Capacity-achieving Sparse Regression Codes via Approximate Message Passing Decoding

Cynthia Rush
Yale University, USA
Email: cynthia.rush@yale.edu

Adam Greig
University of Cambridge, UK
Email: ag611@cam.ac.uk

Ramji Venkataramanan
University of Cambridge, UK
Email: ramji.v@eng.cam.ac.uk

***Abstract*—Sparse superposition codes were recently introduced by Barron and Joseph for reliable communication over the AWGN channel at rates approaching the channel capacity. In this code, the codewords are sparse linear combinations of columns of a design matrix. In this paper, we propose an approximate message passing decoder for sparse superposition codes. The complexity of the decoder scales linearly with the size of the design matrix. The performance of the decoder is rigorously analyzed and it is shown to asymptotically achieve the AWGN capacity. We also provide simulation results to demonstrate the performance of the decoder at finite block lengths, and introduce a power allocation that significantly improves the empirical performance.**

I. Introduction

This paper considers the problem of constructing low-complexity, capacity-achieving codes for the memoryless additive white Gaussian noise (AWGN) channel. The channel generates output y from input x according to

$$y = x + w, \tag{1}$$

where the noise w is a Gaussian random variable with zero mean and variance σ^2. There is an average power constraint P on the input x: if $x_1, \dots, x_n$ are transmitted over n uses of the channel, then we require that $\frac{1}{n}\sum_{i=1}^{n} x_i^2 \leq P$. The signal-to-noise ratio $\frac{P}{\sigma^2}$ is denoted by snr. The goal is to construct codes with computationally efficient encoding and decoding, whose rates approach the channel capacity given by

$$\mathcal{C} := \tfrac{1}{2}\log(1 + \mathsf{snr}). \tag{2}$$

Sparse superposition codes, also called Sparse Regression Codes (SPARCs), were recently introduced by Barron and Joseph [1] for communication over the channel in (1). They proposed an efficient decoding algorithm called 'adaptive successive decoding', and showed that for any fixed rate $R < \mathcal{C}$, the probability of decoding error decays to zero exponentially in $\frac{n}{\log n}$, where n is the block length of the code. Subsequently, a soft-decision iterative decoder was proposed by Cho and Barron [2], [3], with theoretical guarantees similar to the decoder in [1] but improved empirical performance.

In this paper, we propose an approximate message passing (AMP) decoder for SPARCs. We analyze its performance and prove that the probability of decoding error goes to zero with growing block length for all fixed rates $R < \mathcal{C}$.

Approximate Message Passing (AMP): AMP refers to a class of algorithms [4]–[9] that are Gaussian or quadratic approximations of loopy belief propagation algorithms (e.g., min-sum, sum-product) on dense factor graphs. AMP has proved particularly effective for the compressed sensing problem [10], which is described by the model

$$y = A\beta + w. \tag{3}$$

Here A is an $n \times N$ measurement matrix with $n < N$, $\beta \in \mathbb{R}^N$ is a sparse vector to be reconstructed from the observed vector $y \in \mathbb{R}^n$, and $w \in \mathbb{R}^n$ is the measurement noise. One popular class of reconstruction algorithms is ℓ_1-norm based convex optimization, e.g. [11], [12]. Though these algorithms have strong theoretical guarantees, the computational cost makes it challenging to implement the convex optimization procedures for problems where N is large.

The factor graph corresponding to the model in (3) is dense, hence it is infeasible to implement message passing algorithms in which the messages are complicated real-valued functions. AMP circumvents this difficulty by passing only scalar parameters corresponding to these functions. The references [5], [7]–[9] describe how various flavors of AMP for the compressed sensing model can be obtained by approximating the standard message passing equations. These approximations reduce the message passing equations to a set of simple rules for computing successive estimates of β.

In [4], it was demonstrated via numerical experiments that: i) the empirical performance of AMP for a large class of measurement matrices is similar to convex optimization based methods at significantly lower computational cost, and ii) the mean-squared reconstruction error of these estimates of β could be tracked by a simple scalar iteration called *state evolution*. In [6], it was rigorously proved that state evolution is accurate in the large system limit when the measurement matrix A has i.i.d. Gaussian entries.

