ISIT 2016 Tutorial: Sparse Regression Codes

Ramji Venkataramanan
Dept. of Engineering, University of Cambridge
ramji.v@eng.cam.ac.uk

Andrew Barron
Dept. of Statistics, Yale University
andrew.barron@yale.edu

July 8, 2016

Abstract

This handout summarizes the key ideas covered in the tutorial on Sparse Regression Codes (SPARCs). It describes the basic code construction, and the main ideas behind the various encoding and decoding algorithms. The handout contains three sections: the first describes the SPARC construction and its use for AWGN channel coding; the second discusses SPARCs for lossy compression; the final section discusses how to implement binning and superposition in order to construct SPARCs for Gaussian multi-terminal source and channel coding problems. We list some open questions at the end of each section.

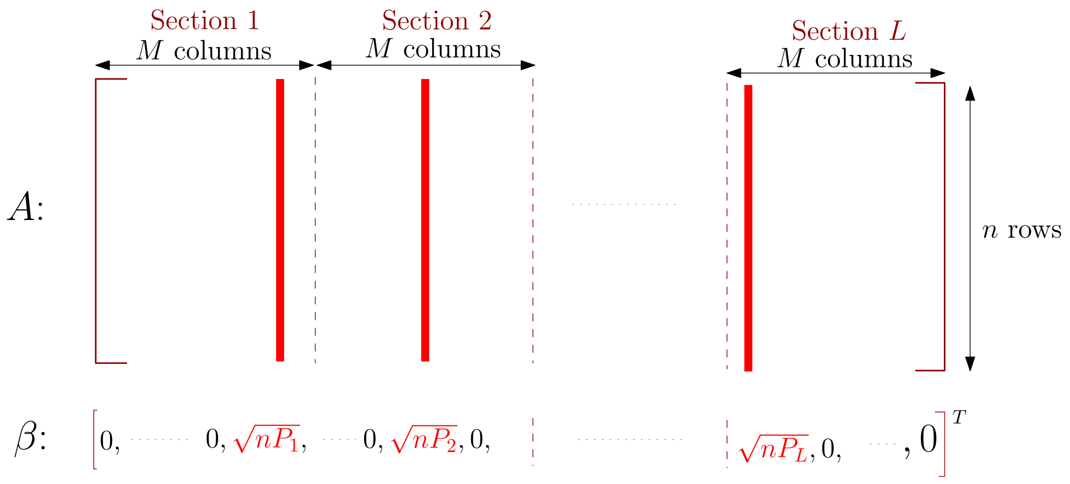
The treatment here is brief and informal — the goal is to provide a quick summary to complement the tutorial slides. We point the reader to references which cover each of the topics in depth. The tutorial slides and handout are available at: `https://goo.gl/8H8wrk`

1 SPARC Construction and AWGN Channel Coding

In this section, we consider communication over the AWGN channel

$$y = x + \varepsilon,$$

where the noise vector ε is i.i.d $\sim \mathcal{N}(0, \sigma^2)$. There is an average power constraint P on the input. The goal is to construct computationally efficient codes to reliably communicate at rates approaching the capacity $\mathcal{C} = \frac{1}{2}\log\left(1 + \frac{P}{\sigma^2}\right)$.


Section 1
M columns
Section 2
M columns
Section L
M columns
A:
n rows
β:
$[0, \cdots 0, \sqrt{nP_1}, \cdots 0, \sqrt{nP_2}, 0, \cdots \sqrt{nP_L}, 0, \cdots, 0]^T$

In the SPARC construction shown above:

- Design matrix A is $n \times ML$, where n is the block length. The entries A_{ij} are iid $\sim \mathcal{N}(0, 1/n)$.
- Codewords are of the form $A\beta$.

- Message vector β contains exactly one non-zero in each of the L sections. The message is specified by the *location* of the non-zero in each section.
- Location of the non-zero in each section is specified by a chunk of $\log_2 M$ input bits $\Rightarrow$ Rate R is determined by $L \log M = nR$.
- For polynomial-sized design matrix A, we choose $M \sim n^{\kappa}$, which gives $L \sim n/\log n$.
- The values of the non-zeros in β are pre-specified. They can be chosen arbitrarily as long as the power constraint is satisfied, i.e., need $\sum_{\ell} P_{\ell} = P$.
- Examples of power allocation: 1) Flat: $P_{\ell} = \frac{P}{L}$ for each ℓ; 2) Exponentially decaying: $P_{\ell} \propto e^{-a\ell/L}$ for some constant a. For all sensible power allocations, $P_{\ell} = \Theta(1/L)$ for all ℓ; thus the non-zero coefficients of β given by $\{\sqrt{nP_{\ell}}\}$ are $\Theta(\sqrt{\log M})$.

