Sparse Regression Codes

Andrew Barron  Ramji Venkataramanan
Yale University  University of Cambridge

Joint work with Antony Joseph, Sanghee Cho, Cynthia Rush, Adam Greig, Tuhin Sarkar, Sekhar Tatikonda

ISIT 2016
Part III of the tutorial:

- SPARCs for Lossy Compression
- SPARCs for Multi-terminal Source and Channel Coding
- Open questions

(Joint work with Sekhar Tatikonda, Tuhin Sarkar, Adam Greig)
Lossy Compression

\[ S = S_1, \ldots, S_n \]

\[ \hat{S} = \hat{S}_1, \ldots, \hat{S}_n \]

- Distortion criterion:
  \[ \frac{1}{n} \| S - \hat{S} \|^2 = \frac{1}{n} \sum_k (S_k - \hat{S}_k)^2 \]

- For i.i.d. \( \mathcal{N}(0, \nu^2) \) source, min distortion = \( \nu^2 e^{-2R} \)
Lossy Compression

\[ S = S_1, \ldots, S_n \]

\[ \hat{S} = \hat{S}_1, \ldots, \hat{S}_n \]

- Distortion criterion: \( \frac{1}{n} \| S - \hat{S} \|^2 = \frac{1}{n} \sum_k (S_k - \hat{S}_k)^2 \)
- For i.i.d \( \mathcal{N}(0, \nu^2) \) source, min distortion = \( \nu^2 e^{-2R} \)
- Can we achieve this with low-complexity codes?
  - Storage & Computation
SPARC Construction

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

Choose $M$ polynomial of $N = \log n$

Storage Complexity

Size of $A$: polynomial in $n$

$n$ rows, $ML$ columns, $A_{ij} \sim \mathcal{N}(0, 1/n)$
**SPARC Construction**

![Diagram showing A, M columns, n rows, M^L columns, \( A_{ij} \sim \mathcal{N}(0, 1/n) \)]

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 \( A \): polynomial in \( n \)
Optimal Encoding

\[ A: \begin{bmatrix} \beta: 0, \cdots, 0, c_1, \cdots, 0, c_2, 0, \cdots, c_L, 0, \cdots, 0 \end{bmatrix}^T \]

Minimum Distance Encoding: \( \hat{\beta} = \arg \min_{\beta \in \text{SPARC}} \| S - A\beta^2 \| \)

Theorem [Venkataramanan, Tatikonda ’12, ’14]:
For source \( S \) i.i.d. \( \sim \mathcal{N}(0, \nu^2) \), the sequence of rate \( R \) SPARCs with \( n, L, M = L^b \) with \( b > b^*(R) \):

\[ P \left( \frac{1}{n} \| S - A\hat{\beta} \|^2 > D \right) < e^{-n \left( E^*(R,D) + o(1) \right)}. \]

Achieves the optimal rate-distortion function with the optimal error exponent \( E^*(R, D) \).
Successive Cancellation Encoding

\[ A: \begin{bmatrix} \text{Section 1} \\ \vdots \\ \beta: \begin{bmatrix} 0, & 0, & c_1, & \vdots \end{bmatrix}^T \end{bmatrix} \]

Step 1: Choose column in section 1 that minimizes \( \| S - c_1 A_j \|^2 \)

- Max among \( M \) inner products \( \langle S, A_j \rangle \)
**Successive Cancellation Encoding**

\[ \beta: 0, 0, c_1, \ldots \]

\( A: \) 

\[ \begin{bmatrix} \text{Section 1} \\ \vdots \end{bmatrix} \]

\[ M \text{ columns} \]

\[ n \text{ rows} \]

---

**Step 1:** Choose column in section 1 that minimizes \( \| S - c_1 A_j \| ^2 \)

- Max among \( M \) inner products \( \langle S, A_j \rangle \)
- \( c_1 = \sqrt{2\nu^2 \log M} \)
- residual \( R_1 = S - c_1 \hat{A}_1 \)
Successive Cancellation Encoding

\[ \beta: 0, c_2, 0, \ldots \]

\[ A: \begin{bmatrix} \vdots & \vdots & \end{bmatrix} \]

\[ M \text{ columns} \]

\[ n \text{ rows} \]

