List Decoding Reed-Solomon Codes in the Lee, Euclidean, and Other Metrics

Math Crypto Workshop at FAU

October 11-12, 2025

Alexandra Veliche Hostetler

(University of South Florida)

Chris Peikert

(University of Michigan)

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \ldots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \ldots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}}$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}} \quad \begin{array}{|c|} \hline f(\alpha_1) \\ \hline f(\alpha_2) \\ \hline \vdots \\ \hline f(\alpha_n) \\ \hline \end{array}$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$

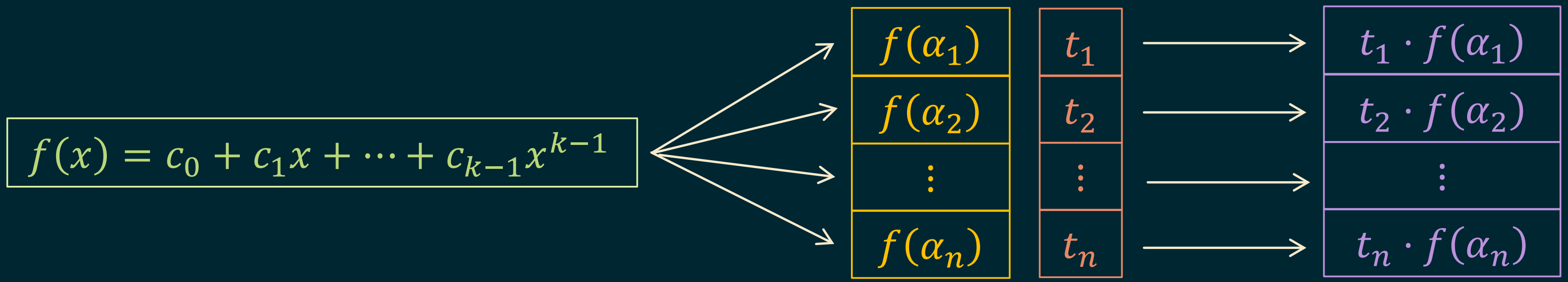
$$\boxed{f(x) = c_0 + c_1 x + \cdots + c_{k-1} x^{k-1}} \;\longrightarrow\; \begin{bmatrix} f(\alpha_1) \\ f(\alpha_2) \\ \vdots \\ f(\alpha_n) \end{bmatrix} \quad \begin{bmatrix} t_1 \\ t_2 \\ \vdots \\ t_n \end{bmatrix}$$

Generalized Reed-Solomon Codes

GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \{(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n)) : f \in \mathbb{F}_q[x], \deg(f) < k\} \subseteq \mathbb{F}_q^n.$$

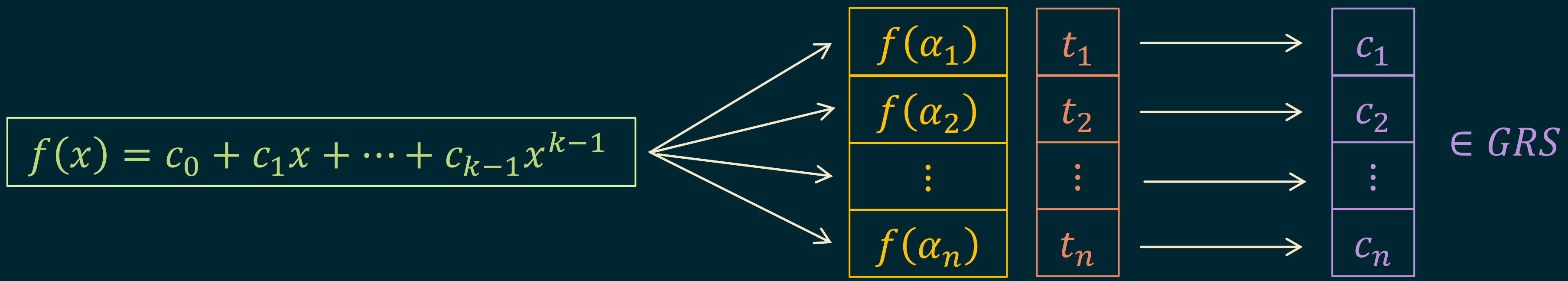


Generalized Reed-Solomon Codes

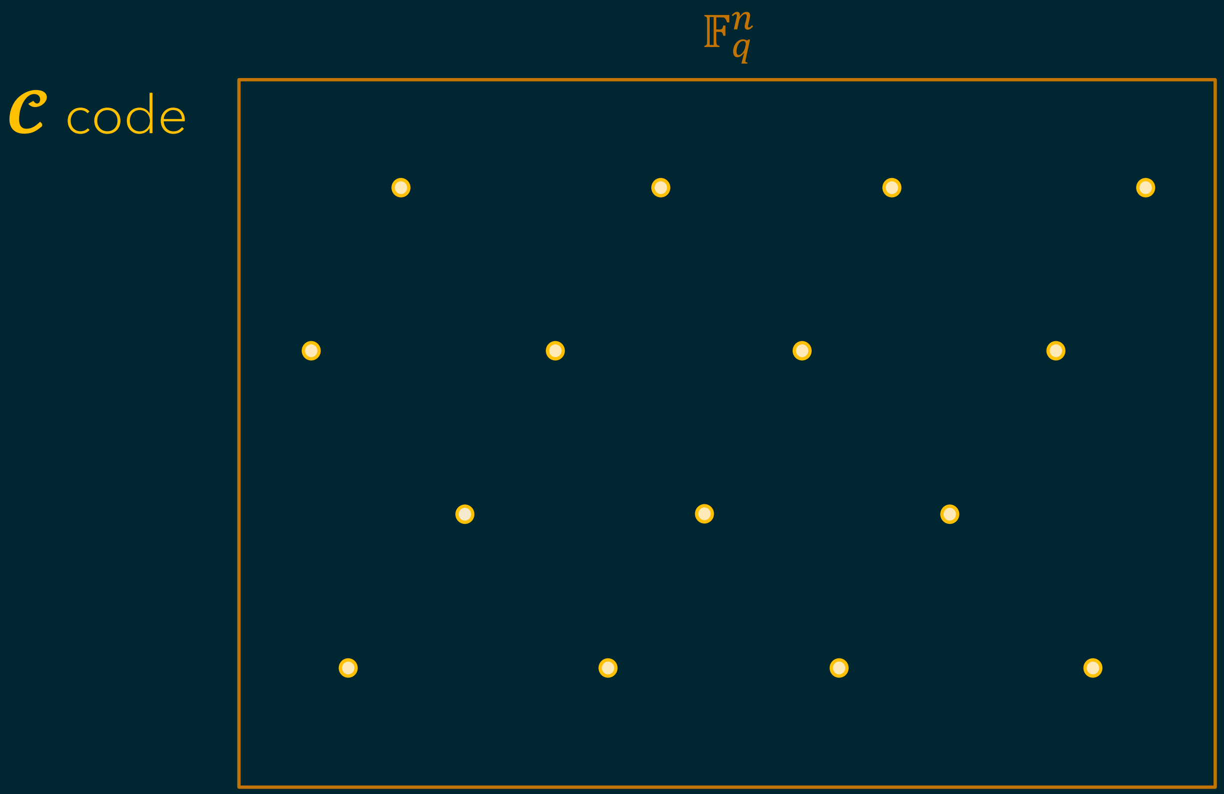
GRS Code: n blocklength, $\mathbb{F}_q$ finite field of size $q \geq n$, k dimension,

$\boldsymbol{\alpha} = (\alpha_1, \dots, \alpha_n) \in \mathbb{F}_q^n$ evaluation points, $\boldsymbol{t} = (t_1, \dots, t_n) \in \mathbb{F}_q^n$ non-zero twist factors