1.1 AWGN Decoding

The goal is to recover the SPARC message vector β from $y = A\beta + \varepsilon$.

Optimal Decoding: The least-squares (ML) decoder produces $\hat{\beta}_{ML} = \arg\min_{\hat{\beta} \in SPARC} \left\| y - A\hat{\beta} \right\|^2$. The optimal decoder was analyzed in [1], and shown to achieve an error exponent of the similar order as the Shannon-Gallager random coding error exponent. In particular, for any rate $R < \mathcal{C}$ and $\epsilon > 0$, the probability that the section error rate of the ML decoder exceeds ϵ decays as $e^{-\kappa n \min\{\epsilon, (\mathcal{C}-R)^2\}}$ for some constant κ.

Feasible Decoders: The feasible SPARC decoders described below sequentially produce estimates β^t for $t > 0$, starting with $\beta^0 = 0$. These estimates are produced based on *test statistics* denoted by stat_t, where for $t > 0$, stat_t is generated using $A, y, \beta^1, \dots, \beta^t$. We briefly describe three feasible decoders below. See the tutorial slides and the references for more details.

1) **Adapative, Successive *Hard-thresholding*** [2]: Given β^t, the statistic stat_t is the inner product of the normalized residual $\mathsf{res}_t := \frac{y - A\beta^t}{\|y - A\beta^t\|/\sqrt{n}}$ with each column of A. Columns for which the statistic exceeds a fixed threshold (slightly larger than $\sqrt{2\log M}$) are picked to generate the updated estimate β^{t+1}.

Iterative *Soft-Decision* Decoding: The decoders 2) and 3) below are both *soft-decision* decoders which iteratively update the posterior probabilities of each entry of β being the true non-zero in its section. The goal in both decoders is to generate statistics that satisfy $\mathsf{stat}_t \approx \beta + \tau_t Z_t$, i.e., stat_t is essentially the true signal observed in independent additive Gaussian noise with variance τ_t^2 that can be computed. If this is true, then the Bayes optimal estimate for β in the next step is

$$\beta^{t+1}(\mathsf{stat}_t) = \mathbb{E}[\beta | \beta + \tau_t Z_t = \mathsf{stat}_t]$$

The expression for this conditional expectation is given in slide 23 of Part I. The variances τ_t^2 can be iteratively computed as

$$\tau_t^2 = \sigma^2 + P(1 - x(\tau_{t-1}))$$

where $x_t := x(\tau_{t-1})$ is defined in slides 25/26. Under the assumed distribution for stat_t, we have $\frac{1}{n}\mathbb{E}[\beta^T \beta^t] = \frac{1}{n}\mathbb{E}\left\|\beta^t\right\|^2 = Px_t$ and $\frac{1}{n}\mathbb{E}\left\|\beta - \beta^t\right\|^2 = P(1-x_t)$. Thus the scalar x_t can be interpreted as the expected (power-weighted) success rate, and $P(1-x_t)$ as the expected interference contribution (due to the undecoded sections) to the noise variance τ_t^2.

Hence, the key question is: *how do we iteratively generate statistics stat_t that are well-approximated as $\beta + \tau_t Z_t$?* The two decoders below achieve this via seemingly very different approaches.

2) **Adapative, Successive Soft-thresholding** [3–5]: The statistics for this decoder are defined using the *fits* $Y, A\beta^1, \ldots, A\beta^t$. With $G_0 = Y$, recursively define G_t to be the part of the fit $A\beta^t$ that is orthogonal to $G_0, G_1, \ldots, G_{t-1}$. The ingredients of stat_t are the vectors $\mathcal{Z}_0, \ldots, \mathcal{Z}_t$, defined as

$$\mathcal{Z}_k = \sqrt{n}\,\frac{A^T G_k}{\|G_k\|}, \qquad k \geq 0.$$

The statistic stat_t is then defined as $\mathsf{stat}_t = \tau_t \mathcal{Z}_t^{comb} + \beta^t$, where the combined vector $\mathcal{Z}_t^{comb}$ is defined as a convex combination of $\mathcal{Z}_0, \ldots, \mathcal{Z}_t$. In [4, 5], two different ways to choose the weights of the convex combination are proposed. Both choices are based on the Cholesky decomposition of a matrix generated based on the estimates $\{\beta^1, \ldots, \beta^t\}$ (see slides 41–53). The key result in the theoretical analysis is a distributional characterization of the ingredients $\mathcal{Z}_k$. The analysis shows that both the proposed choices of convex weights lead to stat_t being close to the desired representation.

