Decoding Balanced Linear Code with Preprocessing

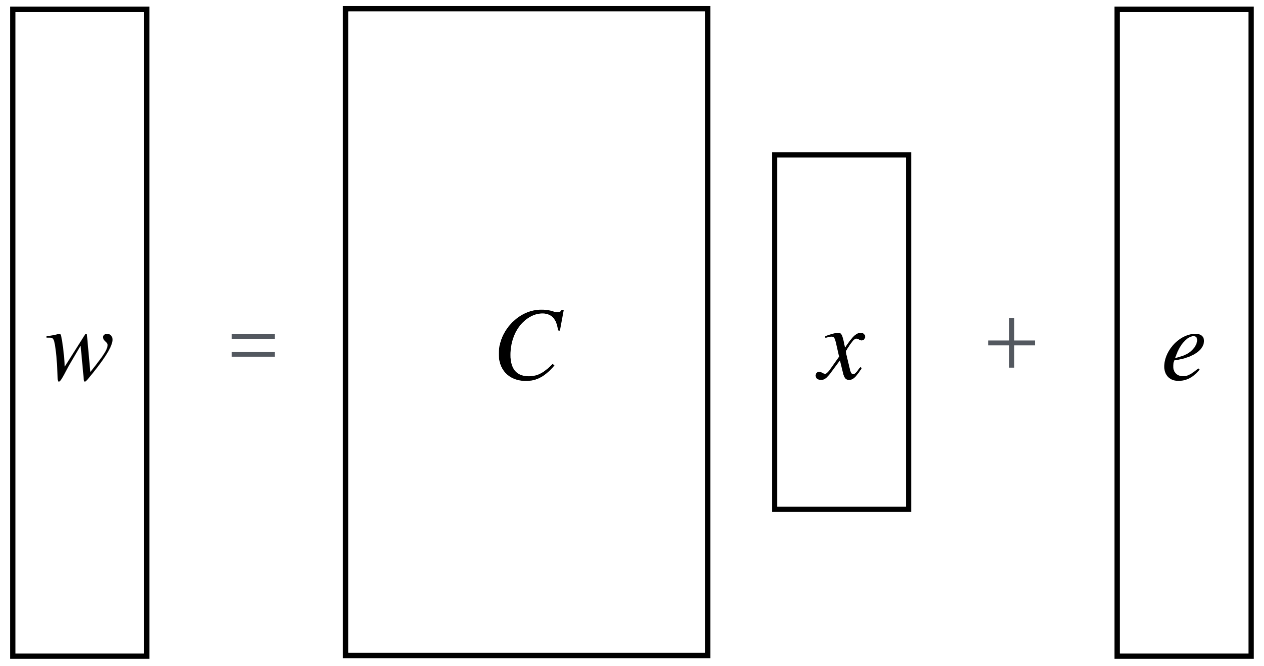
Andrej Bogdanov[1] Rohit Chatterjee[2] **Yunqi Li**[2] Prashant Nalini Vasudevan[2]

[1] University of Ottawa
[2] National University of Singapore

ITCS 2026, Milan

Intro

Nearest Codeword Problem (NCP)



Input: (C, w)

- Generator matrix $C \in \mathbb{F}_2^{m \times n}$ representing a linear code $\mathscr{C} = \{C \cdot x : x \in \mathbb{F}_2^n\}$
- Target vector $w \in \mathbb{F}_2^m$

Search problem find x, minimize the distance $||C \cdot x - w||$

Error rate: $\frac{1}{m} \min_x ||C \cdot x - w||$

Decision problem decide whether w is close to or far from $\mathscr{C}$

Intro

Balanced Nearest Codeword Problem (BNCP)

Input: (C, w)

$\mathscr{C}$ is β-balanced: all non-zero $v \in \mathscr{C}$, the hamming weight $\|v\| \in \frac{1}{2}(1 \pm \beta)m$