**Step 2:** Choose column in section 2 that minimizes \( \| R_1 - c_2 A_j \|^2 \)

- Max among inner products \( \langle R_1, A_j \rangle \)

- \( c_2 = \sqrt{2 (\log M) \nu^2 \left( 1 - \frac{2R}{L} \right)} \)

- Residual \( R_2 = R_1 - c_2 \hat{A}_2 \)
Successive Cancellation Encoding

\[ A: \]

\[ \beta: \]

\[ c_L, 0, \ldots, 0 \]

\[ M \text{ columns} \]

\[ n \text{ rows} \]

Step L: Choose column in section \( L \) that minimizes \( \| R_{L-1} - c_L A_j \|_2^2 \)

- \( c_L = \sqrt{2 (\log M) \nu^2 \left( 1 - \frac{2R}{L} \right)^L} \)

- Final residual \( R_L = R_{L-1} - c_L \hat{A}_L \)

Final Distortion = \( \frac{1}{n} \| R_L \|_2^2 \)
Performance

Theorem [Venkataramanan, Sarkar, Tatikonda '13]:
For an ergodic source $S$ with mean 0 and variance $\nu^2$, the encoding algorithm produces a codeword $A\hat{\beta}$ that satisfies the following for sufficiently large $M, L$:

$$P \left( \frac{1}{n} \| S - A\hat{\beta} \|^2 > \nu^2 e^{-2R} + \Delta \right) < e^{-\kappa n \left( \Delta - \frac{c \log \log M}{\log M} \right)}$$

Deviation between actual distortion and the optimal value is $O\left( \frac{\log \log n}{\log n} \right)$
Performance

Theorem [Venkataramanan, Sarkar, Tatikonda '13]:
For an ergodic source $S$ with mean 0 and variance $\nu^2$, the encoding algorithm produces a codeword $A\hat{\beta}$ that satisfies the following for sufficiently large $M, L$:

$$P \left( \frac{1}{n} \| S - A\hat{\beta} \|^2 > \nu^2 e^{-2R} + \Delta \right) < e^{-\kappa n \left( \Delta - \frac{c \log \log M}{\log M} \right)}$$

Deviation between actual distortion and the optimal value is $O\left( \frac{\log \log n}{\log n} \right)$

*Encoding Complexity:*
$ML$ inner products and comparisons $\Rightarrow$ *polynomial* in $n$
Numerical Experiment

Gaussian source: Mean 0, Variance 1

Parameters: \( M = L^3 \), \( L \in [30,100] \)
Why does the algorithm work?

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

- Step 1: \( S \rightarrow R_1 = S - c_1 \hat{A}_1 \)

\[
|R_1|^2 \approx \nu^2 e^{-2R/L} \approx \nu^2 \left( 1 - \frac{2R}{L} \right) \quad \text{for } c_1 = \sqrt{2\nu^2 \log M}
\]
Why does the algorithm work?

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

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

$$|\mathbf{R}_1|^2 \approx \nu^2 e^{-2R/L} \approx \nu^2 \left( 1 - \frac{2R}{L} \right) \quad \text{for} \quad c_1 = \sqrt{2\nu^2 \log M}$$

- Step 2: ‘Source’ $\mathbf{R}_1 \rightarrow \mathbf{R}_2 = \mathbf{R}_1 - c_2 \hat{A}_2$
Why does the algorithm work?

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

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

With $c_i^2 = \frac{2R\nu^2}{L}(1 - \frac{2R}{L})^{i-1}$,

$$|\mathbf{R}_i|^2 \approx |\mathbf{R}_{i-1}|^2 \left(1 - \frac{2R}{L}\right) \approx \nu^2 \left(1 - \frac{2R}{L}\right)^i$$
Why does the algorithm work?

