Algebraic Geometry Codes

Ignacio Manzur Tomasini

Follow

Nethermind.eth

--

Listen

Share

By

[Ignacio Manzur

](https://www.linkedin.com/in/ignacio-manzur-0220121b7/). Special thanks to

[Albert Garreta

](https://twitter.com/0xAlbertG),

[Yevgeny Zaytman

](https://www.linkedin.com/in/yevgeny-zaytman-726640238/),

[Michal Zajac

](https://twitter.com/mpfzajac),

[Ahmet Ramazan Agirtas

](https://www.linkedin.com/in/aragirtas), and

[Aikaterini-Panagiota Stouka

](https://www.linkedin.com/in/aikaterini-panagiota-stouka/)for their helpful comments and suggestions.

This is a more technical blog post introducing algebraic geometry codes over curves. It assumes some level of familiarity with ring and field theory, as well as vector spaces over fields.

Table of contents:

- References.
- Disclaimer.
- About us.

Introduction: messages and information

Data is usually stored and transferred as a sequence of 0's and 1's, where each individual element in this sequence is a bit. Let's say you want to send the following message to your friend via Wi-Fi: "Hello!". If we picture what happens very naively, you are sending this string of bits through the air with radio waves:

"01001000 01100101 01101100 01101100 01101111 00100001"

Your friend receives it, her device understands the radio signal representing that string of bits, and displays "Hello!". But what guarantees that this radio signal arrives unaltered to your friend's device? Couldn't this signal interact with our atmosphere as it travels and be modified or lost? The answer is that it absolutely can, and it happens every day. If only two bits in the preceding message are flipped, you send "Hell!" — a much more sinister message.

Now, say you stored all your most precious information in a hard drive. Unfortunately, the area where you live has been hit by a shower of cosmic radiation, probably after a violent solar flare. The resulting cosmic particles went through the transistors storing your data, flipping an unknown number of bits. The information you keep on your hard drive is now slightly different and might not even be readable anymore!

These problems are real. They come up all the time and can have serious consequences. Though cosmic ray bit flips are extremely rare (if any exist), think about a scratched DVD or corrupted USB drive. There are naive ways to deal with this problem: one that comes to mind is to send or store the same information multiple times and use the majority of received/stored information to decide what the correct data should be. It is clear that such a method quickly becomes inefficient. One way to efficiently deal with this problem is by using a type of encoding called linear code. When correctly designed, it allows users to encode information efficiently and detect and correct errors in received/stored data.

In this blog post, starting from the ubiquitous example of Reed-Solomon encoding, we explore another way of constructing such codes using the algebraic theory of curves over a finite field. These are particularly useful when the amount of data needing to be encoded becomes larger and larger. In fact, they are the best codes

we know in these cases. We will focus on their construction rather than the decoding methods. We'll also explore asymptotically good sequences of Algebraic Geometry (AG) codes at a very high level.

What is a good (linear) code?

Notation.

For

q =

p^nan integer power of a prime number

p, we denote by $\mathbb{F}$

qthe unique (up to isomorphism)

[field

](https://en.wikipedia.org/wiki/Field_(mathematics))with

qelements. We can describe it as the splitting field of the polynomial X^

q-X over $\mathbb{F}_p$, where $\mathbb{F}_p$ is the set of integers {0, …,

p — 1} with addition and multiplication modulo

p. Alternatively, one can see $\mathbb{F}$

qas the quotient $\mathbb{F}_p[X]/(f(X))$ where f is a monic irreducible polynomial over $\mathbb{F}_p$ with degree

n.

A linear code C of length n

and dimension k

≤ n

is an $\mathbb{F}q$

-sub-vector space of $\mathbb{F}q^n$

of dimension k

. Elements of C are called codewords. Many times, we realize such subspaces as the image of an $\mathbb{F}q$

-linear map $V \to \mathbb{F}q^n$

, where V is an $\mathbb{F}q$

-vector space of dimension k

. The (minimal) distance d

of the linear code is the minimal Hamming distance between nonzero codewords, or equivalently the minimum number of nonzero coordinates of any nonzero codeword. We say that C is an [n,k,d]

-code (sometimes we emphasize [n,k,d]q

).

