Computing an LLL-reduced basis of the orthogonal lattice

陈经纬

Based on joint work with Damien Stehlé and Gilles Villard

November 11, 2018 @ JNU, Guangzhou

Motivation

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

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

Motivation

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

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

Motivation

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

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

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

$$\mathscr{L}^{\perp}(\mathbf{A}) = \left\{\mathbf{m} \in \mathbb{Z}^n : \mathbf{A}^T\mathbf{m} = \mathbf{0}\right\} = \ker(\mathbf{A}^T) \cap \mathbb{Z}^n,$$

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

Motivation

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

Motivation

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

- 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 $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$.
  - Heuristic: $K > 2^{\Omega(n)} \cdot \|\mathbf{A}\|^{\frac{k}{n-k}}$.

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

- 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 $\|\mathbf{A}\| = \max\|\mathbf{a}_i\|$.
  - 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 $= \mathcal{O}(n^2\log(K\|\mathbf{A}^T\|))$.

Motivation

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

- 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 $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$.
  - 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 $= \mathcal{O}(n^2 \log(K \|\mathbf{A}^T\|))$.

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 $(K \cdot \mathbf{A}, \mathbf{I}_n)^T$.

- 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 $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$.
  - 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 $= \mathcal{O}(n^2 \log(K\|\mathbf{A}^T\|))$.

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 $(K \cdot \mathbf{A}, \mathbf{I}_n)^T$.

- 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 $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$.
  - 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 $= \mathcal{O}(n^2 \log(K\|\mathbf{A}^T\|))$.

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 $(K \cdot \mathbf{A}, \mathbf{I}_n)^T$.

- 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 $\|\mathbf{A}\| = \max \|\mathbf{a}_i\|$.
  - 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 $= \mathcal{O}(n^2 \log(K\|\mathbf{A}^T\|))$.

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 $K = 233$.
- The number of LLL iterations remains for $K > 458$.

Contribution

- [Pohst '87] observed this phenomenon.
- [Havas, Majewski & Matthews '98] proved the case of $k = 1$.

Contribution

- A better bound on #iterations of LLL for computing a reduced basis of the orthogonal lattice $\mathscr{L}^{\perp}(\mathbf{A})$.
  - We prove that #iterations is independent of K for large K.
  - [Pohst '87] observed this phenomenon.
  - [Havas, Majewski & Matthews '98] proved the case of $k = 1$.

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 $\mathscr{L}^{\perp}(\mathbf{A})$.
  - We prove that #iterations is independent of K for large K.
  - [Pohst '87] observed this phenomenon.
  - [Havas, Majewski & Matthews '98] proved the case of $k = 1$.

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 $\mathscr{L}^{\perp}(\mathbf{A})$.
  - We prove that #iterations is independent of K for large K.
  - [Pohst '87] observed this phenomenon.
  - [Havas, Majewski & Matthews '98] proved the case of $k = 1$.

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 $\mathscr{L}^{\perp}(\mathbf{A})$.
  - We prove that #iterations is independent of K for large K.
  - [Pohst '87] observed this phenomenon.
  - [Havas, Majewski & Matthews '98] proved the case of $k = 1$.

Background

Lattices and LLL reduced basis

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

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

LLL-reduced basis

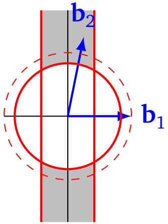
Let $\mathbf{b}_1, \ldots, \mathbf{b}_n$ be a basis for a lattice Λ, $\mathbf{b}_i^*$ the i^{th} GS vector, and $\mu_{i,j}$ the GS coefficients. Then we call the basis is LLL-reduced if

(1) $|\mu_{i,j}| \le \dfrac{1}{2}$ for $1 \le j \le i \le n$,

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

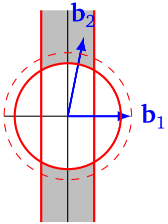
Lattices and LLL reduced basis

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



$n = 2$

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



$n = 2$

LLL-reduced is "nice"

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

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 Λ.

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 Λ.

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 ≈ "#iterations" × "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 Λ.

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 ≈ "#iterations" × "the cost of per iteration"

- #iterations ≤ 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 begining, $\Pi(\mathbf{B})$ 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.

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

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

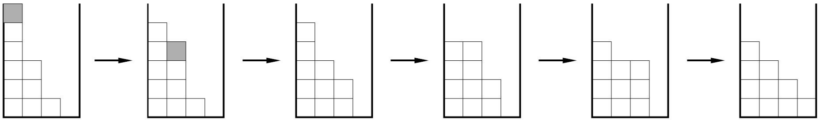


Figure: Sandpile model for LLL (Figure courtesy of Brigitte Vallée)