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

Final Distortion: $|R_L|^2 \approx \nu^2 \left(1 - \frac{2R}{L}\right)^L \leq \nu^2 e^{-2R}$

$L$-stage successive refinement $\ L \sim n/\log n$
**Successive Refinement Interpretation**

- The encoder successively refines the source over $\sim \frac{n}{\log n}$ stages.
- The deviations in each stage can be significant!

\[ |R_i|^2 = \nu^2 \left( 1 - \frac{2R}{L} \right)^i (1 + \Delta_i)^2, \quad i = 0, \ldots, L \]

\[ \text{‘Typical Value’} \]

- **KEY** to result: Controlling the final deviation $\Delta_L$
- Recall: successive cancellation *does not* work for SPARC AWGN decoding
Open Questions in SPARC Compression

- Better encoders with smaller gap to $D^*(R)$? Iterative soft-decision encoding, AMP?

- AWGN decoding AMP doesn’t work when directly used for compression:
  - may need decimation a la LDGM codes for compression

- But recall: With min-distance encoding, SPARCs attain the rate-distortion function with the *optimal* error-exponent

- Compression performance with $\pm 1$ dictionaries

- Compression of finite alphabet sources
Sparse Regression Codes for multi-terminal networks
Codes for multi-terminal problems

Key ingredients:

- **Superposition** (Multiple-access channel, Broadcast channel)
- **Random binning** (e.g., Distributed compression, Channel Coding with Side-Information, ...)

SPARC is based on superposition coding!
Multiple-Access Channel

\[ \frac{\|X_1\|^2}{n} \leq P_1, \quad \frac{\|X_2\|^2}{n} \leq P_2, \quad \text{Noise} \sim \mathcal{N}(0, \sigma^2) \]

Corner points of capacity region are given by

\[ R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{P_2 + \sigma^2} \right), \quad R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right) \]

and

\[ R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{\sigma^2} \right), \quad R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{P_1 + \sigma^2} \right) \]
Successive Decoding

\[ Y = X_1 + X_2 + \text{Noise} \]

The rate-pair

\[ R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{P_2 + \sigma^2} \right), \quad R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right) \]

can be achieved by point-to-point codes:

- \( X_1 \) is decoded from \( Y \) treating \( X_2 \) as noise
- Subtract off \( X_1 \), then decode \( X_2 \) with snr \( P_2/\sigma^2 \)
Successive Decoding

\[ Y = X_1 + X_2 + \text{Noise} \]

The rate-pair

\[
R_1 = \frac{1}{2} \log \left( 1 + \frac{P_1}{P_2 + \sigma^2} \right), \quad R_2 = \frac{1}{2} \log \left( 1 + \frac{P_2}{\sigma^2} \right)
\]

can be achieved by point-to-point codes:

- \( X_1 \) is decoded from \( Y \) treating \( X_2 \) as noise
- Subtract off \( X_1 \), then decode \( X_2 \) with snr \( P_2/\sigma^2 \)

Easy to implement with SPARCs

Rate \( R_1 \) SPARC defined by \( n \times M_1 L_1 \) matrix \( A_1 \)

Rate \( R_2 \) SPARC defined by \( n \times M_2 L_2 \) matrix \( A_2 \)

\[ Y = A_1 \beta_1 + A_2 \beta_2 + \text{Noise} \]
Joint Decoding

\[ Y = \begin{bmatrix} A_1 \ A_2 \end{bmatrix} \begin{bmatrix} \beta_1 \\ \beta_2 \end{bmatrix} + \text{Noise} \]

- One can also decode the message pair \( \beta \) \(:= [\beta_1; \beta_2] \) \textit{directly} using design matrix \( A := [A_1 \ A_2] \)
- Can achieve \textit{all} the points in the capacity region
- Idea extends to > 2 users
- Can achieve capacity region of scalar Gaussian Broadcast channel with similar superposition idea

Codes for MAC and BC are straightforward because SPARC is already based on superposition coding!
Compression with Decoder Side-Information

\[ X \xrightarrow{\text{Encoder}} R \xrightarrow{\text{Decoder}} \hat{X} \]

