

List message passing algorithm for noiseless compressed sensing

Francisco Ramirez-Javega, Meritxell Lamarca
 Dept. of Signal Theory and Communications. (TSC)
 Universitat Politècnica de Catalunya (UPC)
 Email: javega@tsc.upc.edu, meritxell.lamarca@upc.edu

Abstract—We propose a verification-based algorithm for noiseless Compressed Sensing that reconstructs the original signal operating on a sparse graph. The proposed scheme has affordable computational complexity and its performance is significantly better than previous verification-based algorithms and similar to AMP-based algorithms. We also show that the performance of a noiseless compressed sensing scheme when verification-based algorithms and a sparse matrix is employed to reconstruct the original signal can be upper bounded by the performance of a LDPC code employing the same parity matrix when correcting a codeword transmitted through a BEC.

I. INTRODUCTION

The noiseless Compressed Sensing (CS) problem considers the estimation of an unknown and sparse vector $\mathbf{x} \in \mathbb{R}^N$ from a vector of linear observations $\mathbf{y} \in \mathbb{R}^M$, $M < N$, $\mathbf{y} = \mathbf{A}\mathbf{x}$ where $\mathbf{A} \in \mathbb{R}^{M \times N}$ is a fixed randomly generated matrix known as the measurement matrix and only a small number K , $K \ll N$, of elements of \mathbf{x} are non-zero, so the sparsity ratio is $\gamma = K/N$. The set containing the positions of these elements is known as the *support set*, defined as $\mathcal{S} \triangleq \{i \in \{1, \dots, N\} : x_i \neq 0\}$, with $|\mathcal{S}| = K$. The solution to this system of equations is known to be given by the vector $\hat{\mathbf{x}}_0$ that minimizes $\|\mathbf{x}\|_0$ (l_0 -norm) subject to $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}_0$, which is a non-convex optimization problem. The authors in [1], [2] established that the vector $\hat{\mathbf{x}}_1$ with minimum l_1 -norm subject to $\mathbf{y} = \mathbf{A}\hat{\mathbf{x}}_1$ coincides with $\hat{\mathbf{x}}_0$ whenever the measurement matrix satisfies the RIP conditions [1], [3] and guarantees that all sequences with at most k non-null entries can be reconstructed when the number of samples m is large enough [4].

Recently, within this framework, the authors in [5] showed that in the N asymptotic regime a matrix with rate $r > \gamma$ suffices to reconstruct any real-valued signal with an sparsity ratio γ in the almost lossless compression setting, i.e. allowing an arbitrarily low fraction of sequences with reconstruction errors. Simultaneously, in the literature, some algorithms based on message propagation over graph representations of measurement matrices were shown to outperform the theoretical limits of the classical compressed sensing approach [4], e.g. the scheme proposed in [6] based on the approximate message passing (AMP) [7], and the verification-based algorithms [8], [9].

Recently, the authors in [10] showed that capacity achieving coding schemes for the *Binary Erasure Channel* (BEC) are also optimal for the noiseless compressed sensing problem and showed that polar codes/decoding algorithms [11] can be

used to reconstruct a sparse signal with a number of samples $M > K$.

In this paper we do two contributions for the almost lossless compressed sensing problem. First, we propose to exploit the analogy with the BEC from a point of view different than that described in [10]. We show that the performance of the verification-based algorithm for a given sparse graph (matrix) can be upper bounded by the performance an LDPC code in the BEC. Second, we introduce a verification-based algorithm for noiseless compressed sensing based on list message passing [12] that outperforms the previous verification-based algorithms and has a performance similar to one of the Estimate-Maximize AMP-based algorithm with Gaussian-Bernoulli a priori (EM-GB-AMP) [6].

