Lossy Compression via Sparse Regression

Ramji Venkataramanan — University of Cambridge
Tuhin Sarkar — IIT Bombay
Sekhar Tatikonda — Yale University

ISIT 2013

GOAL : Efficient, rate-optimal codes for

- Lossy Compression

GOAL : Efficient, rate-optimal codes for

- Lossy Compression
- Point-to-point communication
- Multi-terminal models:
  - Compression with decoder side info (Wyner-Ziv)
  - Communication with encoder side info (Gelfand-Pinsker)
  - Multiple-access, broadcast, multiple descriptions, ...

Achieving the Shannon limits

Textbook Constructions

- Random codes for *point-to-point* source & channel coding
- Random Binning, Superposition
- High Coding and Storage Complexity
  - *exponential* in block length n

Achieving the Shannon limits

Textbook Constructions

- Random codes for *point-to-point* source & channel coding
- Random Binning, Superposition
- High Coding and Storage Complexity

  - *exponential* in block length n

- WANT: Compact representation + fast encoding/decoding

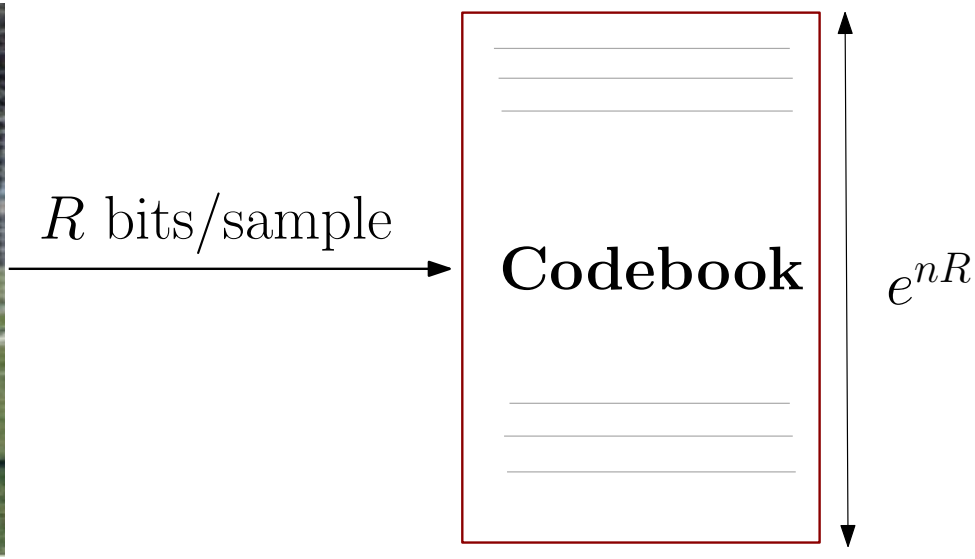
  'Fast' $\Rightarrow$ *polynomial* in n
- LDPC/LDGM codes, Polar codes

  - For finite-alphabet sources & channels

In this talk . . .

- Ensemble of codes based on sparse linear regression
- *Provably* achieve rates close to info-theoretic limits
  - with fast encoding + decoding
- Based on construction of Barron & Joseph for AWGN channel
  - Achieve capacity with fast decoding [ISIT '10, Arxiv '12]

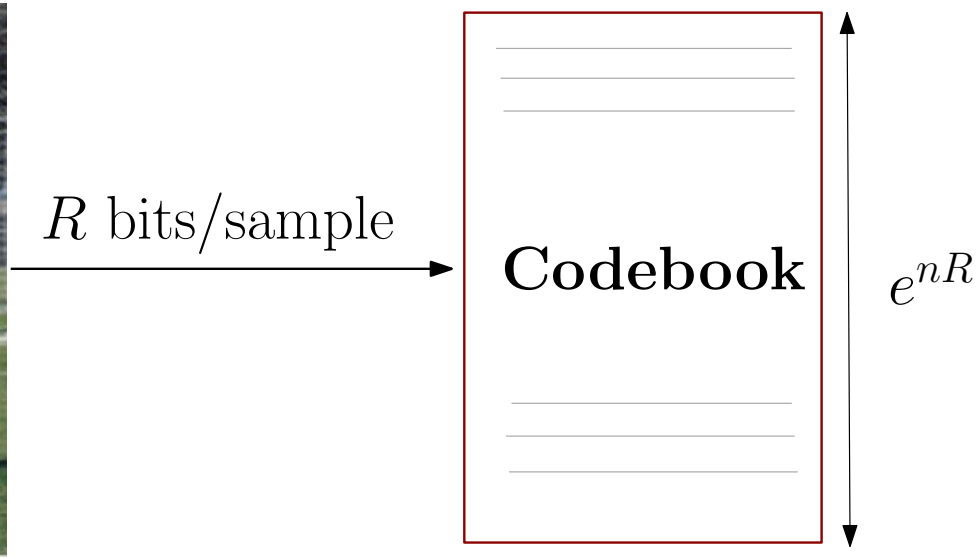
Source Coding



$$\mathbf{S} = S_1, \ldots, S_n \qquad \hat{\mathbf{S}} = \hat{S}_1, \ldots, \hat{S}_n$$

- Distortion criterion: $\frac{1}{n}\|\mathbf{S} - \hat{\mathbf{S}}\|^2 = \frac{1}{n}\sum_k (S_k - \hat{S}_k)^2$
- For i.i.d $\mathcal{N}(0, \sigma^2)$ source, min distortion $= \sigma^2 e^{-2R}$

Source Coding

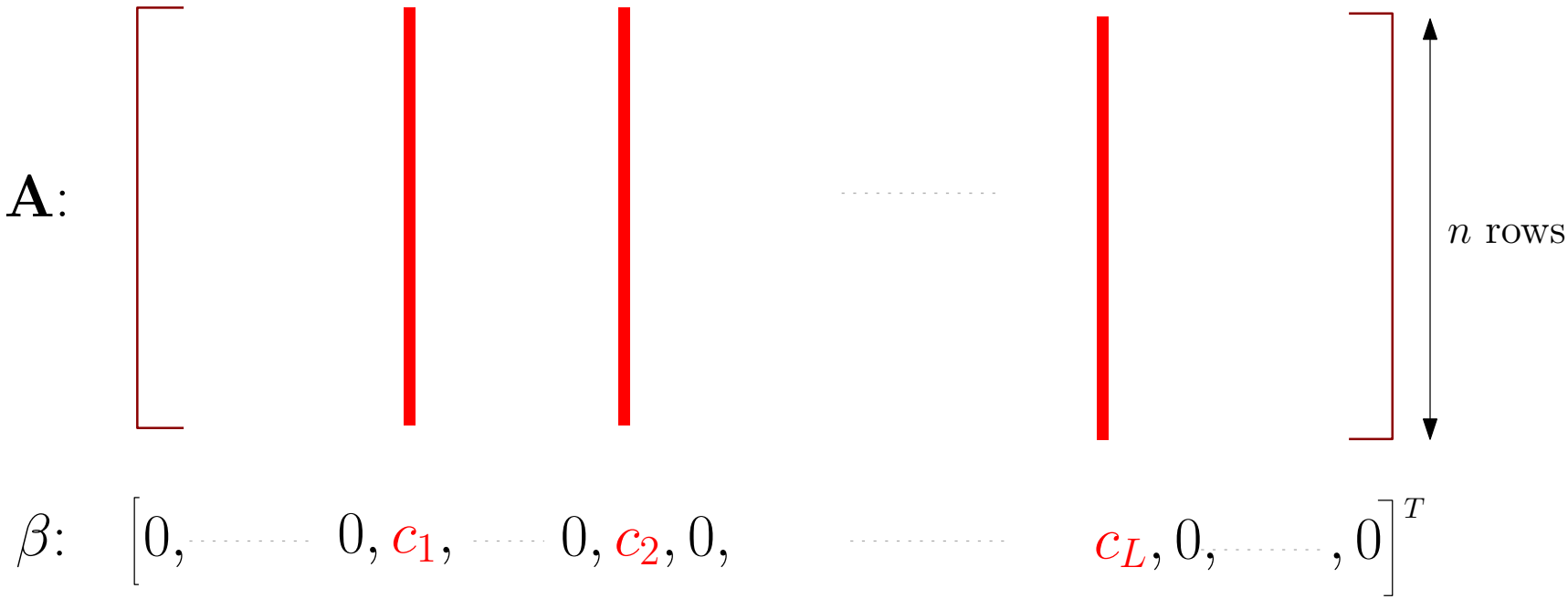


$$\mathbf{S} = S_1, \ldots, S_n \qquad \hat{\mathbf{S}} = \hat{S}_1, \ldots, \hat{S}_n$$

- Distortion criterion: $\frac{1}{n}\|\mathbf{S} - \hat{\mathbf{S}}\|^2 = \frac{1}{n}\sum_k (S_k - \hat{S}_k)^2$
- For i.i.d $\mathcal{N}(0, \sigma^2)$ source, min distortion $= \sigma^2 e^{-2R}$
- Can we achieve this with *low-complexity* algorithms?

