Jet list decoding

D. J. Bernstein
University of Illinois at Chicago

Thanks to:
NSF 1018836
NIST 60NANB10D263

No thanks to:
IEEE violating IEEE policies
and forcing authors
to take papers offline; see
cr.yp.to/writing/ieee.html

Decoding

The $\leq w$-error decoding problem for a linear code $C \subseteq \mathbf{F}_q^n$:

- Output: $c \in C$.
- Input: $v \in \mathbf{F}_q^n$ with $|v - c| \leq w$.

Note that output is unique if $w < \frac{1}{2} \min\{|c| : c \in C - \{0\}\}$.

Notation:

$|v| = \#\{i : v_i \neq 0\}$

$\quad = \text{Hamming weight of } v$;

e.g. $|v - c| = \#\{i : v_i \neq c_i\}$

$\qquad = \text{Hamming distance from } v \text{ to } c$.

Reed–Solomon decoding

Choose integer $t \geq 0$,
integer $n \geq t$, prime power $q \geq n$,
distinct $a_1, \ldots, a_n \in \mathbf{F}_q$.

Define $C \subseteq \mathbf{F}_q^n$ as the code
$\{\mathrm{ev}\, f : f \in \mathbf{F}_q[x], \deg f < n - t\}$
where $\mathrm{ev}\, f = (f(a_1), \ldots, f(a_n))$.

$\min\{|c| : c \in C - \{0\}\} = t + 1$.
Exception: ∞ if $t = n$.

1960 Peterson in some cases,
1961 Gorenstein–Zierler in more,
1965 Forney in general:
$\leq\lfloor t/2 \rfloor$-error decoding for C
takes time $n^{O(1)}$ if $q \in n^{O(1)}$.

Big research direction #1:
Decode faster.

1968 Berlekamp:
$\le\lfloor t/2\rfloor$-error decoding for C
costs $O(nt)$ operations in $\mathbf{F}_q$
plus root-finding in $\mathbf{F}_q$.
Time $n^{2+o(1)}$ for typical t, q.

1976 Justesen,
independently 1977 Sarwate:
Faster algorithm for large n,
$n(\lg n)^{2+o(1)}$ instead of $O(nt)$.
Time $n^{1+o(1)}$ for typical t, q.

Extensive literature
on further speedups.

Decoding more codes

Big research direction #2:
Modify C to expand and improve tradeoffs between q, n, $\#C$, w.

e.g. Replace $C \subseteq \mathbf{F}_q^n$, $q = 2^m$, with $\mathbf{F}_2$-subfield subcode $\mathbf{F}_2^n \cap C$.
$\#C = q^{n-t} \Rightarrow \#(\mathbf{F}_2^n \cap C) \geq 2^{n-mt}$.
Any $\leq w$-error decoder for C also works for $\mathbf{F}_2^n \cap C$.

Can take $\mathbf{F}_2^n \cap C$ where C is RS, but better to twist carefully.
Obtain classical $\mathbf{F}_2$ Goppa codes decoding twice as many errors.

Better for large n: AG codes.

List decoding

Big research direction #3:
Decode more errors *for same* C.

Maybe output c isn't unique.
Decoding problem asks for
some c with $|v - c| \le w$.
List-decoding problem asks for
all c with $|v - c| \le w$.

Trivial approach: Brute force.
e.g. guess $w - \lfloor t/2 \rfloor$ errors and
use any $\le \lfloor t/2 \rfloor$-error decoder.
(For list decoding,
use a covering set of guesses.)
Very slow for large $w - \lfloor t/2 \rfloor$.

Reed–Solomon list decoding

1996 Sudan for smaller w, 1998 Guruswami–Sudan in general:
If $w < n - \sqrt{n(n-t-1)}$ then $\leq w$-error list decoding for $C = \left\{\text{ev}\, f : f \in \mathbf{F}_q[x], \deg f < n - t\right\}$ takes time $n^{O(1)}$ if $q \in n^{O(1)}$.

Reed–Solomon list decoding

1996 Sudan for smaller w, 1998 Guruswami–Sudan in general:
If $w < n - \sqrt{n(n-t-1)}$ then $\le w$-error list decoding for $C = \{\operatorname{ev} f : f \in \mathbf{F}_q[x], \deg f < n - t\}$ takes time $n^{O(1)}$ if $q \in n^{O(1)}$.

2001 Koetter–Vardy:
Assume $q = 2^m$; write $n' = n/2$.
If $w < n' - \sqrt{n'(n'-t-1)}$ then $\le w$-error list decoding for $\mathbf{F}_2^n \cap C$ takes time $n^{O(1)}$ if $q \in n^{O(1)}$.

$n - \sqrt{n(n-t-1)} \approx t/2 + t^2/8n$.
$n' - \sqrt{n'(n'-t-1)} \approx t/2 + t^2/4n$.

Guruswami–Sudan cost analysis: $O(n^3\ell^6)$ operations in $\mathbf{F}_q$ where ℓ is an algorithm parameter.

Extensive literature on speedups and adaptations to more codes.

Critical Howgrave-Graham idea, with state-of-the-art subroutines: $n^{1+o(1)}k^{1+o(1)}\ell^{<3}$ where k is another parameter; $k < \ell$.

For Howgrave-Graham analysis see 2010 Cohn–Heninger (which also adapts to AG etc.), 2011 Bernstein "simplelist" (combining with Koetter–Vardy).

What are these parameters k, ℓ?
Obviously critical for speed.
Why not take k, ℓ to be small?

Answer: Decreasing k, ℓ forces
gap between w and its limit.
Almost all list-decoding methods
have essentially the same gap.

What are these parameters k, ℓ?
Obviously critical for speed.
Why not take k, ℓ to be small?

Answer: Decreasing k, ℓ forces gap between w and its limit.
Almost all list-decoding methods have essentially the same gap.

But not all!
Much better k, ℓ, w tradeoff in "rational" list-decoding methods:
2007 Wu "New list decoding";
2008 Bernstein "goppalist";
2011 Bernstein "jetlist".

Jets

The set of 1-jets over $\mathbf{R}$
is the quotient ring $\mathbf{R}[\epsilon]/\epsilon^2$.

Analogous to the set of complex
numbers $\mathbf{C} = \mathbf{R}[i]/(i^2 + 1)$,
but $\epsilon^2 = 0$ while $i^2 = -1$.

Multiplication of jets:
$(a + b\epsilon)(c + d\epsilon) = ac + (ad + bc)\epsilon$.

Typical construction of a jet:
differentiable $f : \mathbf{R} \to \mathbf{R}$ induces
jet $f(x + \epsilon) = f(x) + f'(x)\epsilon$
for each $x \in \mathbf{R}$.
e.g. $\sin(x + \epsilon) = \sin x + (\cos x)\epsilon$.

