

# Computing an LLL-reduced Basis of the Orthogonal Lattice

**Jingwei Chen**   Damien Stehlé   Gilles Villard

**Chongqing Institute of Green and Intelligent Technology,  
Chinese Academy of Sciences**

Laboratoire LIP

(UMR CNRS - ENS Lyon - UCB Lyon 1 - INRIA 5668)

The 43rd ISSAC @ CUNY

# Motivation

The problem: Given  $\mathbf{A} \in \mathbb{Z}^{n \times k}$ , consider using **LLL** to reduce

$$\begin{pmatrix} \mathbf{K} \cdot \mathbf{a}_{1,1} & \mathbf{K} \cdot \mathbf{a}_{1,2} & \cdots & \mathbf{K} \cdot \mathbf{a}_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K} \cdot \mathbf{a}_{k,1} & \mathbf{K} \cdot \mathbf{a}_{k,2} & \cdots & \mathbf{K} \cdot \mathbf{a}_{k,n} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix}$$

# Motivation

The problem: Given  $\mathbf{A} \in \mathbb{Z}^{n \times k}$ , consider using **LLL** to reduce

$$\begin{pmatrix} \mathbf{K} \cdot a_{1,1} & \mathbf{K} \cdot a_{1,2} & \cdots & \mathbf{K} \cdot a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K} \cdot a_{k,1} & \mathbf{K} \cdot a_{k,2} & \cdots & \mathbf{K} \cdot a_{k,n} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix} \xrightarrow[\mathbf{K} \text{ large enough}]{\text{rank}(\mathbf{A})=k, \text{ LLL}} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{C}_{n \times (n-k)} & * \end{pmatrix}.$$

# Motivation

The problem: Given  $\mathbf{A} \in \mathbb{Z}^{n \times k}$ , consider using **LLL** to reduce

$$\begin{pmatrix} \mathbf{K} \cdot a_{1,1} & \mathbf{K} \cdot a_{1,2} & \cdots & \mathbf{K} \cdot a_{1,n} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{K} \cdot a_{k,1} & \mathbf{K} \cdot a_{k,2} & \cdots & \mathbf{K} \cdot a_{k,n} \\ 1 & 0 & \cdots & 0 \\ 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \cdots & 1 \end{pmatrix} \xrightarrow[\mathbf{K} \text{ large enough}]{\text{rank}(\mathbf{A})=k, \text{LLL}} \begin{pmatrix} \mathbf{0} & * \\ \mathbf{C}_{n \times (n-k)} & * \end{pmatrix}.$$

Then  $\mathbf{C}$  gives short vectors of

$$\mathcal{L}^\perp(\mathbf{A}) = \{\mathbf{m} \in \mathbb{Z}^n : \mathbf{A}^\top \mathbf{m} = \mathbf{0}\} = \ker(\mathbf{A}^\top) \cap \mathbb{Z}^n,$$

which we call the **orthogonal lattice** of  $\mathbf{A}$  (**kernel lattice** of  $\mathbf{A}^\top$ ).

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$ .

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$ .

- How large should the **scaling parameter  $K$**  be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$ .

- How large should the **scaling parameter K** be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .
- How does K impact the complexity bound of LLL?
  - [Lenstra, Lenstra, Lovász '82]:  $\#iterations = \mathcal{O}(n^2 \log(K \|\mathbf{A}^\top\|))$ .

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^T$ .

- How large should the **scaling parameter K** be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|a_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .
- How does K impact the complexity bound of LLL?
  - [Lenstra, Lenstra, Lovász '82]:  $\#iterations = \mathcal{O}(n^2 \log(K \|\mathbf{A}^T\|))$ .

Example:  $n = 4, k = 2$ .

$$\mathbf{A} = \begin{pmatrix} 8 & 69 & 99 & 29 \\ 44 & 92 & -31 & 67 \end{pmatrix}^T$$

- sufficient  $K > 253\,600$ ;



# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^T$ .

- How large should the **scaling parameter K** be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|a_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .
- How does K impact the complexity bound of LLL?
  - [Lenstra, Lenstra, Lovász '82]:  $\#iterations = \mathcal{O}(n^2 \log(K \|\mathbf{A}^T\|))$ .

Example:  $n = 4, k = 2$ .

$$\mathbf{A} = \begin{pmatrix} 8 & 69 & 99 & 29 \\ 44 & 92 & -31 & 67 \end{pmatrix}^T$$

- sufficient  $K > 253\,600$ ; heuristic  $K > 2\,015$ ;

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^T$ .

- How large should the **scaling parameter K** be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|a_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .
- How does K impact the complexity bound of LLL?
  - [Lenstra, Lenstra, Lovász '82]:  $\#iterations = \mathcal{O}(n^2 \log(K \|\mathbf{A}^T\|))$ .

Example:  $n = 4, k = 2$ .

$$\mathbf{A} = \begin{pmatrix} 8 & 69 & 99 & 29 \\ 44 & 92 & -31 & 67 \end{pmatrix}^T$$

- sufficient  $K > 253\,600$ ; heuristic  $K > 2\,015$ ; **best**  $K = 233$ .

# Motivation

The problem: LLL reducing  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^T$ .

- How large should the **scaling parameter K** be?
  - Sufficient:  $K > 2^{\frac{n-1}{2}} \cdot (n-k)^{\frac{n-k}{2}} \cdot \|\mathbf{A}\|^k$ , where  $\|\mathbf{A}\| = \max \|a_i\|$ .
  - Heuristic:  $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$ .
- How does K impact the complexity bound of LLL?
  - [Lenstra, Lenstra, Lovász '82]:  $\#iterations = \mathcal{O}(n^2 \log(K \|\mathbf{A}^T\|))$ .

Example:  $n = 4, k = 2$ .

$$\mathbf{A} = \begin{pmatrix} 8 & 69 & 99 & 29 \\ 44 & 92 & -31 & 67 \end{pmatrix}^T$$

- sufficient  $K > 253\,600$ ; heuristic  $K > 2\,015$ ; **best**  $K = 233$ .
- When  $K > 458$ , **the number of LLL iterations remains**.

