Some recent results on local testing of sparse linear codes*

Swastik Kopparty† and Shubhangi Saraf‡

February 24, 2010

Abstract

We study the local testability of linear codes. We first reformulate this question in the language of tolerant linearity testing under a non-uniform distribution. We then study the question of linearity testing under non-uniform distributions directly, and give a sufficient criterion for linearity to be tolerantly testable under a given distribution. We show that several natural classes of distributions satisfy this criterion (such as product distributions and low Fourier-bias distributions), thus showing that linearity is tolerantly testable under these distributions. This in turn implies that the corresponding codes are locally testable.

For the case of random sparse linear codes, we show the testability and decodability of such codes in the presence of very high noise rates. More precisely, we show that any linear code in $\mathbb{F}_2^n$ which is:

- sparse (i.e., has only poly(n) codewords)
- unbiased (i.e., each nonzero codeword has Hamming weight in $(1/2 - n^{-\gamma}, 1/2 + n^{-\gamma})$ for some constant $\gamma > 0$)

can be locally tested and locally list decoded from $(1/2 - \epsilon)$-fraction errors using only poly$(\frac{1}{\epsilon})$ queries to the received word. This simultaneously simplifies and strengthens a result of Kaufman and Sudan, who gave a local tester and local (unique) decoder for such codes from some constant fraction of errors.

For the case of Dual BCH codes, these algorithms can also be made to run in sublinear time. We also give sub-exponential time algorithms for list-decoding arbitrary unbiased (but not necessarily sparse) linear codes in the high-error regime.

1 Introduction

We begin by setting up some notation. A linear code $\mathcal{C}$ in $\mathbb{F}_2^N$ is simply a linear subspace of $\mathbb{F}_2^N$. The elements of $\mathcal{C}$ are often referred to as "codewords". We say that $\mathcal{C}$ is sparse if $|\mathcal{C}| \leq N^c$ for some

*The current extended abstract is a summary of some of the results that appeared in "Tolerant linearity testing and locally testable codes" that appeared in RANDOM 2009 and "Local list-decoding and testing of random linear codes from high-error" that will appear in STOC 2010.

†Computer Science and Artificial Intelligence Laboratory, MIT, swastik@mit.edu. Part of this work was done while the author was a summer intern at Microsoft Research New England.

‡Computer Science and Artificial Intelligence Laboratory, MIT, shubhangi@csail.mit.edu. Part of this work was done while the author was a summer intern at Microsoft Research New England.

constant c. For a string $x \in \mathbb{F}_2^N$, we define its (normalized Hamming) weight $\mathrm{wt}(x)$ to equal $\frac{1}{n}\times$ (the number of nonzero coordinates of x). We define the *bias* of the code $\mathcal{C}$ as $\max_{y \in \mathcal{C}\setminus\{0\}} \left|\frac{1-\mathrm{wt}(y)}{2}\right|$. Thus each nonzero codeword y of a code of bias β has $\mathrm{wt}(y) \in [(1-\beta)/2, (1+\beta)/2]$. For two strings $x, y \in \mathbb{F}_2^N$, we define the (normalized Hamming) distance between x and y, $\Delta(x,y)$, to be $\frac{1}{n}\times$ (the number of coordinates $i \in [N]$ where x_i and y_i differ). We then define the distance of a string x from the code $\mathcal{C}$, $\Delta(x,\mathcal{C})$ to be the minimum distance of x to some codeword of $\mathcal{C}$:

$$\Delta(x,\mathcal{C}) = \min_{y \in \mathcal{C}} \Delta(x,y).$$

The basic algorithmic tasks associated with codes are error-detection and error-correction. Here we are given an arbitrary received word $w \in \mathbb{F}_2^N$, and we want to (1) determine if $\Delta(w,\mathcal{C}) > \delta$, and (2) find all codewords $y \in \mathcal{C}$ such that $\Delta(w,\mathcal{C}) \leq \delta$. In recent years, there has been much interest in developing sublinear time algorithms (and in particular, highly query-efficient algorithms) for these tasks. In what follows, we will describe our results on various aspects of these questions.

2 Locally Testable Codes and Tolerant Linearity Testing

Informally, a local tester for $\mathcal{C}$ is a randomized algorithm A, which when given oracle access to a received word $w \in \mathbb{F}_2^N$, makes very few queries to the received word w, and distinguishes between the case where $w \in \mathcal{C}$ and the case where w is "far" (in Hamming distance) from all codewords of $\mathcal{C}$. A code $\mathcal{C}$ is said to be *locally testable* if there exists a constant query local tester for $\mathcal{C}$.

The first local tester (for any code) came from the seminal work of Blum, Luby and Rubinfeld [BLR93], which gave an efficient, 3-query local tester for the Hadamard code.

In order to study local testability for linear codes, we first reformulate the question in the language of testing under *distributions*. Let $\mathcal{C} \subseteq \mathbb{F}_2^N$ be a linear code of dimension n, and let G be an $n \times N$ generator matrix for $\mathcal{C}$. Let $S = \{v_1, v_2, \ldots, v_N\} \subset \mathbb{F}_2^n$ denote the set of columns of G. We associate to $\mathcal{C}$ the distribution μ over $\mathbb{F}_2^n$ which is uniform over S. Every word w in $\mathbb{F}_2^N$ can be viewed as a function $f_w : S \to \mathbb{F}_2$, where $f_w(v_i) = w_i$. Under this mapping, every *codeword* of $\mathcal{C}$ gets associated with a *linear* function.