Recap for late sleepers

50 years ago: Polynomial-time decoding of $\leq \lfloor t/2 \rfloor$ errors in length-n Reed–Solomon code $\{\text{ev}\, f : f \in \mathbf{F}_q[x], \deg f < n - t\}$.

Big research directions since then:

3. Decode more errors.
Output might not be unique: have list of possible codewords.

2. Improve choice of code: classical Goppa codes, AG, et al.

1. Decode faster.

Lattice-basis reduction

Define $L = (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z}$
$= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

What is the shortest
nonzero vector in L?

Lattice-basis reduction

Define $L = (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z}$ $= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

What is the shortest nonzero vector in L?

$L = (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z}$

Lattice-basis reduction

Define $L = (0,24)\mathbf{Z} + (1,17)\mathbf{Z}$
$= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

What is the shortest
nonzero vector in L?

$$L = (0,24)\mathbf{Z} + (1,17)\mathbf{Z}$$
$$\quad = (-1,7)\mathbf{Z} + (1,17)\mathbf{Z}$$

Lattice-basis reduction

Define $L = (0,24)\mathbf{Z} + (1,17)\mathbf{Z}$
$= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

What is the shortest nonzero vector in L?

$$\begin{aligned} L &= (0,24)\mathbf{Z} + (1,17)\mathbf{Z} \\ &= (-1,7)\mathbf{Z} + (1,17)\mathbf{Z} \\ &= (-1,7)\mathbf{Z} + (3,3)\mathbf{Z} \end{aligned}$$

Lattice-basis reduction

Define $L = (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z}$
$= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

What is the shortest nonzero vector in L?

$$
\begin{aligned}
L &= (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z} \\
&= (-1, 7)\mathbf{Z} + (1, 17)\mathbf{Z} \\
&= (-1, 7)\mathbf{Z} + (3, 3)\mathbf{Z} \\
&= (-4, 4)\mathbf{Z} + (3, 3)\mathbf{Z}.
\end{aligned}
$$

