code is expected to do better than projection.

[2]Ties arising in minimum weight decoding are broken in some unspecified manner.

Theorem 14: $D(p,\delta,\varepsilon_{\max})$ is positive for any $\delta, p \in (0,1/2)$.

Proof: Fix $\delta \in (0,1/2)$, and let $f(p) := D(p,\delta,\varepsilon_{\max})$. It is easy to check that:

$$\lim_{p\to 0^+} f(p) = 0,$$
$$\lim_{p\to 1/2^-} f(p) = 0,$$
$$\lim_{p\to 0^+} f'(p) > 0,$$
$$\lim_{p\to 1/2^-} f'(p) < 0,$$

Therefore, it suffices to show that $f'(p)$ has only one zero in $(0,1/2)$. Observe that $\varepsilon_{\max}$ is chosen so that $\frac{\partial D}{\partial \varepsilon}(\delta,p,\varepsilon_{\max}) = 0$. Hence

$$\begin{aligned} f'(p) &= \frac{\partial D}{\partial p}(\delta,p,\varepsilon_{\max}) \\ &= \frac{\varepsilon_{\max}}{p\log(2)} - \frac{1-\varepsilon_{\max}}{(1-p)\lg(2)} + \frac{1-H(\delta)}{(1-p)\log(2)}, \end{aligned}$$

so

$$\log(2) f'(p) = \frac{\varepsilon_{\max}}{p} - \frac{1-\varepsilon_{\max}}{1-p} + \frac{1-H(\delta)}{1-p}.$$

Therefore $f'(p) = 0$ when $\varepsilon_{\max} = pH(\delta)$. From Theorem 13 we find

$$p = \frac{4\delta(1-\delta) - H(\delta)^2}{2(H(\delta) - H(\delta)^2)}.$$

■

An immediate consequence of Theorem 14 is the non-optimality of projections.

Theorem 15: Fix the error rate $p \in (0,1/2)$. For any $R \in (0,1)$ and n sufficiently large, the expected probability of collision for a random code of rate R is higher than projection.

ACKNOWLEDGEMENTS.

The authors would like to thank William Bradley, David desJardins and David Moulton for stimulating discussions which helped initiate this work. Also, Tom Dorsey and Amit Khetan provided the simpler proof of Theorem 14 given here.

APPENDIX

By Theorem 8, we may find all optimal regions by examining all right-shifted down-sets. Right-shifted down-sets correspond to ideals in the poset whose elements are in $\mathbb{F}_2^n$ and with partial order $\mathbf{x} \preceq \mathbf{y}$ if $\mathbf{x}$ can be obtained from $\mathbf{y}$ by a series of ρ_i and σ_{ij} operations. It turns out that there are not too many such ideals, and they may be computed efficiently.

Our method for producing the ideals is not new, but since the main references are unpublished, we describe them briefly here. In Section 4.12.2 of [15], Ruskey describes a procedure GenIdeal for listing the ideals in a poset $\mathcal{P}$. Let $\downarrow\mathbf{x}$ denote all the elements $\preceq \mathbf{x}$, and $\uparrow\mathbf{x}$ denote all the elements $\succeq \mathbf{x}$.

procedure GenIdeal($\mathcal{Q}$: Poset, I: Ideal)
local $\mathbf{x}$: PosetElement
begin
  if $\mathcal{Q} = \phi$ **then** PrintIt(I);
  else
    $\mathbf{x}$:= some element in $\mathcal{Q}$;
    GenIdeal($\mathcal{Q} - \downarrow\mathbf{x},\ I \cup \downarrow\mathbf{x}$);
    GenIdeal($\mathcal{Q} - \uparrow\mathbf{x},\ I$);
end

The idea is to start with I empty, and $\mathcal{Q} = \mathcal{P}$. Then for each $\mathbf{x}$, an ideal either contains $\mathbf{x}$, in which case it will be found by the first call to GenIdeal, or it does not, in which case the second call will find it.