# Contribution

- [Pohst '87], [Havas, Majewski & Matthews '98]


# Contribution

- A better bound on **#iterations** of LLL for computing a reduced basis of the orthogonal lattice  $\mathcal{L}^\perp(\mathbf{A})$ .
  - We prove that **#iterations** is **independent** of  $K$  for large  $K$ .
  - [Pohst '87], [Havas, Majewski & Matthews '98]

# Contribution

- A new potential function for the LLL algorithm.
- A better bound on #iterations of LLL for computing a reduced basis of the orthogonal lattice  $\mathcal{L}^\perp(\mathbf{A})$ .
  - We prove that #iterations is independent of  $K$  for large  $K$ .
  - [Pohst '87], [Havas, Majewski & Matthews '98]

# Contribution

- A new potential function for the LLL algorithm.  
 a variant of the classic one
- A better bound on #iterations of LLL for computing a reduced basis of the orthogonal lattice  $\mathcal{L}^\perp(\mathbf{A})$ .
  - We prove that #iterations is independent of  $K$  for large  $K$ .
  - [Pohst '87], [Havas, Majewski & Matthews '98]

# Contribution

captures the behavior of LLL more accurately

- A new potential function for the LLL algorithm.

a variant of the classic one

- A better bound on #iterations of LLL for computing a reduced basis of the orthogonal lattice  $\mathcal{L}^\perp(\mathbf{A})$ .
  - We prove that #iterations is independent of  $K$  for large  $K$ .
  - [Pohst '87], [Havas, Majewski & Matthews '98]



# Background

# Lattices and LLL reduced basis

- An  $n$ -dim. lattice:  $\Lambda = \sum \mathbb{Z} \cdot \mathbf{b}_i$  for **linearly indep.**  $(\mathbf{b}_i)_{i \leq n}$ .
- Lattice basis:  $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ .
- SVP: Given a basis of  $\Lambda$ , find a shortest non-zero vector.
  - SVP is hard.
  - But, approximations (e.g., **LLL-reduced bases**) are still useful.

# Lattices and LLL reduced basis

- An  $n$ -dim. lattice:  $\Lambda = \sum \mathbb{Z} \cdot \mathbf{b}_i$  for **linearly indep.**  $(\mathbf{b}_i)_{i \leq n}$ .
- Lattice basis:  $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ .
- SVP: Given a basis of  $\Lambda$ , find a shortest non-zero vector.
  - SVP is hard.
  - But, approximations (e.g., **LLL-reduced bases**) are still useful.

## LLL-reduced basis

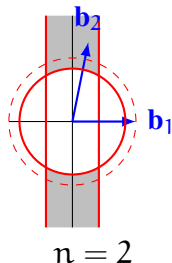
Let  $\mathbf{b}_1, \dots, \mathbf{b}_n$  be a basis for a lattice  $\Lambda$ ,  $\mathbf{b}_i^*$  the  $i^{\text{th}}$  GS vector, and  $\mu_{i,j}$  the GS coefficients. Then we call the basis is LLL-reduced if

$$(1) \quad |\mu_{i,j}| \leq \frac{1}{2} \text{ for } 1 \leq j \leq i \leq n,$$

$$(2) \quad \|\mathbf{b}_i^*\|^2 \leq 2\|\mathbf{b}_{i+1}^*\|^2 \text{ for } 1 \leq i \leq n-1. \text{ [Siegel condition]}$$

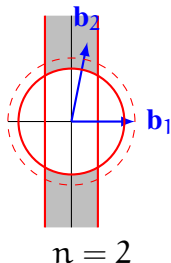
# Lattices and LLL reduced basis

- An  $n$ -dim. lattice:  $\Lambda = \sum \mathbb{Z} \cdot \mathbf{b}_i$  for **linearly indep.**  $(\mathbf{b}_i)_{i \leq n}$ .
- Lattice basis:  $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ .
- SVP: Given a basis of  $\Lambda$ , find a shortest non-zero vector.
  - SVP is hard.
  - But, approximations (e.g., **LLL-reduced bases**) are still useful.



# Lattices and LLL reduced basis

- An  $n$ -dim. lattice:  $\Lambda = \sum \mathbb{Z} \cdot \mathbf{b}_i$  for **linearly indep.**  $(\mathbf{b}_i)_{i \leq n}$ .
- Lattice basis:  $\mathbf{B} = (\mathbf{b}_1, \mathbf{b}_2, \dots, \mathbf{b}_n)$ .
- SVP: Given a basis of  $\Lambda$ , find a shortest non-zero vector.
  - SVP is hard.
  - But, approximations (e.g., **LLL-reduced bases**) are still useful.



## LLL-reduced is “nice”

- not too far from orthogonal
- GS lengths do not drop “too” fast
- short first vector:  $\|\mathbf{b}_1\| \leq 2^{\frac{n-1}{2}} \lambda_1(\Lambda)$ ,  
where  $\lambda_1(\Lambda) = \min\{\|\mathbf{b}\| \mid \mathbf{b} \in \Lambda \setminus \{\mathbf{0}\}\}$ .

# The LLL algorithm [Lenstra, Lenstra, Lovász '82]

**Input:** A basis  $(\mathbf{b}_i)_{i \leq n}$  of a lattice  $\Lambda \subseteq \mathbb{Z}^m$ .

**Output:** An LLL-reduced basis of  $\Lambda$ .

- 1  $k := 1$ .
- 2 While  $k \leq n - 1$  do
  - a. Size-reduce  $\mathbf{b}_{k+1}$  with respect to  $\mathbf{b}_k$ .
  - b. If the Siegel condition holds for  $k$ , then  $k := k + 1$ .
  - c. Else **SWAP**  $\mathbf{b}_k$  and  $\mathbf{b}_{k+1}$ ; set  $k := \max\{k - 1, 1\}$ .
- 3 Return the current basis  $(\mathbf{b}_i)_{i \leq n}$ .

# The LLL algorithm [Lenstra, Lenstra, Lovász '82]

**Input:** A basis  $(\mathbf{b}_i)_{i \leq n}$  of a lattice  $\Lambda \subseteq \mathbb{Z}^m$ .

**Output:** An LLL-reduced basis of  $\Lambda$ .

- ①  $k := 1$ .
- ② While  $k \leq n - 1$  do
  - a. Size-reduce  $\mathbf{b}_{k+1}$  with respect to  $\mathbf{b}_k$ .
  - b. If the Siegel condition holds for  $k$ , then  $k := k + 1$ .
  - c. Else **SWAP**  $\mathbf{b}_k$  and  $\mathbf{b}_{k+1}$ ; set  $k := \max\{k - 1, 1\}$ .
- ③ Return the current basis  $(\mathbf{b}_i)_{i \leq n}$ .

The cost  $\approx$  “#iterations”  $\times$  “the cost of per iteration”

# The LLL algorithm [Lenstra, Lenstra, Lovász '82]

**Input:** A basis  $(\mathbf{b}_i)_{i \leq n}$  of a lattice  $\Lambda \subseteq \mathbb{Z}^m$ .

**Output:** An LLL-reduced basis of  $\Lambda$ .

- ①  $k := 1$ .
- ② While  $k \leq n - 1$  do
  - a. Size-reduce  $\mathbf{b}_{k+1}$  with respect to  $\mathbf{b}_k$ .
  - b. If the Siegel condition holds for  $k$ , then  $k := k + 1$ .
  - c. Else **SWAP**  $\mathbf{b}_k$  and  $\mathbf{b}_{k+1}$ ; set  $k := \max\{k - 1, 1\}$ .
- ③ Return the current basis  $(\mathbf{b}_i)_{i \leq n}$ .