Error detection, error correction

The distance of the code is intimately related to its ability to detect and correct errors in received messages. Intuitively, the larger this distance, the smaller the set of possible codewords close to the received message. We try to picture this in Figure 1 below.

The red points represent codewords; some elements in the ambient linear space are shown in blue. Our code has a minimal distance of 4 (it would be the minimum number of horizontal or vertical steps one has to take to go from one point to another and not the Hamming distance, but we use it to understand what is going on). On the left, there has only been one error while transmitting the message, so the received message is the blue point one step to the right of our bottom left red point. Someone receiving such a message can immediately deduce that the original message is most likely the closest codeword to this point, which is our original red point. On the right, there have been two errors while transmitting the message, and the received word is the blue point in the middle of the figure. Here, someone receiving this message can deduce that there has been an error, but it cannot determine whether the original message was the bottom left or the upper right red point, as both are at distance 2 from the received word. Therefore this code can correct at most 1 error (remark that $1 = \lfloor (4-1)/2 \rfloor$, in fact, this is the relationship between the minimal distance and the maximum number of errors one can correct “uniquely”), as shown by the green circle on the left, representing the “Hamming ball” around our codeword.

This simple example shows that we should aim to have the maximum possible minimal distance. Unfortunately, there is a simple result constraining the possible values of d

:

Lemma (Singleton bound).

If C is an [n,k,d]q linear code, then d≤ n-k+1

proof.

Let C be an arbitrary linear code over $\mathbb{F}q$

with minimal distance d

. All codewords are distinct. If we delete the first d —

1 coordinates of every codeword, they are still pairwise distinct because every pair differs in at least d

coordinates. The number of coordinates of the new codewords is

n — d+1

, but there is the same number of distinct codewords. It follows that $|C| \leq q^{n-d+1}$

. If C is linear of dimension k

, we have $|C|=q^k$

. ■

It follows that the largest possible value of d

is n — k+1

when we attain the singleton bound. Linear codes that attain this bound are called Maximum Distance Separable (MDS).

Decoding received messages

At the same time, we need to encode messages into codewords using a relatively simple procedure. We also need an algorithm to decode received messages: this means translating back received messages m into codewords from C. Such an algorithm should take a message m and return the closest (or a list of close) codewords to m. There are inefficient ways to do this, e.g., to check all codewords. We are interested in efficient decoding algorithms, and this will usually depend on the search distance.

The example of Reed-Solomon (RS) encoding

The Reed-Solomon encoding is one of the most important examples of linear codes. It is everywhere in modern storage and communication, which is all the more remarkable because of its elegant and simple nature.

Fix a prime power q

(this quantity is called the alphabet size

), n

distinct elements of $\mathbb{F}q$

, say $y_1, \ldots, y_n$ (the integer n is called the blocklength

) and suppose you want to send a message that consists of $k < n \leq q$

elements (k

is the dimension

) $x_1, \ldots, x_k$ in $\mathbb{F}q$

. We construct an extension of the message in the following way:

- Consider the vector of elements $(x_1, \ldots, x_k)$ to be sent as the coefficients of a degree at most k

— 1 polynomial $f(X)=x_1+x_2X+ \ldots + x_kX^{k-1}$ over $\mathbb{F}q$

.

- Compute the evaluations of f at all y_i, that is compute $f(y_i)$ for i=1, …, n.
- Send all the computed evaluations by sending the vector $(f(y_1),\ldots, f(y_n))$.