The paper is structured as follows. In section II we introduce the almost lossless compressed sensing signal model and the notation employed in sparse graphs. Next, in section III, we briefly summarize verification-based algorithms and show that the performance of LDPC in the BEC is equal to the one of a verification-based algorithm for compressed sensing when the support set is known. Afterwards, in section IV, we introduce the list message passing for CS (CS-LMP). Finally we illustrate the performance of CS-LMP with some simulation results and we draw some conclusions.

II. PREVIOUS CONCEPTS

A. Almost lossless noiseless compressed sensing setting

In the noiseless compressed sensing set up we assume that the source vector \mathbf{x} consists of N i.i.d. copies of a real valued random variable distributed according to a mixture of discrete and continuous distributions, as follows

$$P_x = (1 - \gamma)\delta_0 + \gamma P_C \quad (1)$$

where δ_0 denotes a unit mass at zero, P_C is a continuous distribution from which the non-zero entries are drawn, and $0 \leq \gamma \leq 1$ parametrizes the sparsity of the signal vector \mathbf{x} . Note that this vector has approximately γN non-null coefficients. The measurement vector $\mathbf{y} \in \mathbb{R}^M$ is generated as the product of \mathbf{x} and the measurement matrix $\mathbf{A} \in \mathbb{R}^{M \times N}$, so the compression rate is $r = M/N$.

This work was supported by Ministerio de Economía y Competitividad¹ of the Spanish Government, and European Regional Development Funds (TEC2013-47020-C2-2-R COMPASS) and by the Catalan Government (2014 SGR 60 AGAUR)

At the decoder side, a reconstruction or decoding algorithm performs an attempt to reconstruct the original signal vector $\hat{\mathbf{x}}$ under a given criteria of fidelity.

In general this problem reduces to the design of measurement matrix and a decoding algorithm with minimum compression rate r for the given criteria of fidelity. It was shown in [5] that it is possible to reconstruct almost all signal vectors with sparsity γ whenever the compression rate is $r \geq \gamma + o(1)$, in the N asymptotic limit.

B. Graph representation of a matrix

A sparse measurement matrix \mathbf{A} can be represented by a bipartite graph \mathcal{A} having two types of nodes: N variable nodes (one per entry of vector \mathbf{x}) and M check nodes (one per entry of vector \mathbf{y}) where the variable (check) nodes are strictly connected to d_v check (d_c variable) nodes. These connections (edges) are indicated by the non-null entries of the measurement matrix \mathbf{A} . We denote as the neighborhood of the j -th check node \mathcal{I}_j to those variable nodes that contribute to generated the y_j measurement. Similarly, we denote as the neighborhood of a variable node to those check nodes that are connected to it by an edge.

Bipartite graphs belong to ensembles that are fundamentally characterized by the polynomials named check and variable node degree distribution, $R(d)$ and $L(d)$ respectively, which define the fraction of check and variable nodes that have d edges in the graph. Similarly, we can define the fraction of edges connected to check or variable nodes of degree d as $\lambda(d)$ and $\rho(d)$, respectively.

We say that a graph is regular whenever all the variable nodes have the same degree d_v and all check nodes have the same degree d_c . It may also happen that the graphs are right (left) regular whenever all check (variable) nodes have the same degree d_c (d_v) and the variable (check) node degree distribution is irregular.

III. VERIFICATION-BASED ALGORITHMS

The Verification-based Algorithms (VA) are local message passing strategies over a graph representation of a measurement matrix. When employed in the CS problem their main features are that (i) variable nodes can infer their values with infinite reliability in some situations, (ii) they can be implemented with $O(N)$ computational complexity and (iii) their performance does not depend of the values of the matrix coefficients whenever the corresponding graph is free of length-four cycles [9], [13].

When applied to the noiseless CS setup, the VA exploit that the graph is sparse and the probability density function of the components of vector \mathbf{x} is a continuous function with a unit mass concentration at zero, as stated before. Due to the latter fact the non-null coefficients of the signal vector \mathbf{x} must be different and furthermore the probability that a sum of a given set of non-null coefficient is equal to a non-null coefficient of the signal vector is also equal to zero.