The cost  $\approx$  “#iterations”  $\times$  “the cost of per iteration”

- #iterations  $\leq 2$ #swaps + n; #swaps =  $\mathcal{O}(n^2 \log \|\mathbf{B}\|)$ .



# The classic potential for LLL

Let  $\mathbf{B}$  be a basis of an  $n$ -dimensional lattice. Define

$$\Pi(\mathbf{B}) = \sum_{i=1}^{n-1} (n - i) \log \|\mathbf{b}_i^*\|.$$

# The classic potential for LLL

Let  $\mathbf{B}$  be a basis of an  $n$ -dimensional lattice. Define

$$\Pi(\mathbf{B}) = \sum_{i=1}^{n-1} (n - i) \log \|\mathbf{b}_i^*\|.$$

## Properties

- At the beginning,  $\Pi(\mathbf{B})$  can be bounded from above.
- Each LLL swap decreases  $\Pi(\mathbf{B})$  by a constant.
- At the end,  $\Pi(\mathbf{B})$  can be bounded from below.

# The classic potential for LLL

Let  $\mathbf{B}$  be a basis of an  $n$ -dimensional lattice. Define

$$\Pi(\mathbf{B}) = \sum_{i=1}^{n-1} (n-i) \log \|\mathbf{b}_i^*\|.$$

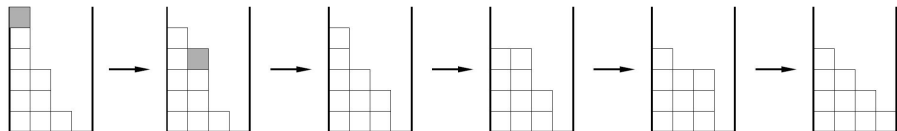


Figure: Sandpile model for LLL (Figure courtesy of Brigitte Vallée)