The new potential

Observations

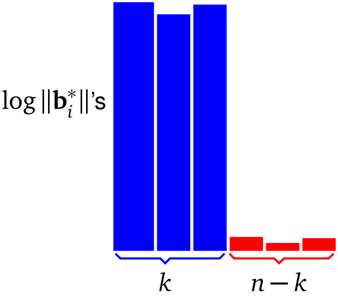


Figure: At the beginning

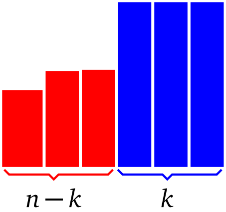


Figure: At the end

Observations

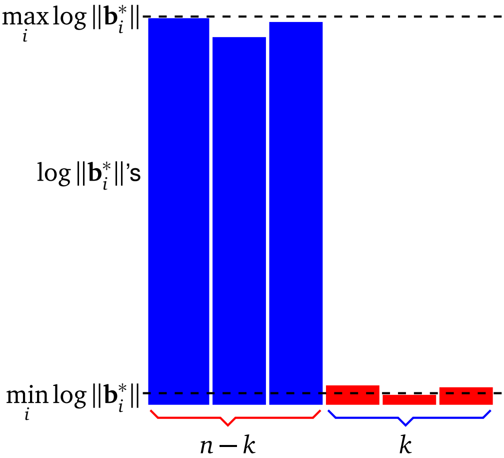


Figure: An example

Observations

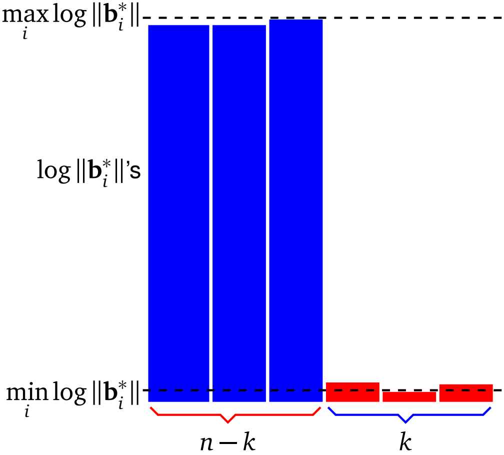


Figure: An example

Observations

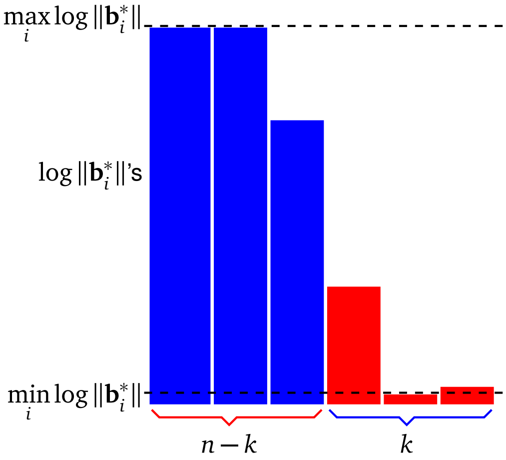


Figure: An example

Observations

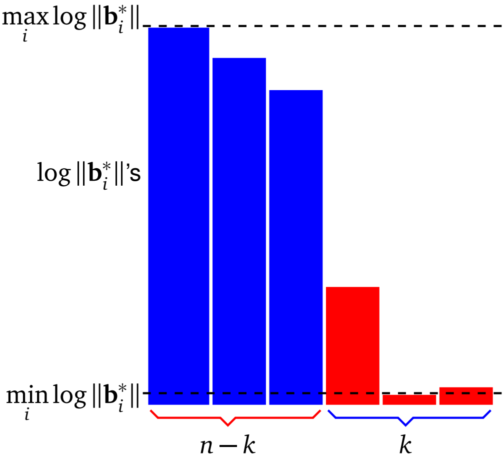


Figure: An example

Observations

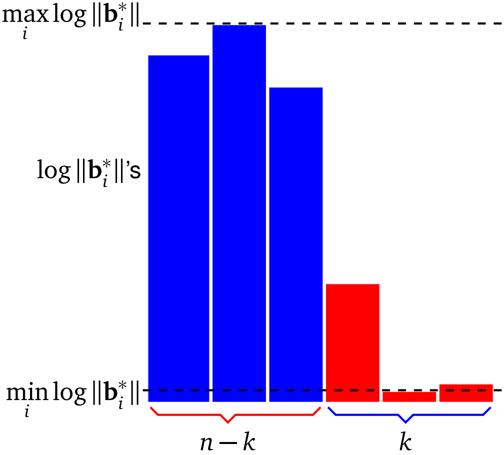


Figure: An example

Observations

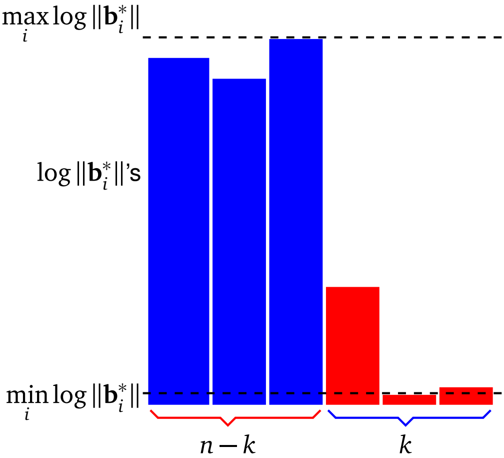


Figure: An example

Observations

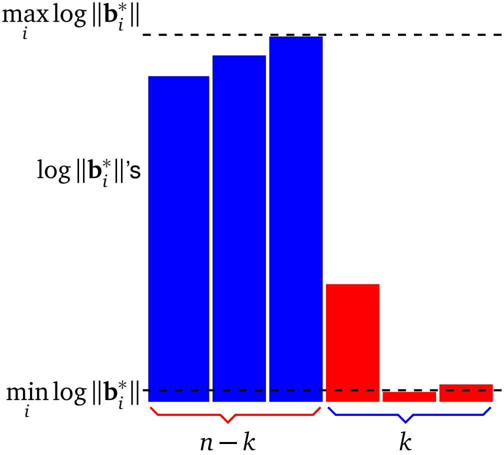


Figure: An example

Observations

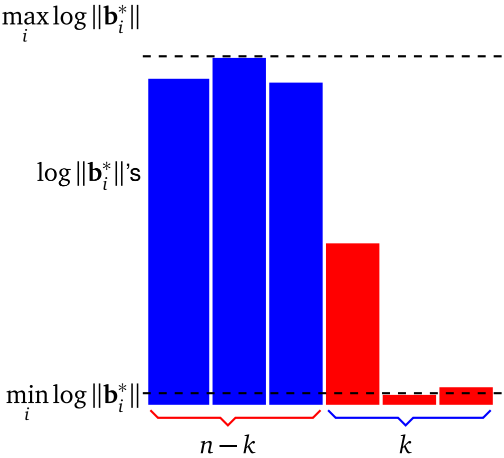


Figure: An example

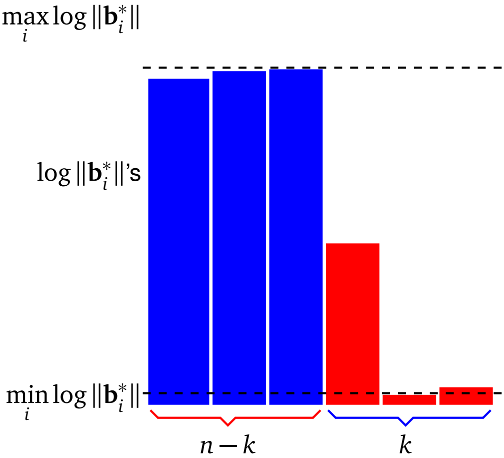


Figure: An example

Observations

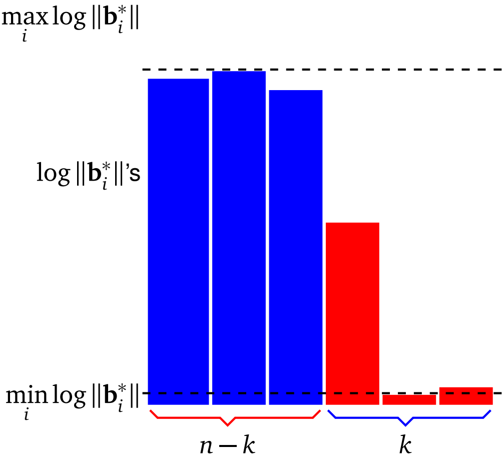


Figure: An example

Observations

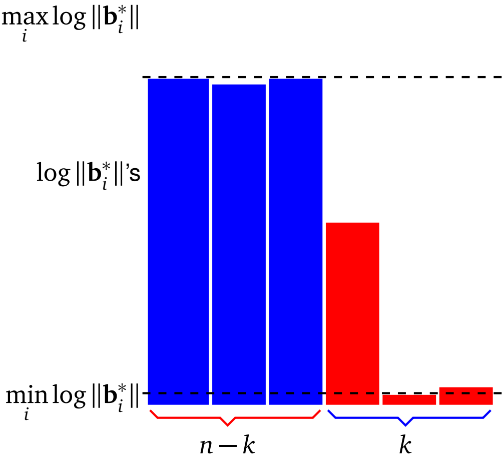


Figure: An example

Observations

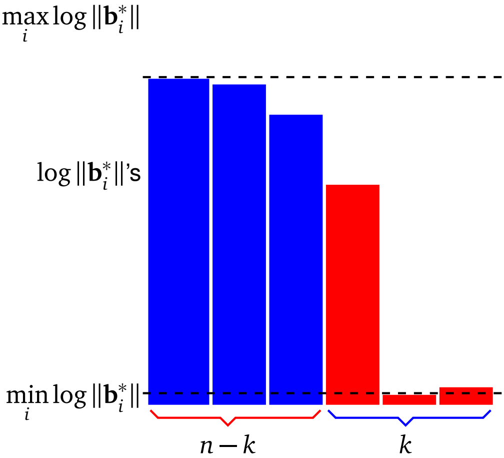


Figure: An example

Observations