Note that via this translation, the problem of testing if w is close to some codeword exactly translates into the problem of testing if f_w is close to some linear function *under the distribution* μ (where the distance of two functions g, h under μ is measured by the probability that g and h differ on a random sample from μ).

In this language, the BLR local tester for the Hadamard code is precisely the problem of testing linearity under μ, where now μ is the uniform distribution over $\mathbb{F}_2^n$.

For a general linear code $\mathcal{C}$, the related distribution μ is uniform over a subset S of $\mathbb{F}_2^n$. For the relationship between testability of the code $\mathcal{C}$ and the testability of linearity under μ to be tight, we must essentially force the tester for linearity under μ to make queries only from the set S. A notion of testing that naturally enforces this requirement is that of *tolerant* linearity testing. We now formally describe this notion and give the connection to locally testable codes.

Let $\mathcal{C}$ be a class of functions from a finite set $\mathcal{D}$ to a finite set $\mathcal{R}$. In the task of *tolerant testing* for $\mathcal{C}$, we are given oracle access to a function $f : \mathcal{D} \to \mathcal{R}$, and we wish to determine using few

queries to f, whether f is well approximable by functions in $\mathcal{C}$; equivalently, to distinguish between the case when f is *close* to some element of $\mathcal{C}$, and the case when f is *far* from all elements of $\mathcal{C}$. Tolerant property testing was introduced by Parnas, Ron and Rubinfeld in [PRR06] as a refinement of the problem of property testing [RS96], [GGR98] (where one wants to distinguish the case of f *in* $\mathcal{C}$ from the case when f is *far* from $\mathcal{C}$), and is now widely studied. The usual notion of closeness considered in the literature is via the distance measure $\Delta(f,g) = \Pr_{x\in\mathcal{D}}[f(x) \neq g(x)]$, where $x \in \mathcal{D}$ is picked according to the *uniform distribution* over $\mathcal{D}$.

We propose to study tolerant property testing under general distributions. Given a probability measure μ on $\mathcal{D}$, the μ-distance of f from g, where $f, g : \mathcal{D} \to \mathcal{R}$, is defined by

$$\Delta_\mu(f,g) = \Pr_{x\in\mu}[f(x) \neq g(x)].$$

Then the measure of how well f can be approximated by elements of $\mathcal{C}$ is via the μ-distance

$$\Delta_\mu(f,\mathcal{C}) = \min_{g\in\mathcal{C}} \Delta_\mu(f,g).$$

The new goal in this context then becomes to approximate $\Delta_\mu(f,\mathcal{C})$ using only a few oracle calls to f. In our context, we study a concrete instance of the above framework. We consider the original problem considered in the area of property testing, namely the classical problem of *linearity testing*.

The problem of linearity testing was introduced by Blum, Luby and Rubinfeld in [BLR93]. In this problem, we are given oracle access to a function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, and want to distinguish between the case that f is a *linear* function and the case that f is *far* from the class $\mathcal{L}$ of all linear functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$. [BLR93] gave a simple 3-query test T that achieves this. In fact, this test also achieves the task of *tolerant linearity testing*; i.e., for any function $f : \mathbb{F}_2^n \to \mathbb{F}_2$, letting $\delta = \Pr[T^f \text{ rejects}]$, we have

$$C_1 \cdot \delta \leq \Delta_{U_n}(f,\mathcal{L}) \leq C_2 \cdot \delta,$$

where C_1 and C_2 are absolute constants, and U_n is the uniform distribution on $\mathbb{F}_2^n$. Hence the test of [BLR93], in addition to *testing* linearity, actually estimates how well f can be *approximated* by functions in $\mathcal{L}$.

Here we study tolerant linearity testing over general probability distributions. Let μ be a probability distribution over $\mathbb{F}_2^n$. In the problem of *tolerant linearity testing under* μ, we wish to estimate the how well f may be approximated under μ by linear functions from $\mathcal{L}$. For a given family $(\mathbb{F}_2^n, \mu_n)_n$, we say *linearity is tolerantly testable* under $\mu = \mu_n$ with q queries, if there exists a q-query tester T_n and constants C_1, C_2 such that for any $f : \mathbb{F}_2^n \to \mathbb{F}_2$, letting $\delta = \Pr[T_n^f \text{ rejects}]$, we have

1. **Perfect completeness:** $\delta = 0$ if and only if $\Delta_\mu(f,\mathcal{L}) = 0$.
2. **Distance approximation:** δ approximates $\Delta_\mu(f,\mathcal{L})$:

$$C_1 \cdot \delta \leq \Delta_\mu(f,\mathcal{L}) \leq C_2 \cdot \delta. \tag{1}$$

The main question is to determine for which μ is linearity tolerantly testable under μ. This seems to be a basic question worthy of further study. Furthermore, the notion of linearity being tolerantly testable under general distributions is intimately connected with the concept of locally testable linear codes, and we now elaborate on this connection.

