Electronic Colloquium on Computational Complexity, Report No. 104 (2005)

On the Robust Testability of Product of Codes

Don Coppersmith[*] Atri Rudra[†]

September 16, 2005

Abstract

Ben-Sasson and Sudan in [4] asked if the following test is robust for the tensor product of a code with another code– pick a row (or column) at random and check if the received word restricted to the picked row (or column) belongs to the corresponding code. Valiant showed that for general *linear* codes, the test is not robust [12]. However the question remained open for the tensor product of a code with *itself*. We resolve this question in the negative. We also show a similar result for *non-linear* codes.

1 Introduction

Locally testable codes (or LTCs in short) are error correcting codes which have the following property– given oracle access to a received word, there is an efficient tester which makes very few oracle queries and ascertains whether the received word is a codeword or is far from being one. Recently, there has been heightened activity in study of these objects [7, 3, 9, 4, 5, 6]. LTCs are an important field of study both for their own sake and their connections with probabilistic checkable proofs (PCPs) [2, 1].

A code is said to be robustly testable if it is a LTC and its tester has an additional desirable property called *robustness*. Informally, a test is robust if for every received word which is far from being a codeword, the tester not only rejects the codeword with high probability but also with high probability the tester's local view of the received word is far from any accepting view. Robust LTCs have been useful in the construction of PCPs [3, 5] and in the construction of tolerant LTCs [8].

Given two codes C_1 and C_2 of block length n_1 and n_2 respectively, their tensor product (denoted by $C_1 \otimes C_2$) consists of $n_2 \times n_1$ matrices with the property that every row is a codeword in C_1 and every column is a codeword in C_2. Ben-Sasson and Sudan [4] using ideas from Raz and Safra [11], show that the code obtained by two or more applications of the tensor product to a code is robustly testable.

A natural test for $C_1 \otimes C_2$ is to uniformly at random pick a row (or column) and check if the received word restricted to that row (or column) is a codeword in C_1 (or C_2). This test is indeed robust in a couple of special cases– for example, when both C_1 and C_2 are Reed-Solomon codes (this is the bivariate-polynomial testing of Polishchuk and Spielman [10]) and when both C_1 and C_2 are themselves tensor product of a code [4].

[*] IBM TJ Watson Research Center. *Email*: dcopper@us.ibm.com

[†] University of Washington. *Email*: atri@cs.washington.edu

ISSN 1433-8092

However, the question whether the natural tester described in the above paragraph is a robust tester for $C_1 \otimes C_2$ was left open by Ben-Sasson and Sudan [4]. Recently, Valiant answered this question in negative by showing that there are linear codes C_1 and C_2 such that $C_1 \otimes C_2$ is not robustly testable [12]. Basically, Valiant constructs linear codes C_1, C_2 and a matrix v such that every row (and column) of v is "close" to some codeword in C_1 (and C_2) while v is "far" from every codeword in $C_1 \otimes C_2$ (where close and far are in the sense of hamming distance).

However, Valiant's construction *does not* work when C_1 and C_2 are the same code. In this note, we show a reduction from Valiant's construction to exhibit a code C such that C^2 (we will use this shorthand for $C \otimes C$ for the rest of this paper) is not robustly testable. We also construct a *non-linear* code C such that C^2 is not robustly testable. An interesting feature of the latter construction is that the received word v has the property that *every* row is a codeword in C and all but one column in a codeword in C– the "errant" column differs from a codeword in C in just *one position*. On the other hand v is from from every codeword in C^2.

2 Preliminaries

A code C over an alphabet Σ is a mapping from k symbols to n symbols. The *distance* of C (denoted by the parameter d) is the minimum hamming distance between any two codewords (where the hamming distance between two vectors $u, v \in \Sigma^n$, denoted by $\Delta(u, v)$, is the number of places they differ). The parameter n is called the *block length* of the code. Given any vector $v \in \Sigma^n$, its (relative) distance from C is given by $\delta_C(v) = \min_{w \in C} \Delta(w, v)/n$. A vector $v \in \Sigma^n$ is said to be ϵ-close to C if $\delta_C(v) \leq \epsilon$, otherwise it is ϵ-far from C.

A local tester with query complexity q is a probabilistic machine that given an oracle access to a string $v \in \Sigma^n$, makes q queries to the oracle v and accepts or rejects the string. A code C is said to be (ϵ, q)-locally testable if there exists a tester T with query complexity q such that for every codeword v, T accepts v with probability 1 and rejects every ϵ-far word v with probability at least 2/3.

