PARAMETER CHOICES AND A BETTER BOUND ON THE LIST SIZE IN THE GURUSWAMI-SUDAN ALGORITHM FOR ALGEBRAIC GEOMETRY CODES

NATHAN DRAKE AND GRETCHEN L. MATTHEWS
DEPARTMENT OF MATHEMATICAL SCIENCES
CLEMSON UNIVERSITY
CLEMSON, SC 29634-0975
U.S.A.
E-MAIL: NDRAKE@CLEMSON.EDU, GMATTHE@CLEMSON.EDU

Abstract. Given an algebraic geometry code $C_{\mathcal{L}}(D, \alpha P)$, the Guruswami-Sudan algorithm produces a list of all codewords in $C_{\mathcal{L}}(D, \alpha P)$ within a specified distance of a received word. The initialization step in the algorithm involves parameter choices that bound the degree of the interpolating polynomial and hence the length of the list of codewords generated. In this paper, we use simple properties of discriminants of polynomials over finite fields to provide improved parameter choices for the Guruswami-Sudan list decoding algorithm for algebraic geometry codes. As a consequence, we obtain a better bound on the list size as well as a lower degree interpolating polynomial.

1. Introduction

Algebraic geometry codes (sometimes called AG codes) were first defined by V. D. Goppa in the late 1970's [3, 4]. They are generalizations of Reed-Solomon codes which are among the most popular codes used in practice. Moreover, algebraic geometry codes have more flexible parameters than Reed-Solomon codes. Indeed, Tsfasman, Vlădut, and Zink [12] proved in the early 1980's that there are AG codes which perform better than the Gilbert-Varshamov bound (see also [1]).

A major advance in the study of algebraic geometry codes came with the generalization of Sudan's algorithm for list decoding Reed-Solomon codes [11] to one-point AG codes. The Guruswami-Sudan algorithm [7] is a polynomial time algorithm for list decoding one-point AG codes, enabling error correction beyond half the minimum distance (see also [6, 9]). In particular, for a received word $w \in \mathbb{F}^n$, the Guruswami-Sudan algorithm produces a list of codewords in an algebraic geometry code $C_{\mathcal{L}}(D, \alpha P)$ which agree with w in at least t coordinates, given that $t^2 > \alpha n$. The length of the list is bounded by a parameter s which is chosen in the initialization step of the algorithm.

In this paper, we provide new parameter choices which give a tighter bound on the list size generated by the Guruswami-Sudan algorithm. This is accomplished by providing a lower degree interpolating polynomial. This is especially desirable as the final step of the Guruswami-Sudan algorithm involves finding the roots of

Key words and phrases. AG code; list decoding; one-point code; polynomial reconstruction.
This project was supported by NSF DMS-0201286 and NSA H-98230-06-1-0008.

this polynomial. The method employed here parallels that of M. Wang's for Reed-Solomon codes [13] and is similar to the parameter choices made in Sudan's original algorithm.

This paper is organized as follows. This section concludes with notation to be used in the remainder of the paper. Section 2 is a brief review of the Guruswami-Sudan algorithm. Section 3 contains the main result on parameter selection. The final section of the paper, Section 4, contains examples illustrating the improvements given by our parameter choices.

Notation Let X be a projective curve of genus g over a finite field $\mathbb{F}$. Let $\mathbb{F}(X)$ denote the field of rational functions on X defined over $\mathbb{F}$. The divisor of a rational function f will be denoted by (f). Given a divisor A on X defined over $\mathbb{F}$, let $\mathcal{L}(A)$ denote the set of rational functions f on X defined over $\mathbb{F}$ with divisor $(f) \geq -A$ together with the zero function. Let $\ell(A)$ denote the dimension of $\mathcal{L}(A)$ as an $\mathbb{F}$-vector space. A (one-point) AG code $C_{\mathcal{L}}(D, \alpha P)$ can be constructed using divisors $D = \sum_{i=1}^{n} Q_i$ and αP on X where $Q_1, \ldots, Q_n, P$ are pairwise distinct $\mathbb{F}$-rational points and $\alpha \in \mathbb{Z}^+$ is a positive integer. In particular,

$$C_{\mathcal{L}}(D, \alpha P) := \{ev(f) : f \in \mathcal{L}(\alpha P)\}$$

where $ev(f) := (f(Q_1), \ldots, f(Q_n))$. While Goppa's original construction of algebraic geometry codes is more general, we take this as our definition of AG code here as these codes are exactly those considered in [7]. If $\alpha < n$, then $C_{\mathcal{L}}(D, \alpha P)$ has length n, dimension $\ell(\alpha P)$, and designed distance $n - \alpha$. The minimum distance of the code $C_{\mathcal{L}}(D, \alpha P)$ is at least its designed distance. We will use $d(C)$ to denote the minimum distance of a code C. As usual, a code of length n, dimension k, and minimum distance d (resp. at least d) is called an $[n, k, d]$ (resp. $[n, k, \geq d]$) code. Good general references on algebraic geometry codes include [8, 10].