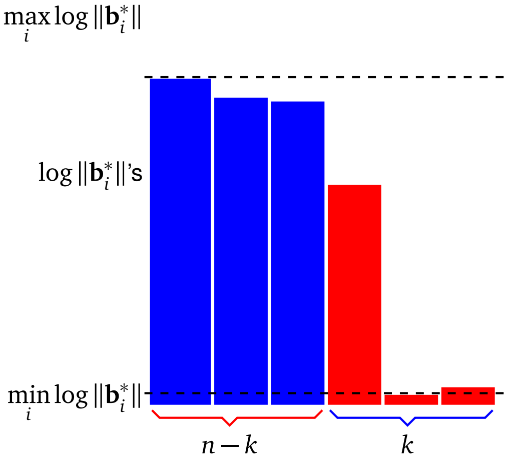


Figure: An example

Observations

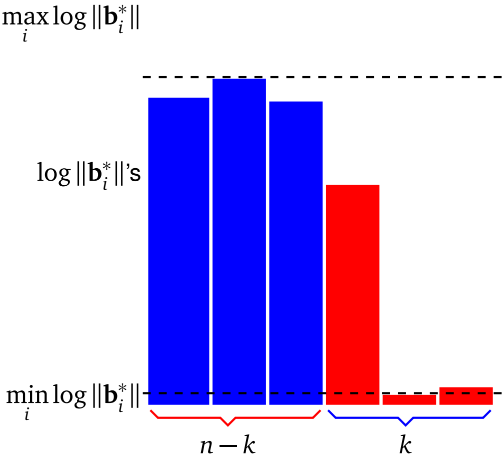


Figure: An example

Observations

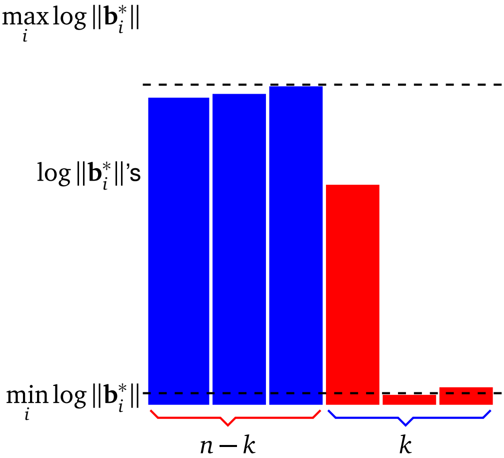


Figure: An example

Observations

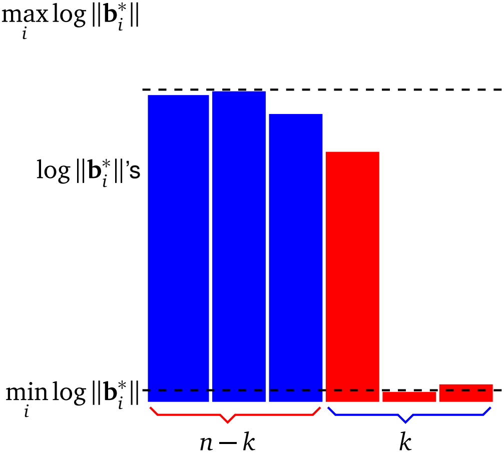


Figure: An example

Observations

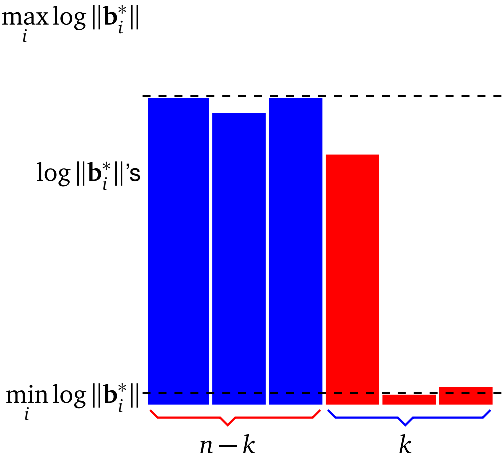


Figure: An example

Observations

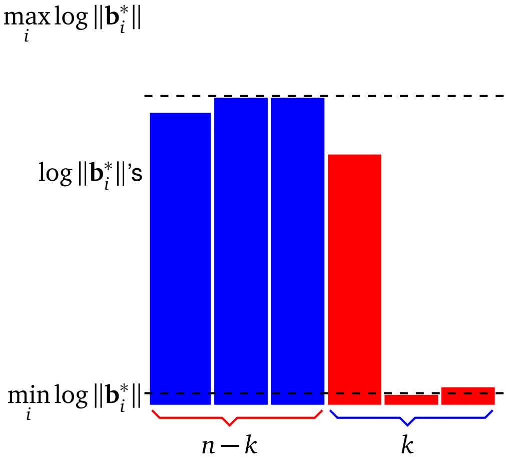


Figure: An example

Observations

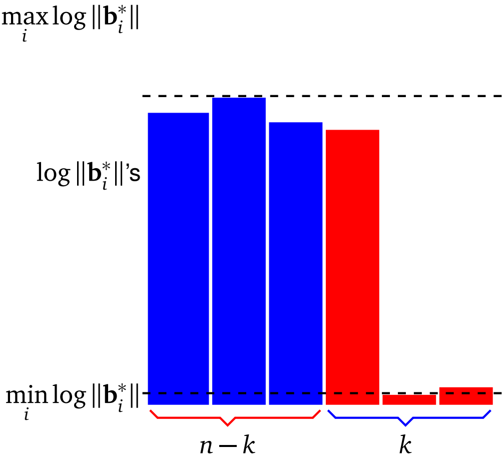


Figure: An example

Observations

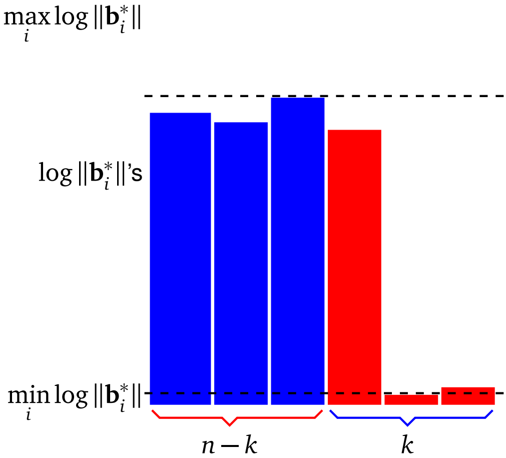


Figure: An example

Observations

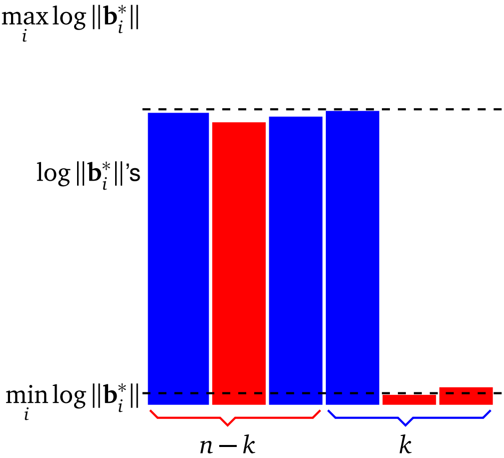


Figure: An example

Observations

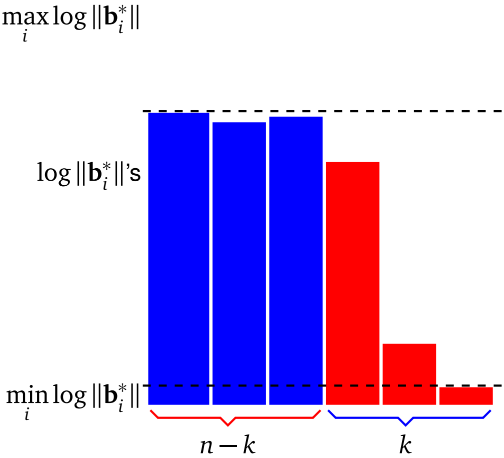


Figure: An example

Observations

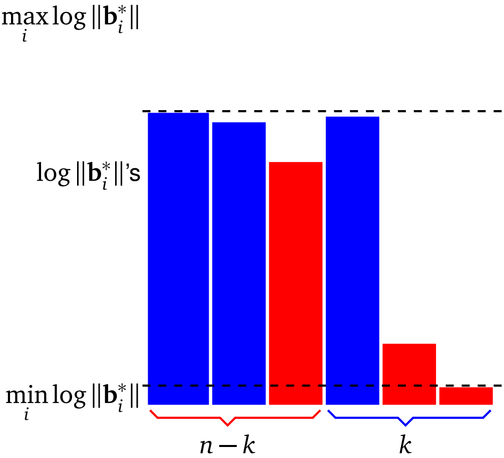


Figure: An example

Observations

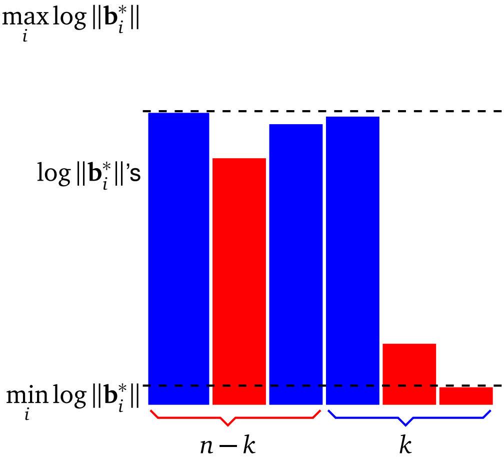


Figure: An example

Observations

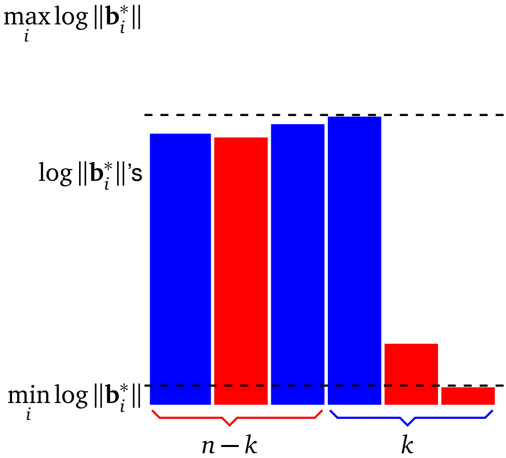


Figure: An example

Observations

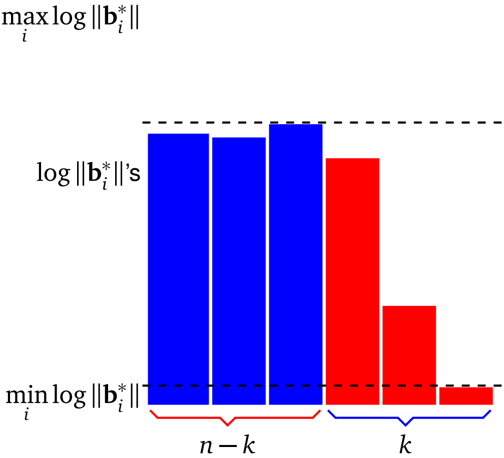


Figure: An example

Observations

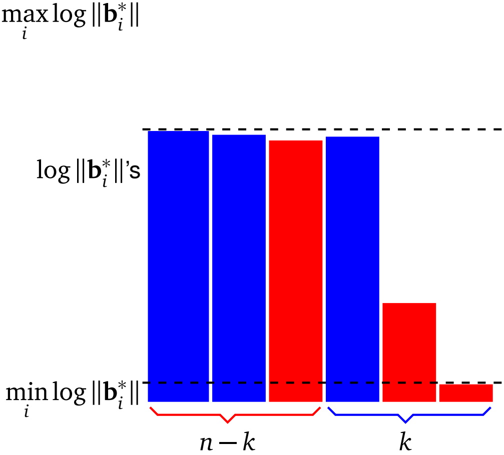


