Adaptive Compressed Sensing Using Sparse Measurement Matrices

Dejan Vukobratović[1] and Aleksandra Pižurica[2]

[1]Department of Power, Electronics and Communications Engineering, University of Novi Sad, Serbia.

[2]Department of Telecommunications and Information Processing, Ghent University, Belgium.

***Abstract*— Compressed sensing methods using sparse measurement matrices and iterative message-passing recovery procedures are recently investigated due to their low computational complexity and excellent performance. The design and analysis of this class of methods is inspired by a large volume of work on sparse-graph codes such as Low-Density Parity-Check (LDPC) codes and the iterative Belief-Propagation (BP) decoding algorithms. In particular, we focus on a class of compressed sensing methods emerging from the Sudocodes scheme that follow similar ideas used in a class of sparse-graph codes called rateless codes. We are interested in the design and analysis of adaptive Sudocodes methods and this paper provides initial steps in this direction.**

1 Introduction

Let $\mathbf{x} = (x_1, x_2, \ldots, x_N) \in \mathbb{F}_q^N$ be a vector of input symbols (message) from q-ary alphabet. Let us transform this message into arbitrarily long sequence of output symbols $\mathbf{y} = (y_1, y_2, \ldots) \in \mathbb{F}_q$, where each output symbol $y_i \in \mathbb{F}_q$ is a "projection" of message $\mathbf{x}$ onto a sparse binary vector $\mathbf{g}_i$ of length N, i.e., $y_i = \mathbf{g}_i \cdot \mathbf{x}^T$. Each vector $\mathbf{g}_i$ is obtained simply by distributing small number d of ones uniformly at random over N possible positions. We assume the number of ones in $\mathbf{g}_i$, called the degree of $\mathbf{g}_i$, is drawn from a degree distribution Ω over integers $d \in [1, 2, \ldots, N]$. LT codes are class of rateless codes providing a recipe for capacity-approaching degree distributions Ω that, combined with a simple iterative recovery procedure, recover $\mathbf{x}$ from any $N + O(\sqrt{N}\ln^2(N/\delta))$ output symbols with probability $1 - \delta$ [1]. The average degree of Ω scales as $O(\ln(N/\delta))$ and thus results in transformation/recovery complexity of $O(N\ln(N/\delta))$.

If $\mathbf{x} \in \mathbb{R}^N$ and is K-sparse (meaning that only $K << N$ input symbols are non-zero) the above rateless methodology for message recovery is still applicable. Sudocodes scheme [2], that independently emerged in compressed sensing community, exploited similar ideas and proposed the iterative recovery procedure which is later identified to be an instance of the verification-decoding for sparse-graph codes [3]. The Sudocodes scheme is extended into a more general framework called Compressed Sensing via Belief Propagation (CSBP) inspired by the iterative Belief-Propagation (BP) decoding of LDPC codes [4]. Sudocodes sparked significant interest of coding community for Compressed Sensing (CS) methods using sparse measurement matrices and iterative message-passing recovery algorithms. Using well-developed tools from coding theory, CS methods combined with verification-based recovery have been analyzed in [5] and [6].

This paper explores Sudocodes scheme using design and analysis tools used in rateless coding. These tools are applied for the design of *adaptive* Sudocodes scheme, where the sequence of output symbols are not mutually independent but designed based on the outcomes of previous output symbols.

2 The Sudocodes: Design and Analysis

The Sudocodes scheme projects the K-sparse message $\mathbf{x} \in \mathbb{R}^N$ into a sequence of output symbols $y_i \in \mathbb{R}$ using random sparse binary projection vectors $\mathbf{g}_i$ of constant degree $d = L$. In other words, the Sudocodes scheme is a rateless code applying the degree distribution[1] $\Omega(x) = x^L$.