Main Contributions: We propose an AMP decoder for sparse regression codes, which is derived via a first-order approximation of a min-sum-like message passing algorithm. The main result of the paper is Theorem 1, in which we rigorously show that for all rates $R < \mathcal{C}$, the probability of decoding error goes to zero as the block length tends to infinity. We also present simulation results to demonstrate the performance of the decoder at finite block lengths, and

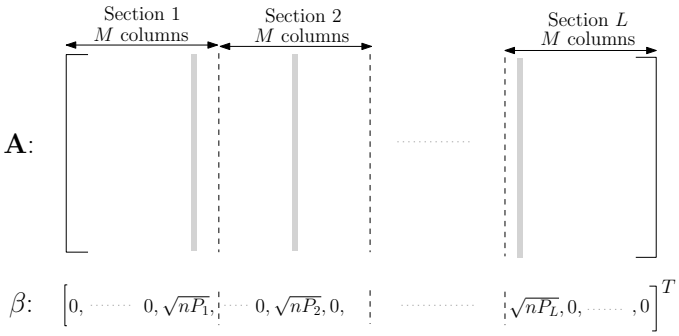

Fig. 1: A is an $n \times ML$ matrix and β is a $ML \times 1$ vector. The positions of the non-zeros in β correspond to the gray columns of A which combine to form the codeword $A\beta$.

introduce a power allocation that significantly improves the empirical performance.

We use the framework of Bayati and Montanari [6] to prove our main result. However, we remark that the analysis of the proposed algorithm does not follow directly from the results in [6], [13]. The main reason for this is that the *undersampling ratio* n/N in our setting goes to zero in the large system limit, whereas previous rigorous analyses of AMP consider the case where the undersampling ratio is a constant.

Related Work on SPARCs: In addition to the the adaptive successive decoder [1] and the iterative soft-decision decoder [2], [3], Barbier and Krzakala recently proposed an approximate message passing decoder for sparse superposition codes in [14]. Their decoder has different update rules from the AMP proposed here. A replica-based analysis of the decoder in [14] suggested it could not achieve rates beyond a threshold which was strictly smaller than $\mathcal{C}$. Very recently, Barbier et al [15] reported empirical results which show that the performance of the decoder in [14] can be improved by using spatially coupled Hadamard matrices to define the code.

Notation: The ℓ_2-norm of vector x is denoted by $\|x\|$. The transpose of a matrix B is denoted by B^*. For any positive integer m, $[m]$ is the set $\{1, \ldots, m\}$. The indicator function of an event $\mathcal{A}$ is denoted by $\mathbf{1}(\mathcal{A})$. log and ln are used to denote logarithms with base 2 and base e, respectively. Rate is measured in bits.

II. The Sparse Regression Codebook

A sparse regression code (SPARC) is defined in terms of a design matrix A of dimension $n \times ML$, whose entries are i.i.d. $\mathcal{N}(0, \frac{1}{n})$. Here n is the block length and M and L are integers whose values will be specified shortly in terms of n and the rate R. As shown in Fig. 1, one can think of the matrix A being composed of L sections with M columns each. Each codeword is a linear combination of L columns, with one column from each section. Formally, a codeword can be expressed as $A\beta$, where β is an $ML \times 1$ vector $(\beta_1, \ldots, \beta_{ML})$ with the following property: there is exactly one non-zero β_j for $1 \le j \le M$, one non-zero β_j for $M+1 \le j \le 2M$, and so forth. The non-zero value of β in section ℓ is set to $\sqrt{nP_\ell}$, where $P_1, \ldots, P_L$ are positive constants that satisfy $\sum_{\ell=1}^{L} P_\ell = P$. Denote the set of all β's that satisfy this property by $\mathcal{B}_{M,L}(P_1, \ldots, P_L)$.

The total number of codewords is M^L. To obtain a rate of R bits/sample, we need

$$M^L = 2^{nR} \quad \text{or} \quad L \log M = nR. \tag{4}$$

There are several choices for the pair (M, L) which satisfy (4). For our constructions, we will choose M equal to L^b, for some constant $b > 0$. Then, (4) becomes

$$bL \log L = nR. \tag{5}$$

Thus $L = \Theta(\frac{n}{\log n})$, and the size of the design matrix A (given by $n \times ML = n \times L^{b+1}$) now grows polynomially in n.