- Generator matrix $C \in \mathbb{F}_2^{m \times n}$ representing a linear code $\mathscr{C} = \{C \cdot x : x \in \mathbb{F}_2^n\}$
- Target vector $w \in \mathbb{F}_2^m$

Search problem find x, minimize the distance $\|C \cdot x - w\|$

Decision problem decide whether w is close to or far from $\mathscr{C}$

Most linear codes are $\Theta(\sqrt{n/m})$-balanced

Intro

Related work

Prange's algorithm [Pra62]

Pick information set and solve linear equations

Error rate $O(\log n/n)$ yields poly-time decoding

Statistical decoding [Jab01, Ove06, CDAMHT22, CDAMHT24]

Use inner products with light dual codewords

Better experimental performance for low-rate codes

Intro

Learning Parity with Noise (LPN) and Cryptography

Learning Parity with Noise [BFKL93]

Distinguish corrupted codeword from random vector

$$C \quad C\,x + e \approx_c C \quad r$$

Average-case decisional NCP

Intro

Learning Parity with Noise (LPN) and Cryptography

Applications in cryptography

Public Key Encryption	[Ale03]
Collision Resistant Hashing	[AHI+17, BLVW19, YZW+19]
Others	[BLSV18, BF22, AMR25]

Main Result

Algorithm with Preprocessing

β-balanced code $\mathscr{C}$ and target w

- Preprocessing phase:

  $H \leftarrow \mathsf{Pre}(C)$ Input: generator matrix C Inefficient

- Online phase:

  YES/NO $\leftarrow \mathsf{Decide}(w; H)$ Input: vector w (either η-close to or β-far)

  $x \leftarrow \mathsf{Search}(w; H)$ Input: noisy codeword $w = C \cdot x + e$ with error rate η

