

# Lower bounds on lattice sieving and information set decoding

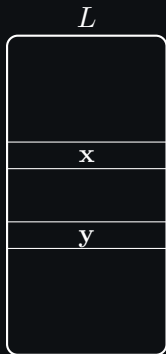
Elena Kirshanova    Thijs Laarhoven

3rd PQC Standardization Conference

June 4, 2021

To appear at Crypto'21

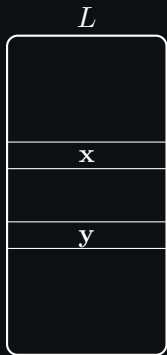
## The closest pairs problem



Given a list  $L$  and  $r > 0$ , find almost all  $\mathbf{x}, \mathbf{y} \in L$  such that

$$\text{dist}(\mathbf{x}, \mathbf{y}) < r.$$

## The closest pairs problem

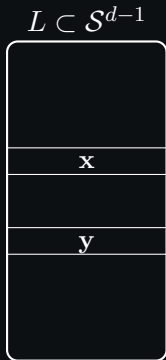


Given a list  $L$  and  $r > 0$ , find almost all  $\mathbf{x}, \mathbf{y} \in L$  such that

$$\text{dist}(\mathbf{x}, \mathbf{y}) < r.$$

- Cases of interest:
  - ◇  $L \subset \mathcal{S}^{d-1}$  a unit sphere,  $r = \Theta(1)$
  - ◇  $L \subset \{0, 1\}^d$ ,  $r = \Theta(d)$
- Often  $|L| = \exp(d)$  (dense setting)
- Elements in  $L$  are uniformly distributed

## Why interesting?



Main subroutine inside sieving algorithms for SVP

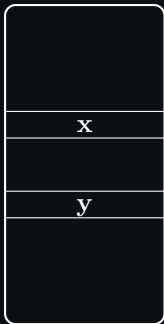
For  $|L| = \left(\frac{4}{3}\right)^{d/2}$ , we can solve this problem in  
 $T = \left(\frac{3}{2}\right)^{d/2}$  time and  $S = \left(\frac{4}{3}\right)^{d/2}$  space.

Some NIST submitters use these complexities to setup their parameters.

All  $o(d)$  factors in the exponents are omitted

## Why interesting?

$$L \subset \{0, 1\}^d$$



Main subroutine inside Information Set Decoding algorithms, [MO15], [BM18]

Relevant to the dense error setting:  $wt(e) = \Theta(d)$

NIST submitters use sparse error:  $wt(e) = o(d)$ .

How to solve the closest pairs problem?

Use locality-sensitive hashing (LSH).