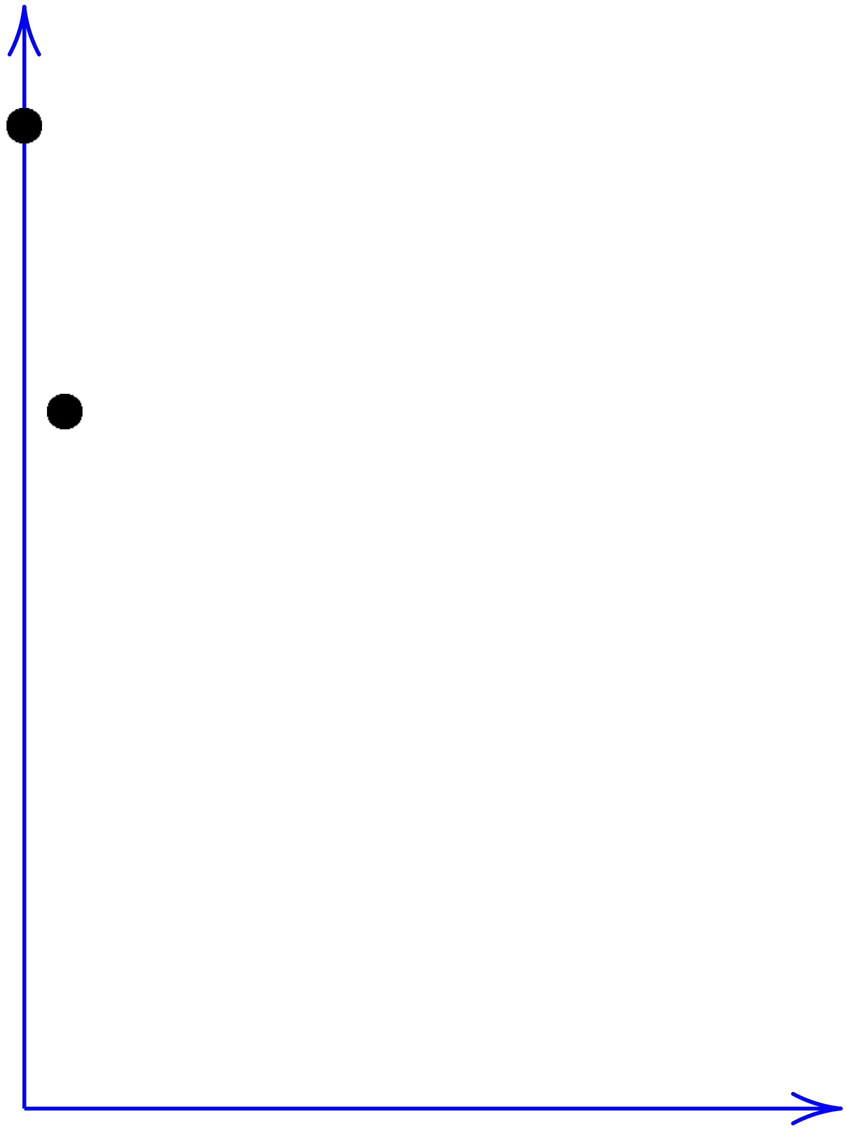
Lattice-basis reduction

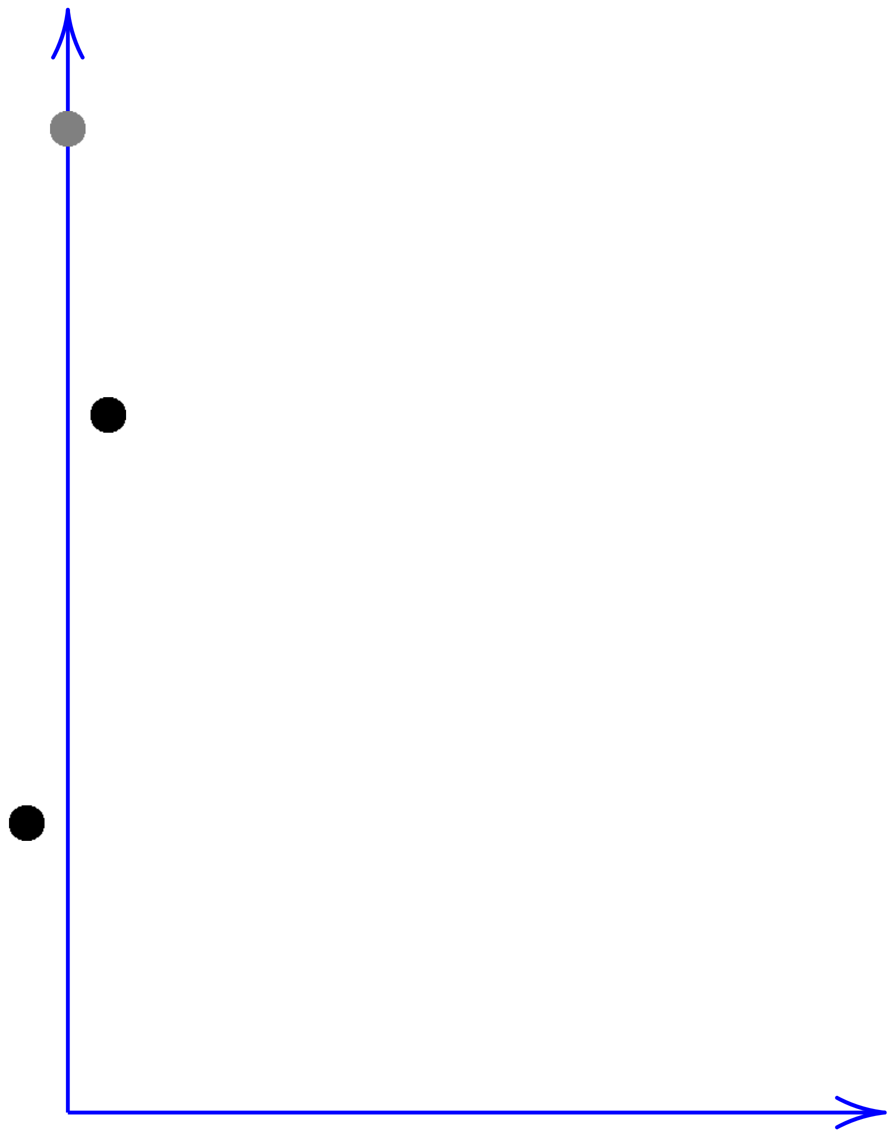
Define $L = (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z}$
$= \{(b, 24a + 17b) : a, b \in \mathbf{Z}\}$.

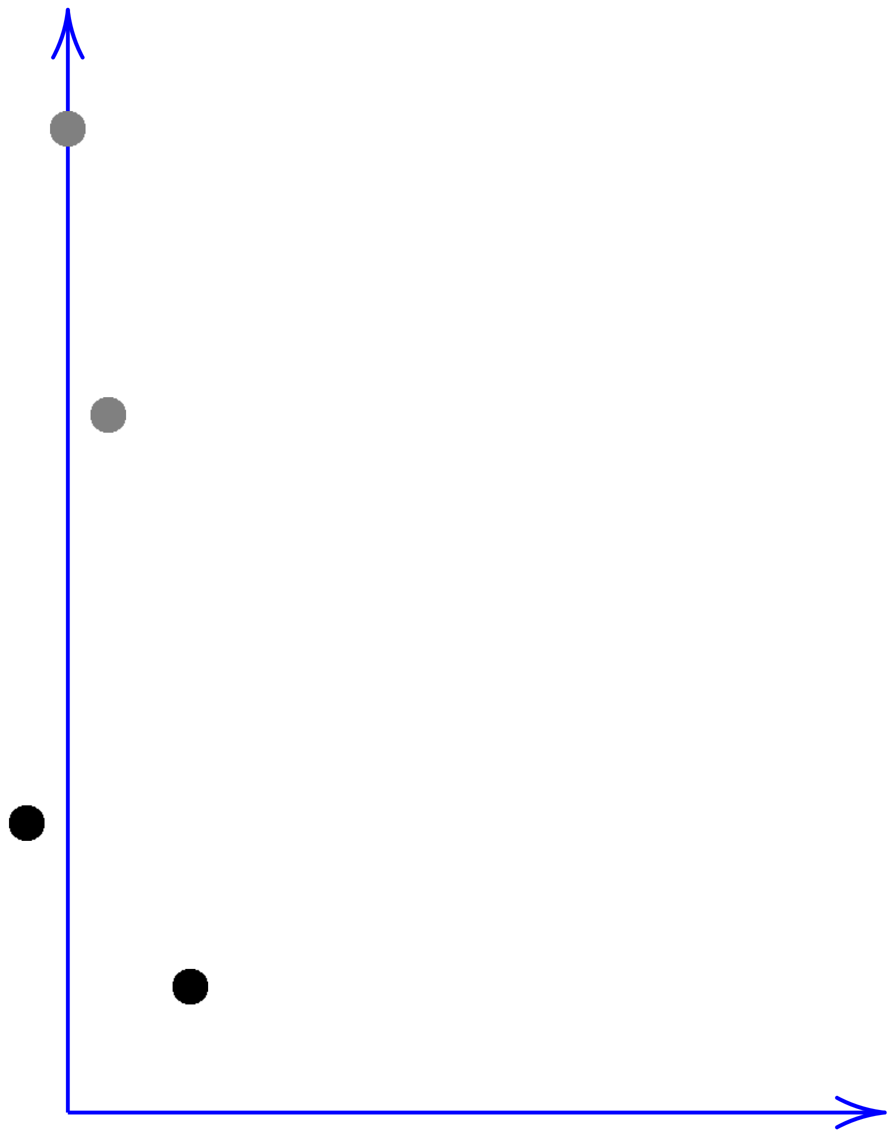
What is the shortest
nonzero vector in L?