Encoding: The encoder splits its stream of input bits into segments of $\log M$ bits each. The message vector β_0 is indexed by L such segments—the decimal equivalent of segment ℓ determines the position of the non-zero coefficient in section ℓ of β_0. The input codeword is then computed as $X = A\beta_0$.

Power Allocation: For our main result, we use an exponentially decaying allocation of the form $P_\ell \propto 2^{-2C\ell/L}$, $\ell \in [L]$. In Section IV-A, we discuss alternative power allocations, and find that an appropriate combination of exponential and flat power allocations yields good decoding performance at finite block lengths. Both the design matrix A and the power allocation are known to the encoder and the decoder before communication begins.

Some more notation: In the analysis, we will treat the message as a random vector β, which is uniformly distributed over $\mathcal{B}_{M,L}(P_1, \ldots, P_L)$, the set of length ML vectors that have a single non-zero entry $\sqrt{nP_\ell}$ in section ℓ, for $\ell \in [L]$. We will denote the true message vector by β_0; β_0 is interpreted as a *realization* of the random vector β.

We will use indices i, j to denote specific entries of β, while the index ℓ will be used to denote the entire section ℓ of β. Thus β_i, β_j are scalars, while β_ℓ is a length M vector. We set $N = ML$, and write $\lim x$ to denote the limit of x as the SPARC parameters $n, L, M \to \infty$ simultaneously, according to $M = L^b$ and $bL \log L = nR$.

III. The AMP Channel Decoder

Given the received vector $y = A\beta_0 + w$, the AMP decoder generates successive estimates of the message vector, denoted by $\{\beta^t\}$, where $\beta^t \in \mathbb{R}^N$ for $t = 1, 2, \ldots$. Set $\beta^0 = 0$, the all-zeros vector. For $t = 0, 1, \ldots$, compute

$$z^t = y - A\beta^t + \frac{z^{t-1}}{\tau_{t-1}^2}\left(P - \frac{\|\beta^t\|^2}{n}\right), \tag{6}$$

$$\beta_i^{t+1} = \eta_i^t(\beta^t + A^* z^t), \quad \text{for } i = 1, \ldots, N = ML, \tag{7}$$

where quantities with negative indices are set equal to zero. The constants $\{\tau_t\}$, and the estimation functions $\eta_i^t(\cdot)$ are defined as follows for $t = 0, 1, \ldots$. Define

$$\tau_0^2 = \sigma^2 + P, \quad \tau_{t+1}^2 = \sigma^2 + P(1 - x_{t+1}), \quad t \ge 0, \tag{8}$$

where

$$x_{t+1} = \sum_{\ell=1}^{L} \frac{P_\ell}{P} \mathbb{E}\left[\frac{\exp\left(\frac{\sqrt{nP_\ell}}{\tau_t}\left(U_1^\ell + \frac{\sqrt{nP_\ell}}{\tau_t}\right)\right)}{\exp\left(\frac{\sqrt{nP_\ell}}{\tau_t}\left(U_1^\ell + \frac{\sqrt{nP_\ell}}{\tau_t}\right)\right) + \sum_{j=2}^{M} \exp\left(\frac{\sqrt{nP_\ell}}{\tau_t} U_j^\ell\right)} \right] \tag{9}$$

In (9), $\{U_j^\ell\}$ are i.i.d. $\mathcal{N}(0,1)$ random variables for $j \in [M],\ \ell \in [L]$. For $i \in [N]$, define

$$\eta_i^t(s) = \sqrt{nP_\ell} \frac{\exp\left(\frac{s_i\sqrt{nP_\ell}}{\tau_t^2}\right)}{\sum_{j\in \text{sec}_\ell} \exp\left(\frac{s_j\sqrt{nP_\ell}}{\tau_t^2}\right)},\ \text{if } i \in \text{sec}_\ell,\ \ell \in [L]. \tag{10}$$

The notation $j \in \text{sec}_\ell$ is used as shorthand for "index j in section ℓ". Notice that $\eta_i^t(s)$ depends on all the components of s in the *section* containing i.