# The new potential

# Observations

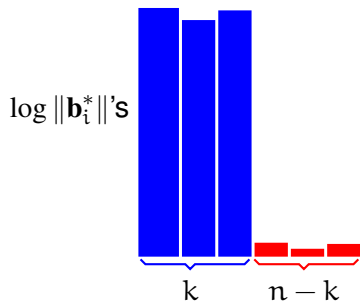


Figure: At the beginning

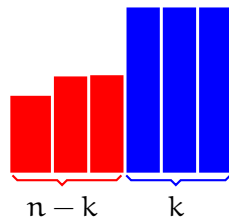


Figure: At the end

# Observations

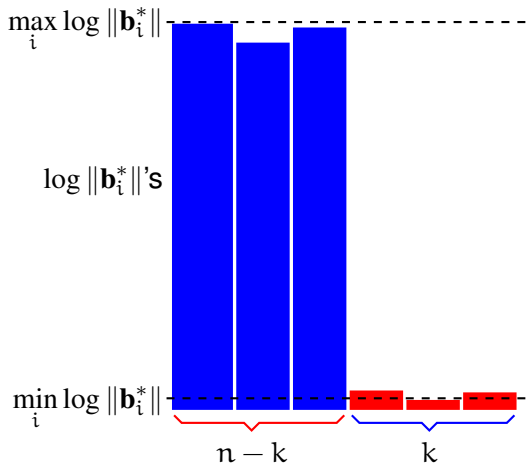


Figure: An example

# Observations

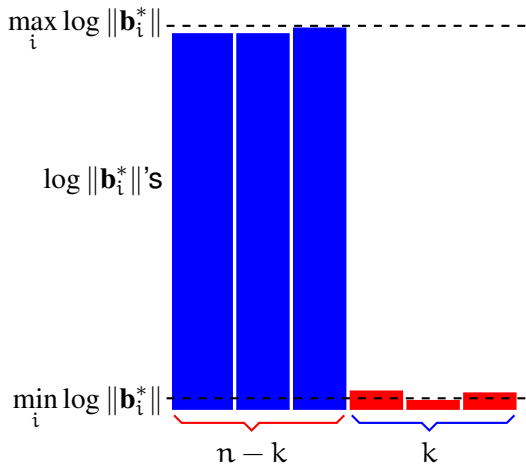


Figure: An example

# Observations

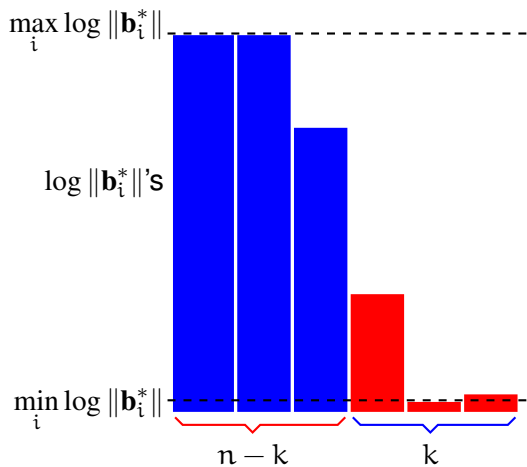


Figure: An example



# Observations

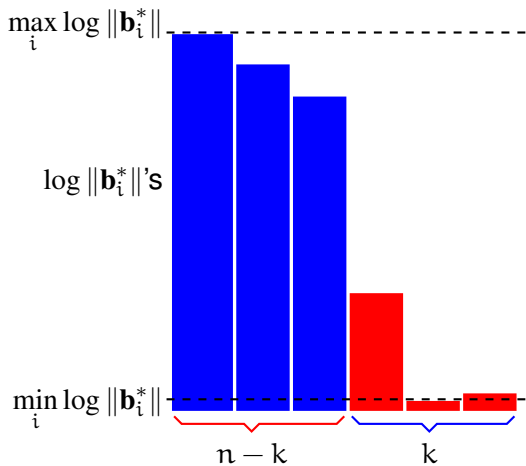


Figure: An example

# Observations

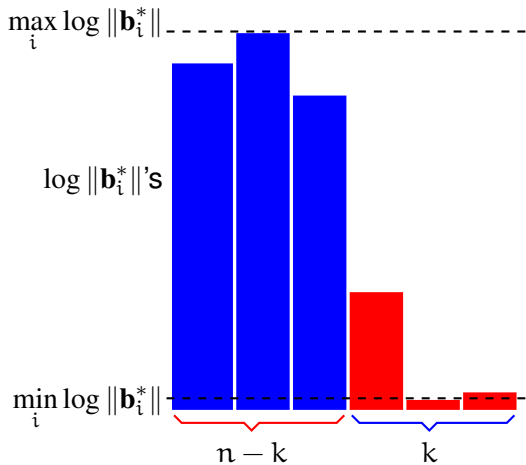


Figure: An example

# Observations

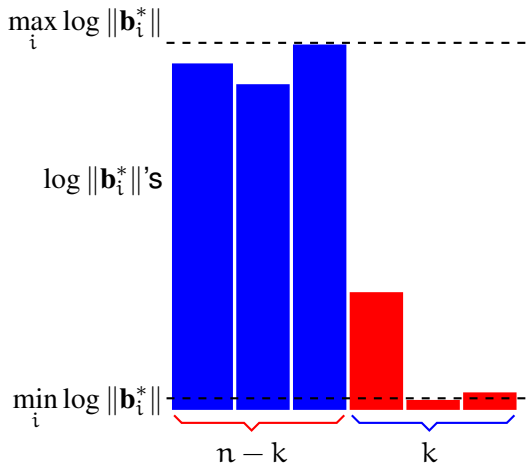


Figure: An example

# Observations

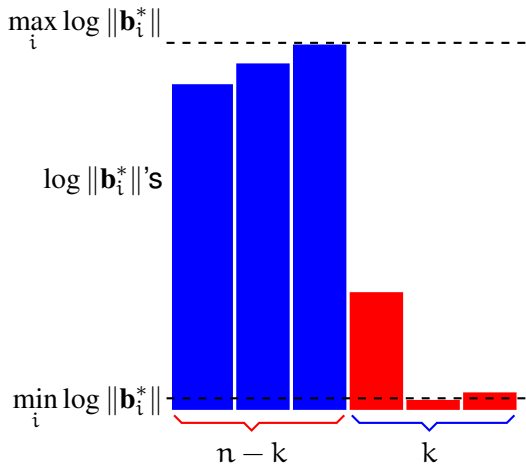


Figure: An example

# Observations

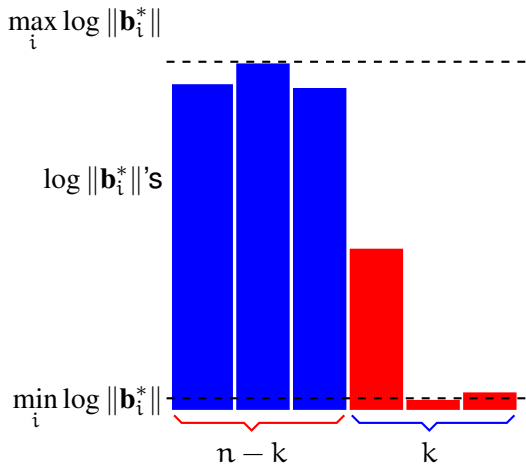


Figure: An example

# Observations

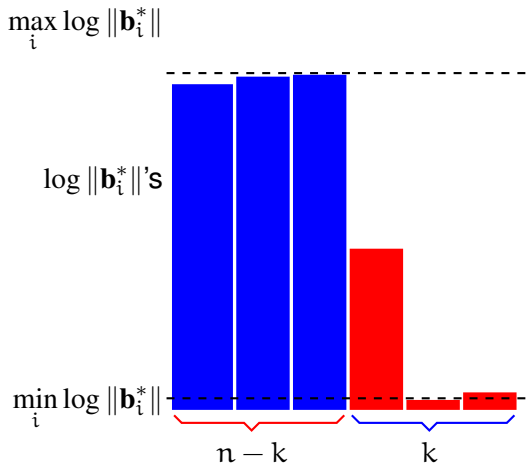


Figure: An example

# Observations

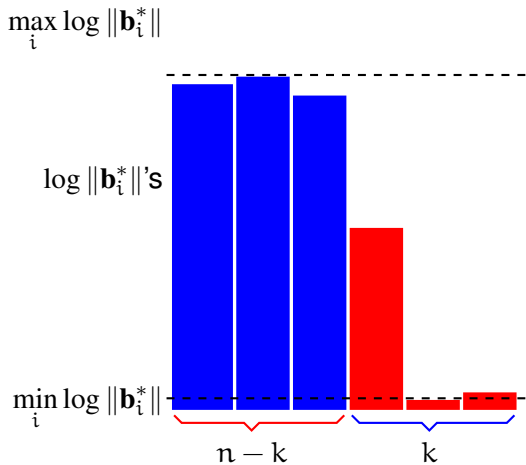


Figure: An example

# Observations

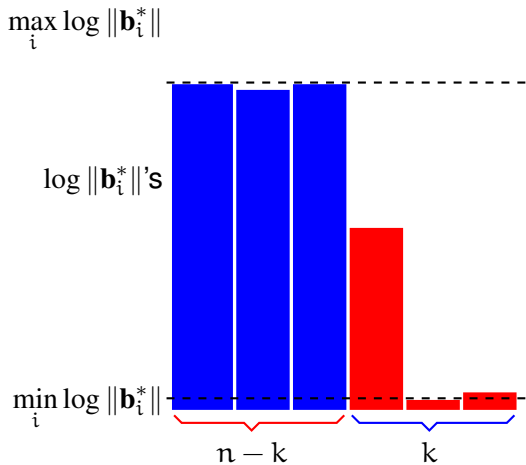


Figure: An example



# Observations

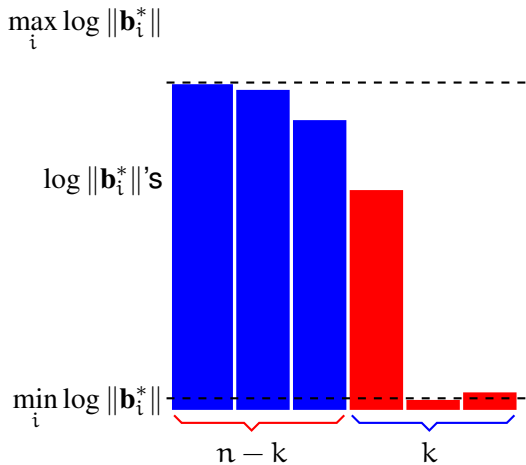


Figure: An example

# Observations

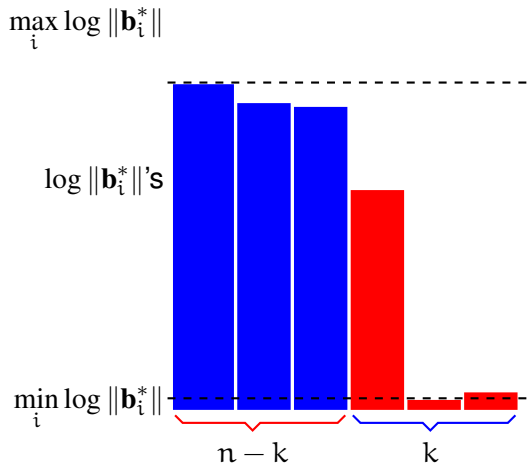


Figure: An example

# Observations

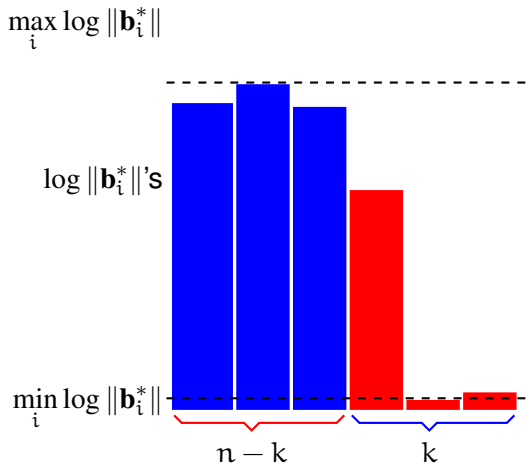


Figure: An example

# Observations

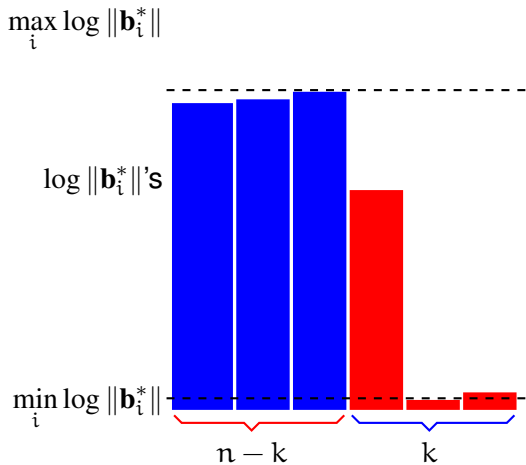


Figure: An example

# Observations

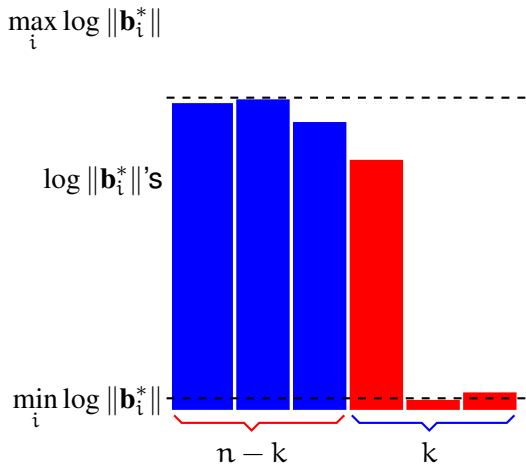


Figure: An example

# Observations

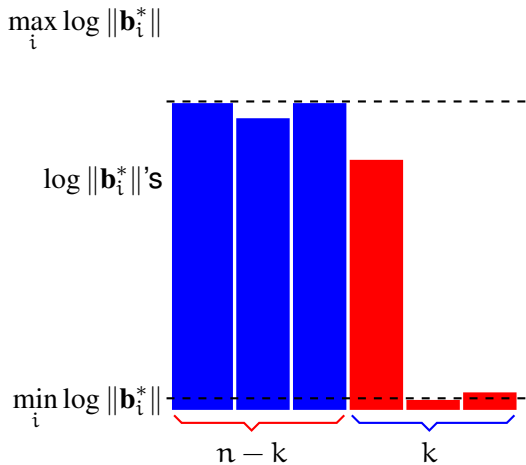


Figure: An example

# Observations

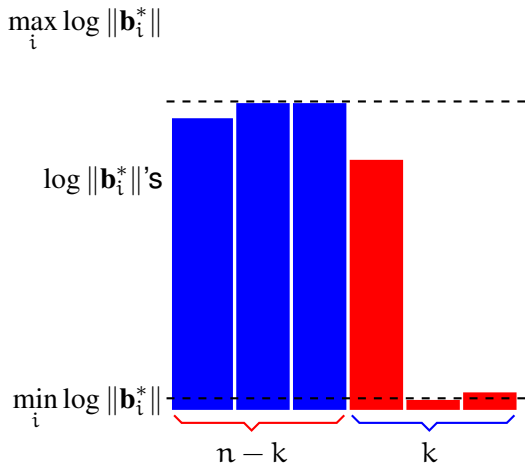


Figure: An example

# Observations

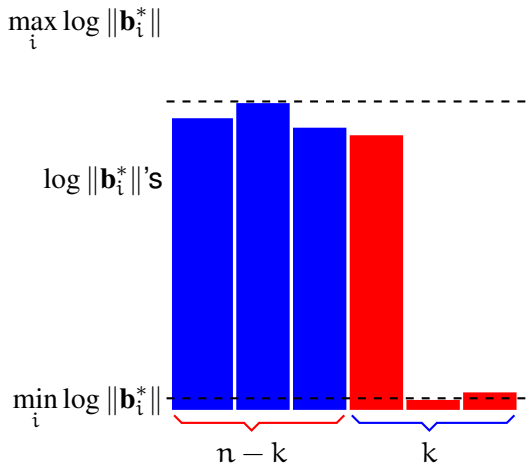


Figure: An example



# Observations

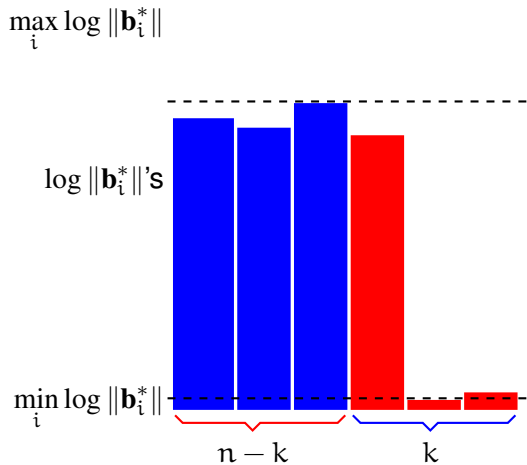


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

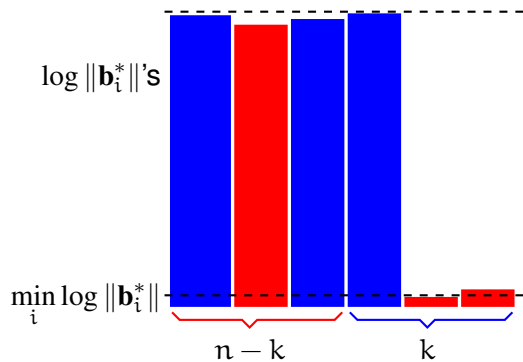


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

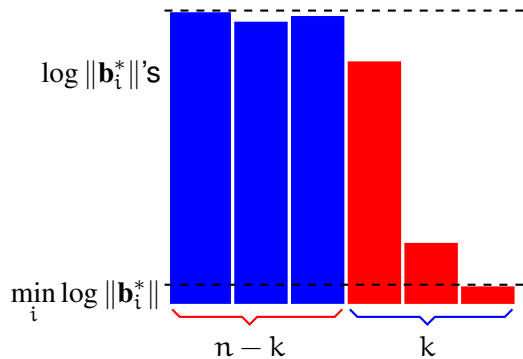