Main Result

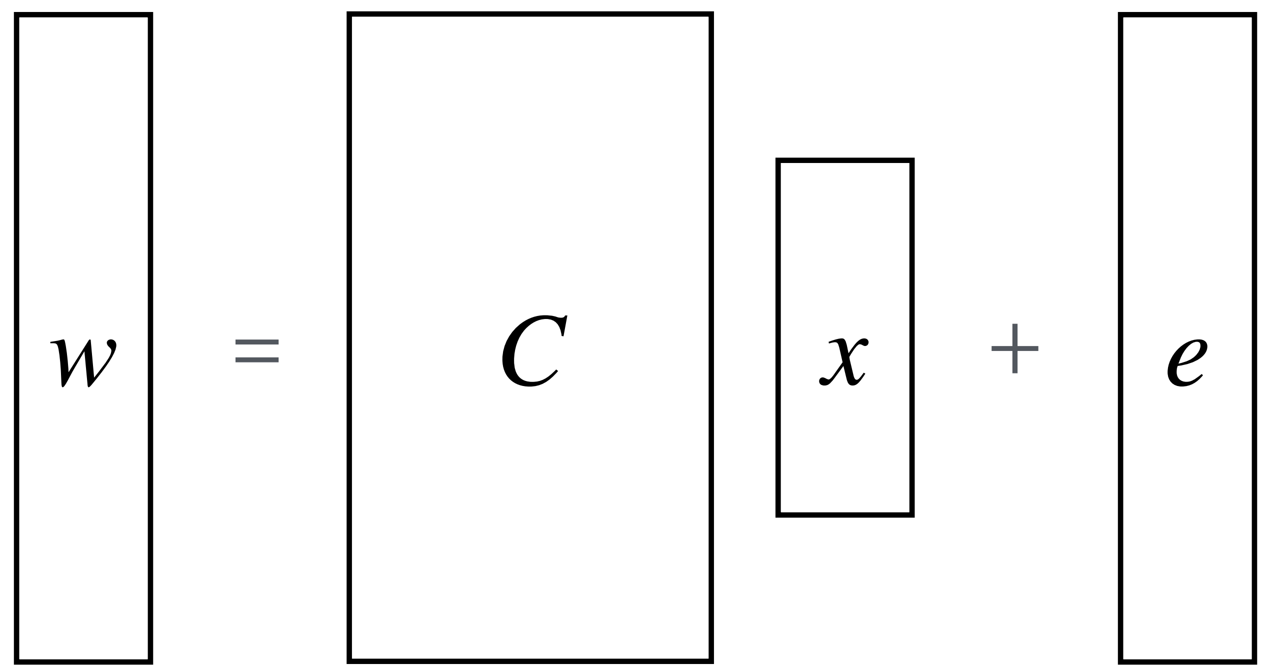
Algorithm with Preprocessing for DBNCP

Theorem 1. There is an algorithm with preprocessing for $\mathsf{DBNCP}_{\beta,\eta}$ with both advice size and running time $m^2 \exp[O(\eta n/\log(1/\beta))]$.

Corollary 1. When $\eta = O(\log^2 n/n)$, there is an algorithm with preprocessing for DBNCP with both advice size and running time polynomial in n.

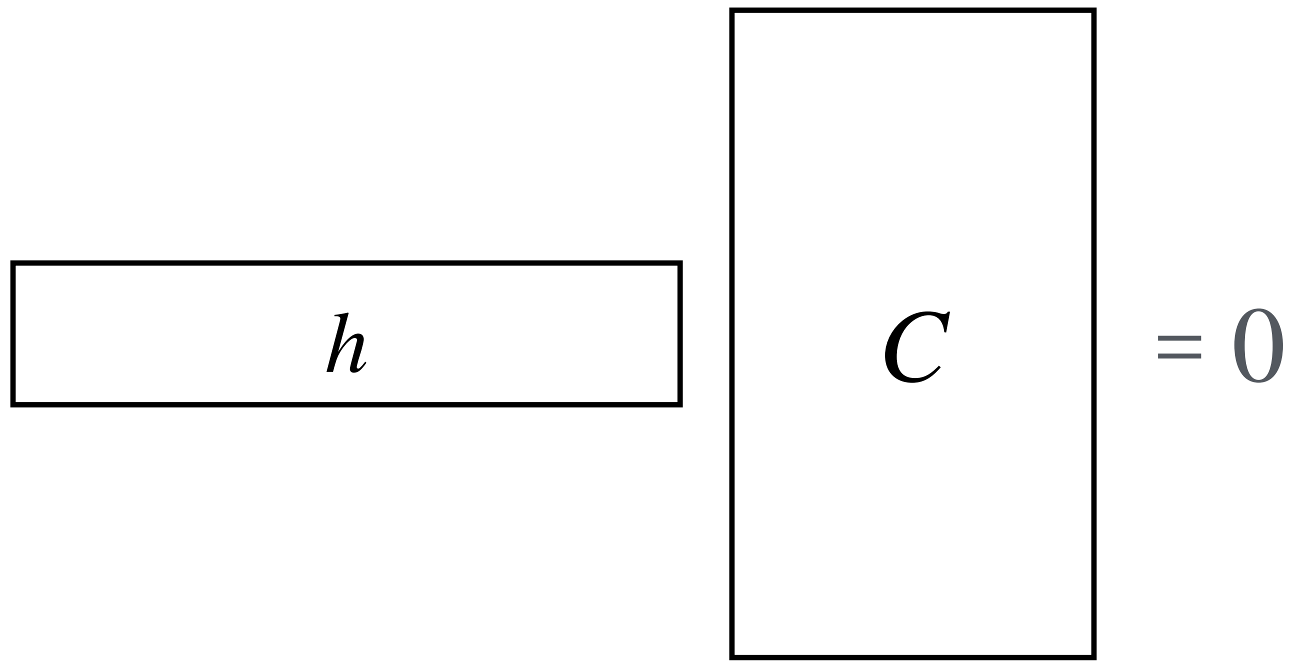
Techniques

High-Level Ideas: light dual vectors



Input (C, w), denote $w = C \cdot x + e$

Dual space $\mathscr{C}^{\perp} = \{h : \langle h, v \rangle = 0, \forall v \in \mathscr{C}\}$