This description of RS codes does not exactly fit our definition of a linear code. To see the link, consider the space V of polynomials with coefficients in $\mathbb{F}q$

with degree strictly less than k

. These can be described by their vector of coefficients of length k

. This is an $\mathbb{F}q$

-sub-vector space of $\mathbb{F}q^n$

because k<n

, and the fact that addition and scalar multiplication are defined component-wise. By using the obviously $\mathbb{F}q$

-linear evaluation map:

we see how any message encoded by using the previous description corresponds to exactly one element in the image of this map.

If your message stays intact during communication or storage, then it is immediate to obtain your original message back by using polynomial interpolation. If not, RS codes possess many properties that allow for efficient decoding. For example, Reed-Solomon codes are MDS. Furthermore, the Berlekamp-Welch algorithm is an example of a decoding algorithm that is much more efficient than exhaustive search and can correct up to ⌊(d —

1)/2⌋ errors.

Algebraic geometry codes

It is no wonder that RS codes satisfy all these nice properties; they are examples of a more general family of codes called Algebraic Geometry codes

(over curves). We will explain later that they can be seen as evaluation codes over the projective line $\mathbb{P}^1$. Algebraic Geometry codes satisfy many desirable properties, including the fact that:

- They are explicitly constructible.
- They are efficiently decodable.
- They have good bounds on their parameters, though they are not MDS when the genus of the curve is greater than 0, they remain close to the singleton bound.

Introduction to algebraic geometry (AG) codes (over curves)

Let us consider a curve $\mathfrak{x}$ over $\mathbb{F}q$,

we can think of it as being the projectivization (and possibly desingularization) of the affine curve

$\mathfrak{x}_a$:={ (x, y) ∈ $\mathbb{F}q$

^{2} | p(x, y)=0 }, where p ∈ $\mathbb{F}q$

[X, Y] is absolutely irreducible (irreducible over the algebraic closure of $\mathbb{F}q$

).

Technical details about homogenization, projectivization, and singularities.

Let p ∈ $\mathbb{F}q$

[X,Y], consider F ∈ $\mathbb{F}q$

[X_0, X_1, X_2] defined as:

We call F the homogenization

of p with respect to X_2. This polynomial is homogeneous, meaning that for any λ ∈ $\mathbb{F}q$

, we have:

The projectivization of the affine curve $\mathfrak{x}_a$ can therefore be defined as:

Here $\mathbb{P}^2$($\mathbb{F}q$

) is the projective plane over $\mathbb{F}q$

, which we define as the quotient of the space $\mathbb{F}q$

^{3}\{(0,0,0)} by the equivalence relation (z_0, z_1, z_2) ∼ (λz_0, λz_1, λz_2), ∀ λ ∈ $\mathbb{F}q$

*. We will encounter the projective line $\mathbb{P}^1$($\mathbb{F}q$

) later on and give a bit more intuition into how to think about it.

Not crucial for understanding the post:

The resolution of singularities part is somewhat more challenging to explain. Recall that a singularity of the projectivization is given by a point [x_0:x_1:x_2] in the projectivization such that $\partial F/\partial X_i(x_0, x_1, x_2)=0$ for i=0, 1, 2. We can assume that this does not happen, i.e., that the projectivization is nonsingular, without harming our argument.

Preliminaries

Consider a “closed point” P of $\mathfrak{X}$, which we can think about as a triple $[x_0: x_1: x_2] \in \mathbb{P}^2(\mathbb{F}q$

) such that $F(x_0,x_1,x_2)=0$, where F is the homogenization of p. We have to be careful here because $\mathbb{F}q$

is not algebraically closed, but this gives a good feeling for what a point might be. For each such point P, there exists a corresponding valuation (see below), or place

, $v_p: K(\mathfrak{X}) \to \mathbb{Z} \cup \{\infty\}$, where $K(\mathfrak{X})=\text{Frac}(\mathbb{F}q$

$[X,Y]/(p(X,Y)))$ is the function field of $\mathfrak{X}$ (here for an integral ring R we denote by Frac(R) its field of fractions). We recall that a valuation is such that:

In analogy with the theory of curves over $\mathbb{C}$ and their meromorphic functions, we can think of $v_p(f)$ as being the order of the “rational function” f at P: if $v_p(f)>0$ we can think of f as having a zero of order $v_p(f)$ at P, if $v_p(f)<0$ we can think of f as having a pole of order $-v_p(f)$ at P.