In this section, first we outline the node-based verification algorithm (NB-VA) to use it as introduction to the *list message*

passing algorithm presented in the following section. Afterwards, we revisit the Genie algorithm proposed in [9] that can be employed as a performance benchmark for the verification-based algorithm and we propose an alternative analysis for the genie algorithm that eases the one previously presented by the authors in [14].

A. Node-based verification algorithm

The Node-based Verification algorithm (NB-VA) [14] gathers all the features of the family of verification algorithms. In the NB-VA check nodes (CN) and variable nodes (VN) exchange messages with two fields $\{state, x\}$ where $state='v'$ indicates that we know for sure that the value of the associated variable node is x (i.e. the variable node is *verified*) whereas $state='nv'$ informs that x is solely an estimate of the variable node value (i.e. the variable node is still *not verified*).

CNs and VNs are activated alternatively. During each VN activation round the variable nodes try to infer (verify) the values of the components of vector \mathbf{x} that they represent, based on all the messages received through their edges, checking whether (VN.i) any of the received messages is verified or (VN.ii) at least two of the received message have the same value. The VNs that accomplish any of these two conditions swap their *state* to *verified* and update their value. Afterwards the CNs also try to infer the value of the VNs in their neighborhoods testing whether (CN.i) the sum of all the verified messages is equal to its measurement (in this case the non-verified VNs in their neighborhood are equal to zero) or (CN.ii) only one VN in its neighborhood is non-verified. For a detailed explanation of the implementation of the algorithm see [8] or [9].

B. Genie Verification Algorithm

The authors of [9] proposed the Genie verification algorithm as a benchmark for the verification algorithms. This benchmark is obtained assuming the support set to be known, i.e. initially the variable nodes that do not belong to the support set are verified. They also derived a set of equations that predicted accurately the evolution of the Genie algorithm on the N -asymptotic regime. However, the resulting density evolution analysis was quite complex.

The complexity of the analysis of the Genie can be reduced if we notice the similarities between the resulting setup (a fraction γ of the variable nodes is non-verified and the remaining variable nodes $1 - \gamma$ is verified) and a LDPC error correcting code correcting a codeword transmitted through a *binary erasure channel* with erasure probability γ ; this resemblance is straightforward if we label the non-verified variable nodes as erasures and, furthermore, check and variable nodes perform the same operations in VA than in for the BEC (see [15] for further details on BP decoding for the BEC). For VA, variable nodes perform VN.i and check nodes the rule CN.ii, as shown in the previous section. Hence, we can employ the density evolution equations for the BEC given in [15] to find out the γ^* threshold (i.e. the maximum γ so as the bit erasure probability is arbitrarily low) of a graph characterized

by the edge degree profiles $\lambda(x)$ and a $\rho(x)$ has or not a fixed point computing iteratively

$$x^{(t)} = \gamma \lambda \left(1 - \rho^* \left(1 - x^{(t-1)} \right) \right) \quad (2)$$

where $x^{(t)}$ is the probability that a check nodes transmit a non-verified message at the t -th iteration when initially all incoming messages to the variable nodes of graph \mathcal{A} are non-verified message, $x^{(0)} = 1$. It can be shown that (2) provides the same thresholds for the genie algorithm given in [9].

IV. LIST MESSAGE PASSING FOR COMPRESSED SENSING

Recently, the authors of [12] proposed a new verification based-algorithm for q -ary channel coding, named *List Message Passing* (LMP). The main differences of LMP with respect to verification-based algorithms are that, (i) whenever the field 'state' of a message is 'non-verified' the field value is a *list of estimates* of the value of the variable node and that (ii) all the nodes exchange extrinsic information. The drawback of this channel coding algorithm is that the size of the list of estimates grows unbounded with the iteration number and the check node degree.