Finding $\uparrow\mathbf{x}$ and $\downarrow\mathbf{x}$ may be done efficiently if we precompute two $|\mathcal{P}| \times |\mathcal{P}|$ incidence matrices representing these sets for each element of $\mathcal{P}$. This precomputation takes time $O(|\mathcal{P}|^2)$, and then the time per ideal is $O(|\mathcal{P}|)$. This is independent of the choice of $\mathbf{x}$. Squire (see [15] for details) realized that, by picking $\mathbf{x}$ to be the middle element of $\mathcal{Q}$ in some linear extension, the time per ideal can be shown to be $O(\lg|\mathcal{P}|)$.

We are only interested in down-sets that are right-shifted and also are of fairly small size. The feasibility of our computations involves both issues. In particular, within GenIdeal we may restrict to $\mathbf{x} \in \mathbb{F}_2^n$ with $\mathrm{Size}(\downarrow\mathbf{x})$ no more than the target size of the region we are looking for. If we were using GenIdeal with the poset whose ideals correspond to down-sets of size 64 in $\mathbb{F}_2^{63}$, there would be 83278001 such $\mathbf{x}$ to consider. However, for our situation with right-shifted down-sets, there are only 257 such $\mathbf{x}$ and the problem becomes quite manageable. Furthermore, instead of stopping when $\mathcal{Q}$ is empty, we stop when I is at or above the desired size.

TABLE II
NUMBER OF RIGHT-SHIFTED DOWN-SETS

size	number
2	1
3	1
4	2
5	2
6	3
7	4
8	6
9	7
10	10

size	number
11	13
12	18
13	23
14	31
15	40
16	54
17	69
18	91
19	118
20	155

size	number
21	199
22	260
23	334
24	433
32	3140
48	130979
64	4384627

Table II gives the number of right-shifted down-sets of different sizes. The computation for size 32 sets took just over a second on one processor of an HP Superdome. Size 64 sets took 23 minutes. Let $R_{s,n}$ refer to an optimal region of size s in $\mathbb{F}_2^n$. Tables III and IV list $R_{2^t,n}$ for all $t \le 6$ and all $n < 2^t$.

Several features of Tables III and IV require explanation. First we identify the binary expansion $x = \sum_{i<n} 2^i x_{n-i}$ with the vector $\mathbf{x} = (x_1,\ldots,x_n)$. Second, for each optimal right-shifted down-set $R_{2^t,n}$ we have listed a minimal set of generators. For example $\langle 2^4 - 1\rangle$ corresponds to the 4-dimensional cube while $\langle 2^{14}\rangle$, as a subset of $\mathbb{F}_2^{15}$, corresponds to the 15-dimensional 1-sphere.

For each region p_{cross} indicates the crossover value for p at which point that region performs better than any preceding

TABLE III
OPTIMAL RIGHT-SHIFTED DOWN-SETS $R_{2^t,n}$ ($t \le 5$).