Now let C be any linear code, and let $s_1, \ldots, s_N \in \mathbb{Z}_2^n$ be the columns of a generator matrix for C. Let μ be the uniform distribution over $\{s_1, \ldots, s_N\}$. Then, if linearity is tolerantly testable under μ, then C is locally testable. Indeed, given any $r : [N] \to \mathbb{Z}_2$, we may define the function $f : \mathbb{Z}_2^n \to \mathbb{Z}_2$ by $f(x) = r(j)$ if $x = s_j$, and $f(x) = 0$ otherwise. By the tolerant testability of linearity under μ, any useful query made by a tolerant linearity tester for μ must be to one of the s_j. The distance of f from linear under μ then translates directly into the Hamming distance of r from C, and the very same tester that tolerantly tests linearity under μ shows that C is locally testable.

2.1 Notions and Results

We highlight a simple criterion, which we call *uniform-correlatability*, that lets us design and analyze tolerant linearity tests under a given distribution. Roughly speaking, a distribution μ over $\mathbb{F}_2^n$ is uniformly-correlatable if one can "design" a distribution of few correlated random variables, with each variable distributed according to μ, while all the sum of the variables if nearly uniformly distributed. In this case, we show that linearity is tolerantly testable under μ with few queries (see Theorem A). We complement this by demonstrating that many natural distributions satisfy this criterion.

Although we state all our results for functions from $\mathbb{F}_2^n$ to $\mathbb{F}_2$, most results carry over to all pairs of abelian groups G, H.

Definition 1 (Uniformly-correlatable distribution) *Let μ be a probability distribution on $\mathbb{F}_2^n$. We say that μ is* (ϵ, k)-uniformly-correlatable *if there is a random variable $\mathrm{X} = (\mathrm{X}_i)_{i \in [k]}$ taking values in $(\mathbb{F}_2^n)^k$ such that:*

1. *For each $i \in [k]$, X_i is distributed according to μ.*
2. *The distribution of the random variable $\sum_{i \in [k]} \mathrm{X}_i$ is ϵ-close to the uniform distribution over $\mathbb{F}_2^n$.*

Our main result in this setting is that uniformly correlatable distributions are tolerantly testable.

Informal Theorem A Let μ be a distribution over $\mathbb{F}_2^n$ that is (ϵ, k)-uniformly-correlatable. Then there is a $4k$ query tester T that tolerantly tests linearity over μ.

We supplement the above theorem by showing that several natural classes of distributions are all (ϵ, k)-uniformly-correlatable for suitable ϵ, k, such as product distributions, symmetric distributions and low Fourier-bias distributions, thus showing that linearity is tolerantly testable under these distributions. This in turn implies that the corresponding codes are locally testable.

3 Overview of Proof for Uniform-Correlatability $\implies$ Testability

We first give some intuition for the uniform correlatability criterion. For T to be a tester for linearity under μ, it needs to satisfy the following minimum requirements: (1) each query made by

the tester needs to be distributed essentially according to μ (so that the probability of rejection is upper bounded by the distance), and (2) the queries need to satisfy some linear relations (so that the tester has something to test). This already indicates that a tester will need to "design" a query distribution very carefully, so that both the above requirements are satisfied. This is where the uniformly-correlatable criterion comes in: given the uniformly-correlated distribution, it allows us to design other correlations quite flexibly, and in particular to produce queries distributed according to μ that satisfy linear relations.

To prove Theorem A, we use "uniform-correlatability" to reduce the problem of linearity testing over μ to linearity testing under the uniform distribution[1], for which we already know the BLR linearity test. In order to carry out the reduction, we use uniform-correlatability to define a "self-corrected" version h of the function f being tested. h has the property that if f is close to a linear function under μ, then h must be close to that same linear function under the uniform distribution! With this property in mind, we design two tests, Test 1 and Test 2. Test 1 is essentially the BLR test (over the uniform distribution) applied to the function h. Hence if Test 1 passes with high probability, then the BLR analysis implies that h is close to a linear function g under the uniform distribution. Test 2 is designed to test the proximity of original function f to the functions h under the μ distribution. Hence if Test 2 also passes, then it implies that f is actually close to the linear function g under μ.

In designing these two tests, we ensure that each query to f made by the tester is distributed essentially according to μ. It follows that the probability of rejection of the tests is at most a fixed multiple (depending on the number of queries made by the tester) of the distance of f from linear, and hence the tester is tolerant.

4 High error

Our main result in the high-error regime is that random sparse linear codes are locally testable and locally list-decodable in the *high-error* regime with only a *constant* number of queries. More precisely, we show that for all constants $c > 0$ and $\gamma > 0$, and for every linear code $\mathcal{C} \subseteq \{0,1\}^N$ which is:

- *sparse*: $|\mathcal{C}| \leq N^c$, and
- *unbiased*: each nonzero codeword in $\mathcal{C}$ has weight $\in (\frac{1}{2} - N^{-\gamma}, \frac{1}{2} + N^{-\gamma})$,

$\mathcal{C}$ is locally testable and locally list-decodable from $(\frac{1}{2} - \epsilon)$-fraction worst-case errors using only $\text{poly}(\frac{1}{\epsilon})$ queries to a received word. We also give *sub-exponential time* algorithms for list-decoding arbitrary unbiased (but not necessarily sparse) linear codes in the high-error regime. In particular, this gives a sub-exponential time algorithm even for the problem of (unique) decoding random linear codes of inverse-polynomial rate from a fixed positive fraction of errors (which does not seem to have been known prior to our work).