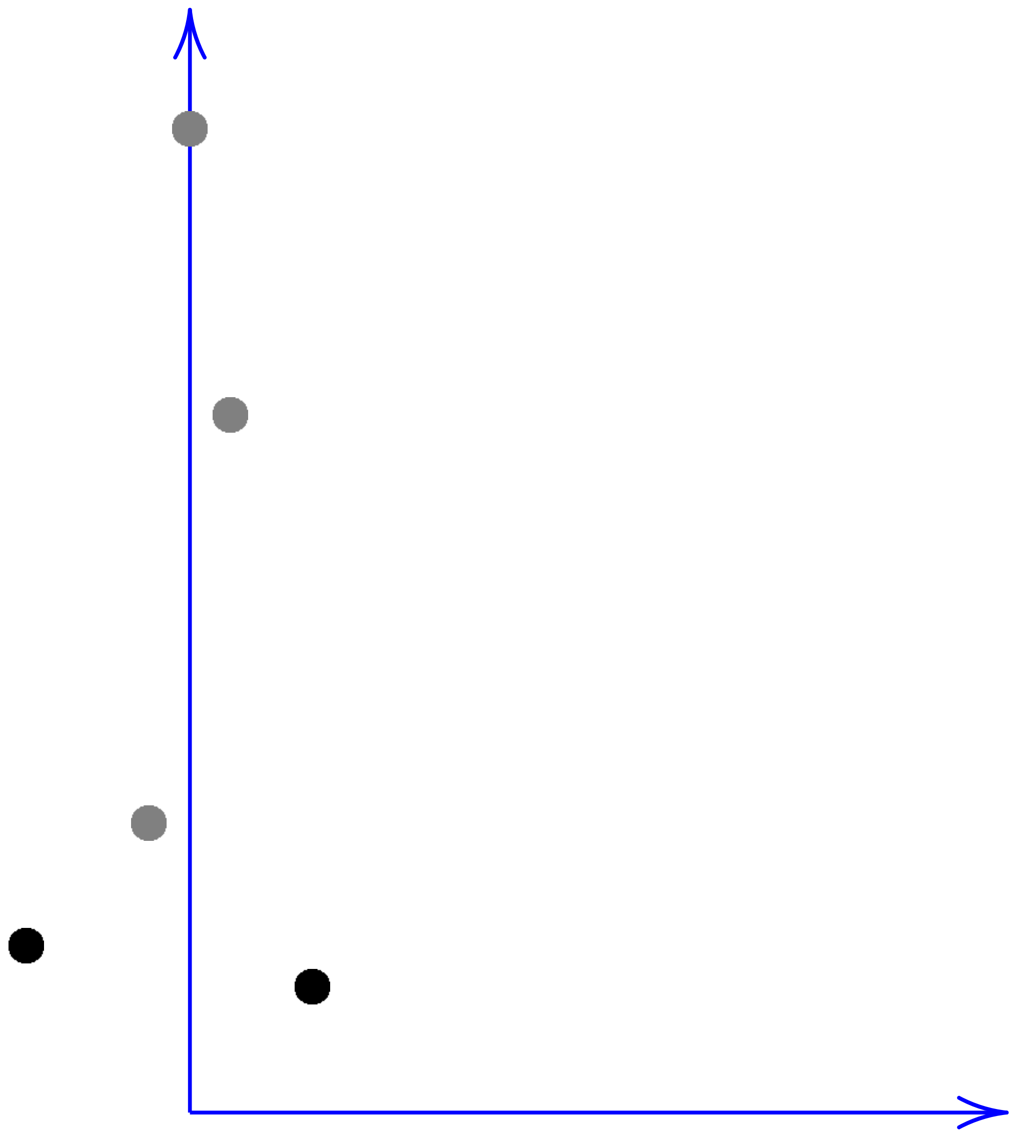
$$\begin{aligned} L &= (0, 24)\mathbf{Z} + (1, 17)\mathbf{Z} \\ &= (-1, 7)\mathbf{Z} + (1, 17)\mathbf{Z} \\ &= (-1, 7)\mathbf{Z} + (3, 3)\mathbf{Z} \\ &= (-4, 4)\mathbf{Z} + (3, 3)\mathbf{Z}. \end{aligned}$$

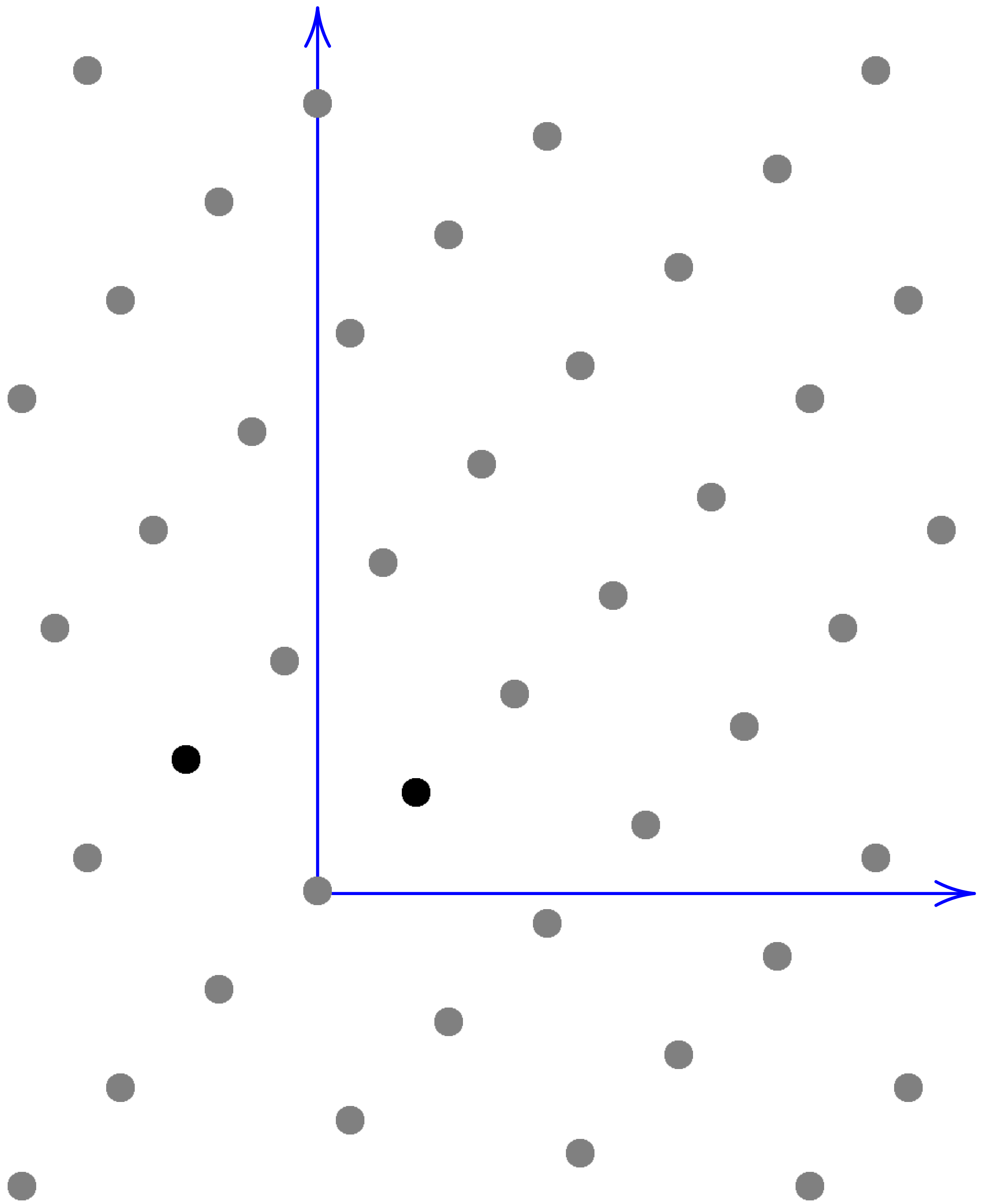
$(-4, 4), (3, 3)$ are orthogonal.
Shortest vectors in L are
$(0, 0)$, $(3, 3)$, $(-3, -3)$.