t	n	p_{cross}	distance distribution function	$R_{2^t,n}$
1	1	0	$2(1+x)$	$\langle 1\rangle$
2	2	0	$4(1+x)^2$	$\langle 2^2-1\rangle$
3	3	0	$8(1+x)^3$	$\langle 2^3-1\rangle$
4	4	0	$16(1+x)^4$	$\langle 2^4-1\rangle$
	12	0.4560	$16+36x+144x^2+60x^3$	$\langle 2^{11}, 2^3+1\rangle$
	"	"	"	$\langle 2^{11}, 3\cdot 2\rangle$
	13	0.3929	$16+34x+162x^2+44x^3$	$\langle 2^{12}, 2^2+1\rangle$
	14	0.3333	$16+32x+184x^2+24x^3$	$\langle 2^{13}, 2+1\rangle$
	15	0.2826	$16+30x+210x^2$	$\langle 2^{14}\rangle$
5	5	0	$32(1+x)^5$	$\langle 2^5-1\rangle$
	12	0.4882	$32+100x+368x^2+380x^3+144x^4$	$\langle 2^{11}+1, 2^9+2\rangle$
	"	"	"	$\langle 2^{11}, 2^{10}+2\rangle$
	13	0.4492	$32+98x+378x^2+396x^3+120x^4$	$\langle 2^{12}+1, 2^7+2\rangle$
	14	0.3929	$2(1+x)(16+34x+162x^2+44x^3)$	$\langle 2^{13}+1, 2^3+3\rangle$
	15	0.3333	$2(1+x)(16+32x+184x^2+24x^3)$	$\langle 2^{14}+1, 7\rangle$
	16	0.2826	$2(1+x)(16+30x+210x^2)$	$\langle 2^{15}+1\rangle$
	19	0.3333	$32+86x+498x^2+408x^3$	$\langle 2^{18}, 2^{12}+1\rangle$
	20	0.2799	$32+84x+512x^2+396x^3$	$\langle 2^{19}, 2^{11}+1\rangle$
	21	0.2724	$32+82x+530x^2+380x^3$	$\langle 2^{20}, 2^{10}+1\rangle$
	22	0.2627	$32+80x+552x^2+360x^3$	$\langle 2^{21}, 2^9+1\rangle$
	23	0.2515	$32+78x+578x^2+336x^3$	$\langle 2^{22}, 2^8+1\rangle$
	24	0.2390	$32+76x+608x^2+308x^3$	$\langle 2^{23}, 2^7+1\rangle$
	25	0.2259	$32+74x+642x^2+276x^3$	$\langle 2^{24}, 2^6+1\rangle$
	26	0.2126	$32+72x+680x^2+240x^3$	$\langle 2^{25}, 2^5+1\rangle$
	27	0.1992	$32+70x+722x^2+200x^3$	$\langle 2^{26}, 2^4+1\rangle$
	28	0.1864	$32+68x+768x^2+156x^3$	$\langle 2^{27}, 2^3+1\rangle$
	"	"	"	$\langle 2^{27}, 3\cdot 2\rangle$
	29	0.1741	$32+66x+818x^2+108x^3$	$\langle 2^{28}, 2^2+1\rangle$
	30	0.1626	$32+64x+872x^2+56x^3$	$\langle 2^{29}, 2+1\rangle$
	31	0.1518	$32+62x+930x^2$	$\langle 2^{30}\rangle$

entry in the table. For example, the 4-dimensional cube $\langle 2^4-1\rangle$ is optimal for all $p \in (0, 0.5)$ if $4 \le n \le 11$ but is only optimal for $p \in (0, 0.4560)$ if $n = 12$. For $(t, n) = (4, 13)$, the 4-dimensional cube is optimal for $p \in (0,\ 0.3929)$ while the right-shifted down-set $\langle 2^{12}, 2^2+1\rangle$ is optimal for $p \in (0.3929,\ 0.5)$.

There are several specific (t, n) for which more than two nonisomorphic right-shifted down-sets are optimal. In several cases the nonisomorphic optimal right-shifted down-sets have the same distance distribution. (The two nonisomorphic regions $R_{2^4,12}$ were originally found by Kündgen [14, pg. 160: Table 1].) In other cases different regions are optimal for different values of p. (Such cases are highlighted with a box $\boxed{\cdot}$.) For example, with $(t, n) = (5, 19)$, the 5-dimensional cube $\langle 2^5-1\rangle$ is optimal for $p \in (0,\ 0.2826)$, $\langle 2^{15}+1\rangle$ is optimal on $(0.2826,\ 0.3333)$, while $\langle 2^{18}, 2^{12}+1\rangle$ is optimal on $(0.3333,\ 0.5)$. Somewhat similar situations involve $t = 6$ and $n \in \{19, 28, 29, 35, 36, 37, 38, 58, 59\}$.[3] For $t \le 6$ and for any n, there are at most three different optimal regions.

Some of the optimal regions $R_{64,n}$ are better than those for any known hash function. Table V gives the best known regions for each k, and their generators. If any new regions were shown to tile their cube, we would have an improvement to Figure 1.

[3]For $n = 28$, the three regions are $\langle 2^6-1\rangle$ on $(0, 0.199)$, $\langle 2^{27}+1, 2^5+3\rangle$ on $(0.199, 0.25)$ and $\langle 2^{27}+1, 2^9+2\rangle$ on $(0.25, 0.5)$.