For each place v_p, there is an associated local ring $\mathcal{O}_p=\{f \in K(\mathfrak{X}) | v_p(f) \geq 0\} \cup \{0\}$ with unique maximal ideal $\mathfrak{m}_p=\{f \in K(\mathfrak{X}) | v_p(f)>0\} \cup \{0\}$ (a “rational function” f in $\mathcal{O}_p$ is invertible “at P” if and only if $v_p(f)=0$).

The quotient $\mathcal{O}_p/\mathfrak{m}_p:=k_p$, called the residue field

at P, is a vector space over $\mathbb{F}q$

, and we define:

The idea now is to consider functions with prescribed zeros and poles because this will allow us to build linear maps from finite-dimensional vector spaces to $\mathbb{F}q$

^{n}, whose images are precisely what we defined as linear codes. Furthermore, it can help us specify the type of functions we are looking for (as we will see, polynomials have a characteristic structure of zeros and poles over $\mathbb{P}^1$). We do this by associating formal sums to rational functions which encode their multiplicities. We set:

It can be proven that for any $f \in K(\mathfrak{X})^*$, there are only finitely many points where f has zeros or poles, so the previous sum makes sense

. Sums of the form (1) are called principal divisors.

We can also show that zeros and poles of rational functions compensate exactly when taken with the correct multiplicity, that is, for all $f \in K(\mathfrak{X})^*$:

The sum is meaningful because all but finitely many valuations are zero. There are also non-principal divisors, which arise when we allow arbitrary finite linear combinations of points $D=\sum_p c_p \cdot P$, where all $c_p \in \mathbb{Z}$ (the degree of D is defined in the same way). By using the properties of valuations above, it is easy to see that $\text{div}(f \cdot g$

$)=\text{div}(f)+\text{div}(g$

), from which it follows that principal divisors form an additive subgroup of the additive group of divisors.

We can impose conditions on the order of zeros and poles of f by taking a divisor $D=\sum_p c_p \cdot P$ and asking that all coefficients of $\text{div}(f) + D = \sum_p (v_p(f) + c_p) \cdot P$ are positive. We denote this $\text{div}(f)+D \geq 0$, or $\text{div}(f) \geq -D$. Taking degrees is compatible with this partial ordering, meaning that if $E \geq D$, then $\deg(E) \geq \deg(D)$.

For an arbitrary divisor D, its associated Riemann-Roch space

is defined to be the following sub-vector space of $K(\mathfrak{X})$:

Surprisingly (because $K(\mathfrak{X})$ is a vector space of infinite dimension over $\mathbb{F}q$

), L(D) is finite-dimensional for all divisors D, with dimension $\ell(D)$. One of the greatest results of the theory of curves is the following:

Theorem (Riemann-Roch).

There exists a divisor K called the canonical divisor such that for all divisors D, the following holds:

The quantity $\ell(K)$ is called the genus of the curve $\mathfrak{X}$, denoted g. We have $\deg(K)=2g-2$. In particular, since $\ell(K-D)$ is the dimension of a vector space, we find:

Remark first that since $\deg(\text{div}(f))=0$, if the degree of D is strictly negative, there are no rational functions such that $\text{div}(f) \geq -D$, and so $\ell(D)=0$. From this, we obtain the following:

Corollary to Riemann-Roch.

If deg(D)>2g — 2, then $\ell(D)=\deg(D)+1$ — g.