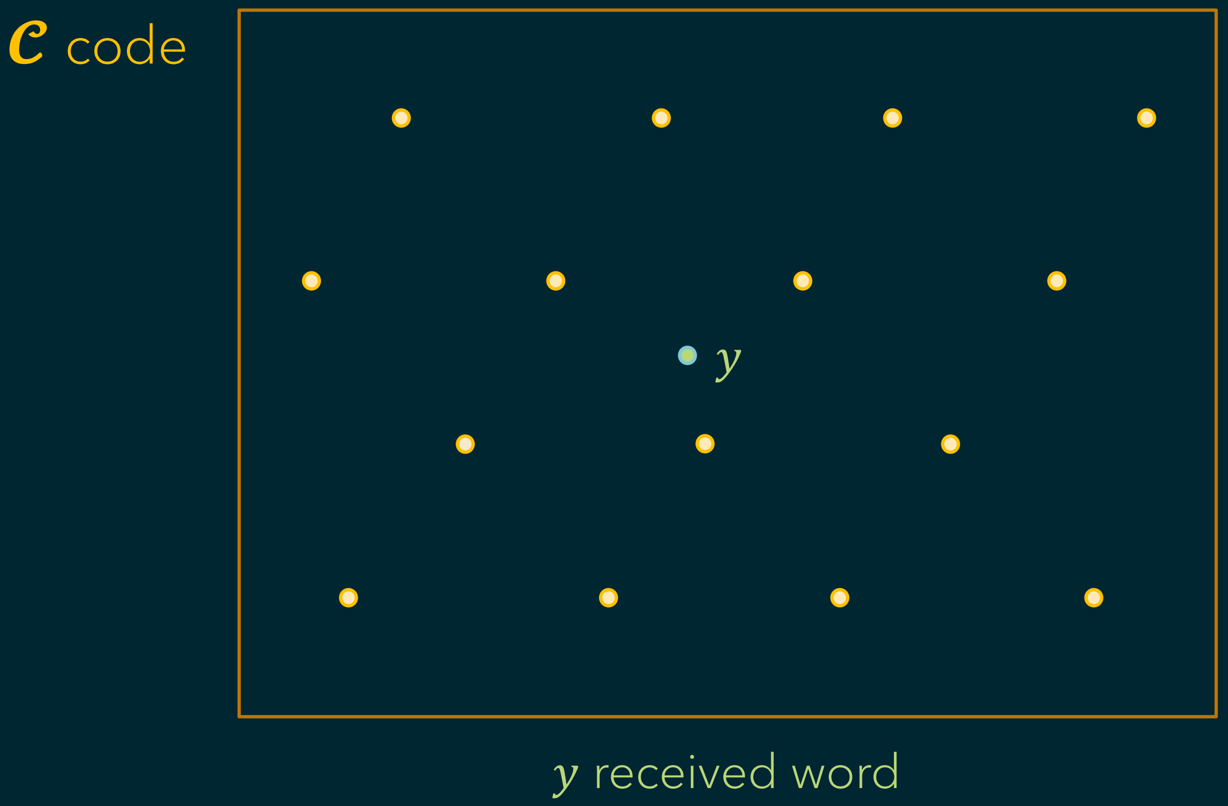
$$GRS_{q,k}(\boldsymbol{\alpha}, \boldsymbol{t}) \coloneqq \left\{ \left(t_1 \cdot f(\alpha_1), \dots, t_n \cdot f(\alpha_n) \right) : f \in \mathbb{F}_q[x], \deg(f) < k \right\} \subseteq \mathbb{F}_q^n.$$