$\langle h, w \rangle = \langle h, C \cdot x + e \rangle = \langle h, e \rangle$

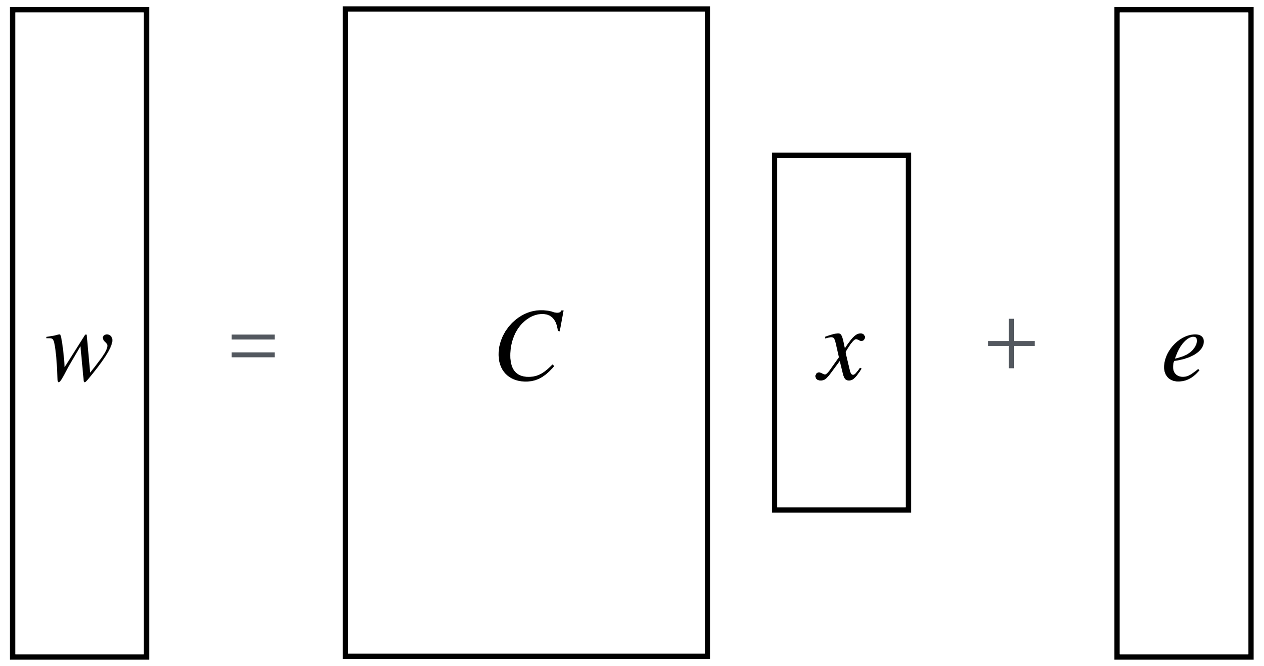
If h is sparse

- $w \leftarrow C \cdot x + e$ $\quad \mathbb{E}_{e \leftarrow \mathsf{Ber}(\eta)^m} (-1)^{\langle h, e \rangle} > 0$
- $w \leftarrow \{0,1\}^m$ $\quad \mathbb{E}_{w \leftarrow \{0,1\}^m} (-1)^{\langle h, w \rangle} = 0$