Figure: An example

Observations

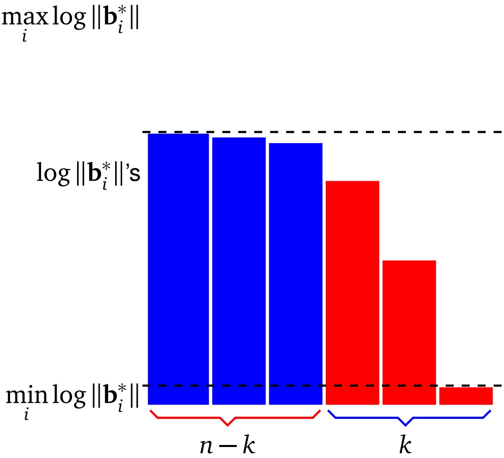


Figure: An example

Observations

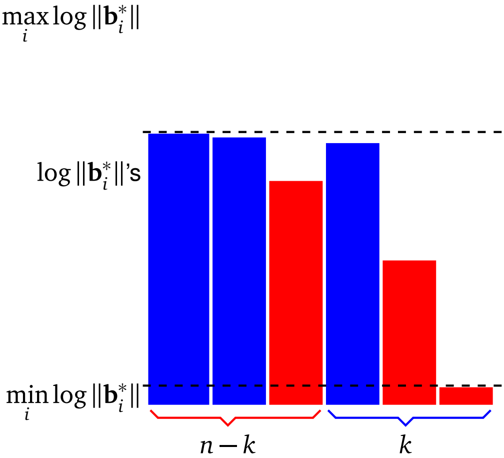


Figure: An example

Observations

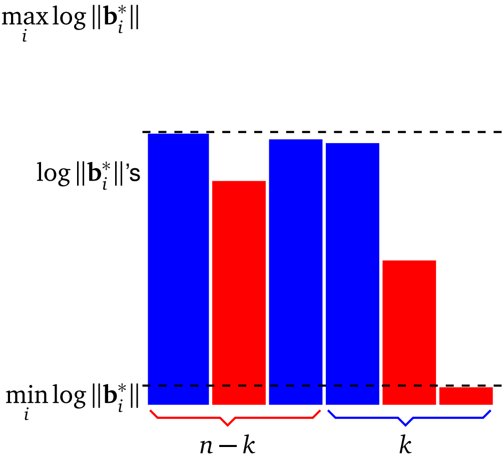


Figure: An example

Observations

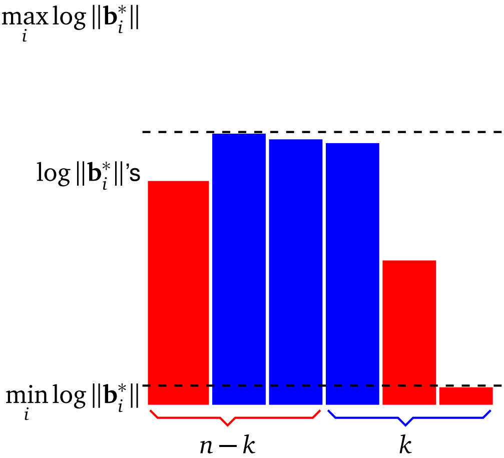


Figure: An example

Observations

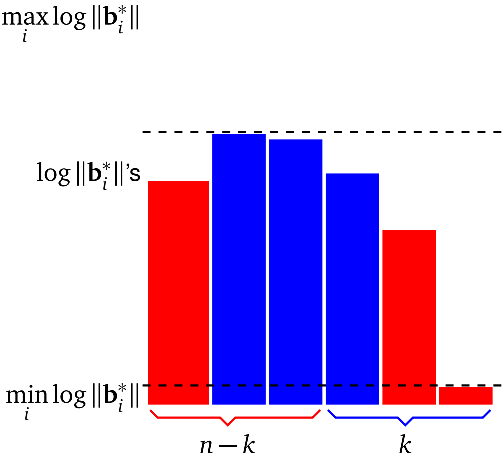


Figure: An example

Observations

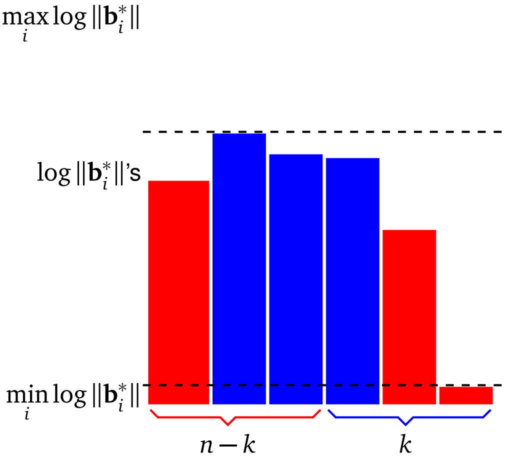


Figure: An example

Observations

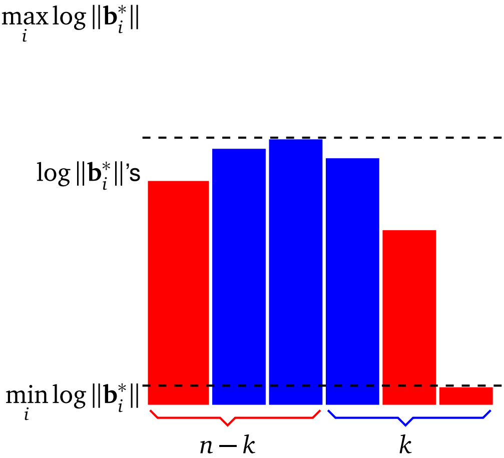


Figure: An example

Observations

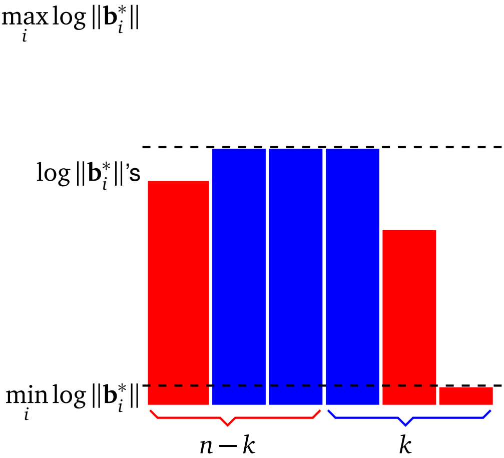


Figure: An example

Observations

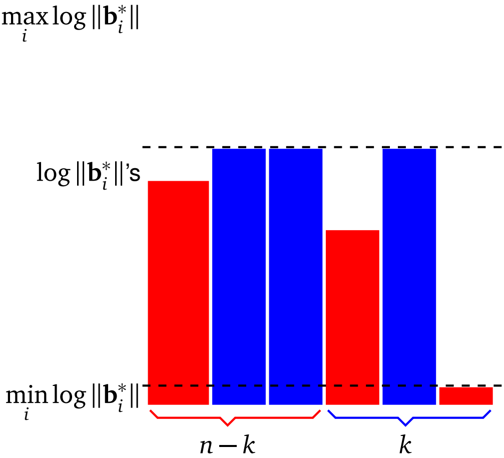


Figure: An example

Observations

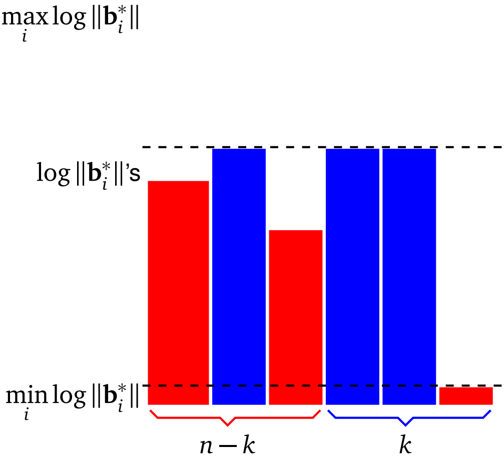


Figure: An example

Observations

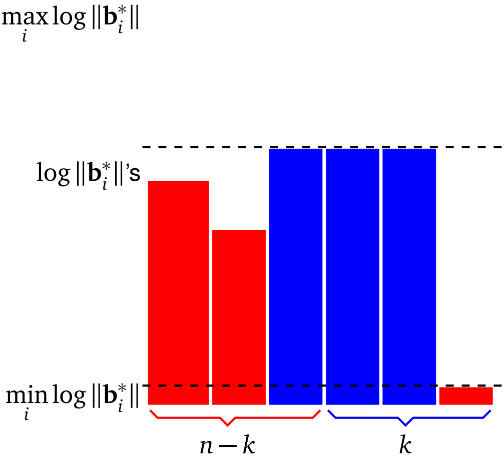


Figure: An example

Observations

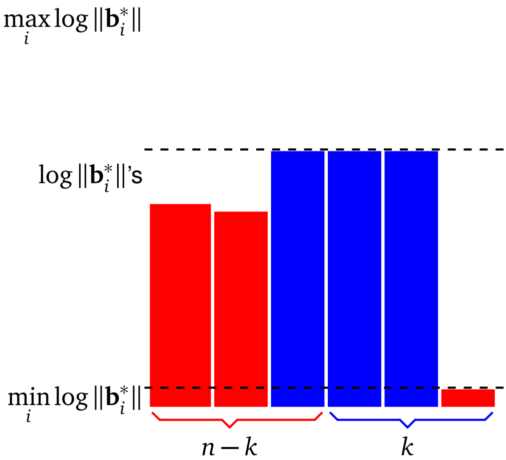


Figure: An example

Observations

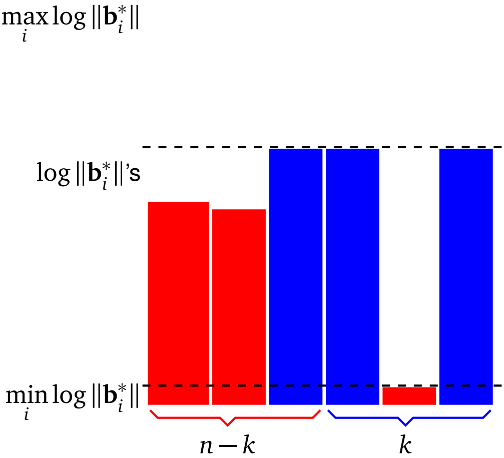


Figure: An example

Observations

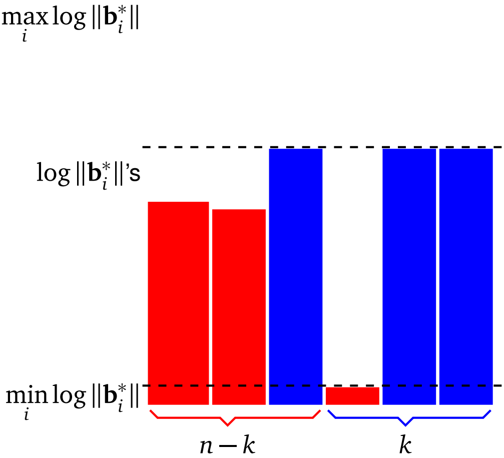