TABLE V
OPTIMAL RIGHT-SHIFTED DOWN-SETS $R_{64,n}$ BEATING KNOWN CODES. (THERE ARE NO SUCH DOWN-SETS $R_{2^t,n}$ FOR $t \le 5$.)

k	n	cross	$R_{64,n}$
6	12	0.487	$\langle 2^{11}, 2^{10}+2^5, 3\cdot 2^8\rangle$
7	13	0.470	$\langle 2^{12}, 2^{10}+2^4, 3\cdot 2^8\rangle$
8	14	0.439	$\langle 2^{13}+2^2, 2^{13}+3, 2^3+2^2+1\rangle$
9	15	0.391	$\langle 2^{14}+3, 2^{10}+2^2\rangle$
16	22	0.244	$\langle 2^{21}+2\rangle$
17	23	0.242	$\langle 2^{22}+1, 2^{19}+2\rangle$
18	24	0.238	$\langle 2^{23}+1, 2^{17}+2\rangle$
19	25	0.231	$\langle 2^{24}+1, 2^{15}+2\rangle$
20	26	0.222	$\langle 2^{25}+1, 2^{13}+2\rangle$
21	27	0.212	$\langle 2^{26}+1, 2^{11}+2\rangle$

REFERENCES

[1] R. Ahlswede and G. O. H. Katona. Contributions to the geometry of Hamming spaces. *Discrete Math.*, 17:1–22, 1977.

[2] A. E. Ashikhmin, G. D. Cohen, M. Krivelevich, and S. N. Litsyn. Bounds on distance distributions in codes

TABLE IV
OPTIMAL RIGHT-SHIFTED DOWN-SETS $R_{64,n}$ ($t = 6$)