2. Review of the Guruswami-Sudan Algorithm

In this section, we outline the decoding algorithm due to Guruswami and Sudan as found in [7, Section IV. B.]. We begin by reviewing the main steps of this algorithm applied to the AG code $C_{\mathcal{L}}(D, \alpha P)$ where $D = Q_1 + \cdots + Q_n$. The Guruswami-Sudan Algorithm consists of three main steps: initialization, interpolation, and factorization. The initialization step consists of choosing parameters r and s so that both the interpolation and the factorization can be performed and are guaranteed to have solutions.

Given a one-point AG code $C_{\mathcal{L}}(D, \alpha P)$ on a curve X, a basis of functions ϕ_i for $\mathcal{L}(\alpha P)$ can be formed so that the following twoároperies hold: 1) $\phi_i \in \mathcal{L}(\alpha_i P)$ for some $\alpha_i \leq \alpha$ and 2) for $i < j$, $\phi_j \notin \mathcal{L}(\alpha_i P)$ [7, Lemma 20]. Thus the pole orders of each of the ϕ_i's is distinct. (Note these functions only have a single pole at the point P.) Moreover for a rational point $Q' \neq P$, each of these functions ϕ_i can be rewritten as a linear combination of functions ψ_j where ψ_j has a zero of degree $j - 1$ at the point Q'. Thus we have $\phi_i = \sum_{j=1}^{m} C_j \psi_j$ where each $C_j \in \mathbb{F}$ depends on the choice of point Q' and the function ϕ_i.

Given such a set of functions, $\{\phi_i\}$, and a received word $w = (w_1, \ldots, w_n) \in \mathbb{F}^n$, the interpolation step seeks to find a polynomial of degree s. The polynomial has the form

$$Q(T) = \sum_{j=0}^{s} \sum_{i=1}^{rt-g-\alpha j} q_{i,j} \phi_i T^j \in K[T]$$

with $q_{i,j}$ as unknowns; that is to say, $Q(T)$ is a polynomial in T whose coefficients lie in the function field K associated with the curve X. Additionally, the polynomial can be rewritten by viewing the functions ϕ_i as linear combinations of ψ_j at each point Q_i. It is then required that Q has a zero of degree at least r at each pair (Q_i, w_i) where Q_i are in the support of D. This causes there to be $\frac{r(r+1)}{2}$ constraints for point Q_i and so there is a total of $n \cdot \left(\frac{r(r+1)}{2}\right)$ constraints for the interpolation problem.

In the final step of the algorithm, the roots of the polynomial $Q(T)$ are calculated. This can be done either through factoring the polynomial or more efficiently using a root finding algorithm such as that in [2]. Each function h such that $d(w, ev(h_i)) \leq n-t$ is a root of $Q(T)$ where $d(w, ev(h_i)) := |\{i : w_i \neq ev(h_i)\}|$. This is ensured by requiring through the choice of r and s that the number of zeros of $Q(h)$ be larger than the number of poles of $Q(h)$.

Algorithm 2.1 (Guruswami-Sudan Algorithm)**.**
Input: n, α, $w \in \mathbb{F}_q^n$, t.
Assumptions: $t^2 > \alpha n$.

(0) Parameter choices: Set $r := \left\lfloor \frac{2gt+\alpha n+\sqrt{(2gt+\alpha n)^2-4(g^2-1)(t^2-\alpha n)}}{2(t^2-\alpha n)} \right\rfloor + 1$, $l := rt-1$, and $s := \left\lfloor \frac{l-g}{\alpha} \right\rfloor$.
(1) Interpolation: Find a polynomial $Q[T]$ of degree s.
(2) Factorization: Find all roots $h \in \mathcal{L}(\alpha P)$ of the polynomial Q. For each such h, if $h(Q_i) = w_i$ for at least t values of i, then add h to the output list.

Output: $h_1, \ldots, h_s$ such that $d(w, ev(h_i)) \leq n-t$

We will focus on Steps (0) and (1) above. Notice that the content of these steps can be rephrased as the following polynomial reconstruction problem over the function field associated with the curve X.