The Sudocode scheme recovers $\mathbf{x}$ from a sequence of M output symbols $\mathbf{y} = (y_1, y_2, \ldots, y_M)$ by applying verification-based recovery across a measurement graph [3]. A measurement graph consists of N input symbol nodes corresponding to $\mathbf{x}$ and M output symbol nodes corresponding to $\mathbf{y}$. Edges of the graph connect each output symbol node y_i to its neighbor set $N(y_i)$ of input symbol nodes determined by non-zero positions in $\mathbf{g}_i$. The measurement graph is usually defined by (edge-oriented) degree distributions for input and output symbol nodes: $\lambda(x) = \sum_i \lambda_i \cdot x^{i-1}$ and $\omega(x) = \sum_i \omega_i \cdot x^{i-1} = \Omega'(x)/\Omega'(1)$ [7]. Two verification-based recovery algorithms called LM1 and LM2 algorithm are proposed in [3]:

LM1: The LM1 operates iteratively over the measurement graph by applying following rules: 1) If $y_i = 0$ then $\forall x_j \in N(y_i) : x_j = 0$; Verify all $x_j : x_j \in N(y_i)$. 2) If $(y_i \neq 0) \wedge (|N(y_i)| = 1)$ then $x_j = y_i$ for the node $x_j \in N(y_i)$; Verify x_j. 3) Remove verified coefficient nodes and their incident edges from the graph; Subtract out verified values from remaining measurements. 4) Repeat until successful signal recovery or makes no progress in two consecutive iterations.

LM2: Besides the above LM1 rules, LM2 adds the additional one: If $(N(y_i) \cap N(y_j) = \{x_k\}) \wedge (y_i = y_j)$ then $x_k = y_i = y_j$ and $\forall x_l \in \{N(y_i) \cup N(y_i) \setminus x_k\} : x_l = 0$; Verify all $x_l \in \{N(y_i) \cup N(y_i) \setminus x_k\}$.

Reconstruction of $\mathbf{x}$ using LM1 algorithm is equivalent to LT decoding [1], thus the AND-OR-tree analysis [7] used to asymptotically evaluate the performance of LT codes can be easily reshaped for LM1 recovery analysis.

Lemma 2.1. *Let $p_l^{(z)}$ and $p_l^{(\bar{z})}$ denote the probabilities that a zero and non zero input symbol, respectively, is not recovered after l iterations of LM1 recovery. Then*

$$p_l^{(\bar{z})} = \lambda\Big(1 - \sum_{i=1}^{d_{max}^{(m)}} \omega_i \cdot \sum_{j=0}^{i-1} \binom{i-1}{j} (\alpha \bar{p}_{l-1}^{(\bar{z})})^j \cdot (\bar{\alpha}\bar{p}_{l-1}^{(z)})^{i-1-j}\Big)$$

$$p_l^{(z)} = \lambda\Big(1 - \sum_{i=1}^{d_{max}^{(m)}} \omega_i \cdot \sum_{j=0}^{i-1} \binom{i-1}{j} (\alpha \bar{p}_{l-1}^{(\bar{z})})^j \cdot \bar{\alpha}^{i-1-j}\Big),$$

where $\alpha = K/N$, we use compact notation $\bar{x} = 1 - x$, and the recursion is initialized at $p_0^{(z)} = p_0^{(\bar{z})} = 1$. Finally, $p_l = \alpha p_l^{(\bar{z})} + \bar{\alpha} p_l^{(z)}$ is the average probability that a signal coefficient is not recovered after l iterations of LM1 recovery.

[1]In rateless coding, the degree distributions are represented as polynomials $\Omega(x) = \sum_i \Omega_i \cdot x^i$, where $\Omega_i = \mathbb{P}(d = i)$.

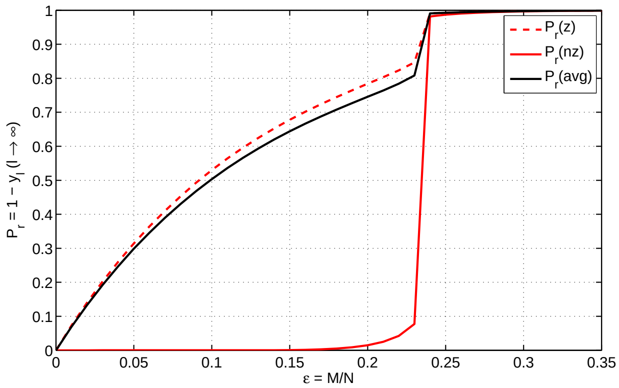

Figure 1: Asymptotic performance of Sudocodes scheme.