Clearly, the complexity of direct application of LMP to CS is non-affordable because all nodes are originally unknown and because the average check node degree is proportional to γ^{-1} . With this in mind, in this section, we present an adaptation of the list message passing algorithm for compressed sensing (CS-LMP) that overcomes this problem and aims at approaching the genie algorithm performance with affordable computational complexity.

A. List message passing for compressed sensing

In this message passing strategy the nodes exchange messages having the following form: $\{state, x\}$ where *state* is either 'nv' or 'v' (as in the previous section) and x is either the variable value (when 'v') or a list of its estimates of length $d_v - 1$ (when 'nv').

The check nodes divide their edges into the sets of verified and non-verified edges, \mathcal{I}_v and \mathcal{I}_{nv} , and subtract the contribution of the verified variable nodes from their measurement to obtain y' as follows

$$y'_j = y_j - \sum_{\forall i \in \mathcal{I}_v} a_{j,i} x_i \quad (3)$$

Now, each check node performs a search in order to check whether any combination of estimates received from the variable nodes indexed in \mathcal{I}_{nv} sum up to y' . Apparently, check nodes must perform up to $d_v^{d_c}$ searches to ensure that no combination sums up to y' . However, this search can be reduced as follows.

In order to reduce the complexity of this search we can exploit the way how the verification algorithms exploit the sparsity of the graph. The non-null entries of the signal \mathbf{x} are activated with probability γ , so the edges of the graph are connected to variable nodes that represent non-null entries of the signal vector also are and thus the edges are *active* also with probability γ . Hence the number of active edges per

Algorithm 1 Check node update rules

for $j = 1, \dots, M$ y_j and their \mathcal{I}_j
Initialize: y', \mathbf{X}_j and \mathcal{I}_{nv}
exec: $[\text{successful}, \hat{\mathbf{x}}'_j] = \text{search}(y', \mathbf{X}_j, \mathcal{I}_{nv})$
if: successful search,
update: $\forall i \in \mathcal{I}_{nv}, \mu_{j \rightarrow i} = \{v, \hat{\mathbf{x}}'_i\}$
else:
update: $\forall i \in \mathcal{I}_{nv}, \mu_{j \rightarrow i} = \{nv, y'\}$
end:
endfor

check node follows a binomial distribution in the N asymptotic regime, as shown below

$$p(|\mathcal{I} \cap \mathcal{S}| = i) = \sum_{\forall d_c} R(d_c) \binom{d_c}{i} \gamma^i (1 - \gamma^{d_c - i}) \quad (4)$$

where we assumed that the check node degree distribution is irregular, (for check regular codes there would be only one term). Obviously a given fraction of the check nodes is generated by the contribution of a single non-null entry of the signal vector (with probability $p(|\mathcal{I} \cap \mathcal{S}| = 1)$). So, if this fraction is high enough the probability that some variable nodes receive the same estimates from different check nodes is higher and hence the variable nodes can decide that the common is their real value (if the graph is free of length four cycles or whenever the weights of the edges are i.i.d. according to some continuous distribution the values y' in equation 3 will be all different).

Thus, VA based codes require a sufficiently large fraction of measurements equal to a single variable node $p(|\mathcal{I} \cap \mathcal{S}| = 1)$ in order to enable the verification process at the variable nodes or to allow them to forward a list of estimates to the check nodes in their neighborhood where the real value of the variable node is between the estimates.

Let us focus on a right regular code with check node of degree d_c . In this case, in average the neighborhood of a check node contains $d_c \gamma$ non-null variable nodes. Thus, as the $p(|\mathcal{I} \cap \mathcal{S}| = 1)$ must be sufficiently high to enable the verification process, there are usually several variable nodes that represent *zeros* contributing to the generation of a given measurement (the mass concentration of the binomial distribution in equation (4) must be concentrated close to $i = 1$). Due to this the check nodes can initially perform the searches assuming that almost all the variable nodes in their neighborhood are zero, hoping to find a combination of estimates that sums up to the y' in equation (3).