At the heart of our algorithms is a natural "self-correcting" operation defined on codes and received words. This self-correcting operation transforms a code $\mathcal{C}$ with a received word w into a simpler

[1]Note that the uniform distribution is $(0,1)$-uniformly-correlatable, and for this case, the test given by Theorem A essentially reduces to the BLR linearity test.

code $\mathcal{C}'$ and a related received word w', such that w is close to $\mathcal{C}$ if and only if w' is close to $\mathcal{C}'$. Starting with a sparse, unbiased code $\mathcal{C}$ and an arbitrary received word w, a constant number of applications of the self-correcting operation reduces us to the case of local list-decoding and testing for the *Hadamard code*, for which the well known algorithms of Goldreich-Levin and Blum-Luby-Rubinfeld are available. This yields the constant-query local algorithms for the original code $\mathcal{C}$.

Our algorithm for decoding unbiased linear codes in sub-exponential time proceeds similarly. Applying the self-correcting operation to an unbiased code $\mathcal{C}$ and an arbitrary received word a super-constant number of times, we get reduced to the problem of *learning noisy parities*, for which non-trivial sub-exponential time algorithms were recently given by Blum-Kalai-Wasserman and Feldman-Gopalan-Khot-Ponnuswami. Our result generalizes a result of Lyubashevsky, which gave sub-exponential time algorithms for decoding random linear codes of inverse-polynomial rate, from *random errors*.

4.1 Local testing

A particular variant of local testability which is of significant interest is local testing in the "high-error" regime. Here, for every constant $\epsilon > 0$, one wants to query-efficiently distinguish between the cases $\Delta(w, \mathcal{C}) < 1/2 - \epsilon$, i.e., w is "close" to $\mathcal{C}$, and $\Delta(w, \mathcal{C}) \approx 1/2$, i.e., w is as far from $\mathcal{C}$ as a random point is (for codes over large alphabets, $1/2$ gets replaced by 1). For the Hadamard code, the existence of such testers follows from the Fourier-analytic proof of the BLR linearity test [BCH$^+$96]. For the code of degree 2 multivariate polynomials over $\mathbb{F}_2$, local testers in the high-error regime were given by Samorodnitsky [Sam07]. For the code of multivariate polynomials over large fields, local-testers in the high-error regime were given by Raz-Safra [RS97], Arora-Sudan [AS03] and Moshkovitz-Raz [MR06]. More recently, Dinur-Goldenberg [DG08] and Impagliazzo-Kabanets-Wigderson [IKW09] gave query-efficient local testing algorithms in the high-error regime for the combinatorial families: the direct-product and XOR codes. These algebraic and combinatorial high-error local-testers led to some remarkable constructions of PCPs with high soundness.

All known locally-testable codes in the high-error regime are for highly structured algebraic or combinatorial codes. Kaufman and Sudan [KS07] showed that a random sparse linear code is locally testable in the low-error regime by studying its weight distribution and the weight distribution of its dual, and in particular their proof was based on the MacWilliams identities and non-trivial information about the roots of Krawtchouk polynomials. In the paper [KS09] (essentially using Theorem A), we gave an alternate (and arguably simpler) proof of this result, as part of a more general attempt to characterize sparse codes that are locally decodable and testable in the low-error regime. Popular belief [Sud09] suggested that local-testability in the *high-error* regime could not be found in random linear codes.

We show that that random codes *can* have query-efficient local testers in the high-error regime. Specifically, sparse and unbiased codes admit high-error local testers with constant query complexity.

Informal Theorem B For every constant $c, \gamma > 0$, every linear code $\mathcal{C} \subseteq \mathbb{F}_2^N$ with N^c codewords and bias $N^{-\gamma}$ can be locally tested from $(1/2 - \epsilon)$-fraction errors using only $\mathrm{poly}(\frac{1}{\epsilon})$ queries to a received word.

We in fact can show something stronger (called *distance estimation in the high-error regime*): for such codes, using constantly many queries, one can distinguish between $\Delta(w, \mathcal{C}) > 1/2 - \epsilon_1$ and $\Delta(w, \mathcal{C}) < 1/2 - \epsilon_2$ for every constants $0 < \epsilon_1 < \epsilon_2 < 1/2$.

4.2 Local list-decoding

Informally, a local list-decoder for $\mathcal{C}$ from δ-fraction errors is a randomized algorithm A that, when given oracle access to a received word $w \in \mathbb{F}_2^N$, recovers the list of all codewords c such that $\Delta(c, w) < \delta$, while querying w in very few coordinates. The codewords thus recovered are "implicitly represented" by randomized algorithms $A_1, \ldots, A_l$ with oracle access to w. Given a coordinate $j \in [N]$, A_i makes very few queries to w and is supposed to output the j^{th} coordinate of the codeword that it implicitly represents.