Another example:

Define $L = (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z}$.

What is the shortest nonzero vector in L?

Another example:

Define $L = (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z}$.

What is the shortest nonzero vector in L?

$L = (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z}$

Another example:

Define $L = (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z}$.

What is the shortest nonzero vector in L?

$$
\begin{aligned}
L &= (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z} \\
&= (-1, 8)\mathbf{Z} + (1, 17)\mathbf{Z}
\end{aligned}
$$

Another example:

Define $L = (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z}$.

What is the shortest nonzero vector in L?

$$
\begin{aligned}
L &= (0, 25)\mathbf{Z} + (1, 17)\mathbf{Z} \\
&= (-1, 8)\mathbf{Z} + (1, 17)\mathbf{Z} \\
&= (-1, 8)\mathbf{Z} + (3, 1)\mathbf{Z}.
\end{aligned}
$$

Another example:

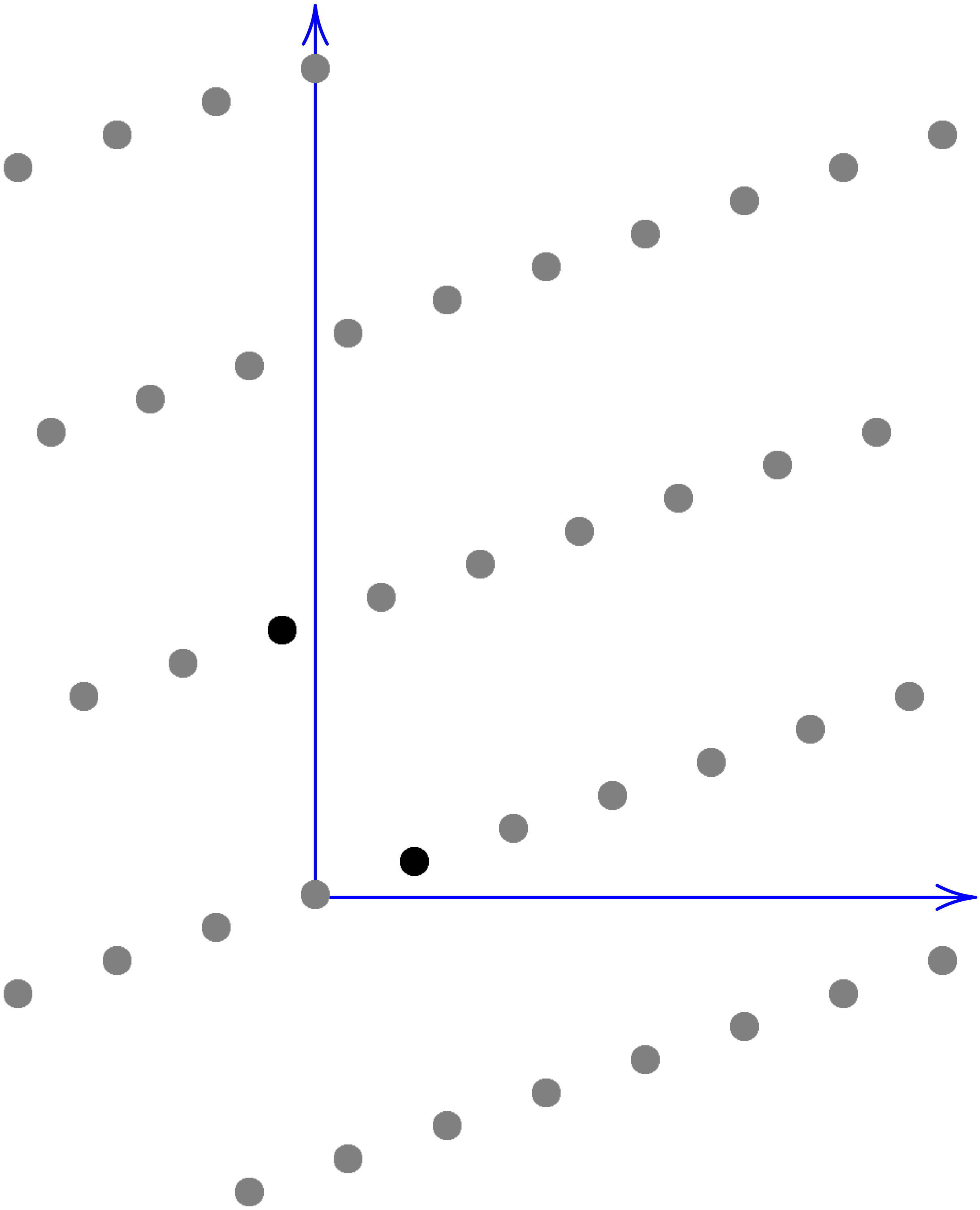
Define $L = (0,25)\mathbf{Z} + (1,17)\mathbf{Z}$.

What is the shortest nonzero vector in L?

$$\begin{aligned} L &= (0,25)\mathbf{Z} + (1,17)\mathbf{Z} \\ &= (-1,8)\mathbf{Z} + (1,17)\mathbf{Z} \\ &= (-1,8)\mathbf{Z} + (3,1)\mathbf{Z}. \end{aligned}$$

Nearly orthogonal.

Shortest vectors in L are $(0,0)$, $(3,1)$, $(-3,-1)$.



Polynomial lattices

Define $R = \mathbf{F}_2[x]$,
$r_0 = (101000)_x = x^5 + x^3 \in R$,
$r_1 = (10011)_x = x^4 + x + 1 \in R$,
$L = (0, r_0)R + (1, r_1)R$.

What is the shortest nonzero vector in L?

Polynomial lattices

Define $R = \mathbf{F}_2[x]$,
$r_0 = (101000)_x = x^5 + x^3 \in R$,
$r_1 = (10011)_x = x^4 + x + 1 \in R$,
$L = (0, r_0)R + (1, r_1)R$.

What is the shortest nonzero vector in L?

$$L = (0, 101000)R + (1, 10011)R$$

Polynomial lattices

Define $R = \mathbf{F}_2[x]$,
$r_0 = (101000)_x = x^5 + x^3 \in R$,
$r_1 = (10011)_x = x^4 + x + 1 \in R$,
$L = (0, r_0)R + (1, r_1)R$.