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

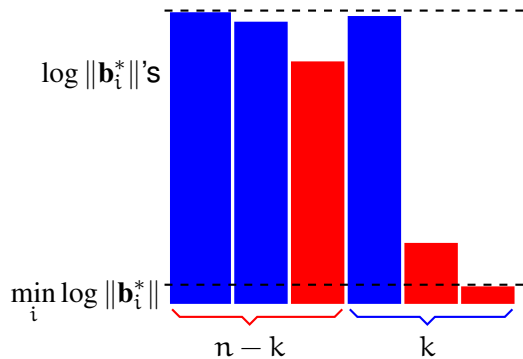


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

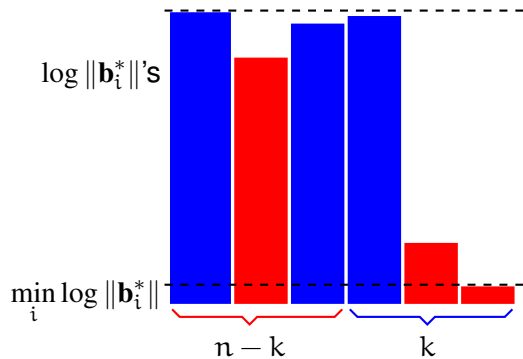


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

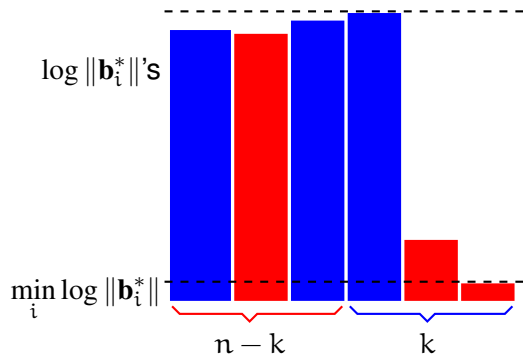


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

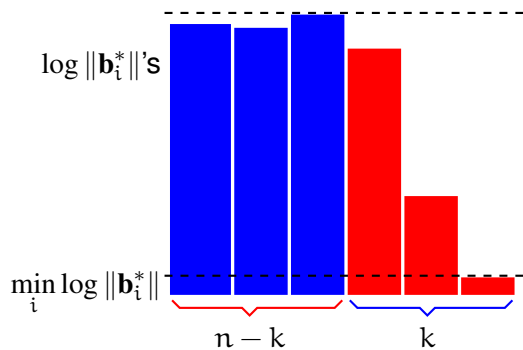


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

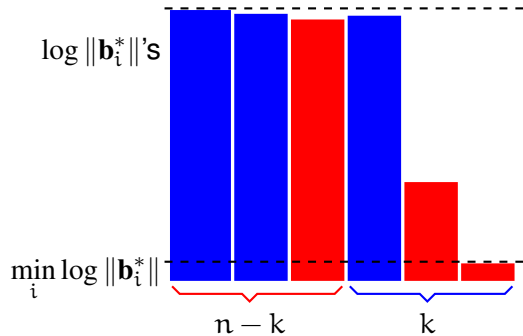


Figure: An example



# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

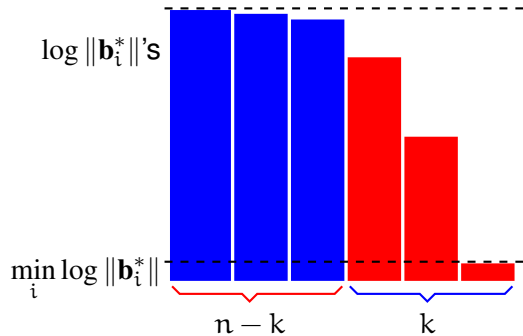


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

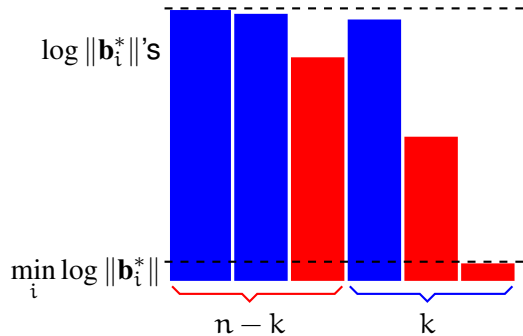


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

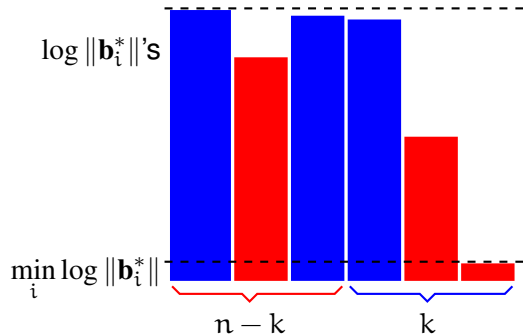


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

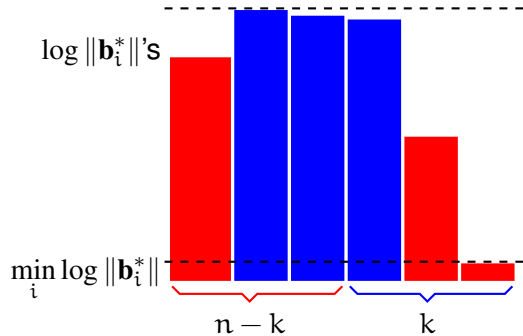


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

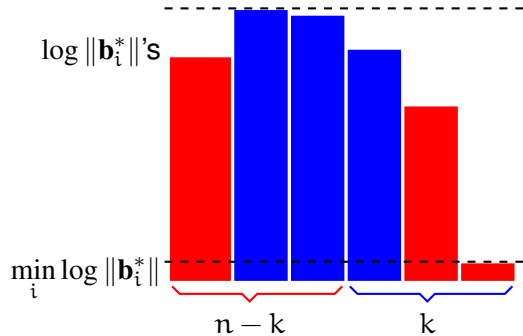


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

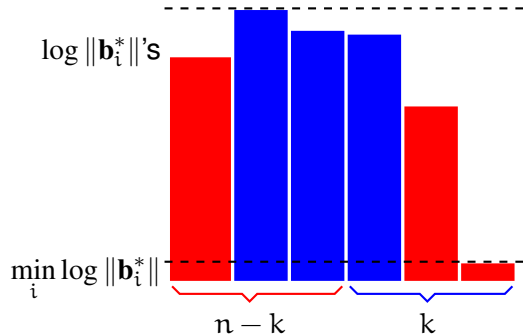


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

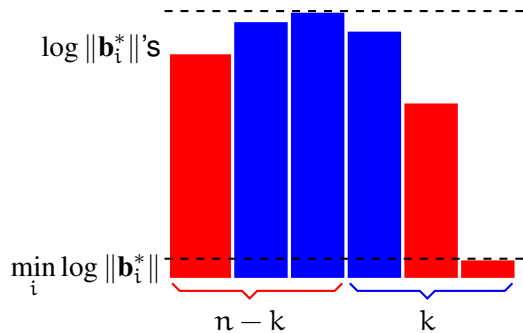


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

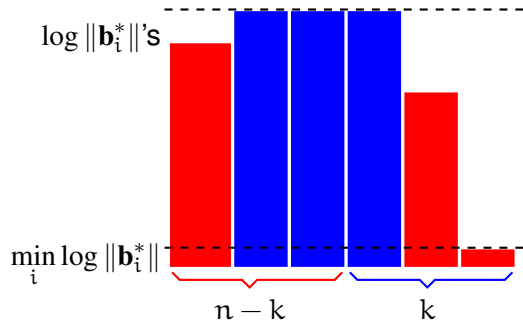


Figure: An example



# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

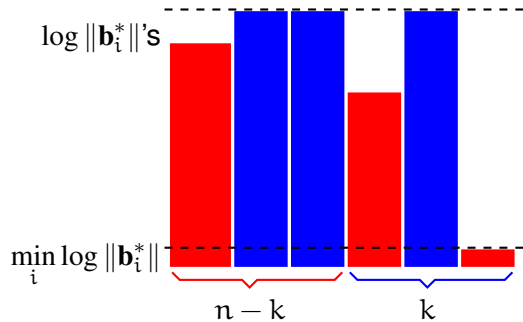


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

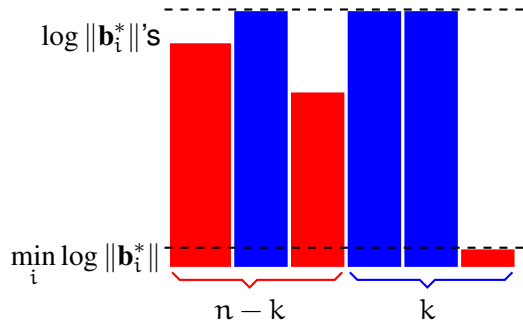


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

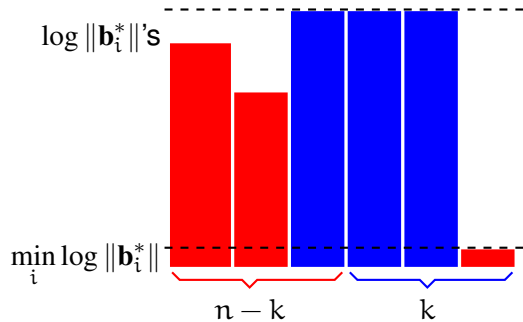


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

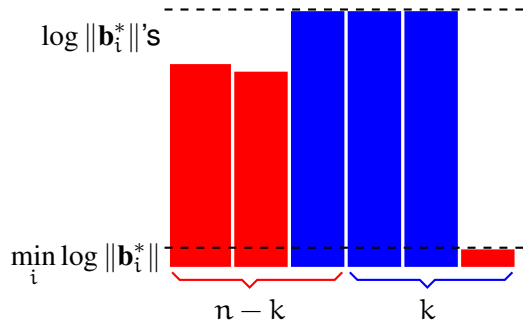


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

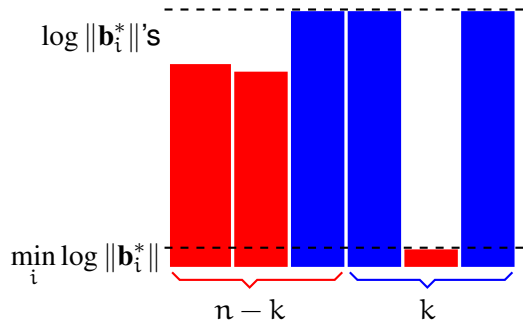


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

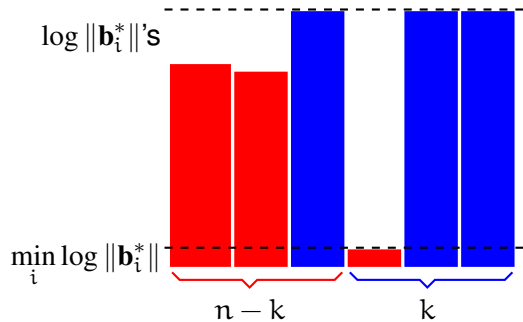