[BGJ15], [BDGL16] for Euclidean metric

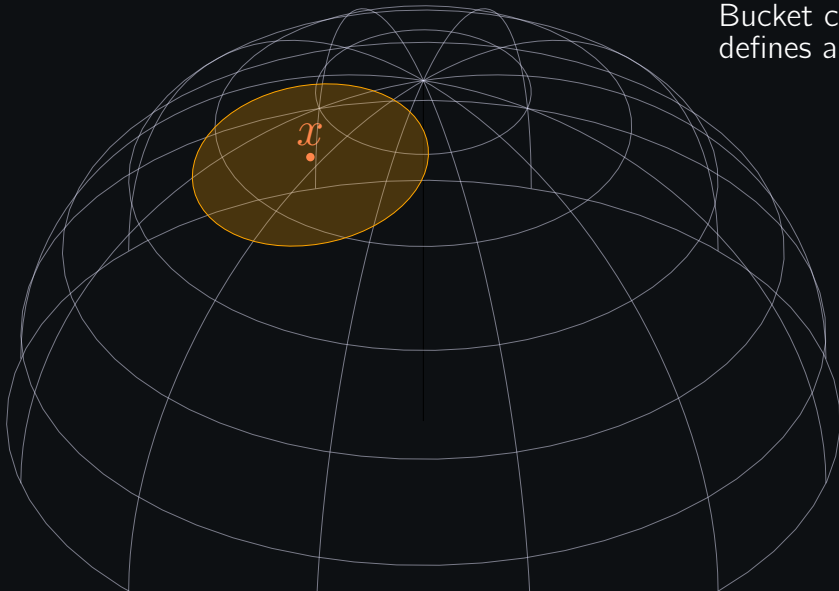
[MO15] for Hamming metric

LSH is built upon a family of hash functions  $h$  such that

$$\Pr_{\substack{\mathbf{x}, \mathbf{y} \sim \mathcal{S}^{d-1} \\ \text{dist}(\mathbf{x}, \mathbf{y}) < r}} [h(\mathbf{x}) = h(\mathbf{y})] \gg \Pr_{\mathbf{x}, \mathbf{y} \sim \mathcal{S}^{d-1}} [h(\mathbf{x}) = h(\mathbf{y})]$$

## LSH on the unit sphere

Bucket center  $\mathbf{x} \in \mathcal{S}^{d-1}$   
defines a region  $B_{\mathbf{x}}$



## LSH on the unit sphere

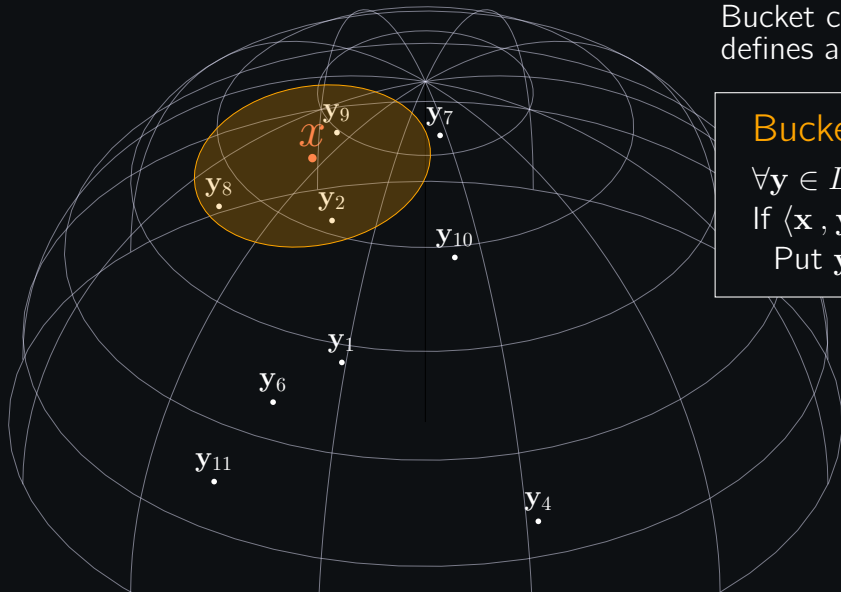
Bucket center  $\mathbf{x} \in \mathcal{S}^{d-1}$   
defines a region  $B_{\mathbf{x}}$

### Bucketing phase

$\forall \mathbf{y} \in L :$

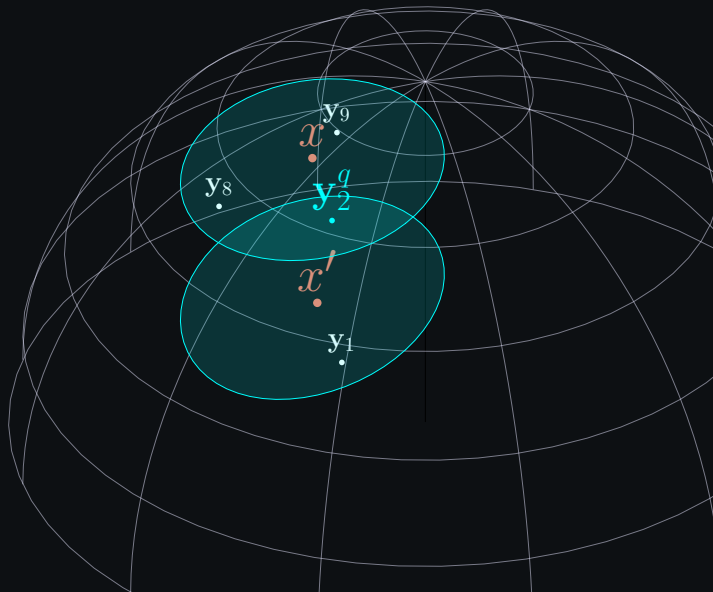
If  $\langle \mathbf{x}, \mathbf{y} \rangle \geq \alpha :$