Before running the AMP decoder, the constants $\{\tau_t\}$ must be iteratively computed using (8) and (9). This is an offline computation, and can be done through Monte Carlo simulation. The relation (8), which describes how τ_{t+1} is obtained from τ_t, is called *state evolution*, following the terminology in [4], [6]. For Theorem 1, we derive closed form expressions for the trajectories of x^{t+1} and τ_t^2 as $n \to \infty$. For now, it suffices to note that for any fixed $R < \mathcal{C}$, τ_t strictly decreases with t for a finite number of steps T_n, at which point we have $\tau_{T_{n+1}} \geq \tau_{T_n}$. Having determined $\tau_0, \tau_1, \ldots, \tau_{T_n}$, the decoder iteratively computes codeword estimates $\beta^1, \ldots, \beta^{T_n}$ using (6) and (7). Finally, in each section ℓ of β^{T_n}, set the maximum value to $\sqrt{nP_\ell}$ and remaining entries to 0 to obtain the decoded message $\hat{\beta}$.

The derivation of the AMP from a min-sum-like message passing algorithm is given in [16]. In the remainder of this section, we give some intuition about the update and state evolution equations (6)–(10) in the large system limit.

A. The Test Statistics $\beta^t + A^* z^t$

Consider the AMP update step (7), in which β^{t+1} is generated from the test statistic

$$s^t := \beta^t + A^* z^t. \tag{11}$$

The update in (7) is underpinned by the following key property of the test statistic: *s^t is asymptotically (as $n \to \infty$) distributed as $\beta + \bar{\tau}_t Z$, where $\bar{\tau}_t$ is the limit of τ_t, and Z is an i.i.d. $\mathcal{N}(0,1)$ random vector independent of the message vector β.* This property is due to the presence of the "Onsager" term

$$\frac{z^{t-1}}{\tau_{t-1}^2}\left(P - \frac{\|\beta^t\|^2}{n}\right)$$

in the residue update step (6); see [6, Section I-C] for a discussion about role of the Onsager term in AMP.

In light of the above property, a natural way to generate β^{t+1} from $s^t = s$ is

$$\beta^{t+1}(s) = \mathbb{E}[\beta \,|\, \beta + \tau_t Z = s], \tag{12}$$

i.e., β^{t+1} is the Bayes optimal estimate of β given the observation $s^t = \beta + \tau_t Z$. For $i \in \text{sec}_\ell$, $\ell \in [L]$, we have

$$\beta_i^{t+1}(s) = \mathbb{E}[\beta_i | \beta + \tau_t Z = s] = \mathbb{E}[\beta_i | \{\beta_j + \tau_t Z_j = s_j\}_{j\in \text{sec}_\ell}] = \sqrt{nP_\ell} \frac{\exp\left(\frac{s_i\sqrt{nP_\ell}}{\tau_t^2}\right)}{\sum_{j\in \text{sec}_\ell} \exp\left(\frac{s_j\sqrt{nP_\ell}}{\tau_t^2}\right)}, \tag{13}$$

which is the expression in (10). The conditional expectation in (13) is computed using the following: β and Z are independent, Z is i.i.d $\sim \mathcal{N}(0,1)$, and the location of the non-zero entry in each section ℓ of β is uniformly distributed within the section.

Thus, under the assumption that $s^t = \beta + \tau_t Z$, β^{t+1} is the estimate of the message vector β (based on s^t) that minimizes the expected squared estimation error. Also, for $i \in \text{sec}_\ell$, $\beta_i^{t+1}/\sqrt{nP_\ell}$ is the posterior probability of β_i being the non-zero entry in section ℓ, conditioned on the observation s^t.

B. State Evolution and its Consequences

The following proposition shows that we can interpret the quantity x_{t+1} defined in (9) as the expectation of the (power-weighted) fraction of correctly decoded sections in step $(t+1)$.