n	p_{cross}	distance distribution function	$R_{64,n}$
6	0	$64 + 384x + 960x^2 + 1280x^3 + 960x^4 + 384x + 64$	$\langle 2^6 - 1\rangle$
12	0.487	$64 + 228x + 1092x^2 + 1020x^3 + 1692x^4$	$\langle 2^{11}, 2^{10} + 2^5, 3 \cdot 2^8\rangle$
13	0.470	$64 + 226x + 1086x^2 + 1100x^3 + 1620x^4$	$\langle 2^{12}, 2^{10} + 2^4, 3 \cdot 2^8\rangle$
14	0.439	$64 + 250x + 1002x^2 + 1508x^3 + 1032x^4 + 240x^5$	$\langle 2^{13} + 2^2, 2^{13} + 3, 2^3 + 5\rangle$
15	0.391	$64 + 248x + 1024x^2 + 1592x^3 + 992x^4 + 176x^5$	$\langle 2^{14} + 3, 2^{10} + 2^2\rangle$
16	0.333	$4(1 + x)^2(16 + 32x + 184x^2 + 24x^3)$	$\langle 2^{15} + 3, 2^4 - 1\rangle$
17	0.283	$4(1 + x)^2(16 + 30x + 210x^2)$	$\langle 2^{16} + 3\rangle$
19	0.36	$64 + 232x + 1184x^2 + 1784x^3 + 832x^4$	$\langle 2^{18} + 2, 2^{10} + 3\rangle$
20	0.277	$64 + 224x + 1240x^2 + 1752x^3 + 816x^4$	$\langle 2^{19} + 2, 2^7 + 3\rangle$
21	0.263	$64 + 216x + 1320x^2 + 1704x^3 + 792x^4$	$\langle 2^{20} + 2, 2^4 + 3\rangle$
22	0.244	$64 + 208x + 1424x^2 + 1640x^3 + 760x^4$	$\langle 2^{21} + 2\rangle$
23	0.242	$64 + 206x + 1426x^2 + 1680x^3 + 720x^4$	$\langle 2^{22} + 1, 2^{19} + 2\rangle$
24	0.238	$64 + 204x + 1440x^2 + 1716x^3 + 672x^4$	$\langle 2^{23} + 1, 2^{17} + 2\rangle$
25	0.231	$64 + 202x + 1466x^2 + 1748x^3 + 616x^4$	$\langle 2^{24} + 1, 2^{15} + 2\rangle$
26	0.222	$64 + 200x + 1504x^2 + 1776x^3 + 552x^4$	$\langle 2^{25} + 1, 2^{13} + 2\rangle$
27	0.212	$64 + 198x + 1554x^2 + 1800x^3 + 480x^4$	$\langle 2^{26} + 1, 2^{11} + 2\rangle$
28	0.199	$2(1 + x)(32 + 70x + 722x^2 + 200x^3)$	$\langle 2^{27} + 1, 2^5 + 3\rangle$
"	0.25	$64 + 196x + 1616x^2 + 1820x^3 + 400x^4$	$\langle 2^{27} + 1, 2^9 + 2\rangle$
29	0.186	$2(1 + x)(32 + 68x + 768x^2 + 156x^3)$	$\langle 2^{28} + 1, 2^4 + 3\rangle$
"	"	"	$\langle 2^{28} + 1, 3 \cdot 2^2 + 1\rangle$
"	0.333	$64 + 194x + 1690x^2 + 1836x^3 + 312x^4$	$\langle 2^{28} + 1, 2^7 + 2\rangle$
30	0.174	$2(1 + x)(32 + 66x + 818x^2 + 108x^3)$	$\langle 2^{29} + 1, 2^3 + 3\rangle$
31	0.163	$2(1 + x)(32 + 64x + 872x^2 + 56x^3)$	$\langle 2^{30} + 1, 7\rangle$
32	0.152	$2(1 + x)(32 + 62x + 930x^2)$	$\langle 2^{31} + 1\rangle$
35	0.1538	$64 + 182x + 2002x^2 + 1848x^3$	$\langle 2^{34}, 2^{28} + 1\rangle$
36	0.1537	$64 + 180x + 2016x^2 + 1836x^3$	$\langle 2^{35}, 2^{27} + 1\rangle$
37	0.153	$64 + 178x + 2034x^2 + 1820x^3$	$\langle 2^{36}, 2^{26} + 1\rangle$
38	0.152	$64 + 176x + 2056x^2 + 1800x^3$	$\langle 2^{37}, 2^{25} + 1\rangle$
39	0.151	$64 + 174x + 2082x^2 + 1776x^3$	$\langle 2^{38}, 2^{24} + 1\rangle$
40	0.150	$64 + 172x + 2112x^2 + 1748x^3$	$\langle 2^{39}, 2^{23} + 1\rangle$
41	0.148	$64 + 170x + 2146x^2 + 1716x^3$	$\langle 2^{40}, 2^{22} + 1\rangle$
42	0.146	$64 + 168x + 2184x^2 + 1680x^3$	$\langle 2^{41}, 2^{21} + 1\rangle$
43	0.144	$64 + 166x + 2226x^2 + 1640x^3$	$\langle 2^{42}, 2^{20} + 1\rangle$
44	0.141	$64 + 164x + 2272x^2 + 1596x^3$	$\langle 2^{43}, 2^{19} + 1\rangle$
45	0.139	$64 + 162x + 2322x^2 + 1548x^3$	$\langle 2^{44}, 2^{18} + 1\rangle$
46	0.136	$64 + 160x + 2376x^2 + 1496x^3$	$\langle 2^{45}, 2^{17} + 1\rangle$
47	0.133	$64 + 158x + 2434x^2 + 1440x^3$	$\langle 2^{46}, 2^{16} + 1\rangle$
48	0.130	$64 + 156x + 2496x^2 + 1380x^3$	$\langle 2^{47}, 2^{15} + 1\rangle$
49	0.127	$64 + 154x + 2562x^2 + 1316x^3$	$\langle 2^{48}, 2^{14} + 1\rangle$
50	0.123	$64 + 152x + 2632x^2 + 1248x^3$	$\langle 2^{49}, 2^{13} + 1\rangle$
51	0.120	$64 + 150x + 2706x^2 + 1176x^3$	$\langle 2^{50}, 2^{12} + 1\rangle$
52	0.117	$64 + 148x + 2784x^2 + 1100x^3$	$\langle 2^{51}, 2^{11} + 1\rangle$
53	0.114	$64 + 146x + 2866x^2 + 1020x^3$	$\langle 2^{52}, 2^{10} + 1\rangle$
54	0.110	$64 + 144x + 2952x^2 + 936x^3$	$\langle 2^{53}, 2^9 + 1\rangle$
55	0.107	$64 + 142x + 3042x^2 + 848x^3$	$\langle 2^{54}, 2^8 + 1\rangle$
56	0.104	$64 + 140x + 3136x^2 + 756x^3$	$\langle 2^{55}, 2^7 + 1\rangle$
57	0.101	$64 + 138x + 3234x^2 + 660x^3$	$\langle 2^{56}, 2^6 + 1\rangle$
58	0.0978	$64 + 138x + 3330x^2 + 452x^3 + 112x^4$	$\langle 2^{57}, 2^3 + 1, 3 \cdot 2\rangle$
"	0.1047	$64 + 136x + 3336x^2 + 560x^3$	$\langle 2^{57}, 2^5 + 1\rangle$
59	0.0946	$64 + 136x + 3440x^2 + 344x^3 + 112x^4$	$\langle 2^{58}, 7\rangle$
"	0.1179	$64 + 134x + 3442x^2 + 456x^3$	$\langle 2^{59}, 2^4 + 1\rangle$
60	0.0920	$64 + 132x + 3552x^2 + 348x^3$	$\langle 2^{59}, 2^3 + 1\rangle$
"	"	"	$\langle 2^{59}, 3 \cdot 2\rangle$
61	0.0891	$64 + 130x + 3666x^2 + 236x^3$	$\langle 2^{60}, 2^2 + 1\rangle$
62	0.0864	$64 + 128x + 3784x^2 + 120x^3$	$\langle 2^{61}, 2 + 1\rangle$
63	0.0838	$64 + 126x + 3906x^2$	$\langle 2^{62}\rangle$