Evaluation codes

We have now obtained a roadmap for building (hopefully good) linear codes: we use linear maps from the finite-dimensional Riemann-Roch spaces to obtain linear codes and apply Riemann-Roch type arguments to obtain bounds on the length, dimension, and distance of the resulting codes. This is exactly what we do.

Let $\mathfrak{X}$ be a curve over $\mathbb{F}q$

, let G be an arbitrary divisor, and let $\mathscr{P}=\{P_1, \ldots, P_n\}$ a set of n

$\mathbb{F}q$

-rational points (with coordinates in $\mathbb{F}q$

, alternatively such that $k_p=\mathbb{F}q$

) such that for all i, the coefficient associated to P_i in the sum defining the divisor G is 0. Associate a very simple divisor to $\mathscr{P}$ as follows:

For any place v_p there is a natural way to “evaluate” a rational function $f \in \mathcal{O}_p$ “at P”: just send it to its image f_p in the residue field $k_p=\mathcal{O}_p/\mathfrak{m}_p$ by the natural quotient map. We define $f_p:=f(P)$. If P is $\mathbb{F}q$

-rational, this means in particular that $k_p=\mathbb{F}q$

, and therefore we find that $f(P) \in \mathbb{F}q$

. Furthermore, in this case, the evaluation map is $\mathbb{F}q$

-linear, that is:

This allows us to make the following definition.

Definition (Evaluation code).

The evaluation code $C(\mathfrak{X},\mathscr{P}, G)$ is defined as the image of the linear map:

For this definition to make sense, in view of our previous observations, we need $f \in \mathcal{O}_p$ for all $f \in L(G)$ and $P=P_i$. This is indeed the case because the coefficient associated with each P_i is 0 in the sum defining the divisor G, therefore any function f verifying $f \geq$ — G is such that $v_p(f) \geq 0$ for all $P=P_i$.

We can now obtain the following basic estimates:

Theorem.

The evaluation code $C(\mathfrak{X},\mathscr{P}, G)$ is a linear code of length $n=\deg(D_p)$ (all degrees $\deg(P_i)$ are 1 because they are $\mathbb{F}q$-rational points) and dimension:

In particular if deg(G)<n, then

And if 2g — 2< deg(G)< n, then

Its minimum distance d satisfies

proof.

The first equality follows from Rank-Nullity, as the map is linear we find:

Now the dimension of the image of ev

is k by definition. If $f \in L(G)$ is such that ev

$(f)=0$

, then this means that $v_p(f) > 0$ for all $P=P_i$. Since in the sum defining the divisor G the P_i’s do not appear, we find that:

or in other words $f \in L(G — D_p)$. The converse follows from the same argument. This shows that ker(ev

$)=L(G — D_p)$, and therefore dim(ker(ev

))=ℓ(G — D_P). The inequalities involving k are a direct application of Riemann-Roch and its Corollary.

For the minimal distance, suppose ev

(f) is a nonzero vector in the image. Let w

be its Hamming weight (the number of nonzero coordinates). Then ev

(f) has n

— w

coordinates equal to 0, w.l.o.g. assume these are the first n

— w

. This means that $f(P)=0$ for $P=P_i$ and i=1, …, n

— w

, and in turn this means that $v_P(f)>0$ for $P=P_i$ and i=1, …, n

— w

. As before, we find:

Taking degrees and using equation (2) from above, together with the fact that the P_i are $\mathbb{F}q$

-rational we obtain:

Since ev

(f) was an arbitrary nonzero vector in the image, the same inequality holds for the minimal distance in place of w

. ■

Combining the above inequalities, it follows that if deg(G) <n

, then:

That is, the singleton defect of $C(\mathfrak{X}, \mathscr{P}, G)$ is at most the genus of $\mathfrak{X}$.

In particular, it follows that if the curve has genus 0, then d=n — k+1 because of the singleton bound. That is, the code $C(\mathfrak{X}$, , G) is MDS.