Proposition 1. *Under the assumption that $s^t = \beta + \tau_t Z$, where $Z \sim$ i.i.d. $\mathcal{N}(0,1)$ and independent of β, the quantity x^{t+1} defined in* (9) *satisfies*

$$x_{t+1} = \frac{1}{nP}\mathbb{E}[\beta^* \beta^{t+1}], \quad \textit{and } 1 - x_{t+1} = \frac{1}{nP}\mathbb{E}[\|\beta - \beta^{t+1}\|^2].$$

The proof is given in [16]. We emphasize that the above interpretation is accurate only in the limit as $n, M, L \to \infty$, when s^t is distributed as $\beta + \bar{\tau}_t Z$, with $\bar{\tau}_t := \lim \tau_t$.

We now state two lemmas which give closed-form expressions for $\bar{\tau}_t := \lim \tau_t$ and $\bar{x}_{t+1} := \lim x_{t+1}$. Treating x_{t+1} in (9) as a function of τ, we can define

$$x(\tau) := \sum_{\ell=1}^{L} \frac{P_\ell}{P} \mathbb{E}\left[\frac{\exp\left(\frac{\sqrt{nP_\ell}}{\tau}\left(U_1^\ell + \frac{\sqrt{nP_\ell}}{\tau}\right)\right)}{\exp\left(\frac{\sqrt{nP_\ell}}{\tau}\left(U_1^\ell + \frac{\sqrt{nP_\ell}}{\tau}\right)\right) + \sum_{j=2}^{M} \exp\left(\frac{\sqrt{nP_\ell}}{\tau} U_j^\ell\right)} \right] \tag{14}$$

where $\{U_j^\ell\}$ are i.i.d. $\sim \mathcal{N}(0,1)$ for $j \in [M],\ \ell \in [L]$.

Lemma 1. *For $t = 0, 1, \ldots$, we have*

$$\bar{x}(\tau) := \lim x(\tau) = \lim_{L\to\infty} \sum_{\ell=1}^{L} \frac{P_\ell}{P} \mathbf{1}\{c_\ell > 2(\ln 2) R \tau^2\} \tag{15}$$

where $c_\ell := \lim_{L\to\infty} LP_\ell$.

The next lemma uses Lemma 1 to obtain the asymptotic state evolution equations for the following exponentially decaying power allocation:

$$P_\ell = P \cdot \frac{2^{2\mathcal{C}/L} - 1}{1 - 2^{-2\mathcal{C}}} \cdot 2^{-2\mathcal{C}\ell/L}, \quad \ell \in [L]. \tag{16}$$

Lemma 2. *For the power allocation in* (16) *and* $t \geq 0$*:*

$$\bar{x}_t := \lim x_t = \frac{(1+\mathit{snr}) - (1+\mathit{snr})^{1-\xi_{t-1}}}{\mathit{snr}}, \tag{17}$$

$$\bar{\tau}_t^2 := \lim \tau_t^2 = \sigma^2 + P(1-\bar{x}_t) = \sigma^2 (1+\mathit{snr})^{1-\xi_{t-1}} \tag{18}$$

where $\xi_{-1} = 0$ *and for* $t \geq 0$*,*

$$\xi_t = \min\left\{ \left(\frac{1}{2\mathcal{C}} \log\left(\frac{\mathcal{C}}{R}\right) + \xi_{t-1} \right),\ 1 \right\}. \tag{19}$$

The proofs of the lemmas are given in [16].

We observe from (19) that for $R < \mathcal{C}$, ξ_t increases in each step by $\frac{1}{2\mathcal{C}} \log\left(\frac{\mathcal{C}}{R}\right)$ until it equals 1. Similarly, (17) implies that $\bar{x}_t$ strictly increases with t until it reaches one, and the number of steps T^* until $\bar{x}_{T^*} = 1$ is

$$T^* = \left\lceil \frac{2\mathcal{C}}{\log(\mathcal{C}/R)} \right\rceil. \tag{20}$$

The constants $\{\xi_t\}_{t \geq 0}$ have a nice interpretation in the large system limit: at the end of step $t+1$, the first ξ_t fraction of sections in β^{t+1} will be correctly decodable with high probability, i.e., the true non-zero entry in these sections will have almost all the posterior probability mass. The other $(1-\xi_t)$ fraction of sections *will not* be correctly decodable from β^{t+1} as the power allocated to these sections is not large enough. An additional $\frac{1}{2\mathcal{C}} \log\left(\frac{\mathcal{C}}{R}\right)$ fraction of sections become correctly decodable in each step until T^*, when all the sections are correctly decodable with high probability.