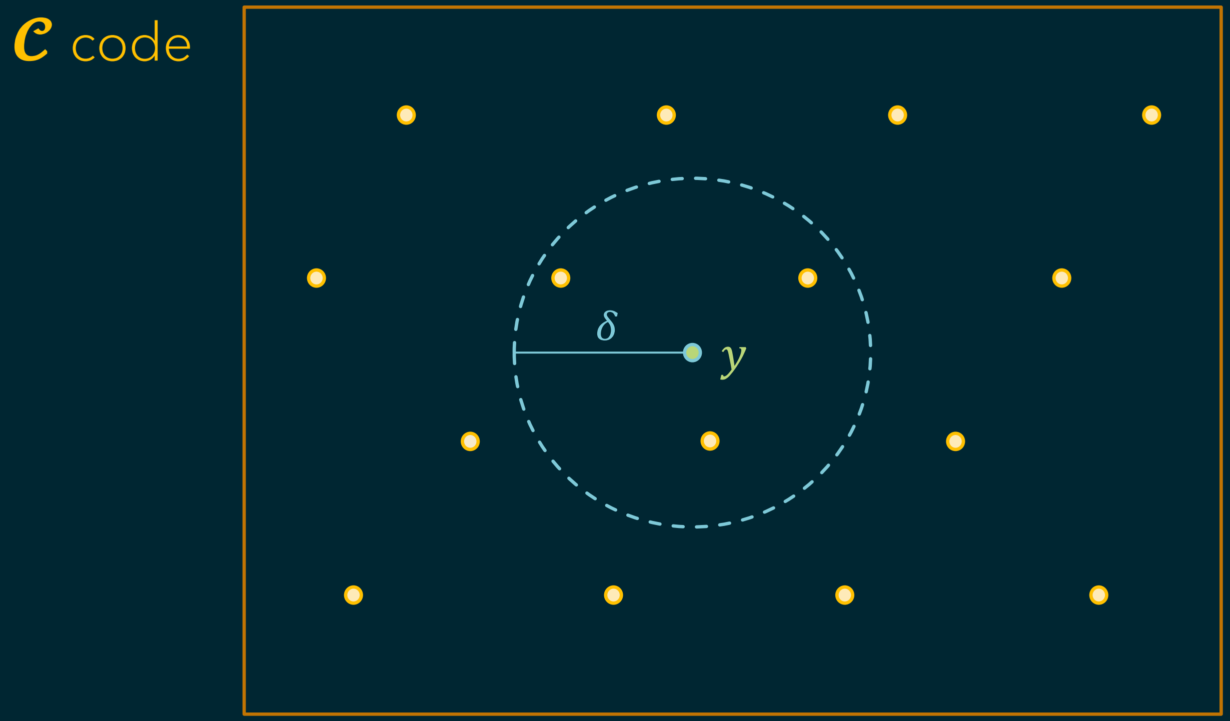
List-Decoding Problem



List-Decoding Problem

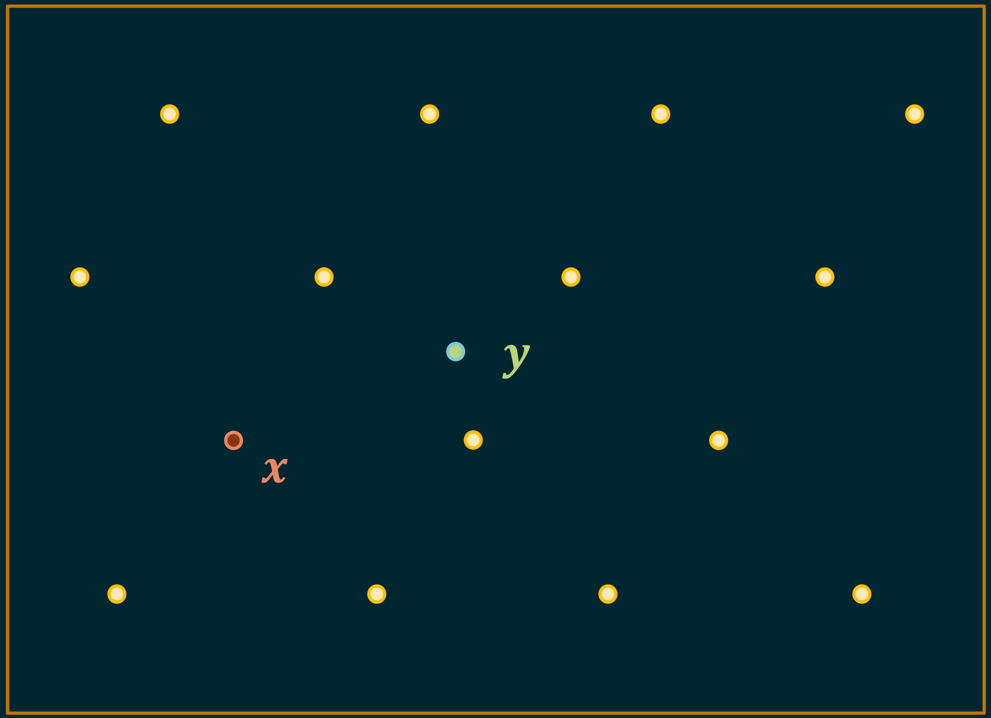


List-Decoding Problem



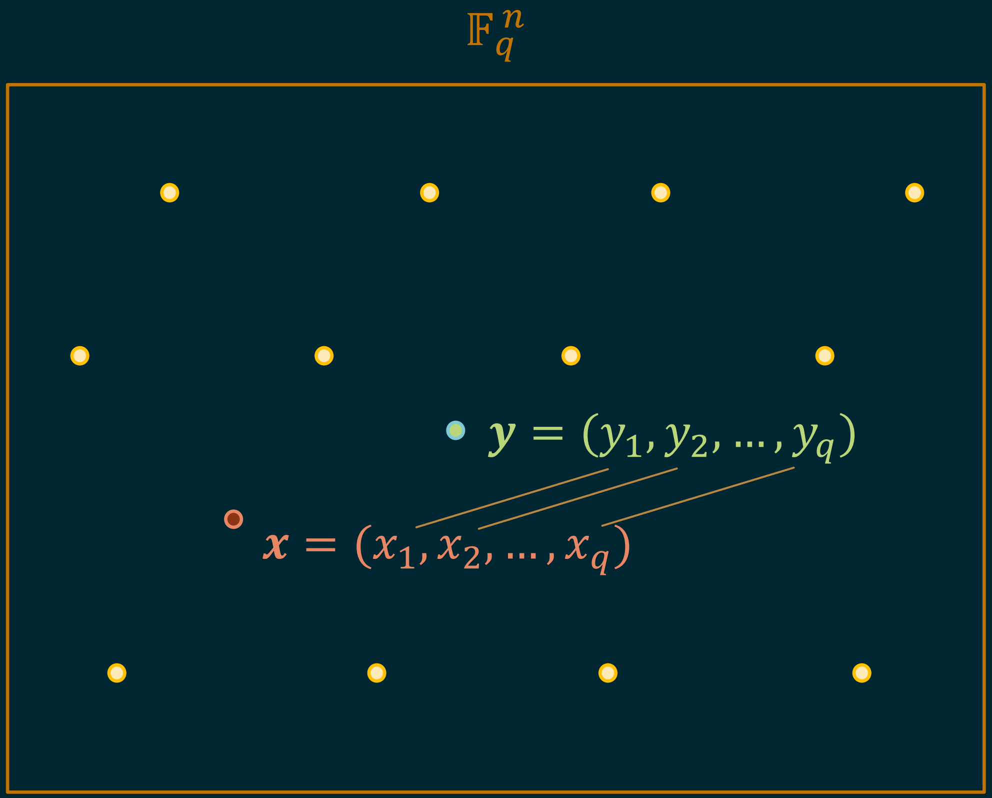
find all codewords within distance δ of y

Measuring Distance

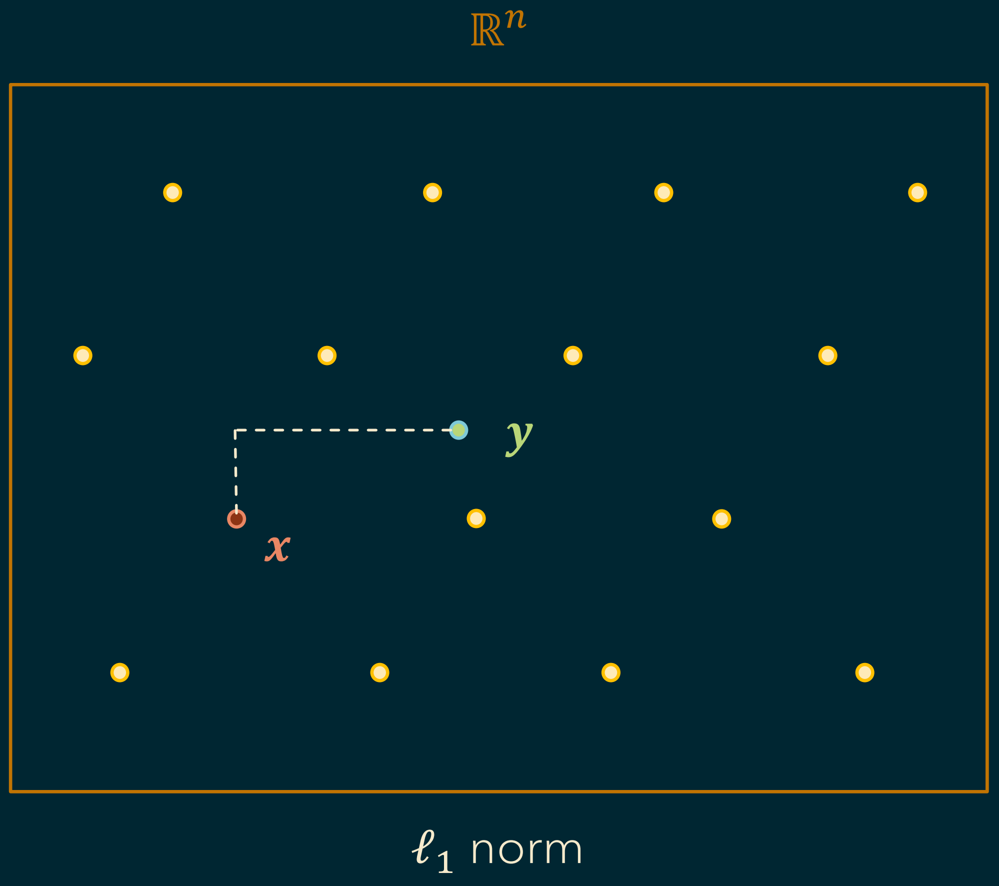


How is distance measured?