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

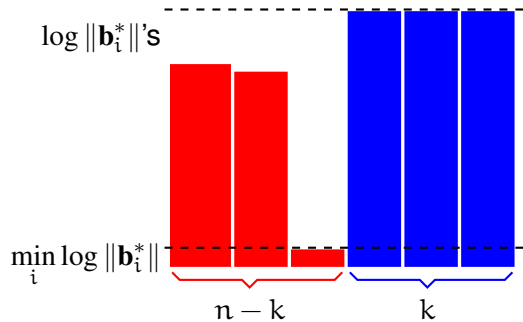


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

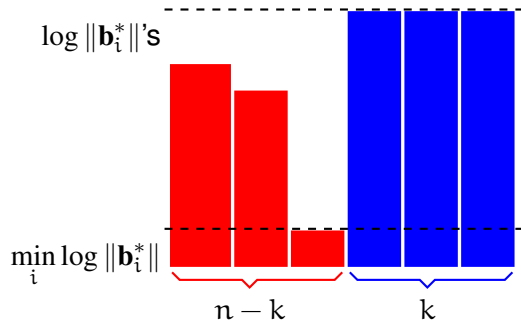


Figure: An example



# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

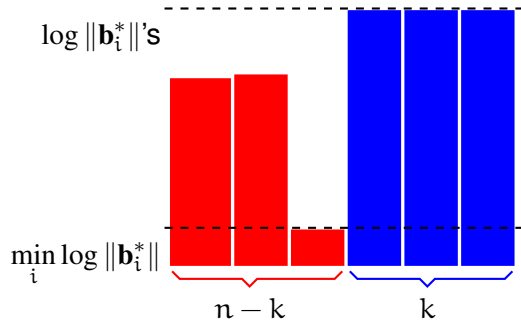


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

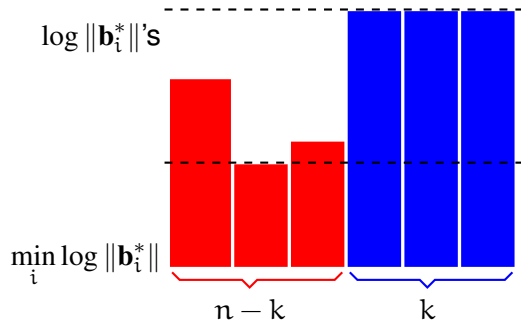


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

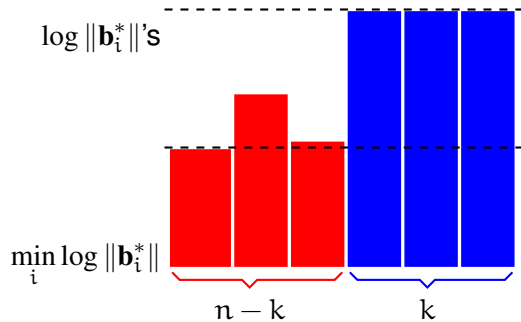


Figure: An example

# Observations

$$\max_i \log \|\mathbf{b}_i^*\|$$

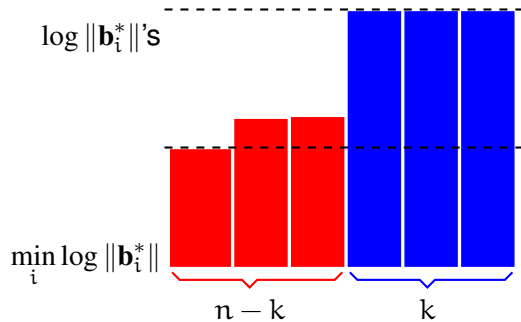


Figure: An example

# Basic idea: capture the unbalance

## Observation

Those vectors with small GS lengths do not interfere much with those vectors with large GS lengths.

# Basic idea: capture the unbalance

## Observation

Those vectors with small GS lengths do not interfere much with those vectors with large GS lengths.

- Partition the vectors into two groups by their GS lengths
  - the  $k$  vectors with larger GS length
  - the other  $n - k$  vectors with smaller GS length

# Basic idea: capture the unbalance

## Observation

Those vectors with small GS lengths do not interfere much with those vectors with large GS lengths.

- Partition the vectors into two groups by their GS lengths
  - the  $k$  vectors with larger GS length
  - the other  $n - k$  vectors with smaller GS length
- Partition all swaps into three kinds
  - small  $\leftrightarrow$  small
  - large  $\leftrightarrow$  large
  - small  $\leftrightarrow$  large
- [van Hoeij & Novocin '10]

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .



# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \cdots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

We define

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k-j) \log \|\mathbf{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\mathbf{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