What is the shortest nonzero vector in L?

$$L = (0, 101000)R + (1, 10011)R$$
$$= (10, 1110)R + (1, 10011)R$$

Polynomial lattices

Define $R = \mathbf{F}_2[x]$,
$r_0 = (101000)_x = x^5 + x^3 \in R$,
$r_1 = (10011)_x = x^4 + x + 1 \in R$,
$L = (0, r_0)R + (1, r_1)R$.

What is the shortest
nonzero vector in L?

$$
\begin{aligned}
L &= (0, 101000)R + (1, 10011)R \\
&= (10, 1110)R + (1, 10011)R \\
&= (10, 1110)R + (111, 1)R.
\end{aligned}
$$

Polynomial lattices

Define $R = \mathbf{F}_2[x]$,
$r_0 = (101000)_x = x^5 + x^3 \in R$,
$r_1 = (10011)_x = x^4 + x + 1 \in R$,
$L = (0, r_0)R + (1, r_1)R$.

What is the shortest nonzero vector in L?

$$\begin{aligned} L &= (0, 101000)R + (1, 10011)R \\ &= (10, 1110)R + (1, 10011)R \\ &= (10, 1110)R + (111, 1)R. \end{aligned}$$

$(111, 1)$: shortest nonzero vector.
$(10, 1110)$: shortest independent vector.

Degree of $(q, r) \in \mathbf{F}_2[x] \times \mathbf{F}_2[x]$ is defined as $\max\{\deg q, \deg r\}$.

Can use other metrics, or equivalently rescale L.

e.g. Define $L \subseteq \mathbf{F}_2[\sqrt{x}] \times \mathbf{F}_2[\sqrt{x}]$ as $(0, r_0\sqrt{x})R + (1, r_1\sqrt{x})R$.

Successive generators for L:
$(0, 101000\sqrt{x})$, degree 5.5.
$(1, 10011\sqrt{x})$, degree 4.5.
$(10, 1110\sqrt{x})$, degree 3.5.
$(111, 1\sqrt{x})$, degree 2.

Warning: Sometimes shortest independent vector is *after* shortest nonzero vector.

e.g. Define
$r_0 = 101000$, $r_1 = 10111$,
$L = (0, r_0\sqrt{x})R + (1, r_1\sqrt{x})R$.

Successive generators for L:
$(0, 101000\sqrt{x})$, degree 5.5.
$(1, 10111\sqrt{x})$, degree 4.5.
$(10, 110\sqrt{x})$, degree 2.5.
$(1101, 11\sqrt{x})$, degree 3.

For any $r_0, r_1 \in R = \mathbf{F}_q[x]$ with $\deg r_0 > \deg r_1$:

Euclid/Stevin computation: Define $r_2 = r_0 \bmod r_1$, $r_3 = r_1 \bmod r_2$, etc.

Extended: $q_0 = 0$; $q_1 = 1$; $q_{i+2} = q_i - \lfloor r_i / r_{i+1} \rfloor q_{i+1}$. Then $q_i r_1 \equiv r_i \pmod{r_0}$.

Lattice view: Have $(0, r_0\sqrt{x})R + (1, r_1\sqrt{x})R = (q_i, r_i\sqrt{x})R + (q_{i+1}, r_{i+1}\sqrt{x})R$.

Can continue until $r_{i+1} = 0$. $\gcd\{r_0, r_1\} = r_i / \operatorname{leadcoeff} r_i$.

Reducing lattice basis for L is a "half gcd" computation, stopping halfway to the gcd.

$\deg r_i$ decreases; $\deg q_i$ increases; $\deg q_{i+1} + \deg r_i = \deg r_0$.

Say j is minimal with $\deg r_j\sqrt{x} \le (\deg r_0)/2$.
Then $\deg q_j \le (\deg r_0)/2$ so $\deg(q_j, r_j\sqrt{x}) \le (\deg r_0)/2$.
Shortest nonzero vector.

$(q_{j+\epsilon}, r_{j+\epsilon}\sqrt{x})$ has degree $\deg r_0\sqrt{x} - \deg(q_j, r_j\sqrt{x})$ for some $\epsilon \in \{-1, 1\}$.
Shortest independent vector.

Proof of "shortest":

Take any $(q, r\sqrt{x})$ in lattice.

$$(q, r\sqrt{x}) = u(q_j, r_j\sqrt{x}) \\ + v(q_{j+\epsilon}, r_{j+\epsilon}\sqrt{x})$$

for some $u, v \in R$.

$q_j r_{j+\epsilon} - q_{j+\epsilon} r_j = \pm r_0$
so $v = \pm(r q_j - q r_j)/r_0$
and $u = \pm(q r_{j+\epsilon} - r q_{j+\epsilon})/r_0$.

If $\deg(q, r\sqrt{x})$
$< \deg(q_{j+\epsilon}, r_{j+\epsilon}\sqrt{x})$
then $\deg v < 0$ so $v = 0$;
i.e., any vector in lattice
shorter than $(q_{j+\epsilon}, r_{j+\epsilon}\sqrt{x})$
is a multiple of $(q_j, r_j\sqrt{x})$.

Classical binary Goppa codes

Parameters determining the code:
integers $n \ge 0$, $m \ge 1$, $t \ge 0$;
distinct $a_1, \ldots, a_n \in \mathbf{F}_{2^m}$;
monic $g \in \mathbf{F}_{2^m}[x]$ of degree t
with $g(a_1) \cdots g(a_n) \ne 0$.

The code: Define $\Gamma \subseteq \mathbf{F}_2^n$
as set of $(c_1, \ldots, c_n)$ with
$\sum_i c_i/(x - a_i) = 0$ in $\mathbf{F}_{2^m}[x]/g$.

$\lg \#\Gamma \ge n - mt$.
$\min\{|c| : c \in \Gamma - \{0\}\} \ge t + 1$.
Better bounds in the BCH case
$g = x^t$ and in many other cases.

Say we receive $v = c + e$.

Define $D, E \in \mathbf{F}_{2^m}[x]$ by

$D = \prod_{i: e_i \neq 0} (x - a_i)$ and

$E = \sum_i De_i/(x - a_i)$.

Say we receive $v = c + e$.
Define $D, E \in \mathbf{F}_{2^m}[x]$ by
$D = \prod_{i:e_i \neq 0}(x - a_i)$ and
$E = \sum_i De_i/(x - a_i)$.

Lift $\sum_i v_i/(x - a_i)$ from $\mathbf{F}_{2^m}[x]/g$
to $s \in \mathbf{F}_{2^m}[x]$ with $\deg s < t$.
Find shortest nonzero
$(q_j, r_j\sqrt{x})$ in the lattice $L =$
$(0, g\sqrt{x})\mathbf{F}_{2^m}[x] + (1, s\sqrt{x})\mathbf{F}_{2^m}[x]$.