Codes from $\mathbb{P}^1$

The projective line is the prototypical example of a genus 0 curve. As a set, we think of it as being the quotient:

Elements of $\mathbb{P}^1$ are therefore represented by equivalence classes of points $(z_1, z_2) \in \mathbb{F}q$

$^2 \setminus \{(0,0)\}$ which we denote $[z_1:z_2]$. Essentially $\mathbb{P}^1$ can be thought of as the line containing the elements of $\mathbb{F}q$

represented by classes [z:1] for $z \in \mathbb{F}q$

, together with a point “at infinity” ∞ = [1:0]. As a curve, its function field can be described as the field of rational functions in one formal variable $\mathbb{F}q$

(X) (using the equivalence between function fields and nonsingular projective curves).

Recall the definition of the Reed-Solomon codes:

Definition (Reed-Solomon codes).

Let $x =(x_1, …, x_n) \in \mathbb{F}q^n$ be a tuple of distinct elements. The Reed-Solomon code $RS_k(x)$ of dimension k <n is the linear evaluation code defined as follows:

It turns out that the nice properties of RS codes can be obtained from the fact that they are a particular type of AG code from the projective line:

Proposition.

Let $x =(x_1, …, x_n) \in \mathbb{F}q^n$ be a tuple of distinct elements.

Set $\mathscr{P}$={ [x_1 :1], …, [x_n :1] } and ∞=[1:0].

Then the evaluation code C($\mathbb{P}^1$, $\mathscr{P}$, (k — 1)∞) is nothing but $RS_k(x)$.

In particular, $RS_k(x)$ is MDS.

proof.

The set L((k

— 1)∞) consists of rational functions in one variable which have a pole of order strictly less than k

at “infinity”. These are precisely the polynomials of degree < k

. This can be shown by using the fact that the valuation corresponding to ∞ is the natural extension to the field of rational functions in one variable of the valuation associated with the prime ideal (1/X) in $\mathbb{F}q$

[1/X]. Since g

($\mathbb{P}^1$)=0, equation (3) shows that $RS_k(x)$ is MDS. ■

There exists an extension of RS codes, called generalized RS codes, defined in the following way:

Definition (Generalized RS codes).

Let x =(x_1, …, x_n), y=(y_1, …, y_n) ∈ $\mathbb{F}q^n$ be two tuples of distinct elements. The Generalized Reed-Solomon code $GRS_k(x)$ of dimension k <n is the linear evaluation code defined as follows:

It is possible to obtain the following result (for proof, we refer to[1], p.15):

Theorem.

Every Generalized Reed-Solomon code is an AG code from the projective line whose evaluation points avoid ∞, and conversely.

Beyond the alphabet size

We have now constructed linear codes by using curves over finite fields. As a particular example, we have obtained RS and even GRS codes. But if RS codes are so efficient and elegant, why do we need other codes? The answer lies in the integer n

, which we call the blocklength of the RS code. Remark that in the definition, we asked for the n

evaluation points to be distinct. Otherwise, we would have redundant information in our encoding. This constrains n

to be, at most q

, the alphabet size. We find that RS codes can encode messages that have a length at most q

— 1.

If we need to send longer messages, we need to work over larger prime numbers or higher degree extensions. This makes decoding more expensive and implies having to implement many different finite fields (in our computers) depending on the lengths of the messages we want to send or store. One natural question, therefore, is the following: for a fixed alphabet size

q

, is it possible to obtain linear codes that can encode arbitrarily large amounts of information while still having error correction capabilities?

This question turned out to be very challenging. It was first understood that those were possible

,

but an explicit construction was lacking. Not only did AG codes provide the first explicit examples of such codes, but they are currently the best that we know of, too.

Asymptotically good codes

In this section, we omit proofs.