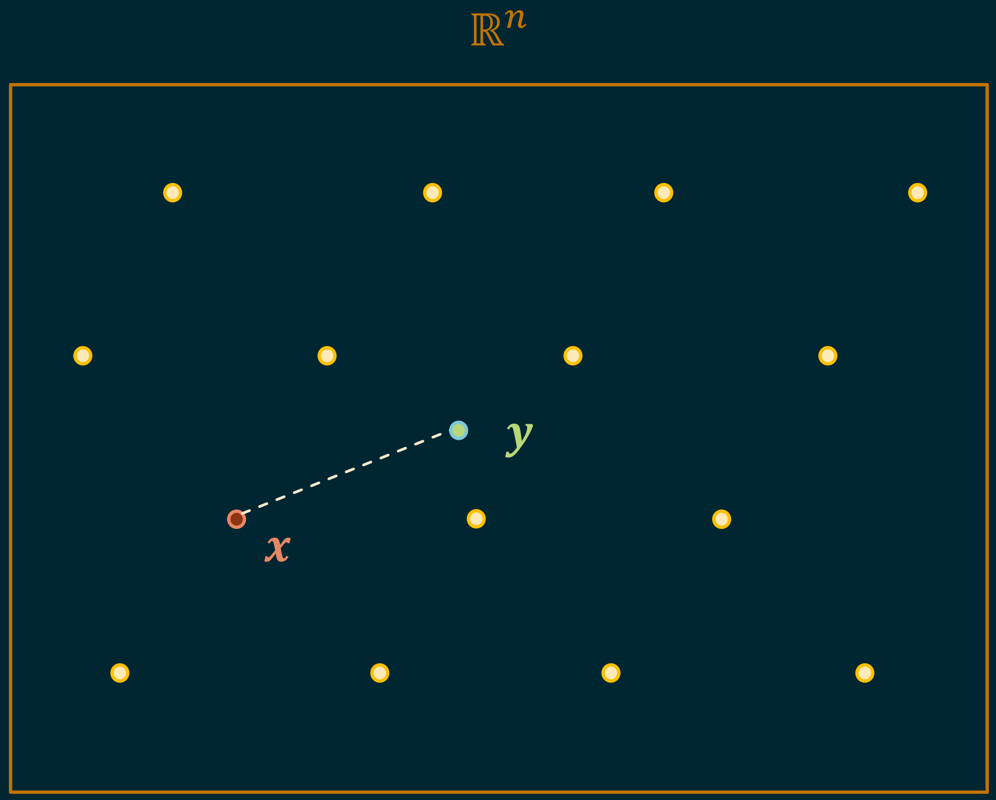
Hamming Distance



Lee Distance



Euclidean Distance



ℓ_2 norm

ℓ_p (Semi)Metric

ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

ℓ_p (Semi)Metric

ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

ℓ_p(Semi)Metric: (on $\mathbb{R}_q^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}_q^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \|\overline{\boldsymbol{x}}\|_p.$$

ℓ_p (Semi)Metric

ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$,

$$\|\boldsymbol{x}\|_p := \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

$\mathbb{R}_q = \mathbb{R}/q\mathbb{Z}$

ℓ_p(Semi)Metric: (on $\mathbb{R}_q^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}_q^n$,

$$\|\boldsymbol{x}\|_p := \|\overline{\boldsymbol{x}}\|_p.$$

ℓ_p (Semi)Metric

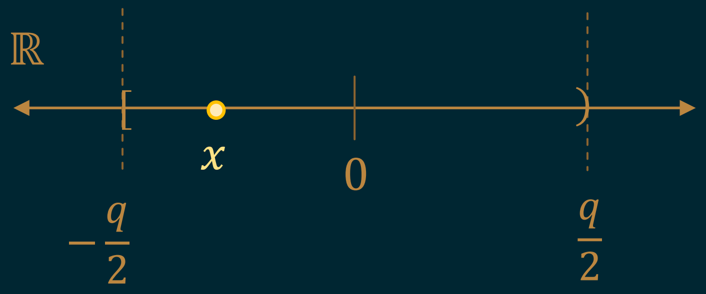
ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

$\mathbb{R}_q = \mathbb{R}/q\mathbb{Z}$

ℓ_p(Semi)Metric: (on $\mathbb{R}_q^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}_q^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \|\overline{\boldsymbol{x}}\|_p.$$



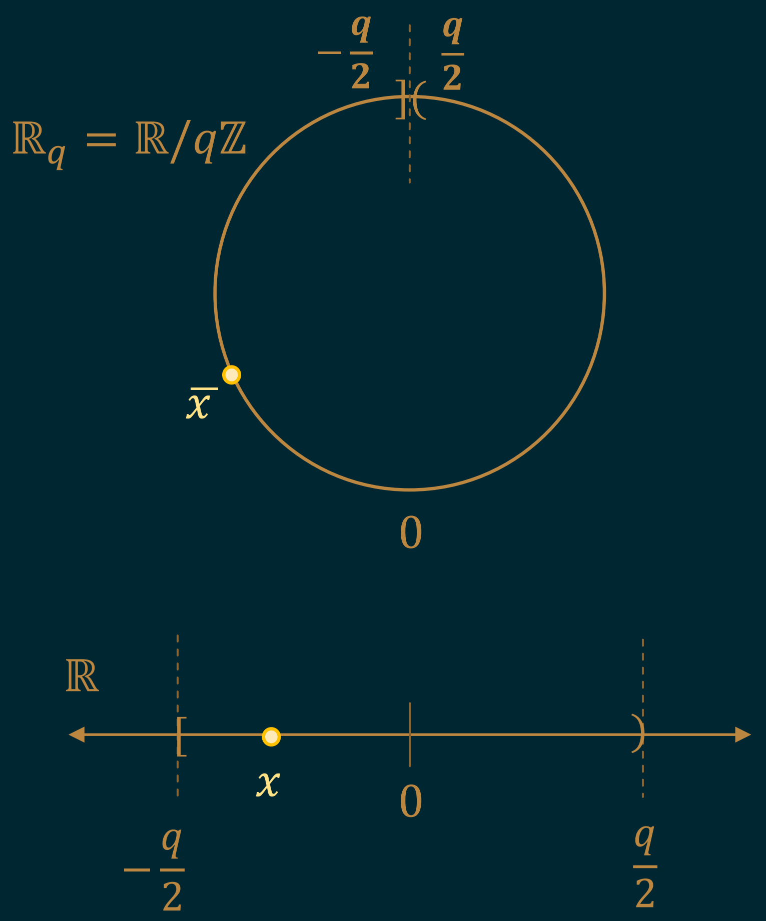
ℓ_p (Semi)Metric

ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$,

$$\|\boldsymbol{x}\|_p := \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

ℓ_p(Semi)Metric: (on $\mathbb{R}_q^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}_q^n$,

$$\|\boldsymbol{x}\|_p := \|\overline{\boldsymbol{x}}\|_p.$$



ℓ_p (Semi)Metric

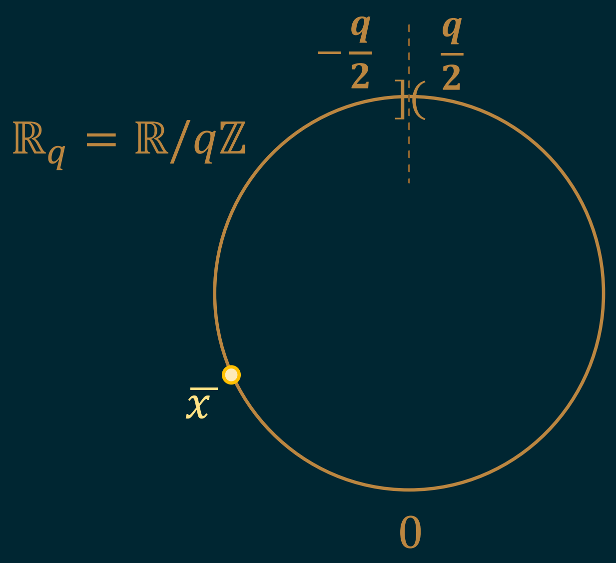
ℓ_p(Quasi)Norm: (on $\mathbb{R}^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}^n$,

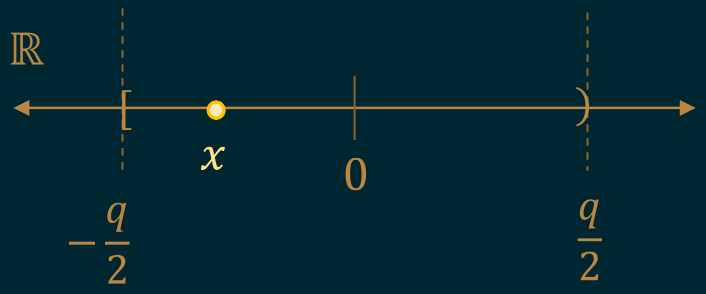
$$\|\boldsymbol{x}\|_p \coloneqq \left(x_1^p + \cdots + x_n^p\right)^{1/p}.$$

ℓ_p(Semi)Metric: (on $\mathbb{R}_q^n$) For any $p > 0$, $\boldsymbol{x} = (x_1, \dots, x_n) \in \mathbb{R}_q^n$,

$$\|\boldsymbol{x}\|_p \coloneqq \|\overline{\boldsymbol{x}}\|_p.$$

Decoding distance: $\delta = d/n^{1/p}$.