Say we receive $v = c + e$.

Define $D, E \in \mathbf{F}_{2^m}[x]$ by
$D = \prod_{i:e_i \neq 0}(x - a_i)$ and
$E = \sum_i De_i/(x - a_i)$.

Lift $\sum_i v_i/(x - a_i)$ from $\mathbf{F}_{2^m}[x]/g$ to $s \in \mathbf{F}_{2^m}[x]$ with $\deg s < t$.
Find shortest nonzero $(q_j, r_j\sqrt{x})$ in the lattice $L = (0, g\sqrt{x})\mathbf{F}_{2^m}[x] + (1, s\sqrt{x})\mathbf{F}_{2^m}[x]$.

Fact: If $|e| \le t/2$
then $E/D = r_j/q_j$ so
D is monic denominator of r_j/q_j.

Say we receive $v = c + e$.

Define $D, E \in \mathbf{F}_{2^m}[x]$ by

$D = \prod_{i:e_i \neq 0}(x - a_i)$ and

$E = \sum_i De_i/(x - a_i)$.

Lift $\sum_i v_i/(x - a_i)$ from $\mathbf{F}_{2^m}[x]/g$ to $s \in \mathbf{F}_{2^m}[x]$ with $\deg s < t$.

Find shortest nonzero $(q_j, r_j\sqrt{x})$ in the lattice $L = (0, g\sqrt{x})\mathbf{F}_{2^m}[x] + (1, s\sqrt{x})\mathbf{F}_{2^m}[x]$.

Fact: If $|e| \leq t/2$ then $E/D = r_j/q_j$ so D is monic denominator of r_j/q_j.

$e_i = 0$ if $D(a_i) \neq 0$.

$e_i = E(a_i)/D'(a_i)$ if $D(a_i) = 0$.

Why does this work?

$\sum_i e_i/(x - a_i) = E/D$ and
$\sum_i c_i/(x - a_i) = 0$ in $\mathbf{F}_{2^m}[x]/g$
so $s = E/D$ in $\mathbf{F}_{2^m}[x]/g$
so $(D, E\sqrt{x}) \in L$.

Why does this work?

$\sum_i e_i/(x - a_i) = E/D$ and
$\sum_i c_i/(x - a_i) = 0$ in $\mathbf{F}_{2^m}[x]/g$
so $s = E/D$ in $\mathbf{F}_{2^m}[x]/g$
so $(D, E\sqrt{x}) \in L$.

$(D, E\sqrt{x})$ is a short vector:
$\deg(D, E\sqrt{x}) \le |e| \le t/2$
$< t + 1/2 - \deg(q_j, r_j\sqrt{x})$.

Why does this work?

$\sum_i e_i/(x - a_i) = E/D$ and
$\sum_i c_i/(x - a_i) = 0$ in $\mathbf{F}_{2^m}[x]/g$
so $s = E/D$ in $\mathbf{F}_{2^m}[x]/g$
so $(D, E\sqrt{x}) \in L$.

$(D, E\sqrt{x})$ is a short vector:
$\deg(D, E\sqrt{x}) \le |e| \le t/2$
$< t + 1/2 - \deg(q_j, r_j\sqrt{x})$.

Recall "shortest" proof:
$(D, E\sqrt{x}) \in (q_j, r_j\sqrt{x})\mathbf{F}_{2^m}[x]$,
so $E/D = r_j/q_j$. Done!

Euclid decoding: 1975 Sugiyama–Kasahara–Hirasawa–Namekawa.

List decoding for these codes

What if $|e| > t/2$?

Find shortest nonzero $(D_0, E_0\sqrt{x})$ and independent $(D_1, E_1\sqrt{x})$ in $(0, g\sqrt{x})\mathbf{F}_{2^m}[x] + (1, s\sqrt{x})\mathbf{F}_{2^m}[x]$, with degrees $t/2 - \delta$ and $t/2 + 1/2 + \delta$ for some $\delta \in \{0, 1/2, 1, 3/2, \ldots\}$.

Know that $(D, E\sqrt{x}) = u(D_0, E_0\sqrt{x}) + v(D_1, E_1\sqrt{x})$;
$v = \pm(ED_0 - DE_0)/g \in \mathbf{F}_{2^m}[x]$,
$u = \pm(DE_1 - ED_1)/g \in \mathbf{F}_{2^m}[x]$,
$\deg v \le |e| - t/2 - 1/2 - \delta$,
$\deg u \le |e| - t/2 + \delta$.

Critical facts about D:

- $D = uD_0 + vD_1$ with known D_0 and D_1, bounded u and v.
- D divides known $N = \prod_i (x - a_i)$.

Critical facts about D:

- $D = uD_0 + vD_1$ with known D_0 and D_1, bounded u and v.
- D divides known $N = \prod_i (x - a_i)$.

Can use these facts to quickly compute all possible D for surprisingly large $|e|$.

Critical facts about D:

- $D = uD_0 + vD_1$ with known D_0 and D_1, bounded u and v.
- D divides known $N = \prod_i (x - a_i)$.

Can use these facts to quickly compute all possible D for surprisingly large $|e|$.

This is essentially 2007 Wu.

2008 Bernstein: combine with Patterson.

1998 Guruswami–Sudan: same $|e|$ limit but much slower.

Algorithm parameters:
"multiplicity" $k \ge 1$;
"lattice dimension" $\ell \ge k + 1$.

Assume $\gcd\{D_1, N\} = 1$.
Otherwise add
constant multiple of D_0 to D_1,
extending field if necessary;
see 2008 Bernstein for analysis.

Lift D_0/D_1 from $\mathbf{F}_{2^m}[x]/N$
to $S \in \mathbf{F}_{2^m}[x]$ with $\deg S < n$.
Then $Su + v \in D\mathbf{F}_{2^m}[x]$.

Note that both u and $x^\theta v$ have
degree $\le \lfloor |e| - t/2 + \delta \rfloor$ where
$\theta = \lfloor t/2 + \delta \rfloor - \lfloor t/2 - 1/2 - \delta \rfloor$.

For $k = 1$: In $\mathbf{F}_{2^m}(x)[y]$ define

$G_0 = N$,

$G_1 = S + x^{-\theta} y$,

$G_2 = (S + x^{-\theta} y) x^{-\theta} y$,

$\vdots$,

$G_{\ell-1} = (S + x^{-\theta} y)(x^{-\theta} y)^{\ell-2}$.