of known size. *IEEE Trans. Info. Theory*, 51:250–258, 2005.

[3] E. Berkovich. Method of and system for searching a data dictionary with fault tolerant indexing. United States Patent: 7,168,025, January 2007. Filed: 10/11/2001 (Appl. No. 09/973,792).

[4] S. Y. Berkovich and E. El-Qawasmeh. Reversing the error-correction scheme for a fault-tolerant indexing. *The Computer Journal*, 43(1):54–64, 1999.

[5] W. Bosma, J. Cannon, and C. Playoust. The Magma algebra system I: The user language. *J. Symb. Comp.*, 24:235–269, 1997. Software version: 2.13-7.

[6] A. Broder. Filtering near-duplicate documents. In *Proc. FUN*, 1998.

[7] D. Dolev, Y. Harari, N. Linial, N. Nisan, and M. Parnas. Neighborhood preserving hashing and approximate queries. In *SODA '94: Proceedings of the fifth annual ACM-SIAM Symposium on Discrete Algorithms*, pages 251–259, 1994.

[8] R. G. Gallager. *Low-density parity-check codes*. MIT Press, Cambridge, MA, 1963.

[9] A. Gionis, P. Indyk, and R. Motwani. Similarity search in high dimensions via hashing. In *Proceedings of the 25th VLDB Conference*, 1999.

[10] L. H. Harper. Optimal assignment of numbers to vertices. *J. Soc. Ind. Appl. Math.*, 12:131–135, 1964.

[11] F. Jaeger, A. Khelladi, and M. Mollard. On shorted cocycle covers of graphs. *J. Combin. Theory Ser. B*, 39:153–163, 1985.

[12] R. M. Karp, O. Waarts, and G. Zweig. The bit vector

intersection problem. In *Proc. 36th Annual Symposium on Foundations of Computer Science*, 1995.

[13] T. Kløve and V. I. Korzhik. *Error Detecting Codes: General Theory and Their Application in Feedback Communication Systems*. Kluwer Academic Publisheres, 1995.

[14] André Kündgen. Minimum average distance subsets in the Hamming cube. *Discrete Math.*, 249:149–165, 2002.

[15] Frank Ruskey. Combinatorial generation. online draft, 2003. available from `http://www.1stworks.-com/ref/RuskeyCombGen.pdf`.

[16] L. Weng. Hashing system utilizing error correction coding techniques. United States Patent: 7,085,988, August 2006. Filed: 3/20/2003 (Appl. No. 10/393,096).