3) **Approximate Message Passing (AMP)** [6]: The AMP decoder is derived by approximating min-sum-like message passing updates for the decoding problem. This leads to statistics of the form $\mathsf{stat}_t = A^T r^t + \beta^t$, where the "modified residual" r^t is defined as follows: $r^0 = y$, and for $t \geq 1$:

$$r^t = y - A\beta^t + \frac{r^{t-1}}{\tau_{t-1}^2}\left(P - \frac{\left\|\beta^t\right\|^2}{n}\right).$$

The analysis in [6] shows that stat_t defined as above asymptotically have close to the desired representation. The techniques of [7] can be used to refine this characterization and extend the result to the case of large but finite n. The analysis of [7] yields concentration inequalities for various inner products and norms of vectors produced by the AMP decoder. For example, it is shown that $\left\|r^t\right\|^2/n$ concentrates around τ_t^2. Therefore, one could just use $\left\|r^t\right\|^2/n$ in the decoder instead of precomputed values of τ_t^2.

Error Performance: With power allocation $P_\ell \propto e^{-2\mathcal{C}\ell/L}$, all three decoders have similar performance guarantees: for any fixed $R < \mathcal{C}$ and $\epsilon > 0$, the probability that the section error rate is larger than ϵ decays as $\exp(-\kappa n \epsilon^2/(\log n)^c)$. The constants κ, c depend on R and the decoding scheme. However, the empirical error performance of the soft-decision decoders is superior to the hard-thresholding one.

Computational Complexity: The complexity of both soft-decision decoders with Gaussian design matrices are of the same order: $O(nN)$, where $N = ML$. However, in practice the AMP is faster as no orthonormalization or Cholesky decomposition is required to compute its test statistic in each step. As described in [6], Hadamard-based design matrices can be used to reduce the computational complexity to $O(N \log N)$. Further there is a huge savings in the memory requirement since a Hadamard-based design matrix does not need to be stored.

Empirical performance at finite block lengths: As discussed in Part II of the tutorial, for rates below capacity and practical block lengths, power allocation plays a key role in determining the section error rate. (See Slides 23–26 for how to design good power allocations.) Another approach to improving performance at finite block lengths is via spatially coupled design matrices [8, 9]. A spatially-coupled design matrix has a band-diagonal structure, with the first few sections of β oversampled to jump-start the decoding progression.

1.2 Open Questions

- Theoretical Guarantees for the Hadamard-based SPARC. Current analysis techniques for the AMP as well as the adaptive successive decoder heavily depend on the Gaussianity of A.

- Can we combine clever power allocation techniques with spatially-coupled design matrices to boost empirical performance at rates close to $\mathcal{C}$ (at reasonable block lengths)?

- The BIG question: Can we design feasible decoders whose gap to capacity is $O\left(\frac{1}{n^a}\right)$ for some $a \in (0, \frac{1}{2})$? Note that with ML decoding, the gap from capacity for SPARCs is close to order $\frac{1}{\sqrt{n}}$. The current analysis suggests that the gap from capacity for the feasible decoders is of order $1/(\log M)^c$, where the constant $c \gtrsim 1$ varies according to the decoder.

2 Lossy Compression with SPARCs

Given a source sequence $S = (S_1, \ldots, S_n)$ with variance ν^2, the goal is to find a SPARC codeword $A\beta$ such that $\|S - A\beta\|^2 \leq D$, with rate R as close as possible to $R^*(D) = \nu^2 e^{-2R}$ nats. Note that there is no power constraint on the non-zero values of β, but they are still fixed a priori, i.e., only the locations of the non-zeros are sent to the compression decoder with rate R.

Optimal Encoding: With optimal (least-squares) encoding, SPARCs achieve the optimal rate-distortion function for i.i.d. Gaussian sources with the optimal error exponent. This was shown in [10, 11]. Contrast this with AWGN communication , where SPARCs with ML decoding have probability of error decreases exponentially in n, but the error exponent is smaller than the Shannon-Gallager random coding error exponent [1].