In general, for irregular check node degree distributions, there are several mass concentrations but it also holds that the one belonging to the check nodes with lowest degree is concentrated close to $i = 1$, and the remaining mass concentrations are situated further to the right, hence is also highly unlikely to find the proper combination of estimates at the check nodes with larger number of edges.

Algorithm 1 summarizes the check node operations. Basi-

Algorithm 2 Variable node update rules

for each non-verified variable node i applies:

- if:** The variable node receives a *verified* message
then: propagates this message through the remaining edges $\{v', y'\}$.
- elseif** The variable node receives at least two *non-verified* messages with the same estimate y' ,
then propagates $\{v', y'\}$ to all the nodes connected to them.
- else:** propagates to each check node j in its neighborhood: $\mu_{i \rightarrow j} = \{nv', y'_j\}$, where y'_j is the list of estimates received from all the other check nodes.

endfor

cally, the j -check node removes from its measurement y_j the contribution of the verified variable nodes in its neighborhood as in equation (3). Then executes an exhaustive search, testing whether any combination sums up to y' . Note that \mathbf{X}_j represents the list of estimates received from the non-verified variable nodes that are indexed in \mathcal{I}_{nv} .¹ After performing the search, if the check nodes found a combination of estimates that sum up to y' , they send to all the variable nodes indexed in \mathcal{I}_{nv} a verified message with the proper value. Otherwise they send a non-verified message with an estimate y' .

Algorithm 2 summarizes variable nodes operation. When they receive a verified message or at least two non-verified messages with the same estimate, their state swaps to verified with the value indicated. Otherwise they keep in a non-verified state and append to the non-verified flag a list with all the estimates received though the remaining edges.

B. Computational complexity of list message passing for CS

In this section we evaluate the computational complexity of the proposed algorithm for a fixed sparsity ratio γ and rate r under the assumptions that the length N is large enough so that the signal vector has approximately γN nonzero entries. Furthermore, theoretically the number of samples M to almost perfectly reconstruct a γ -sparse signal of length N vector is $M = \gamma N + o(N)$, hence M is $O(\gamma N)$. We consider that comparisons, products and sums have the same complexity.

The computational complexity of the activation of a variable node is dominated by the search of at least two estimates with equal value and it requires $\binom{d_v}{2}$ operations. So the activation of all the variable nodes is $O(d_v^2 N)$.

The list of the messages that the variable and check nodes send to the check and variable nodes has at most length $d_v - 1$ and 1, respectively. These messages have to be multiplied/divided by the weight of the edges, so the complexity of this operation is $O(d_v^2 N)$, as there are $d_v N$ edges in the graph.

¹This list has in average $|\mathcal{I}_{nv}| \sum_{\forall d_v} L(d_v) d_v$ values. The wise way to perform the search is first assume that all but one input messages are zero, i.e. perform $(d_v - 1) \binom{1 + \mathcal{I}_{nv}}{1}$ comparisons, then assume that two components are non null, i.e. perform $(d_v - 1)^2 \binom{1 + \mathcal{I}_{nv}}{2}$ comparisons, and so on.

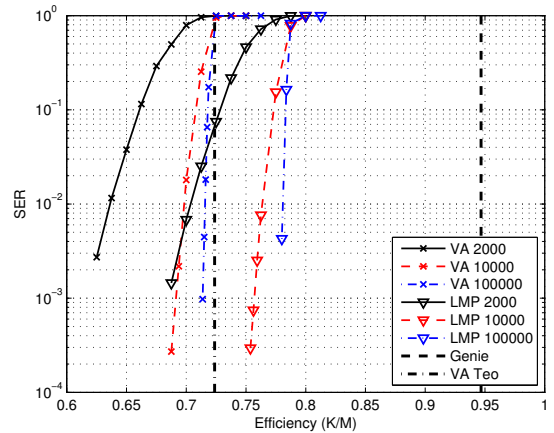


Figure 1. Performance comparison of CS-LMP (LMP) and NB-VA (VA) and different N compared to the thresholds of the NB-VA and the Genie algorithm, for the same graph.