Put  $\mathbf{y}$  into  $B_{\mathbf{x}}$





## LSH on the unit sphere



Bucket center  $\mathbf{x} \in \mathcal{S}^{d-1}$   
defines a region  $B_{\mathbf{x}}$

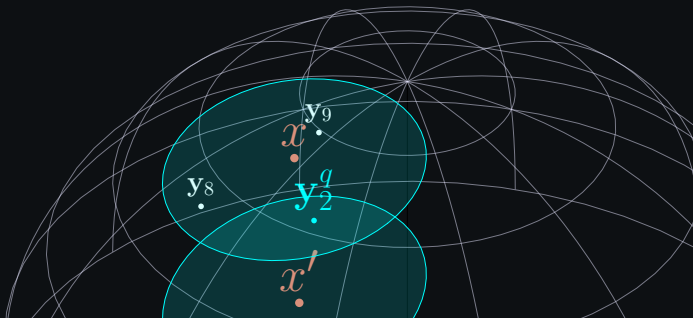
### Bucketing phase

$\forall \mathbf{y} \in L :$   
If  $\langle \mathbf{x}, \mathbf{y} \rangle \geq \alpha :$   
Put  $\mathbf{y}$  into  $B_{\mathbf{x}}$

### Query phase

$\forall B_{\mathbf{x}}$  s.t.  $\mathbf{y}^q \in B_{\mathbf{x}} :$   
 $\forall \mathbf{y} \in B_{\mathbf{x}} :$   
Find  $\mathbf{y}' \in B_{\mathbf{x}}$  s.t.  
 $\text{dist}(\mathbf{y}', \mathbf{y}^q) < r$

## LSH on the unit sphere



Bucket center  $\mathbf{x} \in \mathcal{S}^{d-1}$   
defines a region  $B_{\mathbf{x}}$

### Bucketing phase

$\forall \mathbf{y} \in L :$   
If  $\langle \mathbf{x}, \mathbf{y} \rangle \geq \alpha :$   
Put  $\mathbf{y}$  into  $B_{\mathbf{x}}$

The number of  $\mathbf{x}$ 's determines the runtime  
The shape of  $\mathbf{x}$ 's determines the complexity of  
finding  $\mathbf{x}$  for  $\mathbf{y}$ .

### Query phase

$\forall B_{\mathbf{x}}$  s.t.  $\mathbf{y}^q \in B_{\mathbf{x}} :$   
 $\forall \mathbf{y} \in B_{\mathbf{x}} :$   
Find  $\mathbf{y}' \in B_{\mathbf{x}}$  s.t.  
 $\text{dist}(\mathbf{y}', \mathbf{y}^q) < r$

## Our results (informal)

Instantiating LSH with Spherical caps, i.e.,

$$B_{\mathbf{x}}(\alpha) := \{\mathbf{y} \in \mathcal{S}^{d-1} : \langle \mathbf{x}, \mathbf{y} \rangle \leq \alpha\},$$

is optimal in the **Euclidean metric** and almost optimal in the **Hamming metric**.

Here **optimal** means that choosing hash regions different from spherical caps will not **asymptotically** improve the performance of LSH.

## Our results (informal)

Instantiating LSH with Spherical caps, i.e.,

$$B_{\mathbf{x}}(\alpha) := \{\mathbf{y} \in \mathcal{S}^{d-1} : \langle \mathbf{x}, \mathbf{y} \rangle \leq \alpha\},$$

is optimal in the **Euclidean metric** and almost optimal in the **Hamming metric**.

Here **optimal** means that choosing hash regions different from spherical caps will not **asymptotically** improve the performance of LSH.