A local tester T has two inputs: an oracle for the received word v and a random string s. Depending on s, T generates q query positions $i_1, \cdots, i_q$, fixes a circuit C_s and then accepts if $C_s(v_f(s)) = 1$ where $v_f(s) = \langle v_{i_1}, \cdots, v_{i_q} \rangle$. The robustness of T on inputs v and s, denoted by $\rho^T(v, s)$, is defined to be the minimum, over all strings y such that $C_s(y) = 1$, of $\Delta(v_f(s), y)/n$. The expected robustness of T on v is the expected value of $\rho^T(v, s)$ over the random choices of s and would be denoted by $\rho^T(v)$. A local tester T is said to be c-robust for C if for every $v \in C$, the tester accepts with probability 1, and for every $v \in \Sigma^n$, $\delta_C(v) \leq c \cdot \rho^T(v)$. C is said to be robustly testable if it has a $\Omega(1)$-robust tester.

Given two codes C_1 and C_2 with parameters k_1, n_1, d_1 and k_2, n_2, d_2 their tensor product, denoted by $C_1 \otimes C_2$, consists of $n_2 \times n_1$ matrices such that every row of the matrix is a codeword in C_1 and every column is a codeword in C_2. It is well known that $C_3 = C_1 \otimes C_2$ has the parameters $n_3 = n_1 n_2$, $k_3 = k_1 k_2$ and $d_3 = d_1 d_2$. A natural tester $T_{C_1 \otimes C_2}$ for such a code is the following– flip a coin; if it is heads check if a random row is a codeword in C_1; if it is tails, check if a random column is a codeword in C_2.

Asking whether $T_{C_1 \otimes C_2}$ is a robust tester has the following nice interpretation. The q queries $i_1, \cdots, i_q$ are either rows or columns of the received word v. Let the row or column corresponding to the random seed s be denoted by v^s. Then the robustness of $T_{C_1 \otimes C_2}$ on inputs (v, s), $\rho^{T_{C_1 \otimes C_2}}(v, s)$ is just $\delta_{C_1}(v^s)$ when i_s corresponds to a row and $\delta_{C_2}(v^s)$ when i_s corresponds to a column. Therefore the expected robustness of $T_{C_1\otimes C_2}$ on v is the average of the following two quantities: the average relative distance of the rows of v from C_1 and the average relative distance of the columns of v from C_2.

In particular, if $T_{C_1\otimes C_2}$ is $\Omega(1)$-robust then it implies that for every received word v such that all rows (and columns) of v are $o(1)$-close to C_1 (and C_2), v is $o(1)$-close to $C_1 \otimes C_2$. Ben-Sasson and Sudan [4] asked the question if the tester $T_{C_1\otimes C_2}$ is $\Omega(1)$-robust. Valiant [12] answered the question in the negative.

Theorem 1 *([12]) There exist linear codes C_1 and C_2 with parameters $n_1, k_1, d_1 = n_1/10$ and $n_2 = n_1^2, k_2, d_2 = n_2/10$ and a $n_2 \times n_1$ received word v such that every row of v is a codeword in C_1 and every column of v is $o(1)$-close to C_2 but v is $\Omega(1)$-far from $C_1 \times C_2$.*

Note that in the above construction, $n_2 \neq n_1$ and in particular C_1 and C_2 are not the same code. In Section 3, we show how to construct a linear code C from C_1 and C_2 such that C has linear distance and there exist a received word v' such that every row (and column) of v' is $o(1)$-close to C but v' is $\Omega(1)$-far from C^2. In Section 4 we construct a *non-linear* code C with similar properties.

3 Reduction from the construction of Valiant

In this section, we prove the following result.

Theorem 2 *Let $C_1 \neq C_2$ be linear codes with parameters $n_1, k_1, d_1 = \Omega(n_1)$ and $n_2, k_2, d_2 = \Omega(n_2)$ (with $n_2 > n_1$) and let v be a $n_2 \times n_1$ matrix such that every row (and column) of v is $o(1)$-close to C_1 (and C_2) but v is $\Omega(1)$-far from $C_1 \otimes C_2$. Then there exists a linear code C with parameters $n, k, d = \Omega(n)$ and a received word v' such that such that every row and column of v' is $o(1)$-close to C but v' is $\Omega(1)$-far from C^2.*

Proof: We will first assume that n_1 divides n_2 and let $m = \frac{n_2}{n_1}$. For any $x \in \Sigma^{k_1}$ and $y \in \Sigma^{k_2}$, let

$$C(\langle x, y\rangle) = \langle (C_1(x))^m, C_2(y)\rangle$$

Thus, $k = k_1 + k_2$ and $n = mn_1 + n_2$. Also as $d_1 = \Omega(n_1)$ and $d_2 = \Omega(n_2)$, $d = \Omega(n)$.