A particular case of local list-decoding which is of significant interest is local list-decoding in the "high-error" regime. Specifically, for every constant $\epsilon > 0$, one wants to query-efficiently locally list-decode a code from $(\frac{1}{2} - \epsilon)$-fraction errors (the significance of $\frac{1}{2} - \epsilon$ is that it is just barely enough to to distinguish the received word from a random string in $\mathbb{F}_2^N$; for codes over large alphabets, one considers the problem of decoding from $(1 - \epsilon)$-fraction errors). Local list-decoding in the high-error regime plays a particularly important role in the complexity-theoretic applications of coding theory (see [STV99], for example).

The first known local list-decoder (for any code) came from the seminal work of Goldreich and Levin [GL89], which gave time-efficient, low-query, local list-decoders for the Hadamard code in the high-error regime. In the following years, many highly non-trivial local list-decoders were developed for various codes, including multivariate polynomial based codes (in the works of Goldreich-Rubinfeld-Sudan [GRS00], Arora-Sudan [AS03], Sudan-Trevisan-Vadhan [STV99], and Gopalan-Klivans-Zuckerman [GKZ08]) and combinatorial codes such as direct-product codes and XOR codes (in the works of Impagliazzo-Wigderson and Impagliazzo-Jaiswal-Kabanets-Wigderson [IW97, IJKW08]). Many of these local list-decoders, especially the ones in the high-error regime, play a prominent role in celebrated results in complexity theory on hardness amplification and pseudorandomness [IW97, STV99, IJKW08].

To summarize, all known local list-decoding algorithms were for highly structured algebraic or combinatorial codes. Kaufman and Sudan [KS07] showed that *random* sparse linear codes can be locally (unique-)decoded from a small constant fraction of errors. This was the first result to show that query-efficient decoding algorithms could also be associated with random, unstructured codes. This result was proved by studying the weight distribution of these codes and their duals. Popular belief [Sud09] again suggested that these low-error local decoders for random codes could *not* be extended to the high-error regime.

Here we show that even random codes *can* have query-efficient local list-decoders in the high-error regime. Specifically, we show that linear codes which are *sparse* and *unbiased* (both properties are possessed by sparse random linear codes with high probability) admit high-error local list-decoders with constant query complexity.

Informal Theorem C For every constant c, $\gamma > 0$, every linear code $\mathcal{C} \subseteq \mathbb{F}_2^N$ with N^c codewords and bias $N^{-\gamma}$ can be locally list-decoded from $(1/2 - \epsilon)$-fraction errors using only $\text{poly}(\frac{1}{\epsilon})$ queries to a received word.

4.3 Subexponential time list-decoding

The techniques we develop to address the previous questions turn out to be useful for making progress on another fundamental algorithmic question in coding theory: that of *time-efficient* worst-case decoding of random linear codes. Given a random linear code $\mathcal{C} \subseteq \mathbb{F}_2^N$ and an arbitrary received word $w \in \mathbb{F}_2^N$, we are interested in quickly finding all the codewords $c \in \mathcal{C}$ such that $\Delta(w, c) < \frac{1}{2} - \epsilon$, for constant $\epsilon > 0$. We show that this problem can be solved in *sub-exponential time*, using just the unbiasedness of $\mathcal{C}$. Our algorithm uses some recent breakthroughs on the problem of learning noisy-parities due to Blum-Kalai-Wasserman [BKW03] and Feldman-Gopalan-Khot-Ponnuswami [FGKP06].

Informal Theorem D For all constants $\alpha, \gamma > 0$, for every linear code $\mathcal{C} \subseteq \mathbb{F}_2^N$ with dimension n, where $N = n^{1+\alpha}$, and bias $N^{-\gamma}$, and for every constant $\epsilon > 0$, $\mathcal{C}$ can be list-decoded from $(\frac{1}{2} - \epsilon)$-fraction errors in time $2^{O(n/\log\log n)}$.

In particular, the above theorem implies that if $\mathcal{C} \subseteq \mathbb{F}_2^N$ is a *random linear code* with dimension $n = N^{\frac{1}{1+\alpha}}$, then for every constant $\epsilon > 0$, $\mathcal{C}$ can be list-decoded from $(\frac{1}{2} - \epsilon)$-fraction errors in time $2^{O(n/\log\log n)}$.

Earlier, it was not even known how to *unique-decode* random linear codes from 0.1-*fraction* worst-case errors in time $2^{o(n)}$. For decoding random linear codes of inverse-polynomial rate from *random errors*, Lyubashevsky [Lyu05] gave a sub-exponential time algorithm, also based on algorithms for the Noisy Parity problem. Our result generalizes his in two ways: we decode from worst-case errors, and we give a natural, explicit criterion (namely low-bias) on the code $\mathcal{C}$ which guarantees the success of the algorithm.

A related result (and one that we use in our proof) is the sub-exponential time worst-case decoding of random linear codes in $\mathbb{F}_2^N$, of dimension $n = O(\log N \cdot \log\log N)$, in a weaker model [FGKP06, Theorem 10]. In this model, the adversary first corrupts the received bit associated to $(1/2 - \epsilon)$-fraction of the 2^n possible linear encoding functions, after which the code is randomly chosen. Our result has a more natural coding theory interpretation: the random code is chosen first, and then the adversary choses an arbitrary received word at distance $(1/2 - \epsilon)$ from the code. In the language of learning theory, the [FGKP06] result concerns learning parities in the presence of *agnostic noise*, while our result deals with the model of learning parities in the presence of *nasty classification noise* [BEK02].