### Consequences:

- Another hashing strategy will not improve the performance of lattice sieving
- Improving only the closest pair subroutine in ISD will not result in a noticeable gain
- Asymptotically fastest algorithm will choose  $\mathbf{x}$ 's from a fast-decodable spherical code (may not exist for arbitrary dimensions).

## Proof technique: Euclidean metric

Convolution on  $\mathcal{S}^{d-1}$  :

$$\mathcal{T}(f, g, h) := \int \int_{\mathcal{S}^{d-1} \times \mathcal{S}^{d-1}} f(\mathbf{x})g(\mathbf{y})h(\langle \mathbf{x}, \mathbf{y} \rangle) d\sigma(\mathbf{x})d\sigma(\mathbf{y}).$$

$f, g : \mathcal{S}^{d-1} \rightarrow \mathbb{R}$ ,  $h : [-1, 1] \rightarrow \mathbb{R}$ ,  $\sigma$  normalized surface measure on  $\mathcal{S}^{d-1}$ .

For which  $f, g, h$  is  $\mathcal{T}(f, g, h)$  maximized?

## Proof technique: Euclidean metric

Convolution on  $\mathcal{S}^{d-1}$  :

$$\mathcal{T}(f, g, h) := \int \int_{\mathcal{S}^{d-1} \times \mathcal{S}^{d-1}} f(\mathbf{x})g(\mathbf{y})h(\langle \mathbf{x}, \mathbf{y} \rangle) d\sigma(\mathbf{x})d\sigma(\mathbf{y}).$$

$f, g : \mathcal{S}^{d-1} \rightarrow \mathbb{R}$ ,  $h : [-1, 1] \rightarrow \mathbb{R}$ ,  $\sigma$  normalized surface measure on  $\mathcal{S}^{d-1}$ .

For which  $f, g, h$  is  $\mathcal{T}(f, g, h)$  maximized?

Baernstein–Taylor rearrangement inequality on  $\mathcal{S}^{d-1}$  [BT76] :

$$\mathcal{T}(f, g, h) \leq \mathcal{T}(f^*, g^*, h),$$

for  $f^*, g^*$  depending only on  $\mathbf{x}_1$  and is non-decreasing in  $\mathbf{x}_1$ ,

$$\sigma(\{f^* > \lambda\}) = \sigma(\{f > \lambda\}), \sigma(\{g^* > \lambda\}) = \sigma(\{g > \lambda\}), \forall \lambda.$$

## Proof technique: Euclidean metric

Take

$U, Q \subset \mathcal{S}^{d-1}$  – arbitrary sets

$$C_Q = \{\mathbf{z} \in \mathcal{S}^{d-1} : \mathbf{z}_1 \geq \alpha\}$$

$$C_U = \{\mathbf{z} \in \mathcal{S}^{d-1} : \mathbf{z}_1 \geq \alpha\}$$

$$\sigma(U) = \sigma(C_U), \sigma(Q) = \sigma(C_Q)$$

$$f = \mathbf{1}(U)$$

$$f^* = \mathbf{1}(C_U)$$

$$g = \mathbf{1}(Q)$$

$$g^* = \mathbf{1}(C_Q)$$

$$h(s) = \mathbf{1}\{s > r\}, r \in [-1, 1]$$

## Proof technique: Euclidean metric

Take

$U, Q \subset \mathcal{S}^{d-1}$  — arbitrary sets

$$C_Q = \{\mathbf{z} \in \mathcal{S}^{d-1} : z_1 \geq \alpha\}$$

$$C_U = \{\mathbf{z} \in \mathcal{S}^{d-1} : z_1 \geq \alpha\}$$

$$\sigma(U) = \sigma(C_U), \sigma(Q) = \sigma(C_Q)$$

$$f = \mathbf{1}(U)$$

$$f^* = \mathbf{1}(C_U)$$

$$g = \mathbf{1}(Q)$$

$$g^* = \mathbf{1}(C_Q)$$

$$h(s) = \mathbf{1}\{s > r\}, r \in [-1, 1]$$

Notice that  $(f, g, h, f^*, g^*)$ , satisfy the BT inequality.