We now construct the $n \times n$ matrix v' from v. The lower left $n_2 \times mn_1$ sub-matrix of v' contains the matrix v^m where v^m is the horizontal concatenation of m copies of v (which is a $n_2 \times n_1$ matrix). Every other entry in v' is 0. See figure 1 for an example with $m = 2$.

Let w be the codeword in $C_1 \otimes C_2$ closest to v and construct w' in the same manner as v' was constructed from v. We first claim that w' is the codeword in[1] C^2 closest to v'. For the sake of contradiction, assume that there is some other codeword w'' in C^2 such that $\Delta(v', w'') < \Delta(v', w')$. For any $2n' \times 2n'$ matrix u let u_{lb} denote the lower left $n' \times n'$ sub-matrix of u. Note that by definition of C, $w''_{lb} = x^m$ where $x \in C_1 \otimes C_2$. Further, as v' (necessarily) has 0 everywhere other than v'_{lb} and $\Delta(v', w'') < \Delta(v', w')$, it holds that $\Delta(v, w) > \Delta(v, x)$ which contradicts the definition of w.

Finally, it is easy to see that

$$\delta_{C^2}(v') = \Delta(v', w')/n^2 = \Delta(v, w)m/(mn_1 + n_2)^2 = \Delta(v, w)/(4n_1n_2) = \delta_{C_1\otimes C_2}(v)/4$$

[1] Note that $w' \in C^2$ as the all zeros vector is a codeword in both C_1 and C_2 and $v \in C_1 \otimes C_2$.

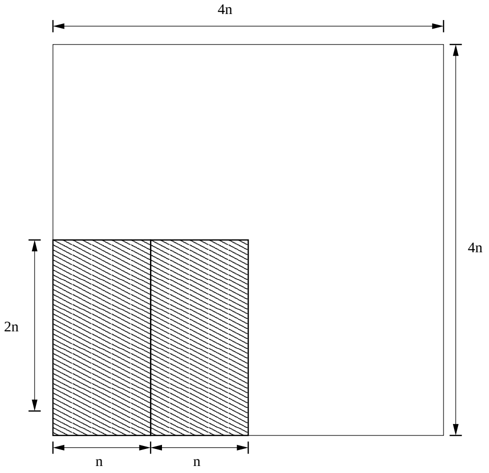

Figure 1: The construction of the new received word v' from v for the case when $n_1 = a$, $n_2 = 2a$ and $m = 2$. The shaded boxes represent v and the unshaded regions has all 0s.

and if for every row (and column), the (relative) distance of v restricted to that row (or column) from C_1 (C_2) is at most α then for every row and column, the relative distance of v' restricted to that every row and column from C is at most $\alpha/2$.

This completes the proof for the case when n_1 divides n_2. For the case when n_1 does not divide n_2 a similar construction works if one defines C in the following manner (for any $x \in \Sigma^{k_1}$ and $x_2 \in \Sigma^{k_2}$)

$$C(\langle x, y\rangle) = \langle (C_1(x))^{\ell/n_1}, (C_2(y))^{\ell/n_2}\rangle$$

where $\ell = \mathrm{lcm}(n_1, n_2)$. The received word v' in this case would have its lower left $\ell \times \ell$ matrix as $v^{(\ell/n_1, \ell/n_2)}$ (where $v^{(m_1,m_2)}$ is the matrix obtained by vertically concatenating m_2 copies of v^{m_1}) and it has 0s everywhere else. ■

Theorem 1 and 2 imply the following result.

Corollary 1 *There exist a linear code C with linear distance such that the tester T_{C^2} is not $\Omega(1)$-robust for C^2.*

4 Product of non-linear codes are not robustly testable

In this section we present a *non-linear* code C such that T_{C^2} is not $\Omega(1)$-robust.

We will use Fibonacci numbers in our construction. Let F_m denote the mth Fibonacci number, with $F_0 = 0$, $F_1 = 1$, and $F_{j+2} = F_j + F_{j+1}$. We will also use the following identity

$$F_j^2 - F_{j+1}F_{j-1} = -(-1)^j \tag{1}$$

and the fact that $F_{j+1}/F_j \approx \phi = (1 + \sqrt{5})/2$.