As $\bar{x}_t$ increases to 1, (18) implies that the variance of the "noise" in the AMP test statistic decreases monotonically from $\bar{\tau}_0^2 = \sigma^2 + P$ down to $\bar{\tau}_{T^*}^2 = \sigma^2$. That is, the initial observation $y = A\beta + w$ is effectively transformed by the AMP decoder into a cleaner statistic $s^{T^*} = \beta + w'$, where w' is Gaussian with the *same* variance as the measurement noise w.

To summarize, in the large system limit:

- For any fixed $R < \mathcal{C}$, the AMP decoder terminates within a finite number of steps T^* given by (20).
- At the termination step T^*, $\lim \frac{1}{n} \mathbb{E} \|\beta - \beta^{T^*}\|^2 = 0$.

For finite-sized dictionaries, the test statistic s^t will not be precisely distributed as $\beta + \tau_t Z$. Nevertheless, computing x_{t+1} numerically via (8) and (9) yields an estimate for the expected weighted fraction of correctly decoded sections after each step.

IV. Performance of the AMP Decoder

Our main result is proved for the following slightly modified AMP decoder, which is run for exactly T^* steps. Set $\beta^0 = 0$ and compute

$$z^t = y - A\beta^t + \frac{z^{t-1}}{\bar{\tau}_{t-1}^2} \left(P - \frac{\|\beta^t\|^2}{n} \right), \tag{21}$$

$$\beta_i^{t+1} = \eta_i^t(\beta^t + A^* z^t), \quad \text{for } i \in [N] \tag{22}$$

where for $i \in \mathrm{sec}_\ell,\ \ell \in [L]$,

$$\eta_i^t(s) = \sqrt{nP_\ell} \frac{\exp\left(s_i \sqrt{nP_\ell}/\bar{\tau}_t^2\right)}{\sum_{j \in \mathrm{sec}_\ell} \exp\left(s_j \sqrt{nP_\ell}/\bar{\tau}_t^2\right)}. \tag{23}$$

The only difference from the earlier decoder described in (8)–(10) is that we now use the limiting value $\bar{\tau}_t^2$ defined in Lemma 2 instead of τ_t^2. The decoded codeword $\hat{\beta}$ is obtained by setting the maximum of β^{T^*} in each section ℓ to $\sqrt{nP_\ell}$ and the remaining entries to 0.

The *section error rate* of a decoder for a SPARC $\mathcal{S}$ is

$$\mathcal{E}_{sec}(\mathcal{S}) := \frac{1}{L} \sum_{\ell=1}^{L} \mathbf{1}\{\hat{\beta}_\ell \neq \beta_{0_\ell}\}. \tag{24}$$

Theorem 1. *Fix any rate* $R < \mathcal{C}$*, and* $b > 0$*. Consider a sequence of rate* R *SPARCs* $\{\mathcal{S}_n\}$ *indexed by block length* n*, with design matrix parameters* L *and* $M = L^b$ *determined according to* (5)*, and an exponentially decaying power allocation given by* (16)*. Then the section error rate of the AMP decoder (described in* (21)–(23)*, and run for* T^* *steps) converges to zero almost surely, i.e., for any* $\epsilon > 0$*,*

$$\lim_{n_0 \to \infty} P\left(\mathcal{E}_{sec}(\mathcal{S}_n) < \epsilon,\ \forall n \geq n_0\right) = 1. \tag{25}$$

The proof of the theorem is given in [16].

Remarks: 1) The probability measure in (25) is over the Gaussian design matrix A, the Gaussian channel noise w, and the uniform prior over the message $\beta \in \mathcal{B}_{M,L}(P_1, \ldots, P_L)$.