Interpreting integrals as probabilities leads to:

$$\Pr_{\mathbf{x}, \mathbf{y} \sim \mathcal{S}^{d-1}} [\mathbf{x} \in U, \mathbf{y} \in Q \mid \langle \mathbf{x}, \mathbf{y} \rangle < \gamma] \leq \Pr_{\mathbf{x}, \mathbf{y} \sim \mathcal{S}^{d-1}} [\mathbf{x} \in C_U, \mathbf{y} \in C_Q \mid \langle \mathbf{x}, \mathbf{y} \rangle < \gamma].$$



## Proof technique: Hamming metric

Andoni-Razenshteyn inequality [AR16]:

For every hash function  $h : \{0, 1\}^d \rightarrow \mathbb{Z}$  and every  $0 < r \leq d/2$ :

$$\Pr_{\substack{\mathbf{x}, \mathbf{y} \sim \{0, 1\}^d \\ \mathbf{E}(\text{dist}(\mathbf{x}, \mathbf{y})) = r}} [h(\mathbf{x}) \neq h(\mathbf{y})] \leq \Pr_{\mathbf{x}, \mathbf{y} \sim \{0, 1\}^d} [h(\mathbf{x}) \neq h(\mathbf{y})]^{\frac{r}{d-r}}.$$

## Proof technique: Hamming metric

Andoni-Razenshteyn inequality [AR16]:

For every hash function  $h : \{0, 1\}^d \rightarrow \mathbb{Z}$  and every  $0 < r \leq d/2$ :

$$\Pr_{\substack{\mathbf{x}, \mathbf{y} \sim \{0, 1\}^d \\ \mathbf{E}(\text{dist}(\mathbf{x}, \mathbf{y})) = r}} [h(\mathbf{x}) = h(\mathbf{y})] \leq \Pr_{\mathbf{x}, \mathbf{y} \sim \{0, 1\}^d} [h(\mathbf{x}) = h(\mathbf{y})]^{\frac{r}{d-r}}.$$

This leads to the lower bound on the runtime of closest pairs problem over  $\{0, 1\}^d$ :

$$\log_2 T \geq \frac{1}{1 - r/d} \log_2 |L|.$$

- [MO15] achieves the lower bound in the sparse setting,
- and comes close to it in the dense setting.

## Proof technique: Hamming metric

Source of the gap: we need a set  $A \subset \{0, 1\}^d$ , for which

$$\Pr_{\substack{\mathbf{x}, \mathbf{y} \sim \{0, 1\}^d \\ \mathbf{E}(\text{dist}(\mathbf{x}, \mathbf{y})) = r}} [\mathbf{x} \in A \mid \mathbf{y} \in A] \leq \left( \frac{|A|}{2^d} \right)^{\frac{r}{d-r}}.$$

is tight. For spherical caps in  $\{0, 1\}^d$  this inequality is not tight.

## Interpretation of the result

- The result does not imply a lower bound on all possible **sieving algorithms**. Another use of the closest pairs problem or a completely different technique is possible.
- E.g., there is not contradiction with the result “Lattice sieving via quantum random walks” (eprint 2021/570).
- It does imply that we have an optimal near neighbor subroutine within sieving algorithms.
- It does imply that in order to noticeably improve ISD, another technique is needed.

## References

- [AR16] A. Andoni, I. Razenshteyn. Tight lower bounds for data- dependent locality-sensitive hashing.
- [BDJ15] A. Becker, N. Gama, A.Joux. Speeding-up lattice sieving without increasing the memory, using sub-quadratic nearest neighbor search
- [BDGL16] A. Becker, L. Ducas, N. Gama, T. Laarhoven. New directions in nearest neighbor searching with applications to lattice sieving.
- [BM18] L.Both, A.May. Decoding linear codes with high error rate and its impact for LPN security.
- [BT76] A. Baernstein, B.A. Taylor. Spherical rearrangements, subharmonic functions, and  $\star$ -functions in  $n$ -space.
- [MO15] A. May, I.Ozerov. On computing nearest neighbors with applications to decoding of binary linear codes.