4.4 Time-efficient local algorithms for dual-BCH codes

For the family of *dual-BCH* codes, perhaps the most important family of sparse, unbiased codes, we show that the constant-query local list-decoding and local testing algorithms can be made to run in a *time-efficient* manner too. The dual-BCH codes form a natural family of polynomial-based codes generalizing the Hadamard code. They have a number of extremal properties which give them an important role in coding theory. For example, the dual-BCH code $\mathcal{C} \subseteq \mathbb{F}_2^N$ with N^t codewords has bias as small as $O(t \cdot N^{-1/2})$, which is *optimal* for codes with N^t codewords!

The key to making our earlier query-efficient local list-decoding and local testing algorithms run in a time-efficient manner for dual-BCH codes, is a time-efficient efficient algorithm for a certain

sampling problem that arises in the local list-decoder and tester. This sampling problem turns out to be closely related to an algorithmic problem that was considered in the context of low-error testing of dual-BCH codes [KL05], that of sampling constant-weight BCH codewords. A variant of the sampling algorithm of [KL05] turns out to suffice for our problem too, and this leads to the following result.

Informal Theorem E For every constant c, the dual-BCH code $\mathcal{C} \subseteq \mathbb{F}_2^N$ with N^c codewords, can be locally list-decoded and locally tested from $(1/2 - \epsilon)$-fraction errors in time $\text{poly}(\log N, \frac{1}{\epsilon})$ using only $\text{poly}(\frac{1}{\epsilon})$ queries to a received word.

The original algorithm for sampling constant-weight BCH codewords given in [KL05], and was based [Lit09] on results on the weight distribution of BCH codes [KL95]. We give an alternate (and possibly simpler) analysis of this result.

5 Overview of proofs in the high error regime

In this section, we give an overview of the main ideas underlying our algorithms in the high error regime.

The main component of our algorithms is a certain "self-correcting" operation which transforms a code $\mathcal{C}$ with a received word w into a simpler code $\mathcal{C}'$ and a related received word w', such that w is close to $\mathcal{C}$ if and only if w' is close to $\mathcal{C}'$. Repeated application of this self-correcting operation will allow us to reduce our list-decoding and testing problems for $\mathcal{C}$ to certain kinds of list-decoding and testing problems for a significantly simpler code $\mathcal{C}^*$ (in our case, $\mathcal{C}^*$ will be the Hadamard code). Query-efficient/time-efficient algorithms for the simpler code $\mathcal{C}^*$ then lead to query-efficient/time-efficient algorithms for the original code $\mathcal{C}$.

In order to simplify the description of the self-correcting operation, we first translate our problems into the language of list-decoding and testing under *distributions*. Let $\mathcal{C} \subseteq \mathbb{F}_2^N$ be a linear code of dimension n, and let G be an $n \times N$ generator matrix for $\mathcal{C}$. Let $S = \{v_1, v_2, \ldots, v_N\} \subset \mathbb{F}_2^n$ denote the set of columns of G. We associate to $\mathcal{C}$ the distribution μ over $\mathbb{F}_2^n$ which is uniform over S. Note that if the code $\mathcal{C}$ has low bias, then the resulting distribution μ has small Fourier bias. Every word w in $\mathbb{F}_2^N$ can be viewed as a function $f_w : S \to \mathbb{F}_2$, where $f_w(v_i) = w_i$. Under this mapping, every *codeword* of $\mathcal{C}$ gets associated with a *linear* function.

Note that via this translation, the problem of testing if w is close to some codeword exactly translates into the problem of testing if f_w is close to some linear function *under the distribution* μ (where the distance of two functions g, h under μ is measured by the probability that g and h differ on a random sample from μ). Similarly, the problem of local list-decoding, i.e. the problem of finding all codewords close to w, translates into the problem of finding all linear functions that are close to f_w under the distribution μ.

We now come to the self-correcting operation on f and μ. The operation has the property that it maintains the property "f correlates with a linear function under μ", and at the same time it results in a distribution that is "simpler" in a certain precise sense.

Define $\mu^{(2)}$ to be the convolution of μ with itself; i.e., it is the distribution of the sum of two independent samples from μ. We define $f^{(2)} : \mathbb{F}_2^n \to \mathbb{F}_2$ to be the (probabilistic) function, where for

a given x, $f^{(2)}(x)$ is sampled as follows: first sample y_1 and y_2 independently and uniformly from μ conditioned on $y_1 + y_2 = x$, and return $f(y_1) + f(y_2)$ (if there are no such y_1, y_2, then define $f^{(2)}(x)$ arbitrarily).

The following two simple facts are key to what follows:

- $\mu^{(2)}$ is "simpler" than μ: the statistical distance of $\mu^{(2)}$ to the uniform distribution on $\mathbb{F}_2^n$ is significantly smaller than the statistical distance of μ to the uniform distribution on $\mathbb{F}_2^n$ (this follows from the low Fourier bias of μ, which in turn came from the unbiasedness of $\mathcal{C}$).
- If f is $(\frac{1}{2} - \epsilon)$-close to a linear function g under μ, then $f^{(2)}$ is $(\frac{1}{2} - 2\epsilon^2)$-close to g under $\mu^{(2)}$: this is a formal consequence of our definition of $f^{(2)}$. In particular, if f is noticeably-close to g under μ, then so is $f^{(2)}$ under $\mu^{(2)}$.