Our Results

Theorem: *(informal)* There is an efficient algorithm that list-decodes GRS codes over a prime field from continuous error in the ℓ_p (semi)metric for any $0 < p \leq 2$ up to *arbitrarily large* (relative) distance $\delta > 0$ for corresponding small enough rates.

Prior Algorithms

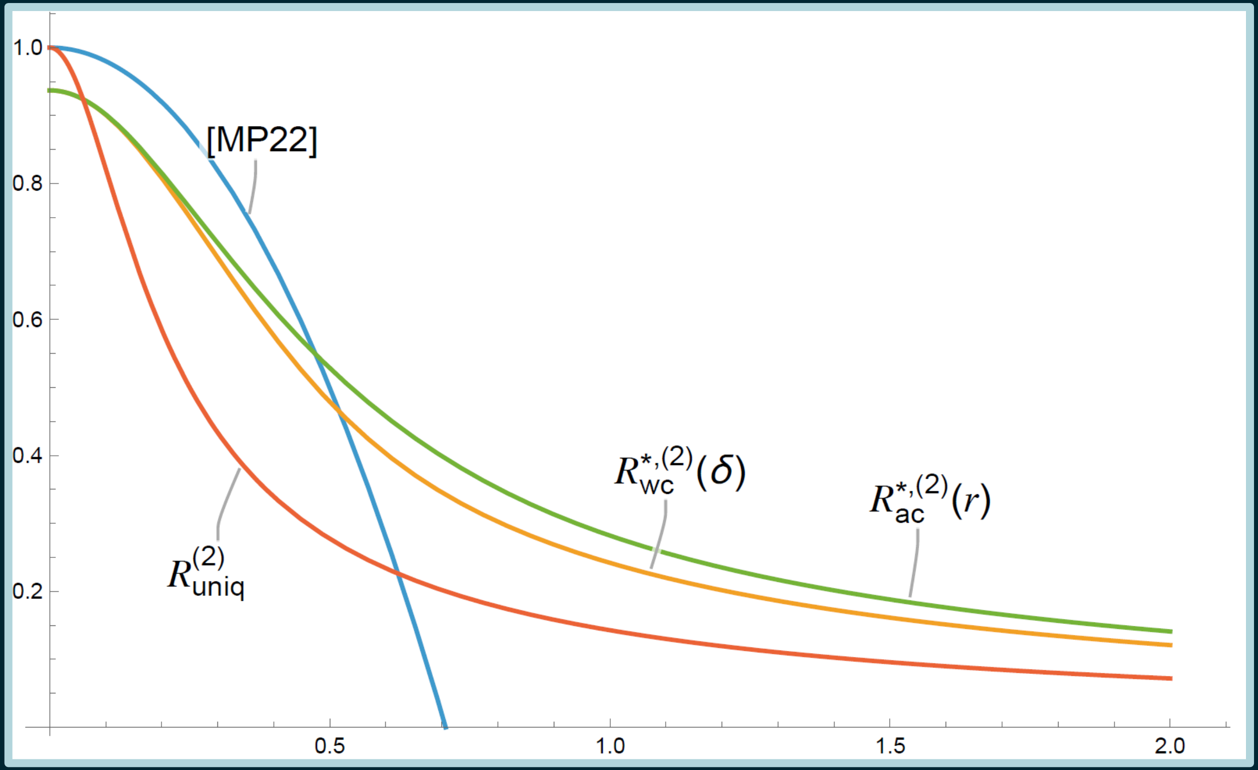
Work	Metric	Codes	Decoder Type	Error Type	Rel. Decoding Distance (δ)
[Guruswami-Sudan, 1999]	Hamming	GRS	list	discrete	$\leq 1 - \sqrt{R}$
[Mook-Peikert, 2022]	Euclidean (ℓ_2)	prime-field GRS	list	continuous	$\leq \sqrt{(1-R)/2}$
[Roth-Siegel, 1994]	Lee (ℓ_1)	subclass of GRS, BCH	unique	discrete	$\leq 1 - R$
[Wu-Kuijper-Udaya, 2003]	Lee (ℓ_1)	prime-field GRS	list	discrete	> 0

Prior Algorithms

Work	Metric	Codes	Decoder Type	Error Type	Rel. Decoding Distance ($\boldsymbol{\delta}$)
[Guruswami-Sudan, 1999]	Hamming	GRS	list	discrete	$\leq 1 - \sqrt{R}$
[Mook-Peikert, 2022]	Euclidean (ℓ_2)	prime-field GRS	list	continuous	$\leq \sqrt{(1-R)/2}$
[Roth-Siegel, 1994]	Lee (ℓ_1)	subclass of GRS, BCH	unique	discrete	$\leq 1 - R$
[Wu-Kuijper-Udaya, 2003]	Lee (ℓ_1)	prime-field GRS	list	discrete	> 0
[Peikert-V.H., 2025]	any ℓ_p, $0 < p \leq 2$	prime-field GRS	list	continuous	$\leq 1/(R \cdot c_p(ep)^{1/p})$