2) As in [1], we can construct a concatenated code with an inner SPARC of rate R and an outer Reed-Solomon (RS) code of rate $(1-2\epsilon)$. If M is a prime power, a RS code defined over a finite field of order M defines a one-to-one mapping between a symbol of the RS codeword and a section of the SPARC. The concatenated code has rate $R(1-2\epsilon)$, and decoding complexity that is polynomial in n. The decoded message $\hat{\beta}$ equals β whenever the section error rate of the SPARC is less than ϵ. Thus for any $\epsilon > 0$, the theorem guarantees that the probability of *message* decoding error for a sequence of rate $R(1-2\epsilon)$ SPARC-RS concatenated codes will tend to zero, i.e., $\lim P(\hat{\beta} \neq \beta) = 0$.

A. Simulation Results and the Effect of Power Allocation

In this section, we make two modifications to the SPARC construction to improve the empirical performance at finite block lengths. First, we introduce a power allocation that yields several orders of magnitude improvement in section error rate for rates R that are not very close to the capacity $\mathcal{C}$. Second, we use a Hadamard design matrix (instead of Gaussian), which facilitates a decoder with $O(N \log N)$ running time and a memory requirement of $O(N)$. In comparsion, with a Gaussian design matrix the running time and memory of the AMP decoder are both $O(nN)$.

Modified Power Allocation: The power allocation is characterized by two parameters a, f. For $f \in [0,1]$, let

$$P_\ell = \begin{cases} \kappa \cdot 2^{-2a\mathcal{C}\ell/L}, & 1 \leq \ell \leq fL \\ \kappa \cdot 2^{-2a\mathcal{C}f}, & fL+1 \leq \ell \leq L \end{cases} \tag{26}$$

where κ is a normalizing constant to ensure that the total power across sections is P. For intuition, first assume that $f = 1$. Then (26) implies that $P_\ell \propto 2^{-a2\mathcal{C}\ell/L}$ for $\ell \in [L]$. Setting

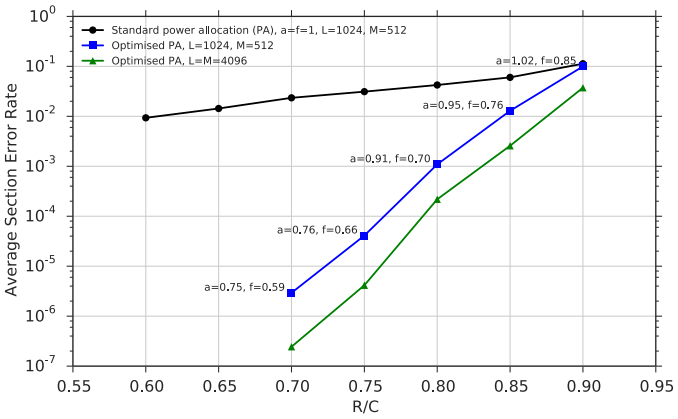

Fig. 2: Section error rate vs $R/\mathcal{C}$ at $\mathsf{snr} = 15, \mathcal{C} = 2$ bits. The top curve shows the average section error rate of the AMP over 1000 trials with $P_\ell \propto 2^{-2\mathcal{C}\ell/L}$. The middle curve shows the section error rate using the power allocation in (26) with the (a, f) values shown. The SPARC parameters for both these curves are $M = 512, L = 1024$. The bottom curve shows the section error rate with the same (a, f) values, but $L = M = 4096$.

$a = 1$ recovers the original power allocation of (16), while $a = 0$ allocates $\frac{P}{L}$ to each section. Increasing a increases the power allocated to the initial sections which makes them more likely to decode correctly, which in turn helps by decreasing the effective noise variance $\bar{\tau}_t^2$ in subsequent AMP iterations. However, if a is too large, the final sections may have too little power to decode correctly.

Hence we want the parameter a to be large enough to ensure that the AMP gets started on the right track, but not much larger. This intuition can be made precise in the large system limit using Lemma 1, which specifies the condition for a section ℓ to be correctly decoded in step $(t+1)$: the limit of LP_ℓ must exceed a threshold proportional to $R\bar{\tau}_t^2$. For rates close to $\mathcal{C}$, we need a to be close to 1 for the initial sections to cross this threshold and get decoding started correctly. On the other hand, for rates such as $R = 0.6\mathcal{C}$, $a = 1$ allocates more power than necessary to the initial sections, leading to poor decoding performance in the final sections.