Successive Cancellation Encoding: A simple SPARC compression encoder based on successive cancellation was proposed in [12]. The encoder starts with $\beta^0 = 0$, and sequentially encodes the position of the non-zero in each section of β. The non-zero location in section ℓ corresponds to the column in ℓth section of A that maximizes the inner product with the residue $S - A\beta^{\ell-1}$. The non zero value in section ℓ is set to be $\sqrt{2(\log M)\nu^2(1 - \frac{2R}{L})^{\ell}}$. (See Slides 6–10 of Part III of the tutorial).

This encoder was shown in [12] to attain the rate-distortion function, with the probability of distortion exceeding $D^*(R) + \Delta$ bounded by $\exp(-n(\Delta - \frac{c \log\log n}{\log n}))$. Note that such a “hard-decision” successive cancellation approach does not work well for AWGN channel decoding, i.e., the section error rate would decay much slower than exponentially with block length n.

2.1 Open Questions

- The main open question is how to design feasible encoders with better compression performance, i.e., whose gap from $D^*(R)$ is smaller than $O\left(\frac{\log\log n}{\log n}\right)$. In particular, can we design soft-decision based encoders, e.g, an AMP encoder?

- Can one extend the results with the optimal encoder and the successive cancellation encoder (which hold for Gaussian design matrices) to Bernoulli dictionaries with i.i.d. ± 1 entries? Note that for AWGN channels, SPARC ML decoding with i.i.d. ± 1 design matrices has been analyzed in [13].

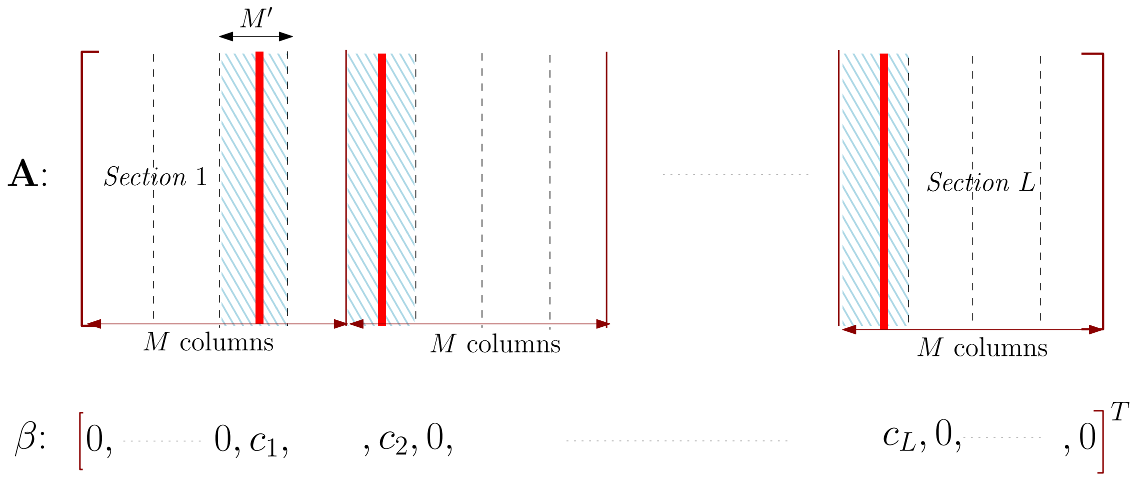
- Can one extend SPARC lossy compression to finite alphabet sources, e.g., binary sources with Hamming distortion?

3 Multi-terminal Source and Channel Coding with SPARCs

Coding schemes that achieve the optimal rate-regions for several multi-terminal source and channel coding models often use the following ingredients: i) rate-optimal point-to-point source and channel codes, and ii) combining or splitting these point-to-point codes via superposition or binning [14].

Since we already have low-complexity SPARC encoding and decoding algorithms for point-to-point source and channel coding, it remains to show how to implement superposition and binning with the SPARC structure. Superposition is easy to implement, as SPARCs are already motivated by the idea of superposition! (See slides 14–16 in part III of the tutorial.)

We now describe how to bin a rate R_1 SPARC (with 2^{nR_1} codewords) into 2^{nR} bins, where $R < R_1$. Fix the parameters M, L, n of the design matrix A such that $L \log M = nR_1$.