We define

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k-j) \log \|\mathbf{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\mathbf{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

large  $\leftrightarrow$  large

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

We define

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k-j) \log \|\mathbf{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\mathbf{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

large  $\leftrightarrow$  large

small  $\leftrightarrow$  small

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

We define

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k-j) \log \|\mathbf{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\mathbf{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

large  $\leftrightarrow$  large      small  $\leftrightarrow$  small      large  $\leftrightarrow$  small

# The new potential function

- Let  $k \leq n \leq m$  and  $\mathbf{B} \in \mathbb{R}^{m \times n}$ .
- $s_1 < \dots < s_{n-k}$ : the indices of the  $n - k$  smallest GS lengths
- $\ell_1 < \dots < \ell_k$ : the indices of the other  $k$  GS lengths

We define

$$\Pi_k(\mathbf{B}) = \sum_{j=1}^{k-1} (k-j) \log \|\mathbf{b}_{\ell_j}^*\| - \sum_{i=1}^{n-k} i \log \|\mathbf{b}_{s_i}^*\| + \sum_{i=1}^{n-k} s_i.$$

large  $\leftrightarrow$  large      small  $\leftrightarrow$  small      large  $\leftrightarrow$  small

- $\Pi_n(\mathbf{B}) = \Pi(\mathbf{B})$ .

## Monotonicity

Let  $\mathbf{B}$  and  $\mathbf{B}'$  be the current  $n$ -dimensional lattice bases before and after an LLL swap. Then for any  $k \leq n$ , we have

$$\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') \geq \log(2/\sqrt{3}).$$



# Properties of $\Pi_k(\mathbf{B})$

## Monotonicity

Let  $\mathbf{B}$  and  $\mathbf{B}'$  be the current  $n$ -dimensional lattice bases before and after an LLL swap. Then for any  $k \leq n$ , we have