In addition, we found that the section error rate can be further improved by *flattening* the power allocation in the final sections. For a given a, (26) has an exponential power allocation until section fL, and constant power for the remaining $(1 - f)L$ sections. The allocation in (26) is continuous, i.e. each section in the flat part is allocated the same power as the final section in the exponential part. Flattening boosts the power given to the final sections compared to an exponentially decaying allocation. The two parameters (a, f) let us trade-off between the conflicting objectives of assigning enough power to the initial sections and ensuring that the final sections have enough power to be decoded correctly.

Experimental Results: Fig. 2 shows the performance of the AMP at different rates. Given the values of M, L, the block length n is determined by the rate R according to (4). The top curve shows the average section error rate of the AMP (over 1000 runs) with an exponentially decaying power allocation where $P_\ell \propto 2^{-2\mathcal{C}\ell/L}$. The middle curve shows the average section error rate with the power allocation in (26), with values of (a, f) obtained via a rough optimization around an initial guess of $a = f = R/\mathcal{C}$. The bottom curve shows the average section error rate with $L = M = 4096$, and the power allocation in (26) with same (a, f) values as before.

In all cases, the decoder described in (21)–(23) was used. The constants $\{\bar{\tau}_t^2\}$ required by the decoder are specified by Lemma 2 for the exponential allocation. For the modified allocation the values $\{\bar{\tau}_t^2\}$ can be similarly derived [16, Sec 4.1.1]. The simulations for Fig. 2 were run using Hadamard design matrices; implementation details can be found in [16, Sec 4.1.2].

It is evident that a judicious power allocation scheme can significantly improve section error rates. An interesting open question is to find good rules of thumb for the power allocation as a function of rate and signal-to-noise ratio.

Acknowledgement

RV would like to acknowledge support from a Marie Curie Career Integration Grant (GA Number 631489). AG is supported by an EPSRC Doctoral Training Award.

References

[1] 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.

[2] A. R. Barron and S. Cho, "High-rate sparse superposition codes with iteratively optimal estimates," in *ISIT*, 2012.

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

[4] D. L. Donoho, A. Maleki, and A. Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.

[5] D. L. Donoho, A. Maleki, and A. Montanari, "Message passing algorithms for compressed sensing: I. motivation and construction," in *IEEE Information Theory Workshop (ITW)*, 2010.

[6] M. Bayati and A. Montanari, "The dynamics of message passing on dense graphs, with applications to compressed sensing," *IEEE Trans. Inf. Theory*, pp. 764–785, 2011.

[7] A. Montanari, "Graphical models concepts in compressed sensing," in *Compressed Sensing* (Y. C. Eldar and G. Kutyniok, eds.), pp. 394–438, Cambridge University Press, 2012.

[8] F. Krzakala, M. Mézard, F. Sausset, Y. Sun, and L. Zdeborová, "Probabilistic reconstruction in compressed sensing: algorithms, phase diagrams, and threshold achieving matrices," *Journal of Statistical Mechanics: Theory and Experiment*, no. 8, 2012.

[9] S. Rangan, "Generalized approximate message passing for estimation with random linear mixing," in *ISIT*, pp. 2168–2172, 2011.

[10] R. Baraniuk, E. Candes, R. Nowak, and M. Vetterli (editors), "Special issue on compressive sampling," *IEEE Signal Processing Magazine*, vol. 25, March 2008.

[11] E. Candes and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, pp. 4203 – 4215, Dec. 2005.

[12] D. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, pp. 1289 –1306, April 2006.

[13] A. Javanmard and A. Montanari, "State evolution for general approximate message passing algorithms, with applications to spatial coupling," *Information and Inference*, p. iat004, 2013.

[14] J. Barbier and F. Krzakala, "Replica analysis and approximate message passing decoder for sparse superposition codes," in *ISIT*, 2014.

[15] J. Barbier, C. Schulke, and F. Krzakala, "Approximate message-passing with spatially coupled structured operators, with applications to compressed sensing and sparse superposition codes." arXiv:1312.1740.

[16] C. Rush, A. Greig, and R. Venkataramanan, "Capacity-achieving sparse superposition codes via approximate message passing decoding." arXiv:1501.05892.