Figure: An example

Observations

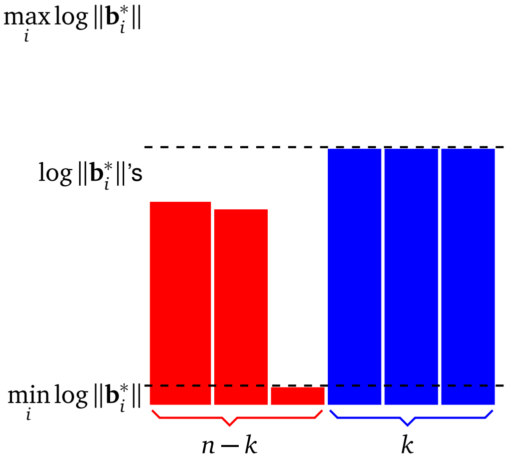


Figure: An example

Observations

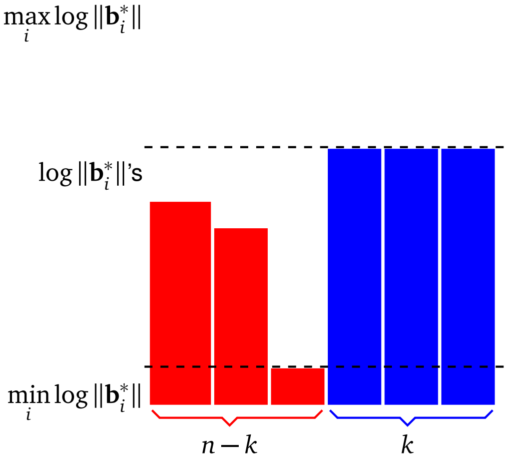


Figure: An example

Observations

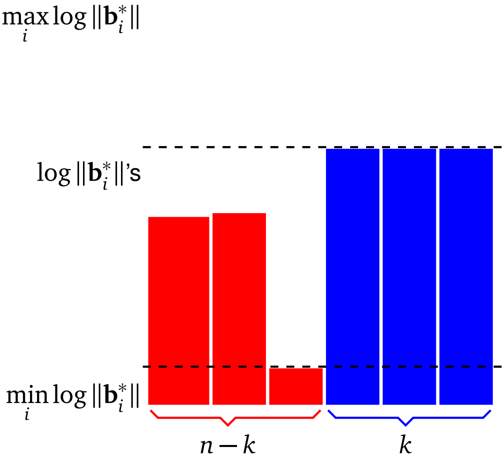


Figure: An example

Observations

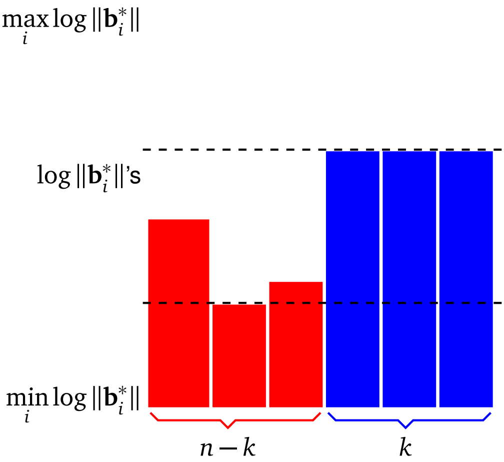


Figure: An example

Observations

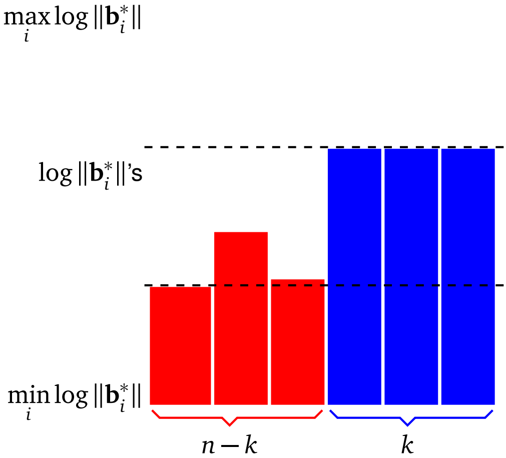


Figure: An example

Observations

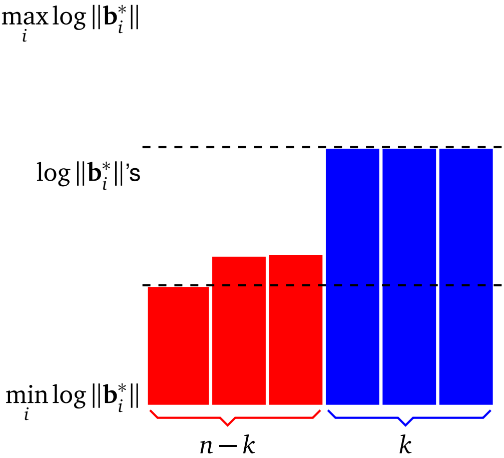


Figure: An example

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 $\longleftrightarrow$ small
  - large $\longleftrightarrow$ large
  - small $\longleftrightarrow$ large
- [van Hoeij & Novocin '10]: remove vectors with small GS length.

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

The new potential function

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

We define

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

The new potential function

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

The new potential function

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

We define