$$\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}') \geq \log(2/\sqrt{3}).$$

## Bounding #swaps

Given full column rank matrix  $\mathbf{B}$  as input, LLL returns  $\mathbf{B}'$ . Then #swaps that LLL performs is no greater than

$$\min_{1 \leq k \leq n} \frac{\Pi_k(\mathbf{B}) - \Pi_k(\mathbf{B}')}{\log\left(\frac{2}{\sqrt{3}}\right)}.$$

# Applying $\Pi_k(\mathbf{B})$ to $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^T$

# Applying $\Pi_k(\mathbf{B})$ to $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$

## The main result

Let  $K$  be a sufficiently large integer. Then, given  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$  as input, LLL computes (as a submatrix of the returned basis) an LLL-reduced basis of  $\mathcal{L}^\perp(\mathbf{A})$  after at most

$$\mathcal{O}(k^3 + k(n - k)(1 + \log \|\mathbf{A}\|))$$

LLL swaps.

## The main result

Let  $K$  be a sufficiently large integer. Then, given  $(K \cdot \mathbf{A}, \mathbf{I}_n)^\top$  as input, LLL computes (as a submatrix of the returned basis) an LLL-reduced basis of  $\mathcal{L}^\perp(\mathbf{A})$  after at most

$$\mathcal{O}(k^3 + k(n - k)(1 + \log \|\mathbf{A}\|))$$

LLL swaps.

## Future work

- Apply to more general bit complexity studies of LLL.
- Apply to more special input bases for LLL.

# Applying $\Pi_k(\mathbf{B})$ to $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$

## The main result

Let  $k$  be a sufficiently large integer. Then, given  $(\mathbf{K} \cdot \mathbf{A}, \mathbf{I}_n)^\top$  as input, LLL computes (as a submatrix of the returned basis) an LLL-reduced basis of  $\mathcal{L}^\perp(\mathbf{A})$  after at most

$$\mathcal{O}(k^3 + k(n - k)(1 + \log \|\mathbf{A}\|))$$

LLL swaps.

## Future work

- Apply to more general bit complexity studies of LLL.
- Apply to more special input bases for LLL.

THANKS