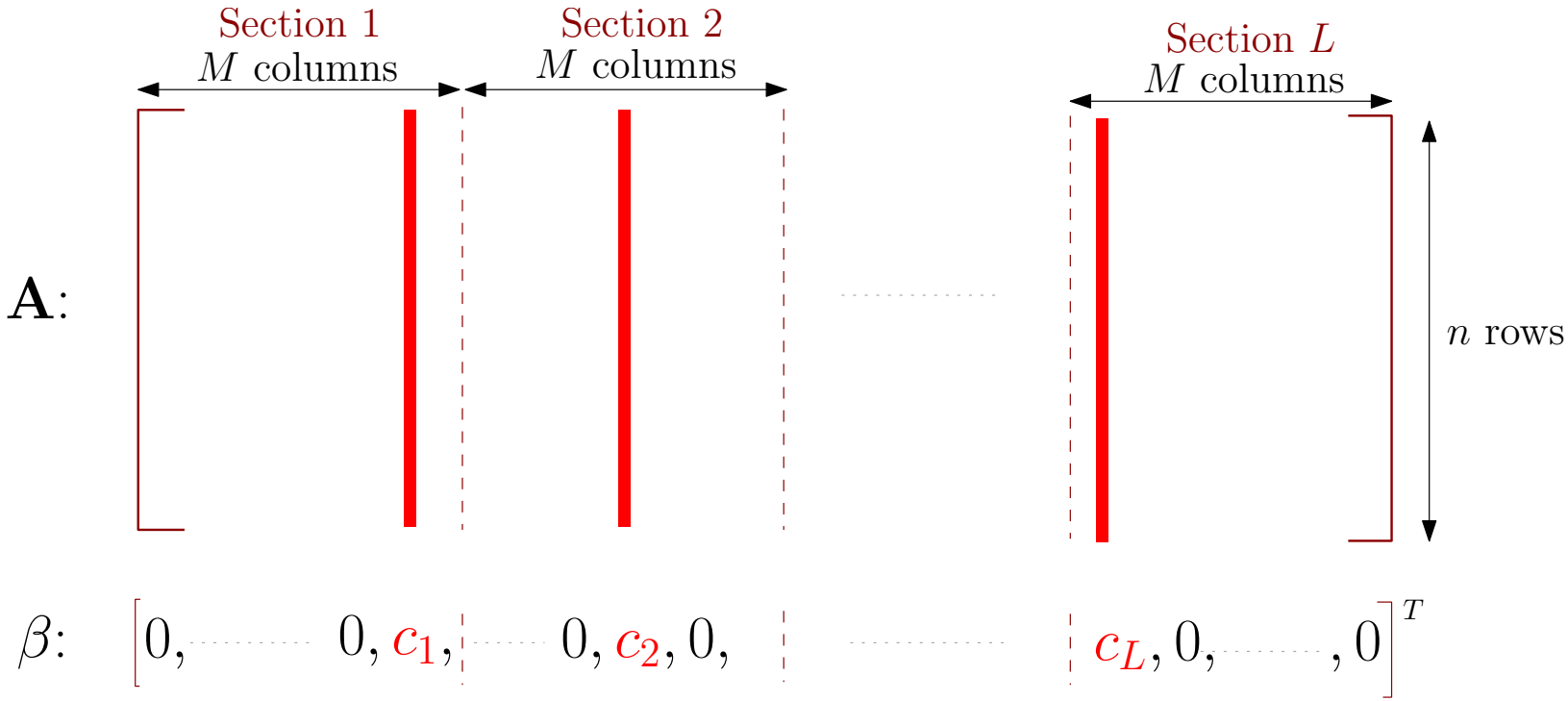
  - Computation & Storage

Sparse Regression Codes (SPARC)



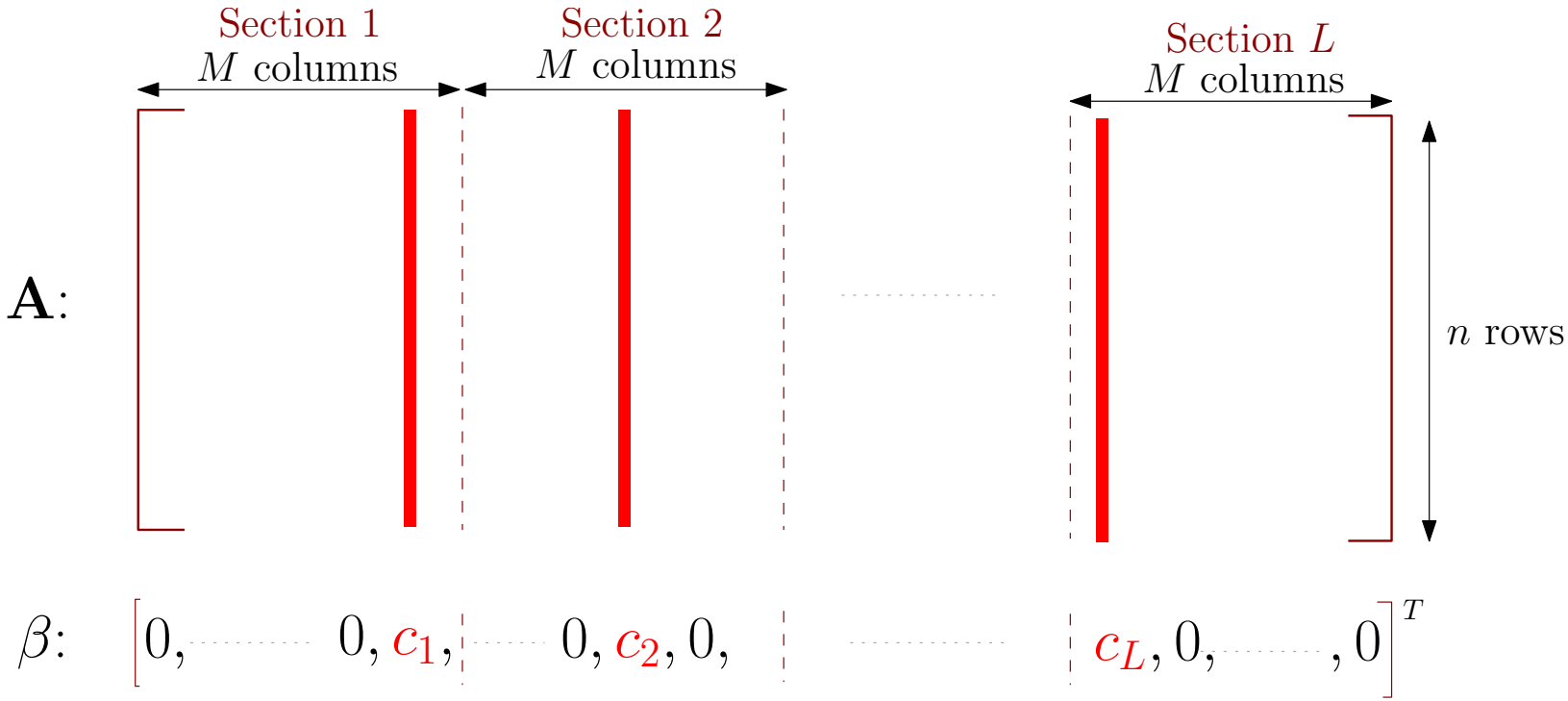
- **A**: design matrix or 'dictionary' with ind. $\mathcal{N}(0,1)$ entries
- Codewords $\mathbf{A}\beta$ - *sparse* linear combinations of columns of **A**