This leads to a general approach for list-decoding/testing for linear functions under μ. First pick k large, and consider the distribution $\mu^{(k)}$ and the function $f^{(k)}$ (defined analogously to $\mu^{(2)}$ and $f^{(2)}$). If k is chosen large enough, then $\mu^{(k)}$ will in fact be 2^{-10n}-close to the uniform distribution in statistical distance. Furthermore, if k is not too large, then $f^{(k)}$ will be noticeably-close under $\mu^{(k)}$ to the same linear functions that f is close to under μ. Thus, if k is suitable (as a function of the initial bias/sparsity of the code) $f^{(k)}$ is noticeably-close *under the uniform distribution* to the same linear functions that f is close to under μ. Now all we need to do is run a local list-decoding/testing algorithm on $f^{(k)}$ under the uniform distribution.

An important issue that was swept under the rug in this discussion, is the query/time-efficiency of working with $f^{(k)}$ and $\mu^{(k)}$. If we ignore running-time, one can simulate oracle access to $f^{(k)}$ using just a factor k larger number of queries to f. This leads to our query-efficient (but time-inefficient) algorithms for sparse, unbiased linear codes in the high-error regime (in this setting k only needs to be a constant). We stress that our proof of this result is significantly simpler and stronger than earlier analyses of local algorithms (in the low-error regime) of sparse, unbiased codes [KL05, KS07].

The bottleneck for implementing these local, query-efficient algorithms in a time-efficient manner is the following algorithmic "back-sampling" problem: given a point $x \in \mathbb{F}_2^n$, produce a sample from the distribution of $y_1, \ldots, y_k$ picked independently from μ conditioned on $\sum y_i = x$. A time-efficient back-sampling algorithm would allow us to time-efficiently simulate oracle access to $f^{(k)}$ given oracle access to f. For random sparse linear codes, solving this problem in time sublinear in N is impossible; however for specific, interesting sparse unbiased codes, this remains an important problem to address. For the special case of dual-BCH codes, perhaps the most important family of sparse, unbiased codes, we observe that the back-sampling problem can be solved using a small variant of an algorithm of Kaufman-Litsyn [KL05]. Thus for dual-BCH codes, we get $\mathrm{poly}\log(N)$-time, constant-query local testing and local list-decoding algorithms in the high-error regime.

For sub-exponential time list-decoding, we follow the same plan. Here too we will self-correct f to obtain a function $f^{(k)}$, such that every linear function that correlates with f under the μ distribution, also correlates with $f^{(k)}$ under the uniform distribution over $\mathbb{F}_2^n$. However, since we are now paying attention to running time (and we do not know how to solve the back-sampling problem for μ efficiently in general), we cannot afford to allow the list-decoder over the uniform distribution over $\mathbb{F}_2^n$ to query the value of $f^{(k)}$ at any point that it desires (since this will force us to back-sample in order to compute $f^{(k)}$ at that point). Instead, we will use some recent remarkable

list-decoders ([FGKP06, BKW03]), developed in the context of learning noisy parities, which can find all linear functions close (under the uniform distribution) to an arbitrary function h in sub-exponential time by simply querying the function h at independent uniformly random points of $\mathbb{F}_2^n$! Using the unbiasedness of μ, it turns out to be easy to evaluate $f^{(k)}$ at independent uniformly random points of $\mathbb{F}_2^n$. This leads to our sub-exponential time list-decoding algorithm.

Relationship to the k-wise XOR on codes: Back in the language of codes, what happened here has a curious interpretation. Given a code $\mathcal{C} \subseteq \mathbb{F}_2^N$, the k-wise XOR of $\mathcal{C}$, $\mathcal{C}^{(\oplus k)}$, is the code contained in $\mathbb{F}_2^{N^k}$ defined as follows: for every codeword $c \in \mathcal{C}$, there is a codeword $c^{(\oplus k)} \in F_2^{[N]^k}$ whose value in coordinate $(i_1, \ldots, i_k)$ equals $c_{i_1} \oplus c_{i_2} \oplus \ldots \oplus c_{i_k}$. In terms of this operation, our algorithms simply do the following: given a code $\mathcal{C}$ and received word w, consider the code $\mathcal{C}^{(\oplus k)}$ with received word $w^{(\oplus k)}$. The crucial observation is, that for k chosen suitably as a function of the bias/sparsity of $\mathcal{C}$, the code $\mathcal{C}^{(\oplus k)}$ is essentially, up to repeating each coordinate a roughly-equal number of times, the Hadamard code! Additionally, $w^{(\oplus k)}$ is close to $c^{(\oplus k)}$ for a codeword c if and only if w is close c. Thus decoding/testing $w^{(\oplus k)}$ for the Hadamard code now suffices to complete the algorithm.

The k-wise XOR on codes is an operation that shows up often as a device for hardness amplification, to convert functions that are hard to compute into functions that are even harder to compute. Our algorithms use the XOR operation for "good": here the XOR operation is a vehicle to *transfer* query-efficient/time-efficient algorithms for the Hadamard code to query-efficient/time-efficient algorithms for arbitrary unbiased codes.