Substituting $y = x^{\theta} v/u$ and multiplying by $u^{\ell-1}$ produces $Nu^{\ell-1}, (Su + v)u^{\ell-2}, \dots, Su + v$, all of which are in $D\mathbf{F}_{2^m}[x]$.

$u^{\ell-1} Q(x^{\theta} v/u) \in D\mathbf{F}_{2^m}[x]$ for any $Q \in G_0 \mathbf{F}_{2^m}[x] + \cdots + G_{\ell-1} \mathbf{F}_{2^m}[x]$.

View all of these polynomials as coefficient vectors in $\mathbf{F}_{2^m}(x)^\ell$.
$G_0, G_1, \ldots, G_{\ell-1}$ have determinant $Nx^{-\ell(\ell-1)\theta/2}$, of degree $n - \ell(\ell-1)\theta/2$.

Use ℓ-dim lattice-basis reduction to find short nonzero Q: $\deg Q_i \le n/\ell - (\ell-1)\theta/2$.

If $|e| > n/\ell + (\ell-1)\lfloor |e| - t/2 + \delta - \theta/2 \rfloor$ then $\deg Q_i (x^\theta v)^i u^{\ell-1-i} < |e|$ so $\deg u^{\ell-1} Q(x^\theta v/u) < |e|$ so $Q(x^\theta v/u) = 0$.
Find u, v by finding roots of Q.

For general k: Redefine G_i to obtain multiples of D^k.

$G_0 = N^k$;

$G_1 = (S + x^{-\theta}y)N^{k-1}$;

$G_2 = (S + x^{-\theta}y)^2 N^{k-2}$;

$\vdots$

$G_k = (S + x^{-\theta}y)^k$;

$\vdots$

$G_{\ell-1} = (S + x^{-\theta}y)^k (x^{-\theta}y)^{\ell-k-1}$.

$\deg Q_i \le nk(k+1)/2\ell - (\ell-1)\theta/2$.

If $k|e| > nk(k+1)/2\ell + (\ell-1)\lfloor |e| - t/2 + \delta - \theta/2 \rfloor$ then $Q(x^\theta v/u) = 0$.

e.g. $t = 0.1n$, $w = 0.051n$:
smallest parameters are
$k = 4$, $\ell = 80$.

For comparison,
Guruswami–Sudan require
multiplicity k and
lattice dimension ℓ to satisfy
$nk(k+1)/2\ell + (\ell-1)(n-t-1)/2$
$< k(n-|e|)$.

e.g. $t = 0.1n$, $w = 0.051n$:
smallest parameters are
$k = 75$, $\ell = 80$.

Jet list decoding

Recall $D = \prod_{i:e_i \neq 0}(x - a_i)$
and $E = \sum_i De_i/(x - a_i)$.

$e_i \in \{0, 1\}$
so $E = \sum_i D/(x - a_i) = D'$.

One consequence:
$\Gamma_2(g) = \Gamma_2(g^2)$ if g is squarefree.
This doubles t, drastically
increasing # errors decoded.

But $\Gamma_2(g^2)$ decoders vary
in effectiveness and efficiency.

1968 Berlekamp decodes
t errors for $\Gamma_2(g^2)$.
1975 Patterson: same, faster.

1998 Guruswami–Sudan:
$\approx t + t^2/2n$ errors.
2007 Wu: same, faster;
the "rational" speedup.
2008 Bernstein: even faster;
"rational" $+$ Patterson.

1968 Berlekamp decodes
t errors for $\Gamma_2(g^2)$.
1975 Patterson: same, faster.

1998 Guruswami–Sudan:
$\approx t + t^2/2n$ errors.
2007 Wu: same, faster;
the "rational" speedup.
2008 Bernstein: even faster;
"rational" $+$ Patterson.

2001 Koetter–Vardy:
$\approx t + t^2/n$ errors.
Can "rational" algorithms
correct $> t + t^2/2n$ errors?

1968 Berlekamp decodes
t errors for $\Gamma_2(g^2)$.
1975 Patterson: same, faster.

1998 Guruswami–Sudan:
$\approx t + t^2/2n$ errors.
2007 Wu: same, faster;
the “rational” speedup.
2008 Bernstein: even faster;
“rational” + Patterson.

2001 Koetter–Vardy:
$\approx t + t^2/n$ errors.
Can “rational” algorithms
correct $> t + t^2/2n$ errors?
Yes! Jet list decoding.

Works for arbitrary $\Gamma_2(g)$.

Notation: $N, D, E, \ldots$ as before.

D divides N so the jet
$D(x+\epsilon) = D + \epsilon D' = D + \epsilon E$
divides $N(x+\epsilon) = N + \epsilon N'$.

$(D+\epsilon E)(D-\epsilon E)$ divides
$(N+\epsilon N')(D-\epsilon E)$ so
D^2 divides $N'D - NE$.

$(D, E) = u(D_0, E_0) + v(D_1, E_1)$
so $N'D - NE =$
$v(N'D_1 - NE_1) + u(N'D_0 - NE_0)$.

Lift $(N'D_0 - NE_0)/(N'D_1 - NE_1)$
from $\mathbf{F}_{2^m}[x]/N^2$ to $S \in \mathbf{F}_{2^m}[x]$.
Then $Su + v \in D^2 \mathbf{F}_{2^m}[x]$.

$G_0 = (N^2)^k$;
$G_1 = (S + x^{-\theta}y)(N^2)^{k-1}$;
$G_2 = (S + x^{-\theta}y)^2(N^2)^{k-2}$;
$\vdots$
$G_k = (S + x^{-\theta}y)^k$;
$\vdots$
$G_{\ell-1} = (S + x^{-\theta}y)^k(x^{-\theta}y)^{\ell-k-1}$.

$u^{\ell-1}Q(x^\theta v/u) \in D^{2k}\mathbf{F}_{2^m}[x]$ if
$Q \in G_0\mathbf{F}_{2^m}[x] + \cdots + G_{\ell-1}\mathbf{F}_{2^m}[x]$.

Roots of shortest nonzero Q
include $x^\theta v/u$
if $2k|e| > nk(k+1)/\ell +$
$(\ell - 1)\lfloor |e| - t/2 + \delta - \theta/2 \rfloor$.

e.g. $t = 0.1n$, $w = 0.051n$:
smallest parameters are
$k = 1$, $\ell = 26$.

e.g. $t = 0.1n$, $w = 0.0521n$:
smallest parameters are
$k = 4$, $\ell = 80$.

Compared to Koetter–Vardy:
same limit on w, but
much smaller k for each w.

Same achieved by 2007 Wu
in one special case, BCH.
Jet list decoding is faster
(thanks to Howgrave-Graham)
and more general.