Each check node performs at most d_v^{dc} comparisons. Hence the activation of all the check nodes in one iteration has a computational complexity $O(\gamma N d_v^{dc})$.

Thus the overall computational complexity of the first iteration is $O(N(\gamma d_v^{dc} + 2d_v^2))$.

V. RESULTS

Figure 1 depicts the performance of the NB-VA and CS-LMP algorithm for different sequence lengths compared with the thresholds of the NB-VA and the BEC benchmark. The horizontal axis represents the sampling efficiency (K/M) and the vertical axis the sequence error rate (SER), $p(\mathbf{x} \neq \hat{\mathbf{x}})$. The plots of the performance of the NB-VA and CS-LMP were obtained by Monte Carlo simulations. Each simulation was stopped when 200, 200 or 20 erroneous reconstructions were obtained, for $N = 2000, 10000$ and 100000 , respectively. A graph degree profile optimized in [9] for the NB-VA of rate 0.8 was employed, it included two degrees for check and variable nodes and the matrix coefficients were either 0 or 1. The non-null entries of the \mathbf{x} are i.i.d. according to a standard distribution. Note that verification algorithms declare that a sequence is perfectly reconstructed whenever all variable nodes are verified and thus, the values of the entries of estimate vector are equal to the ones of the original sequence.

Figure 1 shows that both the performance of the NB-VA and CS-LMP increases as N increases, as expected. It also shows that the performance of the NB-VA approaches theoretical limit obtained by density evolution in [9] as N increases. The figure also shows that the CS-LMP clearly outperforms the NB-VA for all N , being the gain of efficiency approximately of 0.07 irrespective of N . In spite of this gain, the gap to BEC benchmark is still large for this ensemble.

Figure 2 compares the performance of the CS-LMP with the one of the estimate-maximize Gaussian-Bernoulli Approximate Message Passing (EM-GB-AMP) algorithm [6] and the BEC benchmark, for different block lengths. The entries of the measurement matrix of the EM-GB-AMP are i.i.d

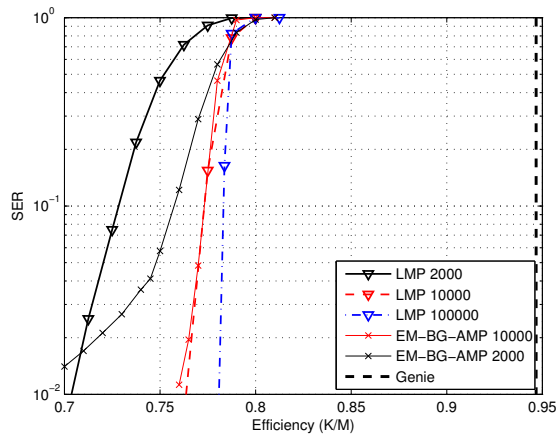


Figure 2. Performance comparison of CS-LMP (LMP) vs. EM-GB-AMP for different N compared to the threshold of the Genie algorithm.

according to a standard distribution. The measurement matrix of the CS-LMP is the same as before. Note that both matrices have the same rate. As before, the horizontal axis represents the sampling efficiency and the vertical axis the SER. The performance plots of the EM-GB-AMP and CS-LMP were obtained by Monte Carlo simulations and each simulation was stopped once 200 erroneous reconstructions were obtained. As before, for the LMP a sequence is declared perfectly reconstructed whenever all variable nodes are verified. For the EM-GB-AMP, a sequence is declared free of errors whenever the normalized minimum square error is $\frac{\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2}{\|\mathbf{x}\|_2^2} < 10^{-4}$.