Polynomial reconstruction problem: Given a set $\{Q_1, \ldots, Q_n, P\}$ of $n+1$ distinct $\mathbb{F}$-rational points on a curve X of genus g, a positive integer α, an agreement parameter $t \in \mathbb{Z}^+$, and $w = (w_1, \ldots, w_n) \in \mathbb{F}^n$, find all functions $h \in \mathcal{L}(\alpha P)$ such that $h(Q_i) = w_i$ for at least t values of i where P is an $\mathbb{F}$-rational point on X not equal to Q_i for all i.

3. Parameters choices in the Guruswami-Sudan algorithm

In this section, we give improved parameter choices which can be used in Step (0) of Algorithm 2.1. Certainly, it is advantageous to choose the parameters that result in a smaller degree interpolating polynomial Q and yield a better bound s on the list size of the output. We show how to do this for any one-point AG code $C_{\mathcal{L}}(D, \alpha P)$ and any agreement parameter $t > \sqrt{\alpha n}$ satisfying either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. The first restriction on t seems necessary to obtain a polynomial time algorithm as Guruswami and Rudra have evidence that a lower agreement parameter may lead to super-polynomially large lists as output [5].

Lemma 3.1. *Suppose n, α, g, and t satisfy (i) $t^2 > \alpha n$ and (ii) either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. Then the following statements are equivalent:*

(1) *There exist positive integers r and s such that*

$$(s+1)(rt-g) - \alpha\binom{s+1}{2} > n\binom{r+1}{2}.$$

(2) *There exist positive integers r and s satisfying the following conditions:*
(a) $r > \frac{\alpha(n-t)+2tg+\sqrt{\Delta_2}}{2(t^2-\alpha n)}$ *or* $r < \frac{\alpha(n-t)+2tg-\sqrt{\Delta_2}}{2(t^2-\alpha n)}$, *and*
(b) $s_1 < s < s_2$,
where

$$\begin{array}{lcl} s_1 & := & \frac{rt-\frac{\alpha}{2}-g-\sqrt{\Delta_1}}{\alpha}, \\ s_2 & := & \frac{rt-\frac{\alpha}{2}-g+\sqrt{\Delta_1}}{\alpha}, \\ \Delta_1 & := & \left(t^2-\alpha n\right)r^2 + \left(\alpha t - \alpha n - 2tg\right)r + \frac{\alpha^2}{4} + g^2 - \alpha g, \text{ and} \\ \Delta_2 & := & \alpha^2 n(n+\alpha-2t) + 4\alpha g n(t+g-\alpha). \end{array}$$

Proof. Assume n, α, g, and t satisfy (i) $t^2 > \alpha n$ and (ii) either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$.

(1) ⇒ (2): Suppose there exist positive integers r and s such that

$$(s+1)\left(rt-g\right) - \alpha\binom{s+1}{2} > n\binom{r+1}{2}.$$

Then

$$\frac{\alpha}{2}s^2 - (rt - g - \frac{\alpha}{2})s + \frac{r^2 n + rn}{2} - rt + g < 0.$$

Set

$$h_1(x) := \frac{\alpha}{2}x^2 - (rt - g - \frac{\alpha}{2})x + \frac{r^2 n + rn}{2} - rt + g.$$

Since $h_1(s) < 0$ and $\frac{\alpha}{2} > 0$, $h_1(x)$ must have two distinct real roots. Let Δ_1 denote the discriminant of $h_1(x)$. Then $\Delta_1 = (t^2 - \alpha n)r^2 + (\alpha t - \alpha n - 2tg)r + \frac{\alpha^2}{4} + g^2 - \alpha g > 0$, and the roots of $h_1(x)$ are $s_1 := \frac{rt-\frac{\alpha}{2}-g-\sqrt{\Delta_1}}{\alpha}$ and $s_2 := \frac{rt-\frac{\alpha}{2}-g+\sqrt{\Delta_1}}{\alpha}$. Consequently, $h_1(s) = (s - s_1)(s - s_2)$ and $s_1 < s < s_2$. Thus, (b) holds.