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

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

small $\longleftrightarrow$ small

large $\longleftrightarrow$ small

The new potential function

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

small $\longleftrightarrow$ small

large $\longleftrightarrow$ small

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

Properties of $\Pi_k(B)$

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 $\Pi_k(B)$

Monotonicity

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

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

Bounding #swaps

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

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

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

The main result

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

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

LLL swaps.

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

The main result

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

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

LLL swaps.

- The result is independent of K.

Comparison

Table: Upper bounds on #swaps for different k, $\alpha = \log \|\mathbf{A}\|$.

	Sufficient K	Heuristic K
$k = 1$	$\mathcal{O}(n^2 \log n + n\alpha)$	$\mathcal{O}(n^2 + n\alpha)$
$k = n/2$	$\mathcal{O}(n^3 \log n + n^3\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$
$k = n-1$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^2\alpha)$

Comparison

Table: Upper bounds on #swaps for different k, $\alpha = \log \|\mathbf{A}\|$.

	Sufficient K	Heuristic K	New analysis
$k = 1$	$\mathcal{O}(n^2 \log n + n\alpha)$	$\mathcal{O}(n^2 + n\alpha)$	$\mathcal{O}(n\alpha)$
$k = n/2$	$\mathcal{O}(n^3 \log n + n^3\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$
$k = n-1$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^3 + n\alpha)$

Comparison

Table: Upper bounds on #swaps for different k, $\alpha = \log \|\mathbf{A}\|$.

	Sufficient K	Heuristic K	New analysis
$k = 1$	$\mathcal{O}(n^2 \log n + n\alpha)$	$\mathcal{O}(n^2 + n\alpha)$	$\mathcal{O}(n\alpha)$
$k = n/2$	$\mathcal{O}(n^3 \log n + n^3\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$
$k = n-1$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^3 + n\alpha)$

- When $k = n-1$ and $\log \|\mathbf{A}\| = o(n)$,

Comparison

Table: Upper bounds on #swaps for different k, $\alpha = \log \|\mathbf{A}\|$.

	Sufficient K	Heuristic K	New analysis
$k = 1$	$\mathcal{O}(n^2 \log n + n\alpha)$	$\mathcal{O}(n^2 + n\alpha)$	$\mathcal{O}(n\alpha)$
$k = n/2$	$\mathcal{O}(n^3 \log n + n^3\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$	$\mathcal{O}(n^3 + n^2\alpha)$
$k = n - 1$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^2\alpha)$	$\mathcal{O}(n^3 + n\alpha)$

- When $k = n - 1$ and $\log \|\mathbf{A}\| = o(n)$,

  - LLL is not a good choice. E.g., one can use [Storjohann '05], ...

Future work

- Apply to more general bit complexity studies of LLL.
- Apply to more kinds of special lattice bases for LLL.
- Apply to design more efficient LLL-type algorithms.

Future work

- Apply to more general bit complexity studies of LLL.
- Apply to more kinds of special lattice bases for LLL.
- Apply to design more efficient LLL-type algorithms.

THANKS