SPARC Construction



n rows, ML columns

SPARC Construction

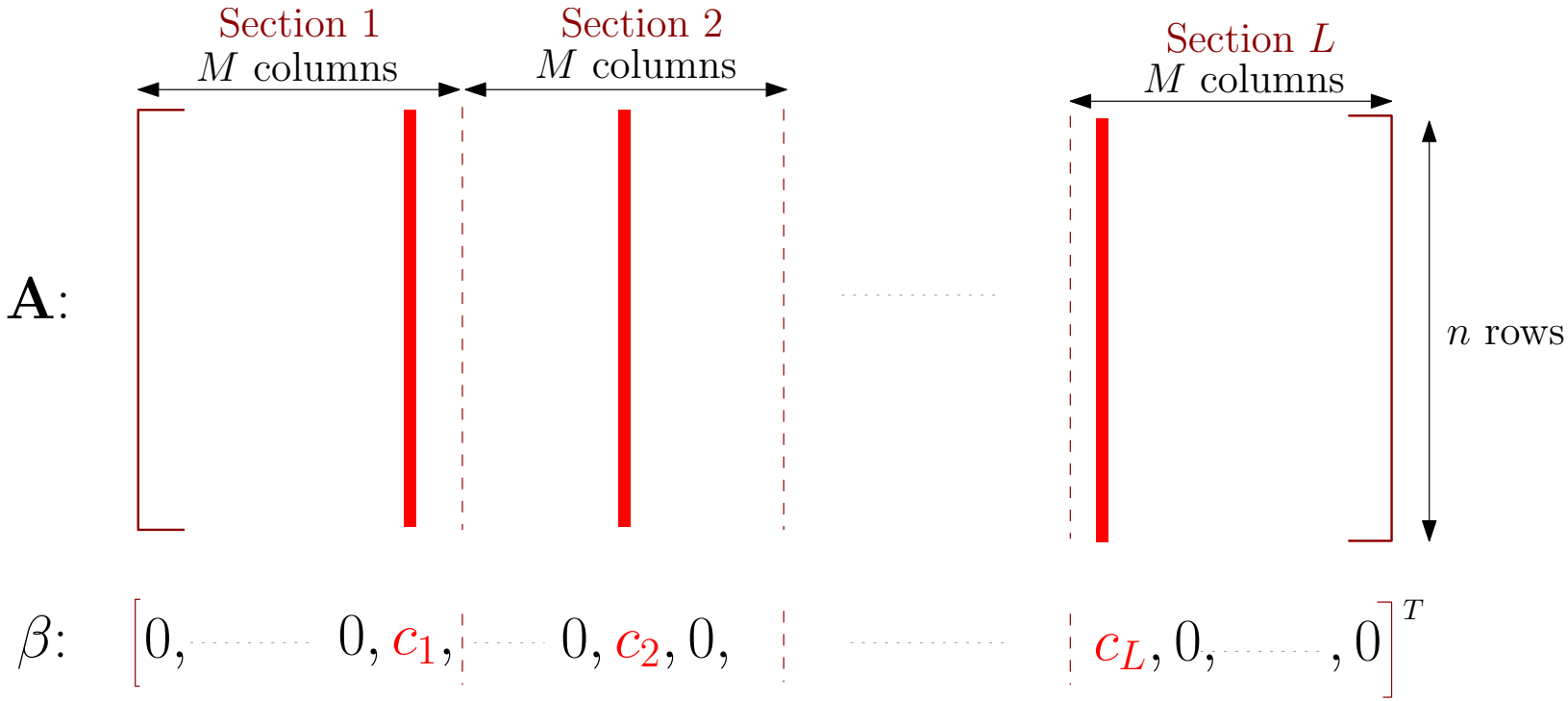


n rows, ML columns

Choosing M and L:

- For rate R codebook, need $M^L = e^{nR}$

SPARC Construction

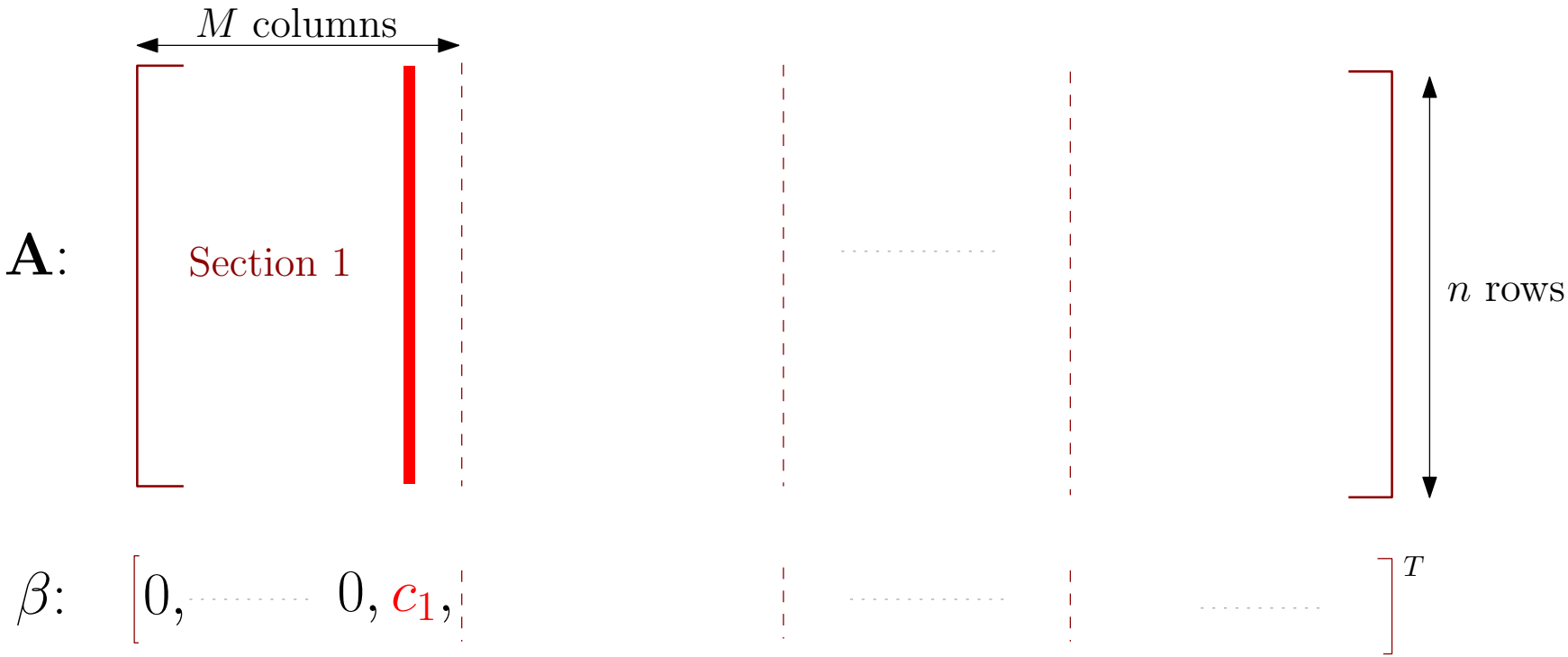


n rows, ML columns

Choosing M and L:

- For rate R codebook, need $M^L = e^{nR}$
- Choose M polynomial of $n \Rightarrow L \sim n/\log n$
- Storage Complexity $\leftrightarrow$ Size of $\mathbf{A}$: polynomial in n