$1/\mathsf{poly}(n)$ gap average-case distinguisher

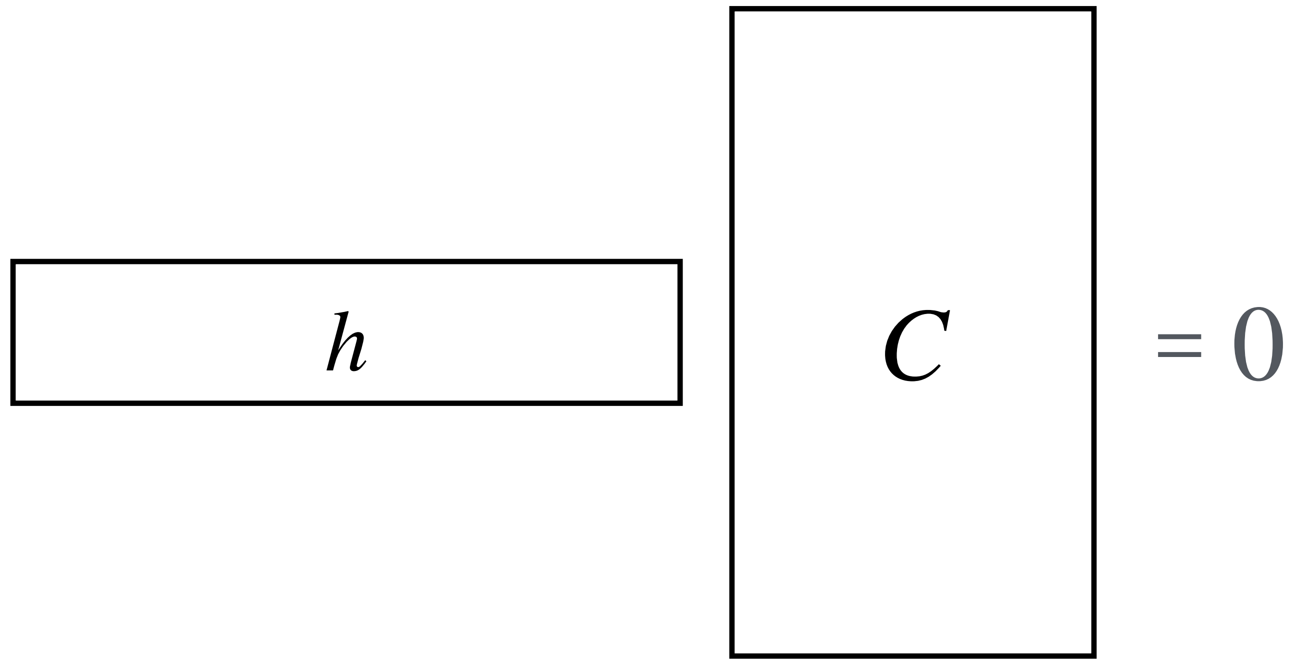
Techniques

High-Level Ideas: light dual vectors



Input (C, w), denote $w = C \cdot x + e$

Dual space $\mathscr{C}^{\perp} = \{h : \langle h, v \rangle = 0, \forall v \in \mathscr{C}\}$



$\langle h, w \rangle = \langle h, C \cdot x + e \rangle = \langle h, e \rangle$

If $h \leftarrow$ distribution of sparse dual vector

- w is η-close $\quad \mathbb{E}_h(-1)^{\langle h,w \rangle} > 1/\mathrm{poly}(n)$
- w is β-far $\quad \mathbb{E}_h(-1)^{\langle h,w \rangle} < \mathrm{negl}(n)$

worst-case distinguisher

Main result

Algorithm with Preprocessing for (Search) BNCP

Theorem 1. There is an algorithm with preprocessing for $\mathrm{DBNCP}_{\beta,\eta}$ with both advice size and running time $m^2 \exp[O(\eta n / \log(1/\beta))]$.

Theorem 2. There is an algorithm with preprocessing for $\mathrm{BNCP}_{\beta,\eta}$ with both advice size and running time $(m^2 \log(1/\alpha)/n)^2 \cdot \exp[O(\eta n / \log(1/\alpha))]$, where $\alpha = \beta + 2\eta$.

Corollary 2. When $\eta = O(\log^2 n / n)$, there is an algorithm with preprocessing for BNCP with both advice size and running time polynomial in n.

Thank you!