For a[n,k,d]q

code C define the information rate R=k/n

and the relative minimal distance δ=d/n

. An asymptotically good sequence of codes is a sequence of codes C_1, C_2,… with lengths n

$_1$, n

$_2$, …, dimensions k

$_1$, k

$_2$, …and minimal distances d

$_1$, d

$_2$, … respectively, such that:

I.e., a sequence of codes with strictly positive asymptotic information rate/relative minimal distance. This condition means that as n_i

grows arbitrarily large, the amount of information k_i

and the minimal distance d_i

do not become negligible with respect to n_i

. This shows that we can encode an arbitrarily large amount of information while still having non-trivial error-correcting capabilities (because the minimal distance is not negligible w.r.t. n_i

).

A non-constructive argument given independently by Gilbert and later Varshamov (using different techniques) shows that an asymptotically good sequence of codes exists over any finite field $\mathbb{F}q$

with 0 <δ <1–1/q

and R

=1-Hq

(δ) where:

is the q

-ary entropy function.

For a long time, it was an open question to determine an explicit/computable construction of asymptotically good codes. Algebraic geometry codes provide a possible answer. Coming back to equation (3), it can be seen as a tradeoff: as n

increases, the minimal distance increases, but as the genus g

increases, the distance decreases. Since n

is ultimately related to the number of distinct $\mathbb{F}q$

-rational points, the aim would be to find curves with low genus and a large number of $\mathbb{F}q$-rational points

. Rewrite (3) as:

Letting n

$\rightarrow \infty$ in (4) motivates the following definition:

Definition (Ihara constant).

The Ihara constant of $\mathbb{F}q$ is defined as follows:

Where $\mathfrak{x}$ runs through all curves over $\mathbb{F}q$ and $n(\mathfrak{x})$ denotes the number of $\mathbb{F}q$-rational points of $\mathfrak{x}$.

It is then possible to obtain the following result:

Theorem (Asymptotic AG codes).

Assume A(q)>1. For any R,δ >0 satisfying R+δ=1 — 1/A(q), there exists asymptotically good codes with asymptotic rate and minimal distance at least R and δ respectively.

For the result to be meaningful, we need estimates on A(q

). Hasse-Weil immediately implies that A(q

)≤ 2 √q

. Several improvements culminated in the following upper bound:

Theorem (Drinfeld-Vlăduţ).

For any q, A(q) ≤ √q — 1.

Using Shimura curves, Ihara showed that when q

is a square, then A(q

) ≥ √q — 1, and so in fact, A(q

)=√q —

1. Ihara's lower bound, combined with the Theorem about asymptotic AG codes, shows that there exist asymptotically good codes with

This is known as the Tsfasman-Vlăduţ-Zink bound, and for q

≥ 49 it gives better asymptotic codes than the Gilbert-Varshamov bound!

And for general

q?

For general

q, Serre showed in

[[9]

](/b7bbef332dbc4feba9330874753ae4cc)that A(

q) ≥ c log(

q). Later, the constant was improved, and we now know that c can be taken to be 1/(96 log(2)). When

qis prime, this is often the best-known estimate.

Related work

Theoretically, we can outline some of the major lines of work in the development of AG codes.

One has focused on improving the minimal distance bound in (3). As explained in [1], these can broadly be categorized based on whether they rely on the use of "base points" of divisors related to G or on the use of a filtration:

with some clever estimates on the number of points in $C(\mathfrak{X}, \mathscr{P}, G_i) \setminus C(\mathfrak{X}, \mathscr{P}, G_{i+1})$. There are also some improvements that rely on the particular embedding of curves in projective space.

Decoding AG codes has also been the subject of extensive research. The first algorithm was proposed by Justesen et al. in [5], and the generalization to arbitrary AG codes has been given by Skorobogatov and Vlăduţ in[6]. These algorithms allow correcting errors up to half of the minimal distance minus some term depending on the genus. A productive direction has been to correct this defect. The key insight of Sudan in [7] is that at the cost of returning a list of possible close codewords, we can decode errors that exceed the usual error-correction bound. Guruswami and Sudan then gave an improved algorithm correcting errors up to the so-called Johnson bound [8].