Theorem 3 *There exists a non-linear code C with parameter $n, k, d = \Omega(n)$ and a received word v such that such that every row (and column) of v is $o(1)$-close to C but v is $\Omega(1)$-far from C^2.*

Proof: Fix an even integer m. Set $n = F_{m+1}$. Define a nonlinear code C, over $\{0,1\}$ as follows. The codewords of C are those words of length n whose Hamming weight is in $\{0, F_{m-1}, F_m\}$. Note that the distance of C, $d \geq F_{m-2} \approx n/\phi^3$.

Construct an $n \times n$ matrix v, in such a way that F_m rows have weight F_m and F_{m-1} rows have weight 0. Further, $n-1$ columns have weight F_{m-1} and the remaining column has weight $F_{m-1}-1$. Such a v can be constructed by entering the 1-elements in a round-robin fashion by rows, so that the columns are filled as evenly as possible. See Section 4.1 for an example of the construction with $m = 8$.

Check that the row and column sums are consistent:

$$F_m(F_m) + F_{m-1}(0) = (F_{m+1} - 1)(F_{m-1}) + 1(F_{m-1} - 1) = F_{m+1}F_{m-1} - 1$$

which is consistent with the identity from (1).

With one exception, each row and each column is a codeword in C. The one exceptional column is at distance 1 from a codeword in C.

Consider a codeword w of $C \times C$ at minimal distance from v. For some $\alpha, \beta, \gamma, \delta$, we know that w has $F_m - \alpha$ rows of weight F_m, and $F_{m-1} - \beta$ rows of weight 0, and $\alpha + \beta$ rows of weight F_{m-1}; and that w has γ columns of weight F_m, and δ columns of weight 0, and $F_{m+1} - (\gamma + \delta)$ columns of weight $Fm - 1$. Set $a = \alpha + \gamma$ and $b = \beta + \delta$. We have $\alpha + \beta \geq 0$, $\gamma \geq 0$, and $\delta \geq 0$, so that $a + b \geq 0$. Ignoring for now the exceptional column in v, and comparing v and w, the number of lines (rows and columns) which have changed their hamming weight is then $|\alpha| + |\beta| + \gamma + \delta \geq a + b$. Of these, $|\beta| + \delta \geq b$ have changed by $\pm F_{m-1}$, and $|\alpha| + \gamma \geq a$ have changed by $\pm(F_m - F_{m-1}) = \pm F_{m-2}$. Each changed matrix element is counted in two changed lines, so accounting for this (and for the odd column in v), the Hamming distance between v and w is at least $(aF_{m-2} + bF_{m-1} - 1)/2$.

In order to get the row sums and column sums to agree, we must have

$$\begin{aligned} (F_m - \alpha)(F_m) \quad &+ (F_{m-1} - \beta)(0) + (\alpha + \beta)(F_{m-1}) \\ &= (\gamma)(F_m) + (\delta)(0) + (F_{m+1} - \gamma - \delta)(F_{m-1}) \end{aligned}$$

$$(\alpha + \gamma)(F_m - F_{m-1}) - (\beta + \delta)(F_{m-1}) = F_mF_m - F_{m+1}F_{m-1} = -1$$

$$aF_{m-2} - bF_{m-1} = -1 \tag{2}$$

The smallest solution to (2), in nonnegative integers, is $a = F_m$, $b = F_{m-1}$:

$$F_mF_{m-2} - F_{m-1}F_{m-1} = -1$$

So the Hamming distance between v and w is at least

$$(aF_{m-2} + bFm - 1 - 1)/2 = (F_mF_{m-2} + F_{m-1}F_{m-1} - 1)/2 = F_mF_{m-2}$$

Recalling that $n = F_{m+1}$, and that $F_{j+1}/F_j \approx \phi = (1 + \sqrt{5})/2$, we find that the Hamming distance between w and v is about n^2/ϕ^4. ■

Remark 1 *We note that in the above construction, the received word v has the property that* every *row is a codeword in C and* all but one *column of v is a codeword in C. Further, the "errant" column differs from a codeword in C in* exactly one position. *Note that in some sense this is the best one can hope for while proving such negative results. By comparison, the construction of Corollary 1 does creates a v such that every row is a codeword in C. However,* at least half *of the columns of v differ from every codeword in C in some non-constant number of positions.*

4.1 A worked example

In this subsection, we will give an example of the construction used in the proof of Theorem 3. $m = 8$, $F_{m-1} = F_7 = 13$, $F_m = F_8 = 21$, $F_{m+1} = F_9 = 34$. $n = 34$.