M'
A:
Section 1
Section L
M columns
M columns
M columns
β: $[0, \cdots 0, c_1, \quad , c_2, 0, \cdots c_L, 0, \cdots , 0]^T$

As shown above, sub-divide each section of A into sub-sections consisting of M' columns each . Now, each *bin* is indexed by picking one sub-section from each section. For example, the collection of shaded subsections in the figure together forms one bin. Since we have (M/M') sub-section choices in each of the L sections, the number of bins is $(M/M')^L$. Choosing M' such that $L \log M' = n(R_1 - R)$, we have 2^{nR} bins as required.

This gives us a way to construct rate-optimal SPARCs for a variety of Gaussian multi-terminal models including broadcast channels, multiple-access channels, as well as source (channel) coding with decoder (encoder) side-information. See the tutorial slides and [15] for more details.

3.1 Open Questions

Using the point-to-point SPARCs for Gaussian channel and source coding together with the binning and superposition techniques described above, construct low-complexity, rate-optimal codes for:

- Distributed Lossy Compression ("Berger-Tung")
- Gaussian Multiple Descriptions
- Fading Channels, MIMO Channels
- Gaussian Relay Channels
- Other Gaussian Multi-terminal Networks

References

[1] A. Barron and A. Joseph, "Least squares superposition codes of moderate dictionary size are reliable at rates up to capacity," *IEEE Trans. on Inf. Theory*, vol. 58, pp. 2541–2557, Feb. 2012.

[2] A. Joseph and A. R. Barron, “Fast sparse superposition codes have near exponential error probability for $R < \mathcal{C}$,” *IEEE Trans. Inf. Theory*, vol. 60, pp. 919–942, Feb. 2014.

[3] A. R. Barron and S. Cho, “High-rate sparse superposition codes with iteratively optimal estimates,” in *Proc. IEEE Int. Symp. Inf. Theory*, 2012.

[4] S. Cho and A. Barron, “Approximate iterative bayes optimal estimates for high-rate sparse superposition codes,” in *Sixth Workshop on Information-Theoretic Methods in Science and Engineering*, 2013.

[5] S. Cho, *High-dimensional regression with random design, including sparse superposition codes.* PhD thesis, Yale University, 2014.

[6] C. Rush, A. Greig, and R. Venkataramanan, “Capacity-achieving sparse regression codes via approximate message passing decoding,” *Proc. IEEE Int. Symp. Inf. Theory*, 2015. Full version: `https://arxiv.org/abs/1501.05892`.

[7] C. Rush and R. Venkataramanan, “Finite sample analysis of approximate message passing,” in *Proc. ISIT 2016*, 2016. Full version: `https://arxiv.org/abs/1606.01800`.

[8] J. Barbier, C. Schülke, and F. Krzakala, “Approximate message-passing with spatially coupled structured operators, with applications to compressed sensing and sparse superposition codes,” *Journal of Statistical Mechanics: Theory and Experiment*, no. 5, 2015.

[9] J. Barbier and F. Krzakala, “Approximate message-passing decoder and capacity-achieving sparse superposition codes,” *arXiv:1503.08040*, 2015.

[10] R. Venkataramanan, A. Joseph, and S. Tatikonda, “Lossy compression via sparse linear regression: Performance under minimum-distance encoding,” *IEEE Trans. Inf. Thy*, vol. 60, pp. 3254–3264, June 2014.

[11] R. Venkataramanan and S. Tatikonda, “The rate-distortion function and error exponent of sparse regression codes with optimal encoding,” in *Proc. ISIT 2014*. Full version: `https://arxiv.org/abs/1401.5272`.

[12] R. Venkataramanan, T. Sarkar, and S. Tatikonda, “Lossy compression via sparse linear regression: Computationally efficient encoding and decoding,” *IEEE Trans. Inf. Theory*, vol. 60, pp. 3265–3278, June 2014.

[13] Y. Takeishi, M. Kawakita, and J. Takeuchi, “Least squares superposition codes with Bernoulli dictionary are still reliable at rates up to capacity,” *IEEE Transactions on Information Theory*, vol. 60, pp. 2737–2750, May 2014.

[14] A. El Gamal and Y.-H. Kim, *Network Information Theory.* Cambridge University Press, 2011.

[15] R. Venkataramanan and S. Tatikonda, “Sparse regression codes for multi-terminal source and channel coding,” in *50th Allerton Conf. on Commun., Control, and Computing*, 2012.