- Side-information \( Y = X + Z \)
  \( X \sim \mathcal{N}(0, \nu^2), \quad Z \sim \mathcal{N}(0, \sigma^2) \)
- Want to compress \( X \) to within squared-distortion \( D \in (0, \text{var}(X|Y)) \), \( \text{var}(X|Y) = \frac{\nu^2 \sigma^2}{\nu^2 + \sigma^2} \)

[Wyner-Ziv '75]: The optimal rate-distortion function is

\[ R^*(D) = \frac{1}{2} \log \frac{\text{var}(X|Y)}{D}, \quad D \in (0, \text{var}(X|Y)) \]

- Want to achieve this with feasible encoding + decoding
Wyner-Ziv Coding Scheme

Encoder \xrightarrow{R} \text{Decoder} \rightarrow \hat{X}

Side-info $Y = X + Z$

$X \sim \mathcal{N}(0, \nu^2), \quad Z \sim \mathcal{N}(0, \sigma^2)$
Wyner-Ziv Coding Scheme

Encoder

- Quantize $X$ to $U$: find $U$ that minimizes $\|X - U\|^2$
- $Z' = X - U$, want $R_1$ large enough that the distortion

$$\frac{\|Z'\|^2}{n} \leq \left( \frac{1}{\nu^2} + \frac{1}{D} - \frac{1}{\text{var}(X|Y)} \right)^{-1}$$

Decoder

Side-info $Y = X + Z$

$$X \sim \mathcal{N}(0, \nu^2), \quad Z \sim \mathcal{N}(0, \sigma^2)$$
Wyner-Ziv Coding Scheme

Encoder

- Quantize $X$ to $U$: find $U$ that minimizes $\|X - U\|^2$
- $Z' = X - U$, want $R_1$ large enough that the distortion
  \[
  \frac{\|Z'\|^2}{n} \leq \left( \frac{1}{\nu^2} + \frac{1}{D} - \frac{1}{\text{var}(X|Y)} \right)^{-1}
  \]
Wyner-Ziv Coding Scheme

Encoder

$X \rightarrow$ Encoder $\rightarrow R \rightarrow$ Decoder $\rightarrow \hat{X}$

Side-info $Y = X + Z$

$X \sim \mathcal{N}(0, \nu^2), \quad Z \sim \mathcal{N}(0, \sigma^2)$

Decoder

$Y = X + Z = U + Z' + Z$

- Find $U$ within bin that minimizes $\|Y - U\|^2$
- Reconstruct $\hat{X} = \mathbb{E}[X | U, Y]$
Binning with SPARCs

A: 

$$\begin{bmatrix}
\text{Section 1} \\
\text{Section 2} \\
\text{Section } L
\end{bmatrix}$$

$$M \text{ columns}$$

$$\beta: \begin{bmatrix} 0, & \ldots & 0, c_1, & \ldots & c_2, 0, & \ldots & c_L, 0, & \ldots & 0 \end{bmatrix}^T$$

- Quantize $X$ to $U$ using $n \times ML$ SPARC (rate $R_1$)
Binning with SPARCs

Quantize $X$ to $U$ using $n \times ML$ SPARC (rate $R_1$)
Divide each section into subsections of $M'$ columns
Encoder sends indices of sub-sections containing the column
Binning with SPARCs

Quantize $X$ to $U$ using $n \times ML$ SPARC (rate $R_1$)

Divide each section into subsections of $M'$ columns

Encoder sends indices of sub-sections containing the column

Each Binning is a collection of $L$ sub-sections

$$(M/M')^L = 2^{nR} \text{ bins}$$
Bin bin with SPARC s

**A:**

- Quantize $X$ to $U$ using $n \times ML$ SPARC (rate $R_1$)
- Divide each section into subsections of $M'$ columns
- Encoder sends indices of sub-sections containing the column
- Each Bin is a collection of $L$ sub-sections

$$(M/M')^L = 2^{nR} \text{ bins}$$

- Decodes $Y$ to $U$ within smaller $n \times M'L$ SPARC
Writing on Dirty Paper

\[ Y = X + S + \varepsilon, \quad S \sim \mathcal{N}(0, \sigma_s^2), \quad \varepsilon \sim \mathcal{N}(0, \sigma^2), \quad \frac{\|X\|^2}{n} \leq P \]