Valid codewords in C have weights 0, 13, 21. v has 21 rows of weight 21 and 13 rows of weight 0, for a total weight of $21 \times 21 + 13 \times 0 = 441 + 0 = 441$. v has 33 columns of weight 13 and one of weight 12, agreeing with total weight $33 \times 13 + 1 \times 12 = 429 + 12 = 441$. To bring all rows and columns into C, we need to change $|\alpha| + |\beta| + \gamma + \delta \geq a + b$ lines, with

$$8a - 13b = -1$$

The best solution is $a = 21, b = 13$, giving

$$8 \times 21 - 13 \times 13 = 168 - 169 = -1$$

The Hamming distance between v and w is

$$(aF_6 + bF_7 - 1)/2 = (21 \times 8 + 13 \times 13 - 1)/2 = (168 + 169 - 1)/2 = 168$$

and this compares to $n^2 = 34^2$ by the fraction

$$168/34^2 = 0.145328\ldots \approx 1/\phi^4 = 0.145898\ldots$$

We could find such w by taking $\beta = \gamma = 0$, $\alpha = a = 21$, $\delta = b = 13$. Then w will have $21 - \alpha = 21 - 21 = 0$ rows of weight 21, and $13 - \beta = 13$ rows of weight 0, and $\alpha + \beta = 21$ rows of weight 13, giving total weight

$$0 \times 21 + 13 \times 0 + 21 \times 13 = 0 + 0 + 273 = 273.$$

Among its columns, w has $\delta = 13$ columns of weight 0 and $34 - \delta = 21$ columns of weight 13, for total weight

$$13 \times 0 + 0 \times 21 + 21 \times 13 = 0 + 0 + 273 = 273.$$

To achieve the minimal change of $8 \times 21 = 168$ cells whose value changed from 1 to 0, we need to concentrate those cells in the intersection of the 21 changed rows and 13 changed columns.

Acknowledgments

The authors would like to thank Venkatesan Guruswami for bringing [12] to their attention.

References

[1] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy. Proof verification and the intractibility of approximation problems. *Journal of the ACM*, 45(3):501–555, 1998.

[2] S. Arora and S. Safra. Probabilistic checking of proofs: A new characterization of NP. *Journal of the ACM*, 45(1):70–122, 1998.

[3] E. Ben-Sasson, O. Goldreich, P. Harsha, M. Sudan, and S. Vadhan. Robust PCPs of proximity, shorter PCPs and application to coding. In *Proceedings of the 36th Annual ACM Symposium on Theory of Computing (STOC)*, pages 1–10, 2004.

[4] E. Ben-Sasson and M. Sudan. Robust locally testable codes and products of codes. In *Proceedings of the 8th International Workshop on Randomization and Computation (RANDOM)*, pages 286–297, 2004.

[5] E. Ben-Sasson and M. Sudan. Simple PCPs with poly-log rate and query complexity. In *Proceedings of 37th ACM Symposium on Theory of Computing (STOC)*, 2005.

[6] O. Goldreich. Short locally testable codes and proofs (Survey). *ECCC Technical Report TR05-014*, 2005.

[7] O. Goldreich and M. Sudan. Locally testable codes and PCPs of almost linear length. In *Proceedings of 43rd Symposium on Foundations of Computer Science (FOCS)*, pages 13–22, 2002.

[8] V. Guruswami and A. Rudra. Tolerant locally testable codes. In *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 306–317, 2005.

[9] T. Kaufman and D. Ron. Testing polynomials over general fields. In *Proceedings of the 45th Annual IEEE Symposium on Foundations of Computer Science (FOCS)*, pages 413–422, 2004.

[10] A. Polishchuk and D. A. Spielman. Nearly-linear size holographic proofs. In *Proceedings of the 26th Annual ACM Symposium on Theory of Computing (STOC)*, pages 194–203, 1994.

[11] R. Raz and S. Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of 29th ACM Symposium on Theory of Computing (STOC)*, pages 475–484, 1997.

[12] P. Valiant. The tensor product of two codes is not necessarily robustly testable. In *Proceedings of the 9th International Workshop on Randomization and Computation (RANDOM)*, pages 472–481, 2005.