We only introduced AG codes over curves, but we can use some more theory to define codes over higher dimensional varieties. For a review, see [4].

AG codes also have been gaining some traction because of their potential in proof systems, where they would replace polynomial encodings and potentially enable more efficient protocols.

Conclusion

So why aren't AG codes everywhere yet? The answer, for now, seems to be that their definition is quite abstract, and, to the best of our knowledge, an efficient computer implementation of all the necessary pieces has not been completed. It is our opinion, however, that this is only a matter of time and that, eventually, general AG codes will be widespread and universally adopted for communication, storage, and maybe even proof systems.

References

1. Couvreur A., Randriambololona H., Algebraic geometry codes and some applications

,

https://arxiv.org/abs/2009.01281

1. Beelen P., A survey on recursive towers and Ihara's constant,

https://arxiv.org/abs/2203.03310

1. Blake I., Heegard C., Høholdt T. and Wei V., Algebraic-geometry codes

, in IEEE Transactions on Information Theory, vol. 44, no. 6, pp. 2596–2618, Oct. 1998, doi: 10.1109/18.720550.

1. Little J. B., Algebraic geometry codes from higher dimensional varieties

,

https://arxiv.org/abs/0802.2349

1. Justesen J., Larsen K. J., Jensen H. E., Havemose A., and Høholdt T. Construction and decoding of a class of algebraic geometry codes

. IEEE Trans. Inform. Theory, 35(4): 811–821, July 1989.

1. Skorobogatov, A.N., & Vladut, S.G. (1990). On the decoding of algebraic-geometric codes

. IEEE Trans. Inf. Theory, 36

, 1051–1060.

1. Sudan M., Decoding of Reed Solomon Codes beyond the Error-Correction Bound

, Journal of Complexity, Volume 13, Issue 1, 1997, Pages 180–193, ISSN 0885–064X,
https://doi.org/10.1006/jcom.1997.0439

1. Guruswami V., Sudan M., Improved decoding of Reed-Solomon and algebraic-geometry codes

, in IEEE Transactions on Information Theory

, vol. 45, no. 6, pp. 1757–1767, Sept. 1999, doi: 10.1109/18.782097.

1. Serre J-P., Rational Points on Curves over Finite Fields

. Cours de Jean-Pierre Serre, no. 6 (1985), (red.), 242 p.http://numdam.org/item/CJPS_1985__6_/

1. https://en.wikipedia.org/wiki/Hamming_distance
2. https://en.wikipedia.org/wiki/Field_(mathematics)
3. https://en.wikipedia.org/wiki/Interpolation
4. https://en.wikipedia.org/wiki/Resolution_of_singularities
5. https://en.wikipedia.org/wiki/Field_of_fractions
6. https://en.wikipedia.org/wiki/Berlekamp–Welch_algorithm
7. https://en.wikipedia.org/wiki/Genus%E2%80%93degree_formula
8. https://en.wikipedia.org/wiki/Riemann%E2%80%93Hurwitz_formula

9. https://en.wikipedia.org/wiki/Hasse's_theorem_on_elliptic_curves

Disclaimer

This article has been prepared for the general information and understanding of the readers. No representation or warranty, express or implied, is given by Nethermind as to the accuracy or completeness of the information or opinions contained in the above article. No third party should rely on this article in any way, including without limitation as financial, investment, tax, regulatory, legal, or other advice, or interpret this article as any form of recommendation.

About us

Nethermind is a team of world-class builders and researchers. We empower enterprises and developers worldwide to access and build upon the decentralized web. Our work touches every part of the Web3 ecosystem, from our Nethermind node to fundamental cryptography research and application-layer protocol development. We're always looking for passionate people to join us in solving Ethereum's most difficult challenges. Are you interested? Check out our

[job board

](https://nethermind.io/company/)and visit our

[DeFi Research page

](https://nethermind.io/research/)for our DeFi analyses.