In terms of zero input symbol recovery probability $\mathbb{P}_r^{(z)} = 1 - p^{(z)}$, it is easy to show that the optimal degree distribution equals $\Omega(x) = x^{d^*}$, where $d^* \approx (\log \frac{1}{1-\alpha})^{-1}$. Optimal degree distributions for average input symbol recovery probability $\mathbb{P}_r = 1 - p$ can be obtained as described in [1] and [8].

Example 2.1. *Fig. 1 shows asymptotic recovery probabilities (as $N \to \infty$) obtained from* Lemma 1 *for Sudocodes scheme that applies $\Omega(x) = x^{20}$ after LM1 recovery of input message of sparsity-factor $\alpha = 0.05$ ($d^*(\alpha = 0.05) = 20$) for zero and non-zero input symbols and average value. Fig. 3 shows asymptotic recovery probability curves for $\Omega(x) = x^d$ with increasing $d = \{5, 10, 15, 20, 25, 30\}$.*

3 Adaptive Sudocodes

Consider the following modification to the Sudocodes scheme:

Modification 1: After generating output symbol $y_i = 0$, remove input symbols $x_j \in N(y_i)$ from consideration while generating following output symbols.

Modification 2: Due to the first modification, the sparsity-factor α decreases as the number of generated output symbols increase. Thus we continuously update the optimal degree $d^* = d^*(\alpha)$ before each output symbol is generated.

We analyze the Adaptive Sudocodes by following the evolution of measurement matrix with the process of generating output symbols (see Fig. 2). Starting from the initial sparsity-factor $\alpha_0 = K/N$, the process runs through a sequence of increasing α-values $\{\alpha_1, \alpha_2, \ldots\}$ at which the optimal d^*-values decrement. The total of m_i measurements are generated using degree d_i^* during which the sparsity-factor increases from α_i to α_{i+1}. By tracking the parameters (dimensions and degree distributions) of the sequence of measurement matrices $\mathbf{\Phi}_i, i \geq 0$, and using *Lemma 1*, we are able to track the evolution of asymptotic recovery probabilities of LM1 reconstruction. The complete analysis is omitted due to lack of space, however, we note that due to approximations of degree distributions, the resulting analysis is not exact (see example below).

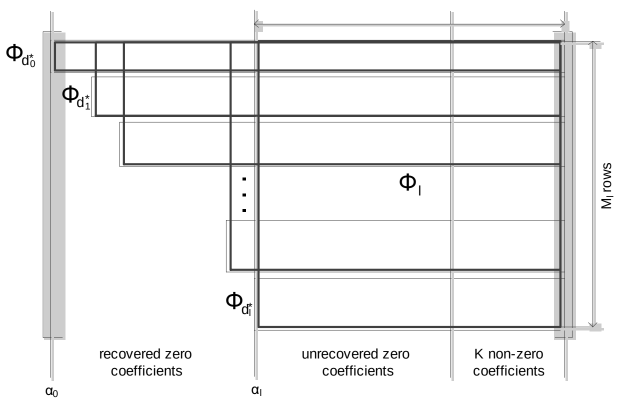

Figure 2: Evolution of measurement matrix in Adaptive Sudocodes.

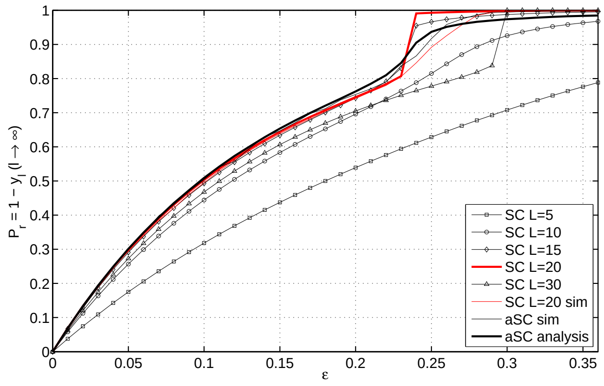

Figure 3: Sudocodes (SC) and adaptive-SC (aSC) performance.