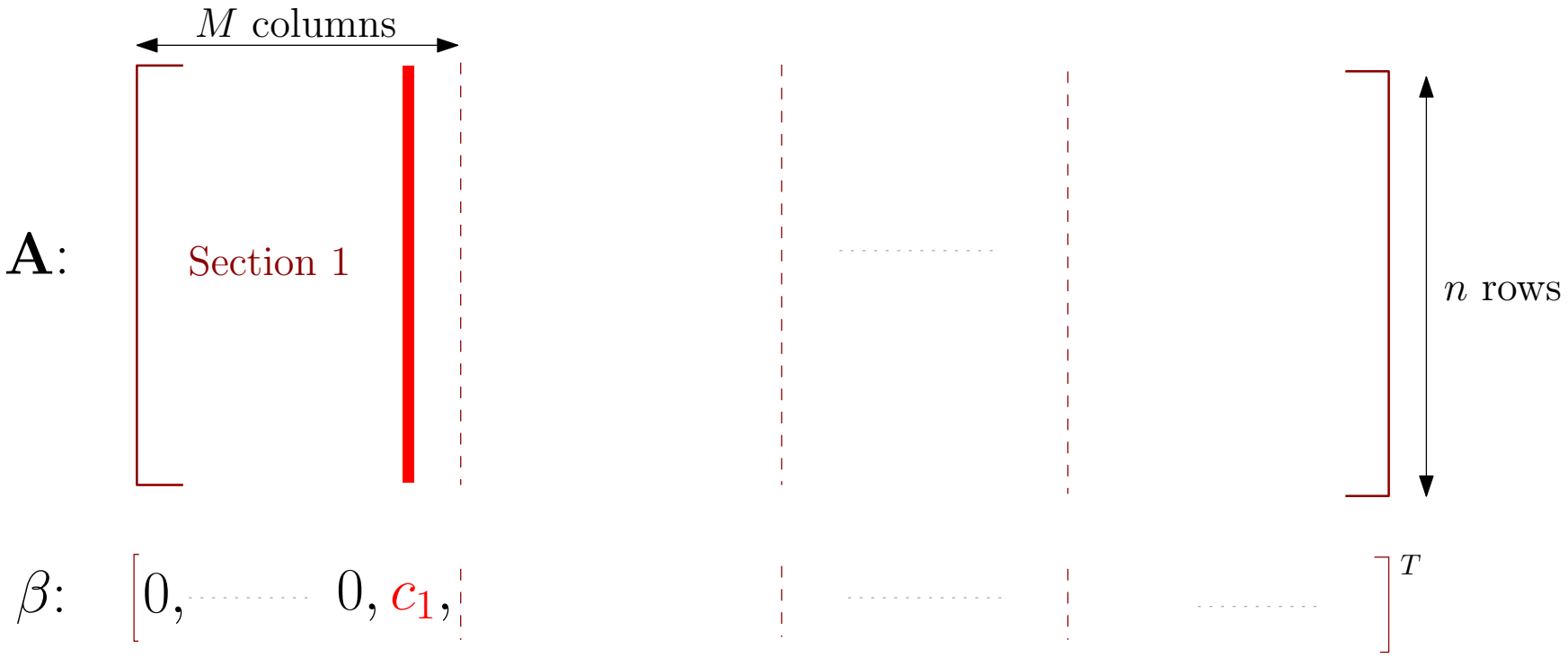
An Encoding Algorithm



Step 1: Choose column in Sec.1 that minimizes $\|\mathbf{S} - c_1\mathbf{A}_j\|^2$

- Max among inner products $\langle \mathbf{S}, \mathbf{A}_j \rangle$

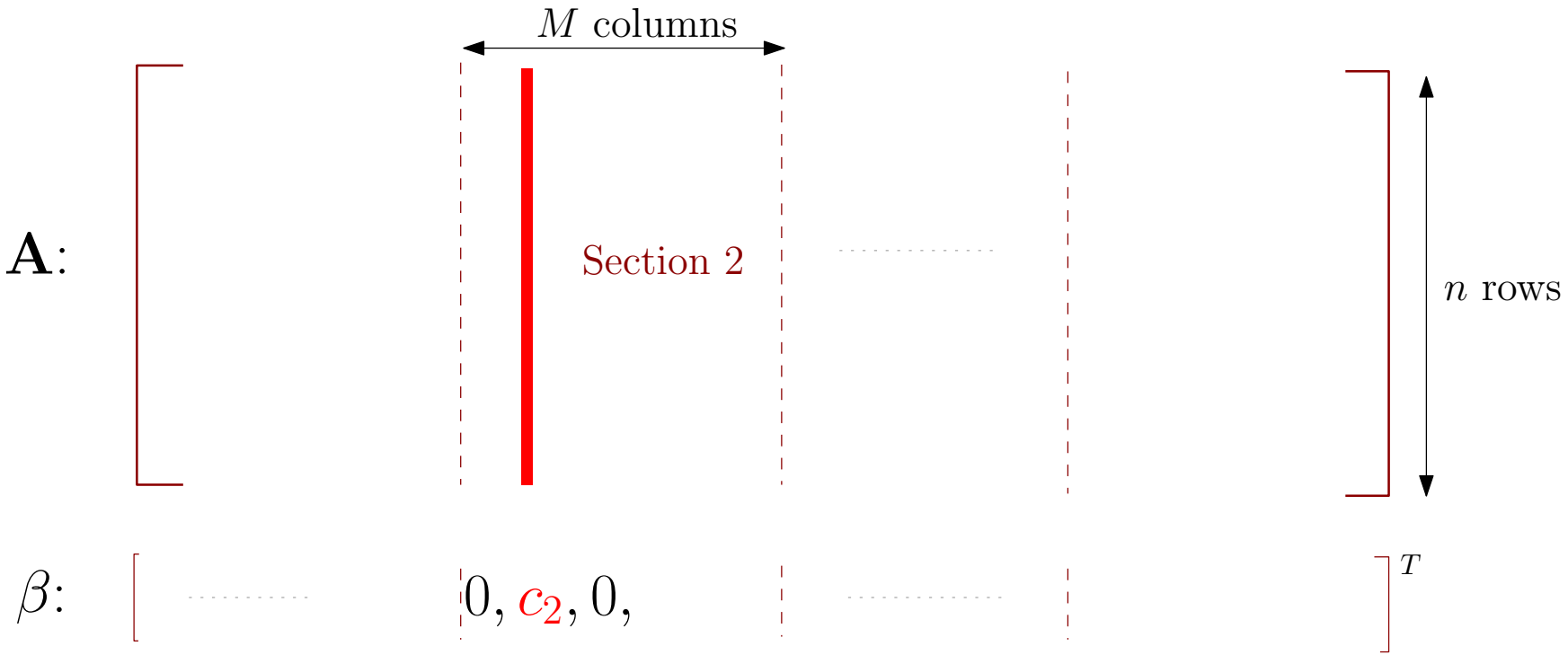
An Encoding Algorithm



Step 1: Choose column in Sec.1 that minimizes $\|\mathbf{S} - c_1\mathbf{A}_j\|^2$

- Max among inner products $\langle \mathbf{S}, \mathbf{A}_j \rangle$
- Residue $\mathbf{R}_1 = \mathbf{S} - c_1\hat{\mathbf{A}}_1$