Theorem [Gelfand-Pinsker ’80, Costa ’83]

The capacity of this channel is \( \frac{1}{2} \log \left(1 + \frac{P}{\sigma^2}\right) \)

High-rate channel code split into bins of lower rate “source” codes
SPARC Construction

\[ Y = X + S + \varepsilon, \quad \frac{\|X\|_2^2}{n} \leq P \]

![Diagram showing A, M', Section 1, Section L, M columns, \( A: \), \( \beta: \), \( \beta: [0, \ldots, 0, c_1, \ldots, c_2, 0, \ldots, c_L, 0, \ldots, 0]^T \)\]

**Encoder**
- \( n \times ML \) SPARC of rate \( R_1 \)
- Divide each section into \( M' \) subsections
  - Defines \((M/M')^L = 2^{nR}\) bins
SPARC Construction

\[ Y = X + S + \varepsilon, \quad \frac{\|X\|^2}{n} \leq P \]

**Encoder**

- **Within message bin** quantize \( S \) to \( U \) using rate \((R_1 - R)\)
- SPARC
- Transmit \( X = U - \alpha S \), for appropriately chosen constant \( \alpha \)
SPARC Construction

\[ Y = X + S + \varepsilon, \quad \frac{\|X\|_2^2}{n} \leq P \]

\[
\begin{pmatrix}
\frac{M}{A}: \\
\frac{\beta}{\beta}: \\
\frac{T}{T_0}, c_1, c_L, 0, M columns \\
Section 1 \\
\end{pmatrix}
\]

\[
\frac{\frac{c_2, 0, M columns}{M columns}}{Section L} \\
\frac{(1 + \kappa)U + \varepsilon'}{Y = (1 + \kappa)U + \varepsilon'} \iff Y = X + S + \varepsilon
\]

- Decode \( U \) from \( Y \) the \textit{big} (rate \( R_1 \)) codebook
Binning with SPARCs

Theorem (Venkataramanan-Tatikonda ’12)

With optimal (ML) encoding + decoding, SPARCs attain the optimal information-theoretic rates for Gaussian Wyner-Ziv and Gelfand-Pinsker models with probability of error exponentially decaying in $n$. 

\[
\beta: \begin{bmatrix} 0, \ldots, 0, c_1, \ldots, c_2, 0, \ldots, c_L, 0, \ldots, 0 \end{bmatrix}^T
\]
Summary

Sparse Regression Codes:

- Rate-optimal for Gaussian point-to-point communication and compression
- Low-complexity encoding and decoding algorithms
- Nice structure that enables binning and superposition
Ongoing Work/Open Questions

– Power Allocation for binning with *feasible* encoding & decoding: Optimal allocation for the source and channel coding parts are different!

– SPARCs for Gaussian channel coding, source coding, binning + superposition ⇒ low-complexity, rate-optimal codes for:
  
  • Distributed Lossy Compression ("Berger-Tung")
  • Gaussian Multiple Descriptions
  • Gaussian Relay Channels
  • Fading Channels, MIMO Channels
  • Gaussian Multi-terminal Networks

– SPARCs for interpreting variables that arise in converses
References

– R. Venkataramanan, A. Joseph and S. Tatikonda, *Lossy Compression via Sparse Linear Regression: Performance under Minimum-distance Encoding*, IEEE Trans. Inf. Theory, June 2014

– R. Venkataramanan, T. Sarkar and S. Tatikonda, *Lossy Compression via Sparse Linear Regression: Computationally Efficient Encoding and Decoding*, IEEE Trans. Inf. Theory, June 2014

– R. Venkataramanan and S. Tatikonda, *The Rate-Distortion Function and Error Exponent of Sparse Regression Codes with Optimal Encoding*, http://arxiv.org/abs/1401.5272 (Short version at ISIT '14)

– R. Venkataramanan and S. Tatikonda, *Sparse Regression Codes for Multi-terminal Source and Channel Coding*, Allerton 2012