References

[AS03] Sanjeev Arora and Madhu Sudan. Improved low degree testing and its applications. *Combinatorica*, 23(3):365–426, 2003. Preliminary version in Proceedings of ACM STOC 1997.

[BCH+96] Mihir Bellare, Don Coppersmith, Johan Håstad, Marcos Kiwi, and Madhu Sudan. Linearity testing over characteristic two. *IEEE Transactions on Information Theory*, 42(6):1781–1795, November 1996.

[BEK02] Nader H. Bshouty, Nadav Eiron, and Eyal Kushilevitz. PAC learning with nasty noise. *Theor. Comput. Sci.*, 288(2):255–275, 2002.

[BKW03] Avrim Blum, Adam Kalai, and Hal Wasserman. Noise-tolerant learning, the parity problem, and the statistical query model. *J. ACM*, 50(4):506–519, 2003.

[BLR93] Manuel Blum, Michael Luby, and Ronitt Rubinfeld. Self-testing/correcting with applications to numerical problems. *Journal of Computer and System Sciences*, 47(3):549–595, 1993.

[DG08] Irit Dinur and Elazar Goldenberg. Locally testing direct product in the low error range. In *FOCS*, pages 613–622. IEEE Computer Society, 2008.

[FGKP06] Vitaly Feldman, Parikshit Gopalan, Subhash Khot, and Ashok Kumar Ponnuswami. New results for learning noisy parities and halfspaces. In *FOCS*, pages 563–574, 2006.

[GGR98] Oded Goldreich, Shafi Goldwasser, and Dana Ron. Property testing and its connection to learning and approximation. *J. ACM*, 45(4):653–750, 1998.

[GKZ08] Parikshit Gopalan, Adam R. Klivans, and David Zuckerman. List-decoding reed-muller codes over small fields. In Ladner and Dwork [LD08], pages 265–274.

[GL89] Oded Goldreich and Leonid Levin. A hard-core predicate for all one-way functions. *Proceedings of the 21st Annual ACM Symposium on Theory of Computing*, pages 25–32, May 1989.

[GRS00] Oded Goldreich, Ronitt Rubinfeld, and Madhu Sudan. Learning polynomials with queries: The highly noisy case. *SIAM Journal on Discrete Mathematics*, 13(4):535–570, November 2000.

[IJKW08] Russell Impagliazzo, Ragesh Jaiswal, Valentine Kabanets, and Avi Wigderson. Uniform direct product theorems: simplified, optimized, and derandomized. In Ladner and Dwork [LD08], pages 579–588.

[IKW09] Russell Impagliazzo, Valentine Kabanets, and Avi Wigderson. New direct-product testers and 2-query pcps. In *STOC*, pages 131–140, 2009.

[IW97] Russell Impagliazzo and Avi Wigderson. P = BPP if E requires exponential circuits: Derandomizing the XOR Lemma. *Proceedings of the 29th Annual ACM Symposium on Theory of Computing*, pages 220–229, May 1997.

[KL95] Ilia Krasikov and Simon Litsyn. On spectra of BCH codes. *IEEE Transactions on Information Theory*, 41(3):786–788, 1995.

[KL05] T. Kaufman and S. Litsyn. Almost orthogonal linear codes are locally testable. In *Proceedings of the Forty-sixth Annual Symposium on Foundations of Computer Science*, pages 317–326, 2005.

[KS07] Tali Kaufman and Madhu Sudan. Sparse random linear codes are locally decodable and testable. In *FOCS*, pages 590–600. IEEE Computer Society, 2007.

[KS09] Swastik Kopparty and Shubhangi Saraf. Tolerant linearity testing and locally testable codes. In *APPROX-RANDOM*, pages 601–614, 2009.

[LD08] Richard E. Ladner and Cynthia Dwork, editors. *Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008*. ACM, 2008.

[Lit09] Simon Litsyn, 2009. Personal Communication.

[Lyu05] Vadim Lyubashevsky. The parity problem in the presence of noise, decoding random linear codes, and the subset sum problem. In *APPROX-RANDOM*, pages 378–389, 2005.

[MR06] Dana Moshkovitz and Ran Raz. Sub-constant error low degree test of almost-linear size. In Jon M. Kleinberg, editor, *STOC*, pages 21–30. ACM, 2006.

[PRR06] Michal Parnas, Dana Ron, and Ronitt Rubinfeld. Tolerant property testing and distance approximation. *J. Comput. Syst. Sci.*, 72(6):1012–1042, 2006.

[RS96] Ronitt Rubinfeld and Madhu Sudan. Robust characterizations of polynomials with applications to program testing. *SIAM Journal on Computing*, 25(2):252–271, April 1996.

[RS97] Ran Raz and Shmuel Safra. A sub-constant error-probability low-degree test, and a sub-constant error-probability PCP characterization of NP. In *Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing*, pages 475–484, New York, NY, 1997. ACM Press.

[Sam07] Alex Samorodnitsky. Low-degree tests at large distances. In *STOC*, pages 506–515, 2007.

[STV99] Madhu Sudan, Luca Trevisan, and Salil Vadhan. Pseudorandom generators without the XOR lemma. *Proceedings of the 31st Annual ACM Symposium on Theory of Computing*, pages 537–546, 1999.

[Sud09] Madhu Sudan, 2009. Personal Communication.