Next, we prove (a). To see this, set

$$h_2(x) := (t^2 - \alpha n)x + (\alpha t - \alpha n - 2tg)x + \frac{\alpha^2}{4} + g^2 - \alpha g.$$

Then $h_2(r) = \Delta_1 > 0$. Let Δ_2 be the discriminant of $h_2(x)$. Then

$$\begin{array}{rcl} \Delta_2 & = & \alpha^2 n(n+\alpha-2t) + 4\alpha g n(t+g-\alpha) \\ & = & \alpha n\left(\alpha n + \alpha^2 + 4g^2 - 4\alpha g - 2t\left(\alpha - 2g\right)\right). \end{array}$$

In the case $\alpha \leq 2g$, we see that

$$\Delta_2 > \alpha n\left(2\alpha^2 + 4g^2 - 4\alpha g - 2t\left(\alpha - 2g\right)\right) = \alpha n\left(2\left(t-\alpha\right)\left(2g-\alpha\right) + 4g^2\right) \geq 0$$

since $\alpha < t$. Otherwise, $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha-2g} + \alpha - 2g\right)$. Here, we have

$$\Delta_2 > \alpha n\left(\alpha n + \alpha^2 + 4g^2 - 4\alpha g - (\alpha - 2g)\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)\right) = 0.$$

Then

$$h_2(r) = \left(r - \frac{\alpha(n-t) + 2tg + \sqrt{\Delta_2}}{2(t^2 - \alpha n)}\right)\left(r - \frac{\alpha(n-t) + 2tg - \sqrt{\Delta_2}}{2(t^2 - \alpha n)}\right)$$

which implies $r > \frac{\alpha(n-t)+2tg+\sqrt{\Delta_2}}{2(t^2-\alpha n)}$ or $r < \frac{\alpha(n-t)+2tg-\sqrt{\Delta_2}}{2(t^2-\alpha n)}$.

(2) ⇒ (1): Suppose there exist positive integers r and s satisfying (a) and (b). Taking $h_1(x)$ and Δ_1 as above, we see that the choice of r guarantees that $\Delta_1 \geq 0$ and the choice of s guarantees $h_1(s) < 0$. As a result, $(s+1)(rt-g) - \alpha\binom{s+1}{2} > n\binom{r+1}{2}$. □

Next, we indicate how Lemma 3.1 can be used in conjunction with Algorithm 2.1 to obtain a better bound on the list size.

Theorem 3.2. *Consider the AG code $C_{\mathcal{L}}(D, \alpha P)$ on a curve X of genus g over $\mathbb{F}$ where $D := Q_1 + \cdots + Q_n$. Suppose (i) $t^2 > \alpha n$ and (ii) either $\alpha < 2g$ or $t < \frac{1}{2}\left(\frac{\alpha n}{\alpha - 2g} + \alpha - 2g\right)$. Then taking*

$$r := \left\lfloor \frac{\alpha(n-t) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)} \right\rfloor + 1 \text{ and } s := \left\lfloor \frac{rt - \frac{\alpha}{2} - g - \sqrt{\Delta_1}}{\alpha} \right\rfloor + 1$$

in Algorithm 2.1 produces a list of s codewords within distance $n-t$ of any received word $w \in \mathbb{F}^n$, where

$$\Delta_3 := \alpha^2\left((n-t)^2 - 4gn\right) + 4\alpha gn\,(t+g)\,.$$

Proof. Notice that $s = \lfloor s_1 \rfloor + 1$. We claim that $s_2 - s_1 > 1$ so that $s_1 < s < s_2$. To see this, observe that $s_2 - s_1 = \frac{2\sqrt{\Delta_1}}{\alpha}$. Thus, it suffices to show that $\Delta_1 > \frac{\alpha^2}{4}$. Since $\Delta_3 = \operatorname{disc}\left(\Delta_1 - \frac{\alpha^2}{4}\right)$, we have that

$$\Delta_1 - \frac{\alpha^2}{4} = \left(r - \frac{\alpha\,(n-t) + 2tg + \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right)\left(r - \frac{\alpha\,(n-t) + 2tg - \sqrt{\Delta_3}}{2(t^2 - \alpha n)}\right).$$

By the choice of r, it follows that $\Delta_1 - \frac{\alpha^2}{4} > 0$. Therefore, $s_1 < s < s_2$ as claimed.

We next check conditions (a) and (b) of Lemma 3.1(2). For condition (a), we note that

$$s\alpha \leq rt + \frac{\alpha}{2} - g - \sqrt{\Delta_1} < rt - g < rt$$

since $\sqrt{\Delta_1} > \frac{\alpha}{2}$ from above. Condition (b) holds, because

$$\Delta_3 - \Delta_2 = \alpha^2\left(t^2 - \alpha n\right) > 0.$$

Now applying Lemma 3.1, we see that r and s are valid parameters for the Guruswami-Sudan algorithm. □

4. Examples

In this section, examples are given to illustrate Theorem 3.2.

Example 4.1. Consider the Hermitian curve of genus 28 defined by $y^8 + y = x^9$ over $\mathbb{F}_{64}$ and the code $C_{\mathcal{L}}(D, 43P_\infty)$ where D is the sum of the 512 $\mathbb{F}_{64}$-rational points on the curve other than P_∞.