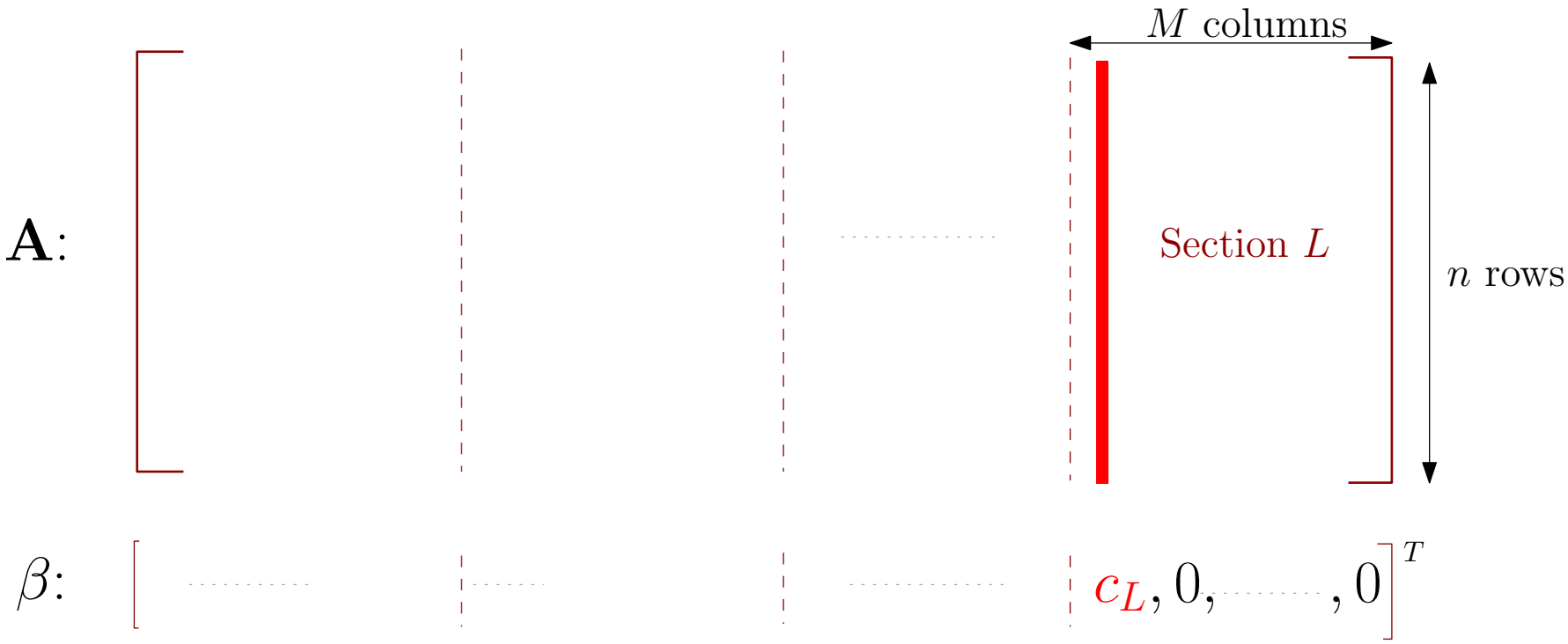
An Encoding Algorithm



Step 2: Choose column in Sec.2 that minimizes $\|\mathbf{R}_1 - c_2\mathbf{A}_j\|^2$

- Max among inner products $\langle \mathbf{R}_1, \mathbf{A}_j \rangle$
- Residue $\mathbf{R}_2 = \mathbf{R}_1 - c_2\hat{\mathbf{A}}_2$

An Encoding Algorithm



Step L: Choose column in Sec.L that minimizes $\|\mathbf{R}_{L-1} - c_L \mathbf{A}_j\|^2$

- Max among inner products $\langle \mathbf{R}_{L-1}, \mathbf{A}_j \rangle$
- Final residue $\mathbf{R}_L = \mathbf{R}_{L-1} - c_L \hat{\mathbf{A}}_L$

Performance

Theorem

For any ergodic source of variance σ^2,

$$P\left(\textit{Distortion} > \sigma^2 e^{-2R} + \Delta \right) \leq e^{-cL\Delta}$$

for

$$\Delta \geq \frac{1}{\log M}.$$

Encoding Complexity

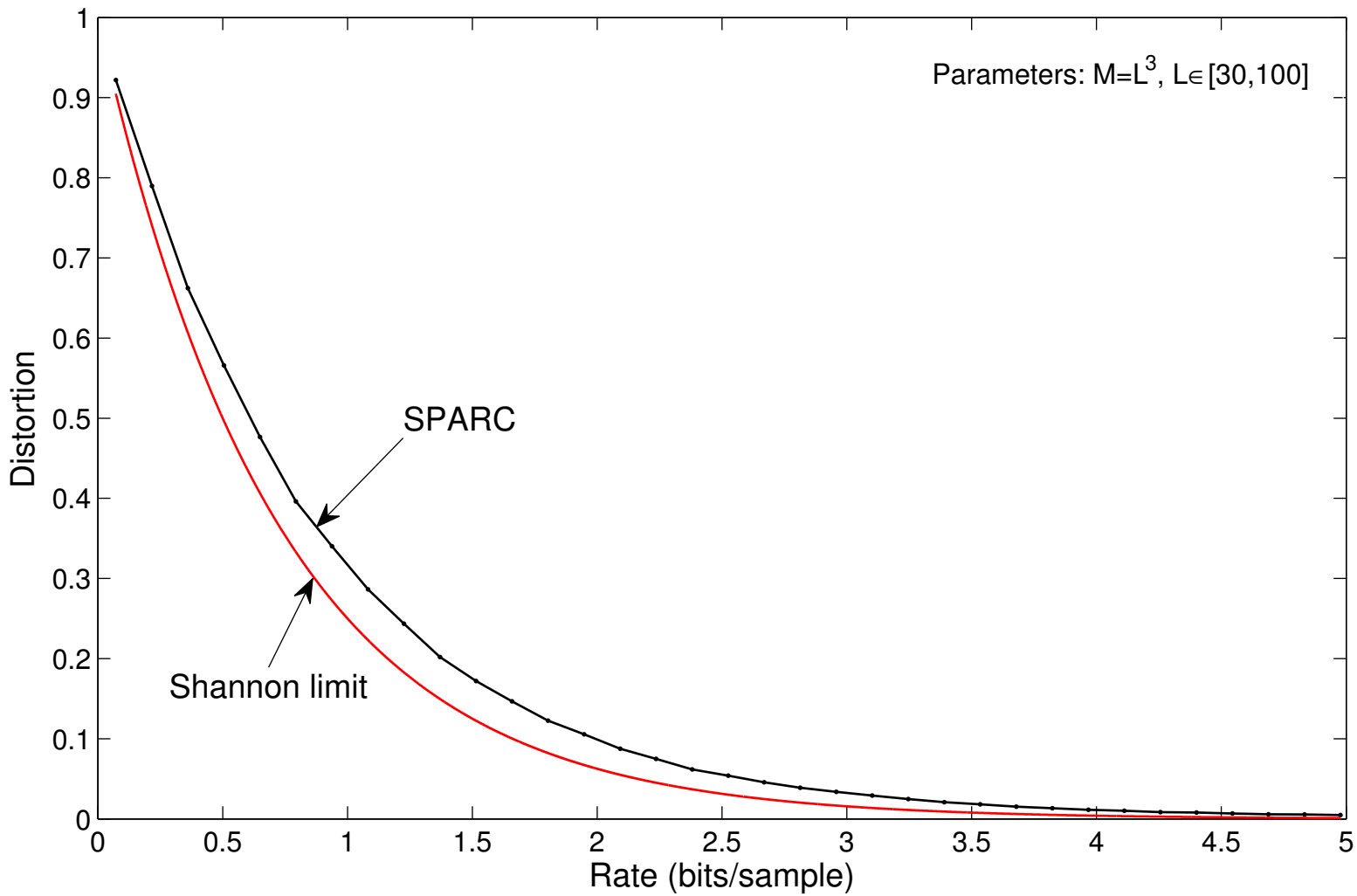
ML inner products and comparisons $\Rightarrow$ *polynomial* in n