Example 3.1. *Fig. 3 shows asymptotic recovery probability for Adaptive Sudocodes for initial sparsity-factor $\alpha_0 = 0.05$. Simulated results for both Sudocodes and Adaptive Sudocodes scheme are provided for $K = 50$ and $N = 1000$ demonstrating that Adaptive Sudocodes performs better with lower complexity since the optimal degree d^* decreases from $d = 20$ down to $d = 2$ at $M/N = 0.35$. We note that our approximate analysis matches well the simulated performance except that it becomes conservative in the region of large M/N-values.*

4 Comments and Future Work

The rateless codes that apply $\Omega(x)$ with constant average degree $\mu = \Omega'(1)$ are affected by upper bound on recovery probability that scales as $e^{-\mu \frac{M}{N}}$. In LT codes, this is solved by using degree distributions whose average degree scales as $O(\log N)$ [1], for the price of increased encoding/decoding complexity of $O(N \log N)$. More advanced solution called Raptor codes preserves linear encoding/decoding complexity by using high-rate precoding combined with the constant average degree $\Omega(x)$ [8].

The Sudocodes scheme suffers from the same error-floor problem due to a constant-degree $\Omega(x)$ employed. This is solved by employing second phase where non-sparse measurements are combined with matrix inversion to recover small fraction of remaining input symbols [2]. However, instead of "postcoding", it is more instructive to use the Raptor-idea of "precoding" the message by adding small number of additional precoded input symbols followed by Sudocodes scheme with constant-degree $\Omega(x)$. This design is part of our ongoing work.

Finally, extending Adaptive Sudocodes to LM2 recovery has strong potential for further performance improvements and knowledge extraction from previous measurements. Note that in LM1 recovery, zero input symbols are learned only from zero-valued output symbols, while in LM2 recovery, the additional recovery rule provides new possibilities for zero input symbols identification. However, asymptotic analysis of LM2 algorithm seems to be considerably more involved [6]. The design and analysis of Adaptive Sudocodes for LM2 recovery is part of our ongoing work.

References

[1] M. Luby: "LT Codes," Proc. IEEE Symp. Found. of Comp. Science (FOCS), Vancouver, Canada, Nov. 2002.

[2] S. Sarvotham, D. Baron and R. G. Baraniuk: "Sudocodes: Fast Measurement and Reconstruction of Sparse Signals," *IEEE Int'l Symp. Information Theory - ISIT 2006*, pp. 2804–2808, Seattle, USA, July 2006.

[3] M. Luby and M. Mitzenmacher:"Verification-based decoding for packet-based low-density parity-check codes," *IEEE Trans. Information Theory*, Vol 51, No 1, pp. 120–127, January 2005.

[4] D. Baron, S., Sarvotham, R. G. Baraniuk: "Bayesian compressive sensing via belief propagation," *IEEE Trans. Signal Processing,* Vol. 58, No. 1, pp. 269–280, January 2010.

[5] F. Zhang and H. D. Pfister: "Verification decoding of high-rate LDPC codes with applications in compressed sensing," *IEEE Trans. Information Theory,* Vol. 58, No. 8, pp. 5042–5058, August 2012.

[6] Y. Eftekhari, A. Heidarzadeh, A. H. Banihashemi and I. Lambadaris: "Density Evolution Analysis of Node-Based Verification-Based Algorithms in Compressed Sensing," *IEEE Trans. Information Theory,* Vol. 58, No. 10, pp. 6616–6645, October 2012.

[7] M. Luby, M. Mitzenmacher and A. Shokrollahi: "Analysis of random processes via and-or tree evaluation," *ACM Symp. on Discrete algorithms - SODA 1998,* pp. 364–373, San Francisco, USA, Jan. 1998.

[8] A. Shokrollahi: "Raptor codes," *IEEE Trans. Info. Theory,* Vol. 52, No. 6, pp. 2551–2567, June 2006.