Let $t = 421$. Using the parameter choices in Algorithm 2.1, we have $r = 1$ and the number of solutions to the reconstruction problem is bounded by $s =$

$\left\lfloor \frac{(1(421)-1)-28}{43} \right\rfloor = 9$. Hence, we are guaranteed that there are at most 9 codewords within distance $n - t = 91$ of a received word $w \in \mathbb{F}_{64}^{512}$. By Theorem 3.2, we see that taking $r = 1$ and $s = 1$ is possible. Thus, applying Algorithm 2.1 with these parameter choices see that there is a unique codeword within distance 91 of w. In this example, we know that this must be the case since $C_{\mathcal{L}}(D, 43P_\infty)$ has minimum distance 469 (according to [14]) and $469 \geq 2 \cdot (512 - 421)$.

Now consider the code $C_{\mathcal{L}}(Q_1 + \cdots + Q_{512}, 217P_\infty)$ on the same curve. Suppose $w \in \mathbb{F}_{64}^{512}$ is a received word, and set $t = 337$. By Theorem 3.2, one can take $r = 24$ and $s = 36$ in the Guruswami-Sudan list decoding algorithm. Applying the algorithm with these parameter choices enables one to work with a degree (at most) 36 interpolating polynomial and yields a list of at most 36 words which agree with w in at least 337 places. The original parameter choices give an upper bound of $s = 83$ on the number of such words.

Example 4.2. Consider the code $C_{\mathcal{L}}(Q_1 + \cdots + Q_{125}, 58P_\infty)$ on the Hermitian curve of genus 10 defined by $y^5 + y = x^6$ over $\mathbb{F}_{25}$. Let $t = 88$. The typical parameters in Algorithm 2.1 are $r = 19$ and $s = 28$. According to Theorem 3.2, we can instead take $r = 9$ and $s = 12$. Hence, there are at most 12 codewords which agree with a received word $w \in \mathbb{F}_{25}^{125}$ in at least 88 places (as opposed to at most 28 which one might expect given by the original parameter choices in the algorithm).

References

[1] A. Garcia and H. Stichtenoth, A tower of Artin-Schreier extensions of function fields attaining the Drinfeld-Vlădut bound, Invent. Math. **121** (1995), 211–222.

[2] S. Gao and M. Shokrollahi, Computing roots of polynomials over function fields of curves, in: Coding Theory and Cryptography: From Enigma and Geheimschreiber to Quantum Theory, Springer, Berlin, 2000, 214–228.

[3] V. D. Goppa, *Algebraico-geometric codes*, Math. USSR-Izv. **21** (1983), 75–91.

[4] V. D. Goppa, *Geometry and Codes*, Kluwer, 1988.

[5] V. Guruswami and A. Rudra, Limits to list decoding Reed-Solomon codes, STOC'05: Proceedings of the 37th Annual ACM Symposium on Theory of Computing, 602–609, ACM, New York, 2005.

[6] V. Guruswami and M. Sudan, On representations of algebraic-geometry codes, IEEE Trans. Inform. Theory **47** (2001), no. 4, 1610–1613.

[7] V. Guruswami and M. Sudan, Improved decoding of Reed-Solomon and algebraic-geometric codes, IEEE Trans. Inform. Theory **45** (1999), no. 6, 1757–1767.

[8] T. Høholdt, J. H. van Lint, and R. Pellikaan, Algebraic geometry codes, in Handbook of Coding Theory, V. Pless, W. C. Huffman, and R. A. Brualdi, Eds., **1**, Elsevier, Amsterdam (1998), 871–961.

[9] M. A. Shokrollahi and H. Wasserman, List decoding of algebraic-geometric codes, IEEE Trans. Inform. Theory **45** (1999), 432–437.

[10] H. Stichtenoth, Algebraic Function Fields and Codes, Springer-Verlag, 1993.

[11] M. Sudan, Decoding of Reed-Solomon codes beyond the error correction bound, J. Compl. **13** (1997), 180–193.

[12] M. A. Tsfasman, S. G. Vlădut, and T. Zink, Modular curves, Shimura curves, and Goppa codes better than the Varshamov-Gilbert bound, Math. Nach., **109** (1982), 21–28.

[13] M. Wang, Parameter choices on Guruswami-Sudan algorithm for polynomial reconstruction, Finite Fields Appl. **13** (2007), 877–886.

[14] K. Yang and P. V. Kumar, On the true minimum distance of Hermitian codes, Coding Theory and Algebraic Geometry, Proceedings, Luminy, 1991, Lecture Notes in Mathematics **1518**, Springer-Verlag, 1992, 99–107.