Storage Complexity

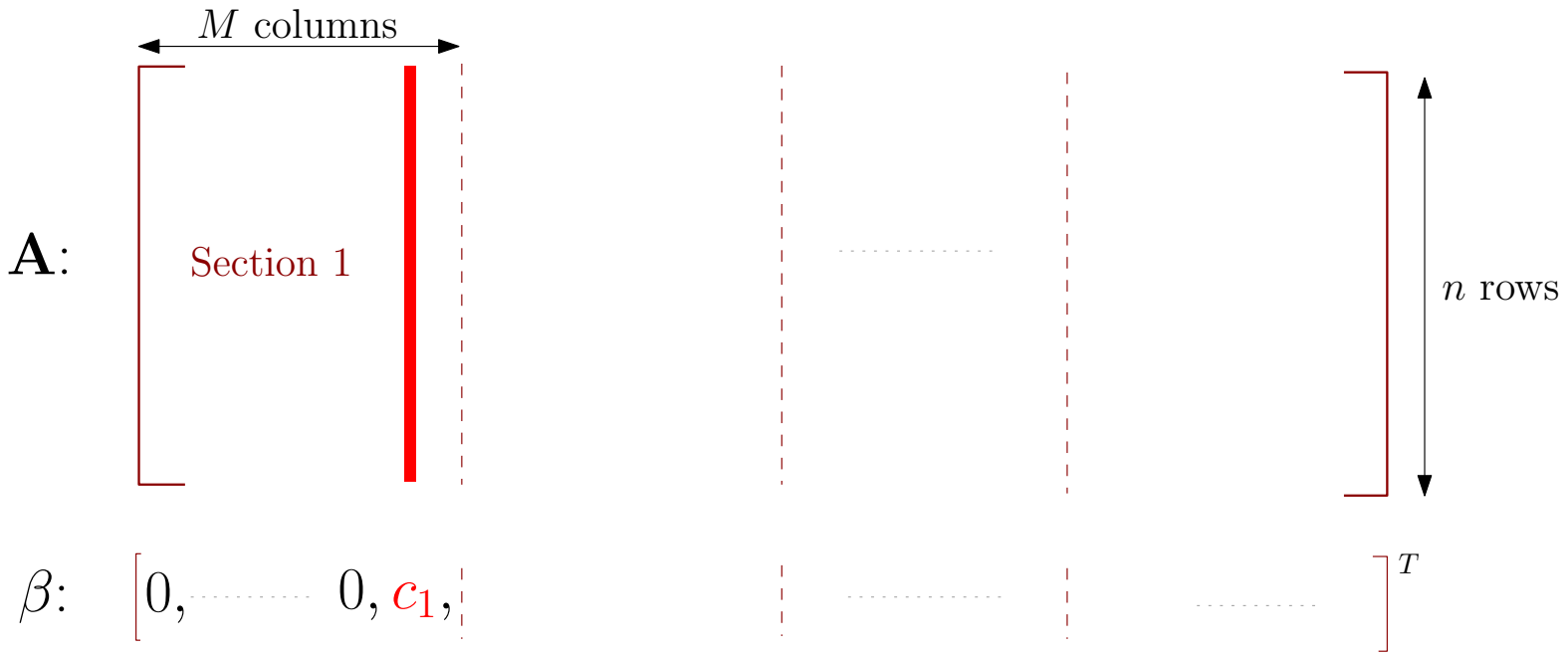
Design matrix $\mathbf{A}$: $n \times ML \Rightarrow$ *polynomial* in n

Simulation

Gaussian source: Mean 0, Variance 1



Why does the algorithm work?

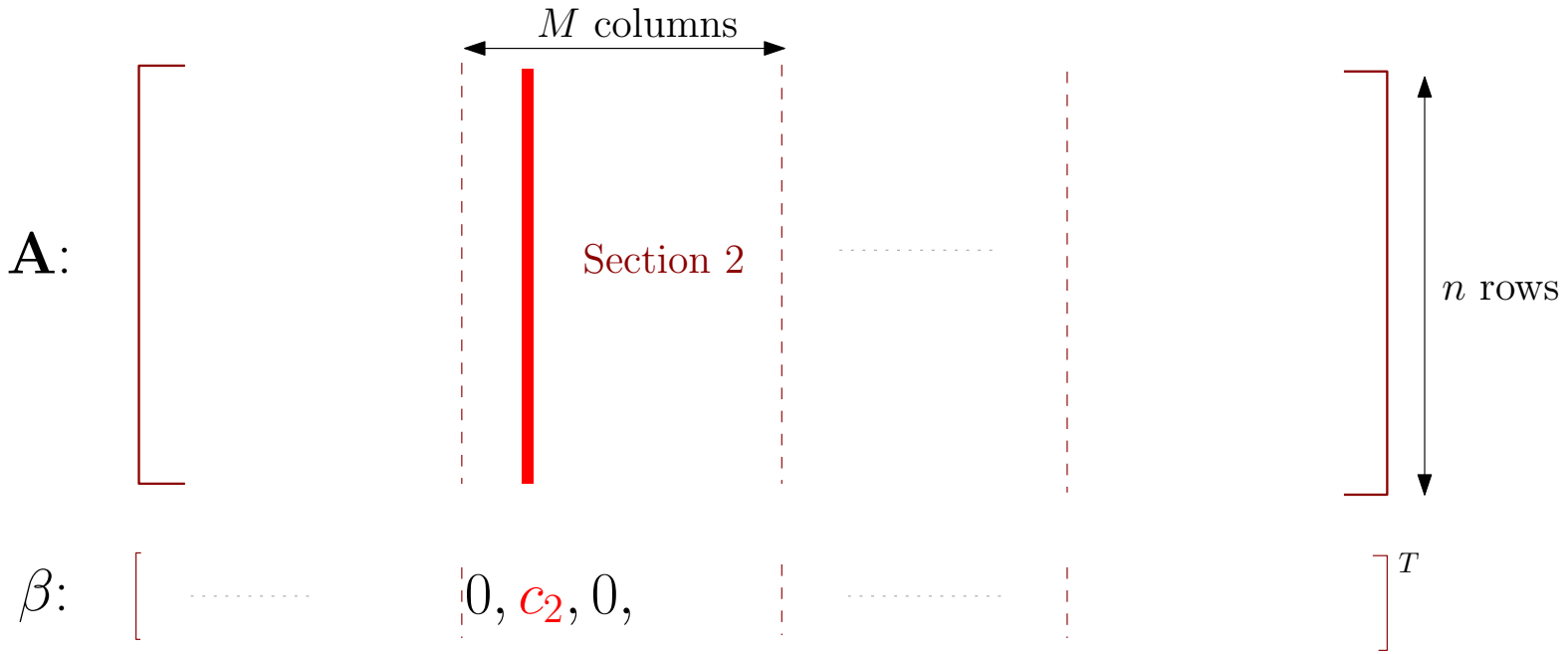


Each section is a code of rate R/L $\quad (L \sim \frac{n}{\log n})$

- Step 1: $\mathbf{S} \longrightarrow \mathbf{R}_1 = \mathbf{S} - c_1 \hat{\mathbf{A}}_1$

$$|\mathbf{R}_1|^2 \approx \sigma^2 \left(1 - \frac{2R}{L}\right) \quad \text{if} \quad c_1 = \sqrt{\frac{2R\sigma^2}{L}}$$

Why does the algorithm work?