Comparison to Prior Algorithms

rate R^*

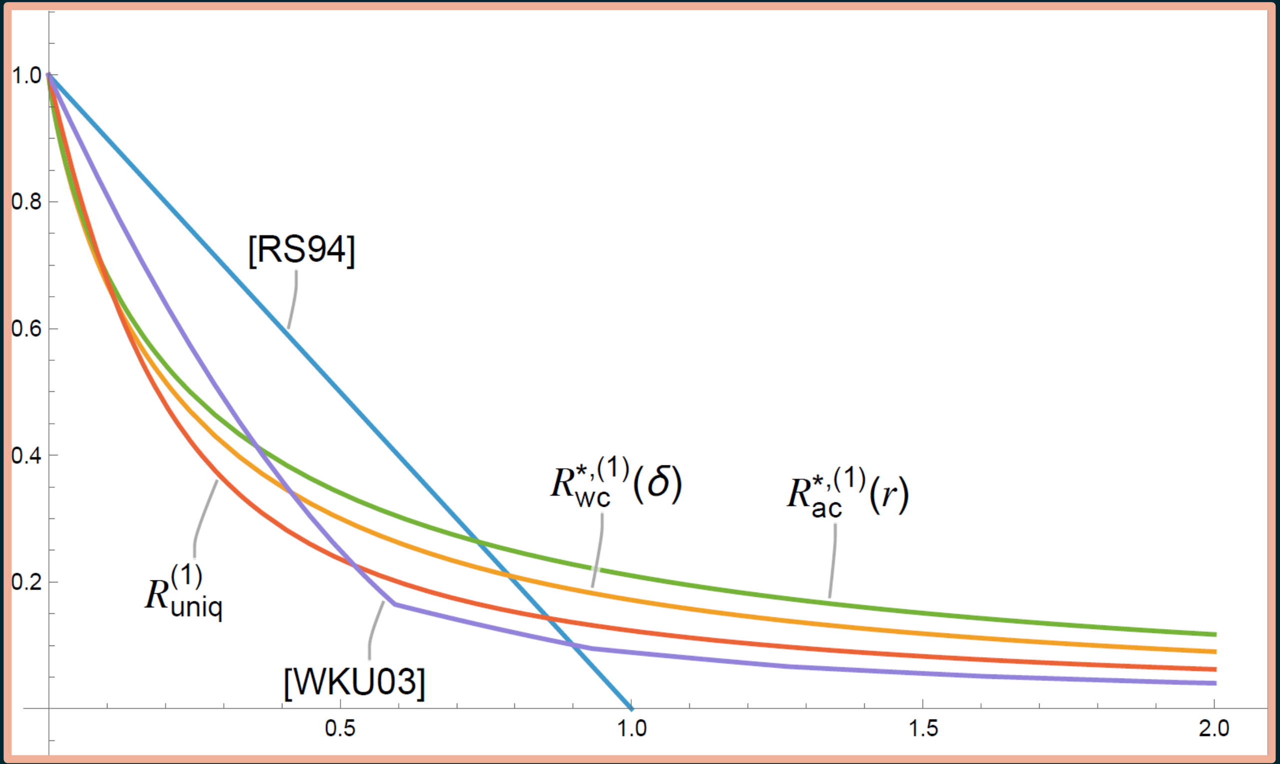


distance $\delta = \frac{r}{\sqrt{2\pi}}$

Rate-distance trade-off for ℓ_2

Comparison to Prior Algorithms

rate R^*



distance $\delta = \frac{r}{2}$

Rate-distance trade-off for ℓ_1

List-Decoding Algorithm

received word
$y \in \mathbb{R}_q^n$

list of close codewords
$c_1, c_2, \dots, c_L \in \mathcal{C}$

Soft-Decision Decoding Algorithm

weight vector

$W = (\overrightarrow{\mathrm{w}_1}, \dots, \overrightarrow{\mathrm{w}_i}, \dots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$

list of close codewords

$c_1, c_2, \dots, c_L \in \mathcal{C}$

Soft-Decision Decoding Algorithm

weight vector

$W = (\overrightarrow{\mathrm{w}_1}, \dots, \overrightarrow{\mathrm{w}_i}, \dots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$

$w_i(x_1)$	$w_i(x_2)$	…	$w_i(x_q)$

$w_i(x)$ specifies the ``likelihood'' that x was the i-th transmitted symbol

list of close codewords
$c_1, c_2, \dots, c_L \in \mathcal{C}$

Guruswami-Sudan Algorithm

[Guruswami-Sudan, 1998], [Koetter-Vardy, 2003], [Guruswami, 2001]

There is a deterministic *soft-decoding* algorithm for (Generalized) Reed-Solomon codes $\mathcal{C} \subseteq \mathbb{F}_q^n$,

dimension k, adjusted rate $R^* = \frac{k-1}{n}$, with

Input: weight vector $\mathrm{W} = (\overrightarrow{\mathrm{w}_1}, \ldots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$,

tolerance parameter $\tau > 0$

Output: list of all codewords $\boldsymbol{c} \in \mathcal{C}$ that are "closely correlated" with W

$$\mathrm{corr}(\mathrm{W}, \boldsymbol{c}) \gtrsim \sqrt{R^*}. + \tau$$

running in $\mathrm{poly}\left(n, q, \frac{1}{\tau \|W\|}\right)$ time.

Guruswami-Sudan Algorithm

[Guruswami-Sudan, 1998], [Koetter-Vardy, 2003], [Guruswami, 2001]

There is a deterministic *soft-decoding* algorithm for (Generalized) Reed-Solomon codes $\mathcal{C} \subseteq \mathbb{F}_q^n$,

dimension k, adjusted rate $R^* = \frac{k-1}{n}$, with

Input: weight vector $\mathrm{W} = (\overrightarrow{\mathrm{w}_1}, \ldots, \overrightarrow{\mathrm{w}_n}) \in [0,1]^{qn}$,

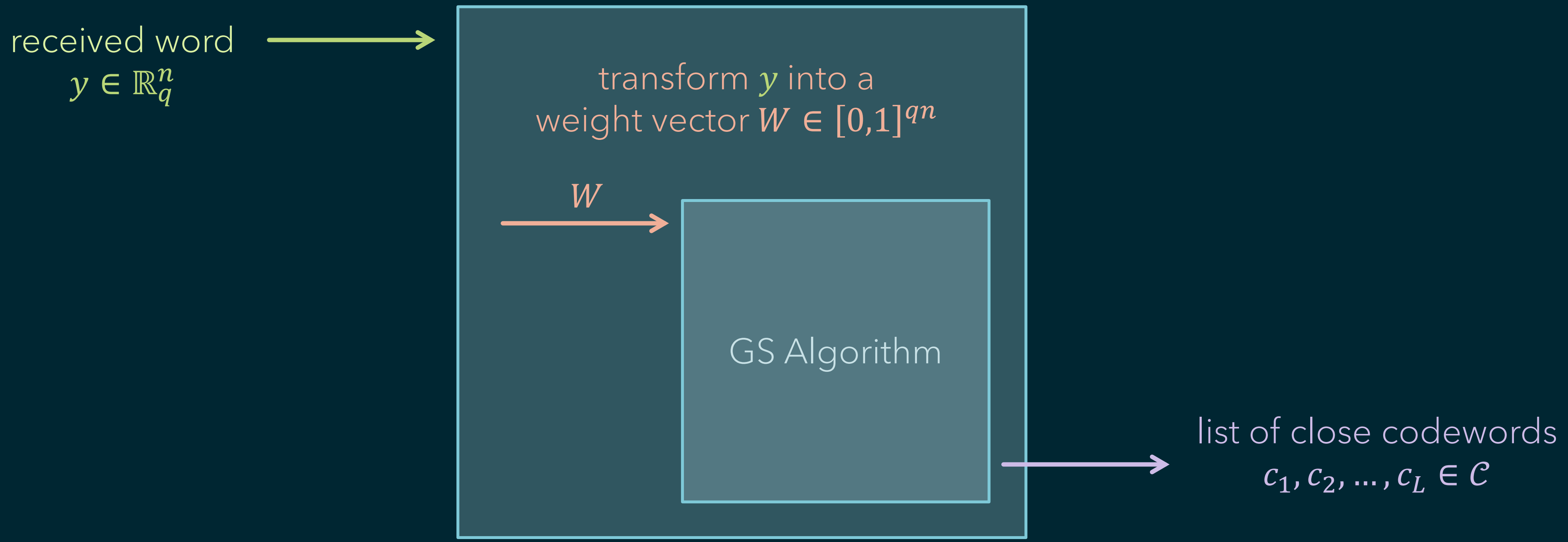
tolerance parameter $\boldsymbol{\tau} > 0$

Output: list of all codewords $\boldsymbol{c} \in \mathcal{C}$ that are "closely correlated" with W

$$\mathrm{corr}(\mathrm{W}, \boldsymbol{c}) := \frac{\langle \mathrm{W}, [\boldsymbol{c}] \rangle}{\|\mathrm{W}\| \cdot \sqrt{n}} \geq \sqrt{R^*} + \boldsymbol{\tau}.$$

running in $\mathrm{poly}\left(n, q, \frac{1}{\tau \|W\|}\right)$ time.