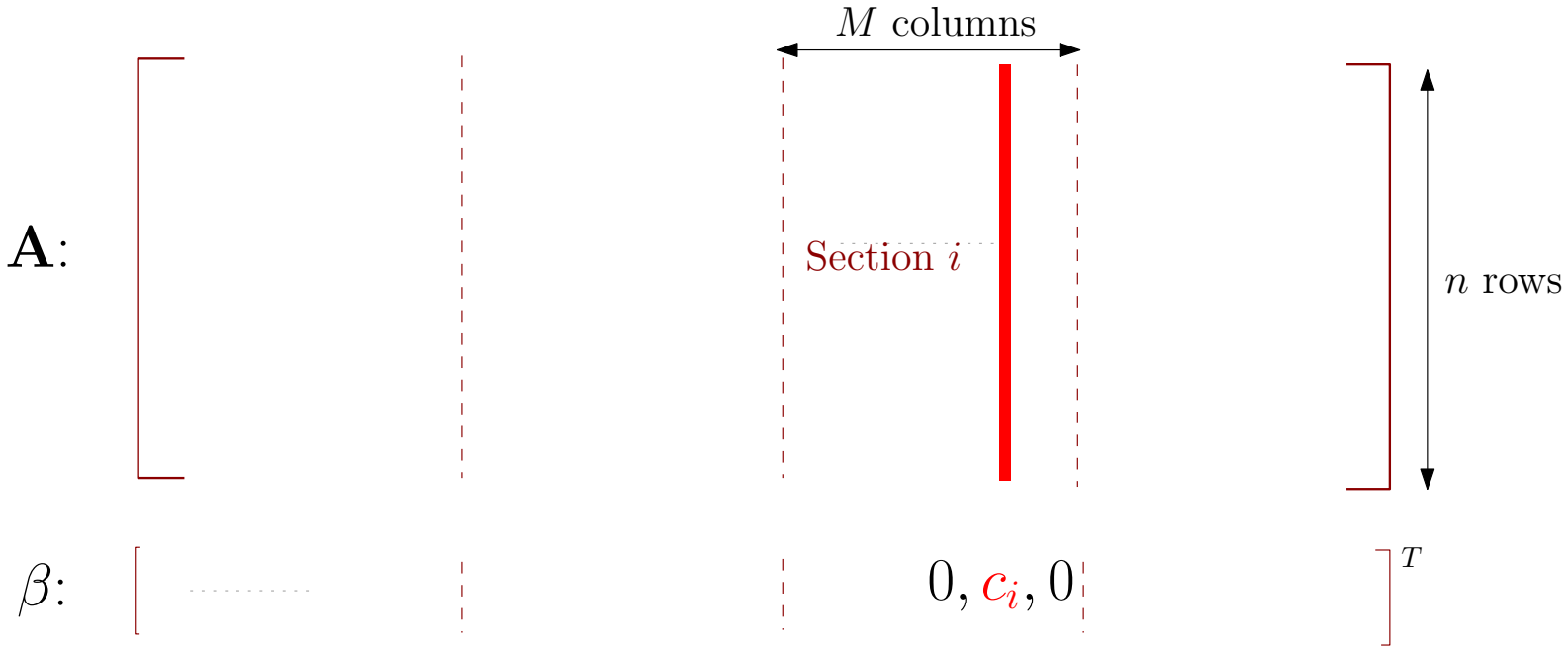
Each section is a code of rate R/L $\quad (L \sim \frac{n}{\log n})$

- Step 1: $\mathbf{S} \longrightarrow \mathbf{R}_1 = \mathbf{S} - c_1 \hat{\mathbf{A}}_1$

$$|\mathbf{R}_1|^2 \approx \sigma^2 \left(1 - \frac{2R}{L}\right) \quad \text{if} \quad c_1 = \sqrt{\frac{2R\sigma^2}{L}}$$

- Step 2: 'Source' $\mathbf{R}_1 \longrightarrow \mathbf{R}_2 = \mathbf{R}_1 - c_2 \hat{\mathbf{A}}_2$

Why does the algorithm work?

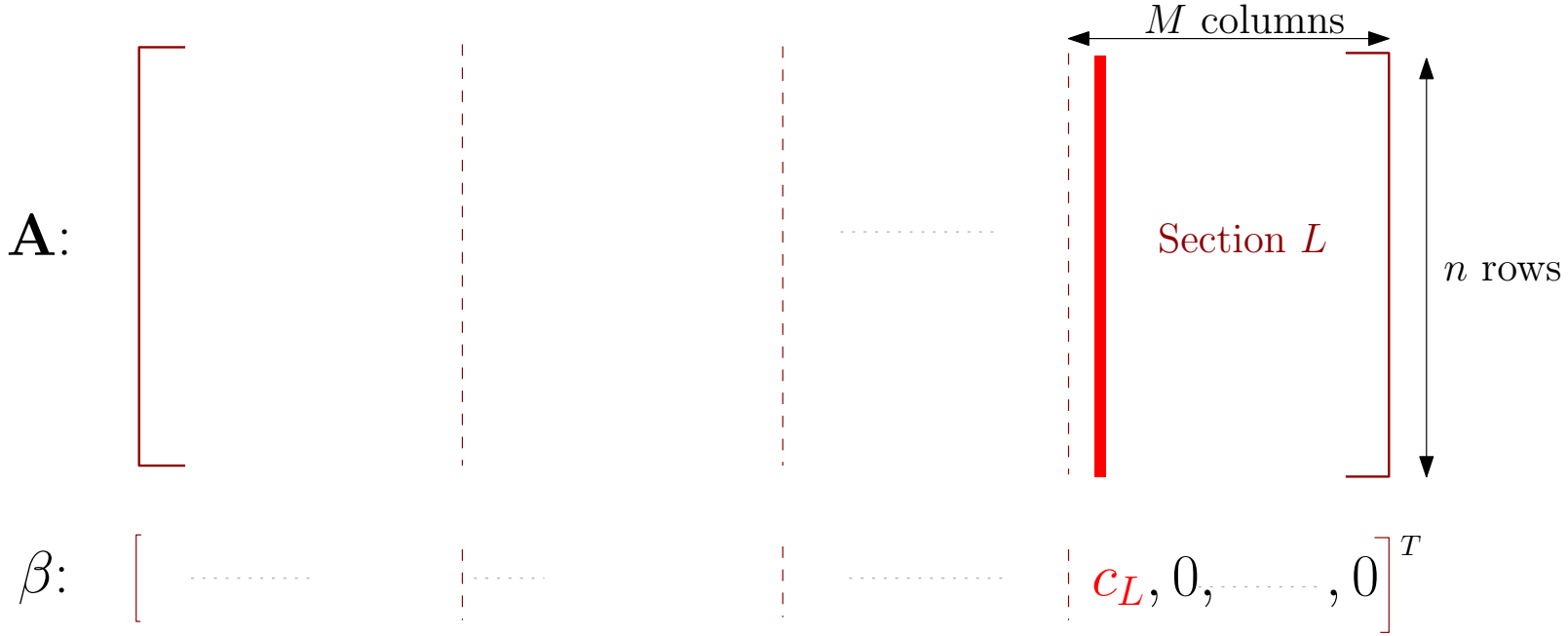


Each section is a code of rate R/L $\quad (L \sim \frac{n}{\log n})$

- Step i: 'Source' $\mathbf{R}_{i-1} \quad \longrightarrow \quad \mathbf{R}_i = \mathbf{R}_{i-1} - c_i \hat{\mathbf{A}}_2$

$$|\mathbf{R}_i|^2 \approx |\mathbf{R}_{i-1}|^2 \left(1 - \frac{2R}{L}\right) \approx \sigma^2 \left(1 - \frac{2R}{L}\right)^i$$

Why does the algorithm work?

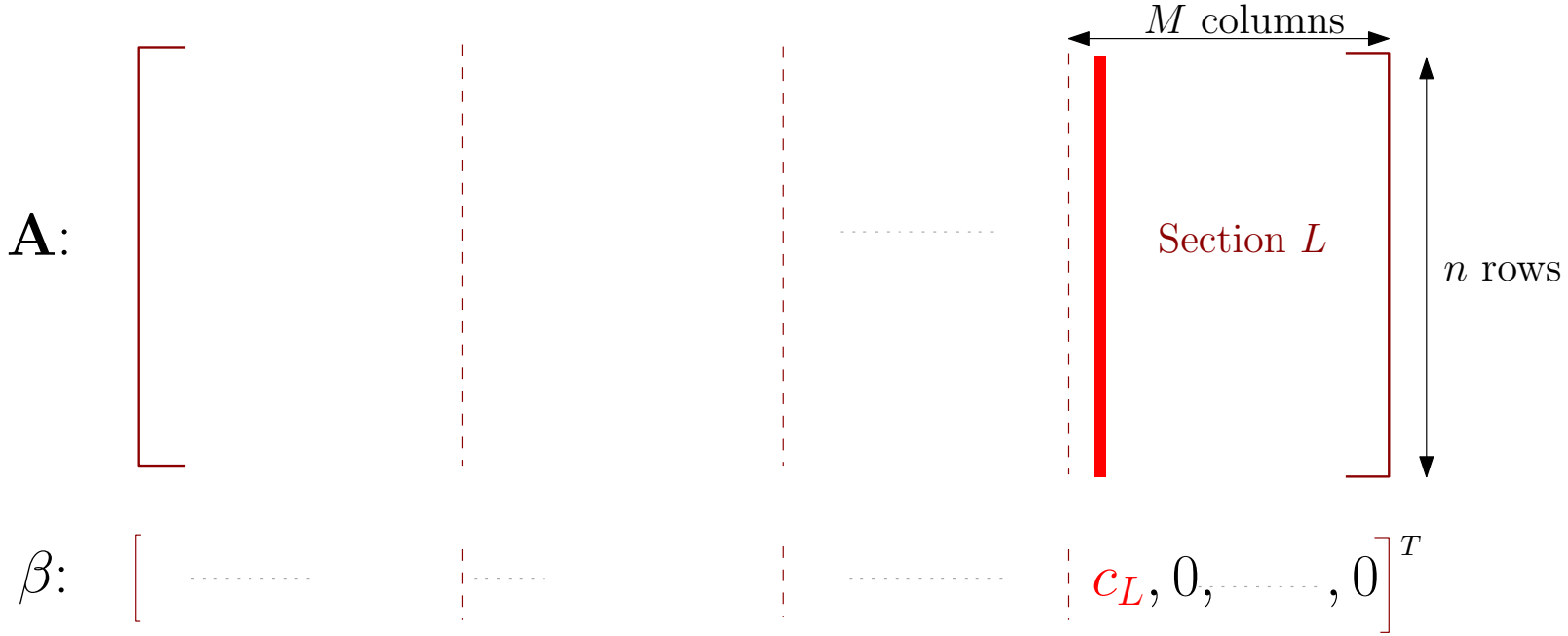


Each section is a code of rate R/L $\quad (L \sim \frac{n}{\log n})$

Final Distortion: $\quad |\mathbf{R}_L|^2 \approx \sigma^2 \left(1 - \frac{2R}{L}\right)^L \leq \sigma^2 e^{-2R}$

L-stage *successive refinement* $\quad L \sim n/\log n$

Why does the algorithm work?



Each section is a code of rate R/L $\quad (L \sim \frac{n}{\log n})$

Final Distortion: $\quad |\mathbf{R}_L|^2 \approx \sigma^2 \left(1 - \frac{2R}{L}\right)^L \leq \sigma^2 e^{-2R}$

The deviations in each step can be significant!

Proof Sketch

$$|\mathbf{R}_i|^2 = \underbrace{\sigma^2 \left(1 - \frac{2R}{L}\right)^i}_{\text{'Typical Value'}} (1 + \Delta_i)^2, \quad i = 0, \ldots, L$$

KEY: Controlling the final deviation Δ_L

Proof Sketch

$$|\mathbf{R}_i|^2 = \underbrace{\sigma^2 \left(1 - \frac{2R}{L}\right)^i}_{\text{'Typical Value'}} (1 + \Delta_i)^2, \quad i = 0, \ldots, L$$

KEY: Controlling the final deviation Δ_L

Deviation due to ...

- *Source*: $|\mathbf{S}|^2 = \sigma^2(1 + \Delta_0)^2$

Proof Sketch

$$|\mathbf{R}_i|^2 = \underbrace{\sigma^2 \left(1 - \frac{2R}{L}\right)^i}_{\text{'Typical Value'}} (1+\Delta_i)^2, \quad i = 0, \ldots, L$$

KEY: Controlling the final deviation Δ_L

Deviation due to ...

- *Source*: $|\mathbf{S}|^2 = \sigma^2(1+\Delta_0)^2$
- *Dictionary columns*: $|\mathbf{A}_j|^2 = 1+\gamma_j, \quad 1 \le j \le ML$

Proof Sketch

$$|\mathbf{R}_i|^2 = \underbrace{\sigma^2 \left(1 - \frac{2R}{L}\right)^i}_{\text{'Typical Value'}} (1+\Delta_i)^2, \quad i = 0, \ldots, L$$

KEY: Controlling the final deviation Δ_L

Deviation due to ...

- *Source*: $|\mathbf{S}|^2 = \sigma^2(1+\Delta_0)^2$
- *Dictionary columns*: $|\mathbf{A}_j|^2 = 1+\gamma_j, \quad 1 \le j \le ML$
- *Computed value*:

$$\max_j \left\langle \frac{\mathbf{R}_{i-1}}{\|\mathbf{R}_{i-1}\|}, \mathbf{A}_j \right\rangle = \sqrt{2\log M}\,(1+\epsilon_i), \quad 1 \le i \le L$$

Proof Sketch

$$|\mathbf{R}_i|^2 = \sigma^2 \left(1 - \frac{2R}{L}\right)^i (1 + \Delta_i)^2, \quad i = 0, \ldots, L$$

Recursion:

$$(1 + \Delta_i)^2 = (1 + \Delta_{i-1})^2 + \frac{2R/L}{1 - 2R/L}(\Delta_{i-1}^2 + \gamma_i - 2\epsilon_i(1 + \Delta_{i-1}))$$

Proof Sketch

$$|\mathbf{R}_i|^2 = \sigma^2 \left(1 - \frac{2R}{L}\right)^i (1 + \Delta_i)^2, \quad i = 0, \ldots, L$$

$$|\Delta_L| \leq c \left[|\Delta_0| + \frac{4R}{(1 - 2R/L)w} \left(\sum_{j=1}^{L} \frac{|\gamma_j|}{L} + \sum_{j=1}^{L} \frac{|\epsilon_j|}{L} \right) \right]$$

Proof Sketch

$$|\mathbf{R}_i|^2 = \sigma^2 \left(1 - \frac{2R}{L}\right)^i (1 + \Delta_i)^2, \quad i = 0, \ldots, L$$

$$|\Delta_L| \leq c \left[|\Delta_0| + \frac{4R}{(1 - 2R/L)w} \left(\sum_{j=1}^{L} \frac{|\gamma_j|}{L} + \sum_{j=1}^{L} \frac{|\epsilon_j|}{L} \right) \right]$$

Large deviations bounds

$$P(|\Delta_0| > \delta_0), \quad P\Big(\sum_{j=1}^{L} \frac{|\gamma_j|}{L} > \delta_1\Big), \quad P\Big(\sum_{i=1}^{L} \frac{|\epsilon_i|}{L} > \delta_2\Big)$$

Fall exponentially in L

Summary

Sparse Regression Codes

- Rate-optimal compression with polynomial-time encoding
- Successive refinement based encoder:
  - Prob. of excess distortion falls exponentially in L
  - But $\Delta \sim \frac{1}{\log M}$

Theorem [RV-Joseph-Tatikonda, ISIT '12]

With ML encoding, SPARCs attain the rate-distortion function with the optimal error-exponent for all $D < D_0$.

Future Directions

- Improved coding algorithms
  - Adaptive successive decoding, ℓ_1 minimization etc.?

Future Directions

- Improved coding algorithms
  - Adaptive successive decoding, ℓ_1 minimization etc.?
- Nice structure that enables
  - Binning (Wyner-Ziv, Gelfand-Pinsker) [Allerton '12]
  - Superposition (Multiple-access, Broadcast)
  - Interference channels, Multiple descriptions . . . ?

Future Directions

- Improved coding algorithms
  - Adaptive successive decoding, ℓ_1 minimization etc.?
- Nice structure that enables
  - Binning (Wyner-Ziv, Gelfand-Pinsker) [Allerton '12]
  - Superposition (Multiple-access, Broadcast)
  - Interference channels, Multiple descriptions . . . ?
- General design matrices

Future Directions

- Improved coding algorithms
  - Adaptive successive decoding, ℓ_1 minimization etc.?
- Nice structure that enables
  - Binning (Wyner-Ziv, Gelfand-Pinsker) [Allerton '12]
  - Superposition (Multiple-access, Broadcast)
  - Interference channels, Multiple descriptions . . . ?
- General design matrices
- Finite-field analogs - binary SPARCs