Our List-decoding Algorithm

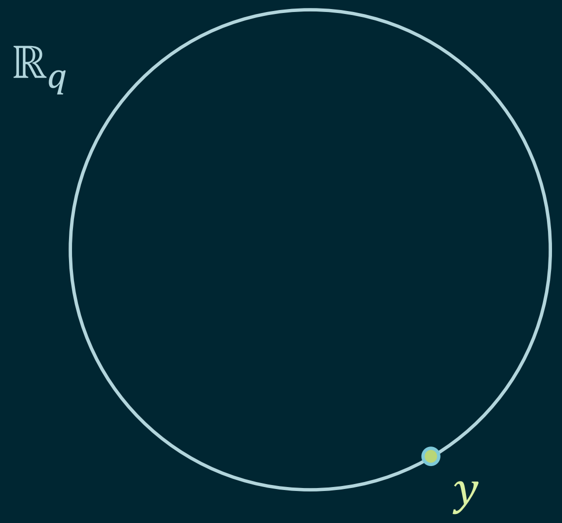


Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \ldots \mid y_n} \in \mathbb{R}_q^n$

Transforming into Weights

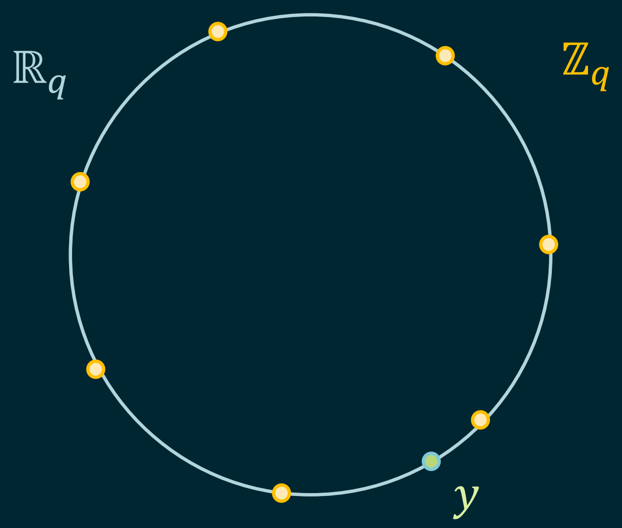
received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$



Transforming into Weights

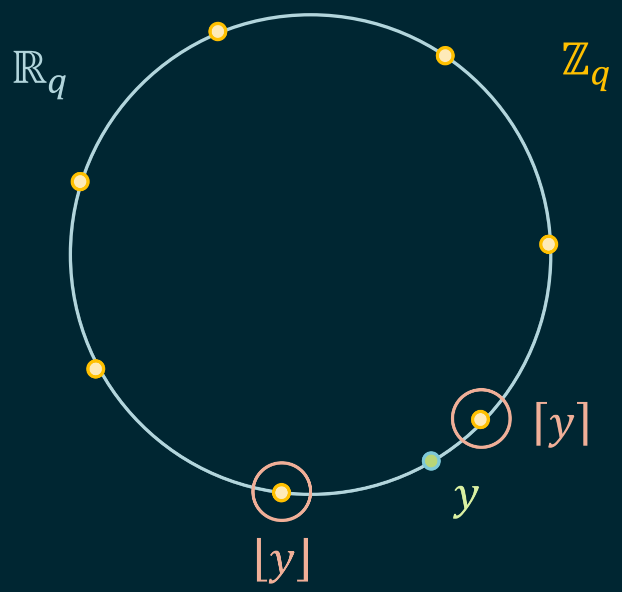
received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$

transmitted codeword must have coordinates in $\mathbb{F}_q \cong \mathbb{Z}_q$



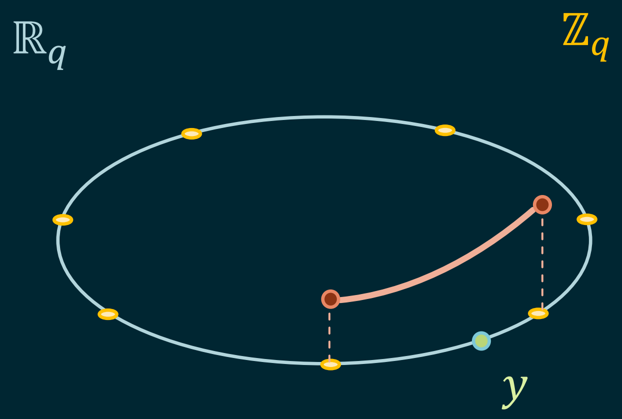
Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \ldots \mid y_n} \in \mathbb{R}_q^n$



Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$

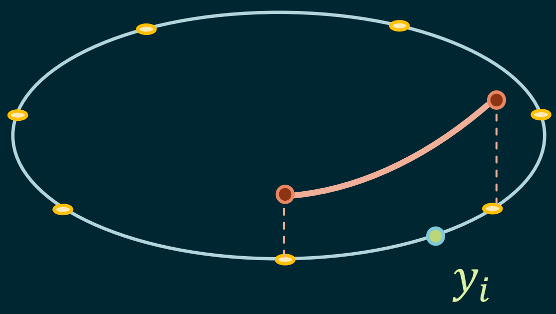


Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$

[Mook-Peikert, 2022] :

$\mathbb{R}_q$ $\mathbb{Z}_q$



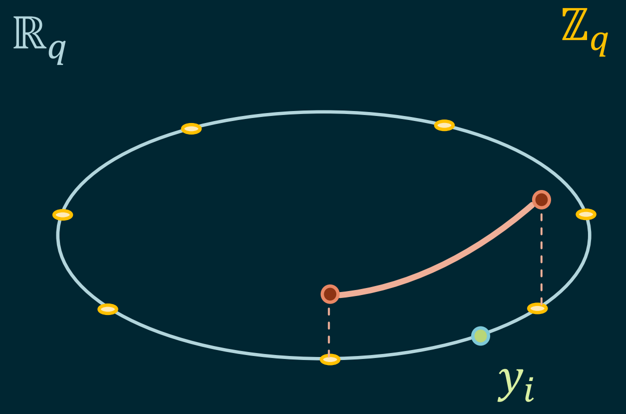
i-th weight vector

$\overrightarrow{\mathrm{w}_i} = \boxed{0 \mid 0 \mid w_i \mid w_i' \mid 0 \mid 0 \mid 0}$

Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$

[Mook-Peikert, 2022] :



weight vector

$$\overrightarrow{w_1} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0 & 0 & w_1 & w_1' & 0 & 0 & 0 \\ \hline\end{array}$$

$$\overrightarrow{w_2} = \begin{array}{|c|c|c|c|c|c|c|}\hline w_2' & 0 & 0 & 0 & 0 & 0 & w_2 \\ \hline\end{array}$$

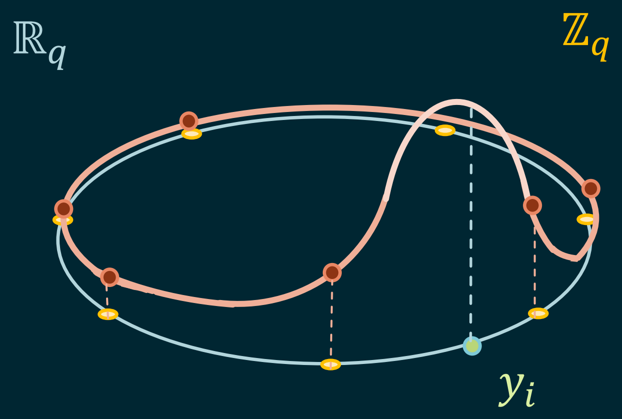
$$\vdots$$

$$\overrightarrow{w_n} = \begin{array}{|c|c|c|c|c|c|c|}\hline 0 & 0 & 0 & 0 & w_n & w_n' & 0 \\ \hline\end{array}$$

Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \ldots \mid y_n} \in \mathbb{R}_q^n$

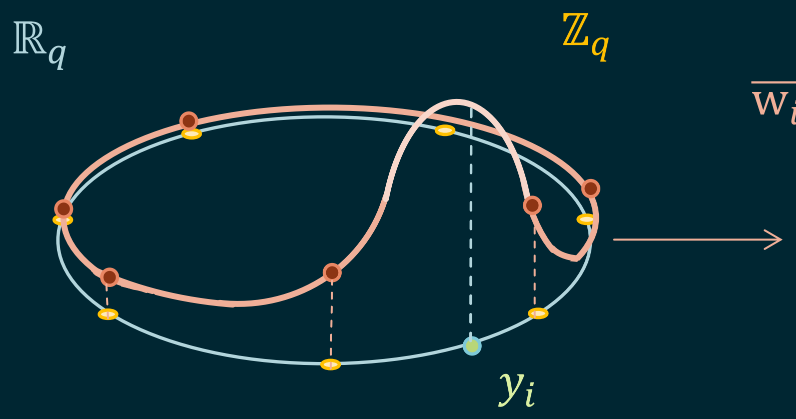
Our weight vector :



Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \ldots \mid y_n} \in \mathbb{R}_q^n$

Our weight vector :



i-th weight vector

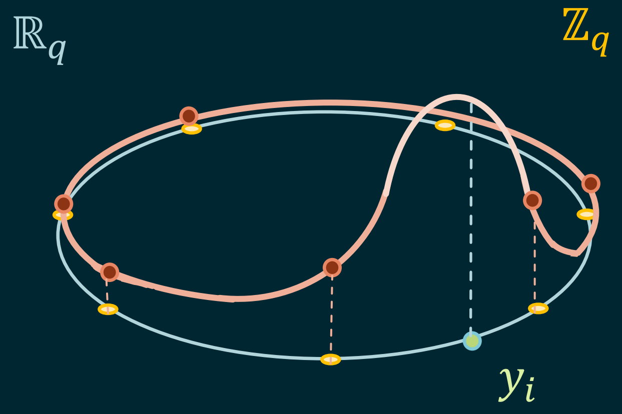
$\overrightarrow{\mathrm{w}_i} = \boxed{w_{y_i}(x_1) \mid \ldots \mid w_{y_i}(x_q)}$

weights given by a function f_s of width $s > 0$

Transforming into Weights

received word $\boldsymbol{y} = \boxed{y_1 \mid y_2 \mid \dots \mid y_n} \in \mathbb{R}_q^n$

Our weight vector :



i-th weight vector

$\overrightarrow{w_i} = \boxed{w_{s,y_i}(x_1) \mid \dots \mid w_{s,y_i}(x_q)}$

$w_{s,y_i}(x) = f_s(y_i - x + q\mathbb{Z})$

Choosing the Weight Function

We can choose any nicely behaved function f that satisfies certain properties.

But some functions perform better for specific metrics…

Choosing the Weight Function

For distances measured in the ℓ_p metric, we choose

$$f_s^{(p)}(x) \coloneqq \exp(-(c_p \cdot |x/s|)^p)$$

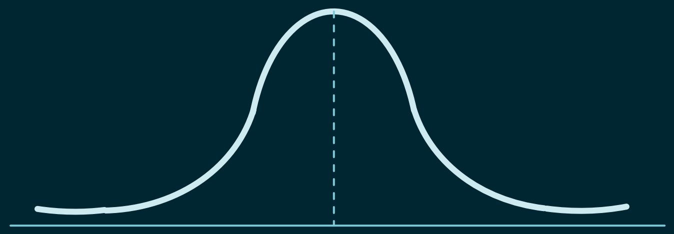
normalizing constant

Choosing the Weight Function

For distances measured in the ℓ_2 metric:

$$f_s^{(2)}(x) \coloneqq \exp(-(\pi \cdot |x/s|)^2)$$

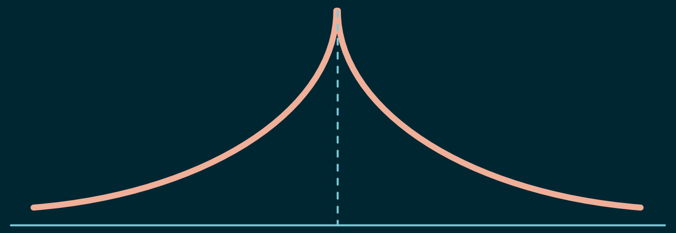
Gaussian function



For distances measured in the ℓ_1 metric:

$$f_s^{(1)}(x) \coloneqq \exp(-(2 \cdot |x/s|)^1)$$

Laplacian function



Our Results

Theorem: *(worst-case)* For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)}$$

in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$.

Our Results

Theorem: For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)}$$

in time polynomial in n, q, and $\exp(1/s^p)\ /\overbrace{(f_s(\delta)/\sqrt{f_s(\mathcal{L}_q)} - \sqrt{R^*}}^{\text{gap between rate and upper bound}})$.

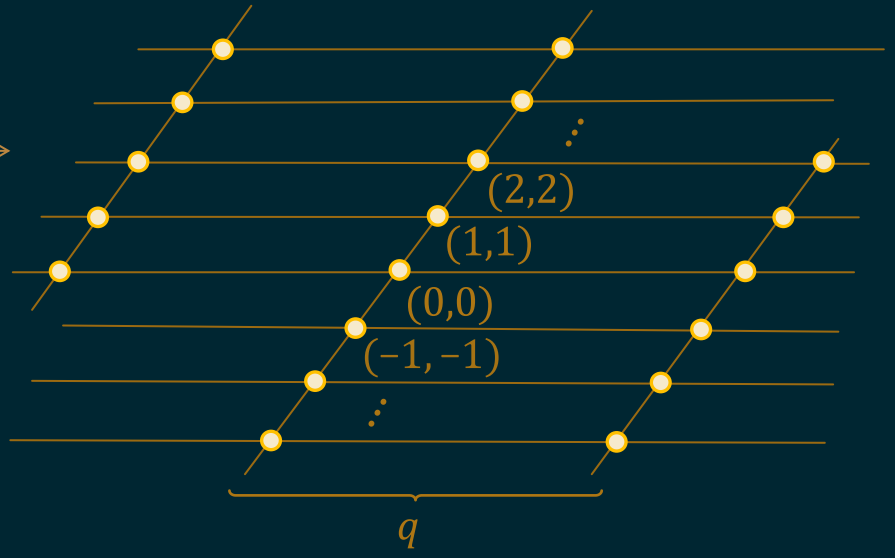
Our Results

Theorem: For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)}$$



in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$.

Our Results

Theorem: For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s)$$

in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$.

Our Results

Theorem: For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s) \xrightarrow{s, q/s \to \infty} \frac{1}{\delta \cdot c_p \cdot (ep)^{1/p}}$$

in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$.

Our Results

Theorem: For any metric parameter $0 < p \leq 2$, prime field size q, and distance $\delta > 0$,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes up to ℓ_p distance $d = \delta \cdot n^{1/p}$ any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

$$R^* < \frac{f_s(\delta)^2}{f_s(\mathcal{L}_q)} =: W_{q,\delta}^{(p)}(s) \xrightarrow{s, q/s \to \infty} \frac{1}{\delta \cdot c_p \cdot (ep)^{1/p}}$$

$\parallel$

volume of the n-dim. ℓ_p ball of radius $n^{1/p}$
(dimension-normalized) !

in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$.

Our Results

Theorem: *(average-case)* For any metric type $0 < p \leq 2$, $\alpha \in (0,1)$, prime q, channel parameter $r > 0$, :

Under a memoryless additive (continuous or discrete) channel whose distribution is D_r,

the GS soft-decision algorithm using weight vectors defined by $f_s^{(p)}$ for any $s > 0$,

list-decodes any GRS code $\mathcal{C} \subseteq \mathbb{F}_q^n$ with adjusted rate

pdf: $D_r(x) = f_r(x)/r$
pmf: $D_r(x) = f_r(x)/f_r(\mathbb{Z})$

$$R^* < \frac{s^2/\|(r,s)\|_p^2}{f_s(\mathcal{L}_q)} =: A_{q,r}^{(p)}(s)$$

in time polynomial in n, q, and $\exp(1/s^p)\ /\tau$,

except with probability $< \exp\left(-2n \cdot f_s(\mathcal{L}_q) \cdot \alpha^2 \cdot \tau^2\right)$.

Open Directions

- The product of the rate R^* and distance δ for which our algorithm works approaches

  $R^* \cdot \delta \to$ 1 / volume of the n-dim. ℓ_p ball of radius $n^{1/p}$ (dimension-normalized).

  Why should this be the case?

- What is the list-decoding capacity for decoding over general ℓ_p norms?

  How do our algorithmic bounds compare?

Questions?