Figure 2 shows that for the shortest block length, $N = 2000, 10000$, initially the EM-GB-AMP algorithm outperforms the CS-LMP algorithm but as the sparsity of the signal decreases, the SER of the EM-GB-AMP algorithm shows an error floor whereas the CS-LMP SER keeps decreasing and eventually the CS-LMP outperforms the EM-GB-AMP.

For the largest block length, both algorithms (with their own different measurement matrices) have the same performance. Notice that for this block length, the plot of the EM-GB-AMP algorithm also seems to depict an error floor.

VI. CONCLUSIONS

We showed the list message passing for compressed sensing outperforms the best verification-based algorithm know up to the date, irrespectively of the block length N . We also showed that LMP for CS has a similar performance of the EM-GB-AMP algorithm for block lengths of 10^4 . The losses of the LMP for the shorter block lengths can be attributed to the inherent difficulties of generating sparse graphs with a large fraction of check and variable nodes with high degrees. In case of the EM-GB-AMP algorithm the presence of the error floor can be attributed to deficiencies in the generation of the measurement matrix. It is remarkable that even for average

block lengths ($\sim 10^4$) the AMP-based algorithm presents error floor for low sequence error probabilities. The reader must note that the performance of LMP-based algorithms can be further increased by using sparse matrices designed for them.

We also showed that the performance of a noiseless compressed sensing scheme using a sparse graph $\mathcal{A} \in \Lambda(\lambda, \rho)$ in the N asymptotic regime can be upper bounded by the performance of a LDPC code of the same ensemble correcting a codeword transmitted through a BEC as the latter problem is equivalent to the former when the support set of the sparse signal is known and a verification algorithm is employed as a reconstruction algorithm.

REFERENCES

- [1] D.L. Donoho, "Compressed sensing," *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, April 2006.
- [2] E.J. Candes and T. Tao, "Near-optimal signal recovery from random projections: Universal encoding strategies?," *IEEE Trans. Inform. Theory*, vol. 52, no. 12, pp. 5406–5425, December 2006.
- [3] R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," *Constructive Approximation*, vol. 28, pp. 253–263, 2008.
- [4] D. L. Donoho and J. Tanner, "Counting faces of randomly-projected polytopes when the projection radically lowers dimension," *J. of the American Mathematical Soc.*, pp. 1–53, 2009.
- [5] W. Yihong and S. Verdu, "Renyi information dimension: Fundamental limits of almost lossless analog compression," *IEEE Trans. Inform. Theory*, vol. 56, no. 8, pp. 3721–3748, August 2010.
- [6] J. Vila and P. Schniter, "Expectation-maximization bernoulli-gaussian approximate message passing," in *Signals, Systems and Computers (ASILOMAR), 2011 Conference Record of the Forty Fifth Asilomar Conference on*, nov. 2011, pp. 799–803.
- [7] David L. Donoho, Arian Maleki, and Andrea Montanari, "Message-passing algorithms for compressed sensing," *Proceedings of the National Academy of Sciences*, vol. 106, no. 45, pp. 18914–18919, 2009.
- [8] F. Ramirez-Javega, M. Lamarca, and J. Villares, "Binary graphs and message passing strategies for compressed sensing in the noiseless setting," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, july 2012, pp. 1867–1871.
- [9] Y. Eftekhari, A.H. Banihashemi, and I. Lambadaris, "Analysis and design of irregular graphs for node-based verification-based recovery algorithms in compressed sensing," in *Information Theory Proceedings (ISIT), 2012 IEEE International Symposium on*, july 2012, pp. 1872–1876.
- [10] L. Li and I. Kang, "A Structured Construction of Optimal Measurement Matrix for Noiseless Sparse Recovery, Utilizing Channel Capacity and Polarization of Analog Transmission," *ArXiv e-prints*, Dec. 2012.
- [11] E. Arıkan, "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," *Information Theory, IEEE Transactions on*, vol. 55, no. 7, pp. 3051–3073, july 2009.
- [12] Fan Zhang and H.D. Pfister, "List-message passing achieves capacity on the q -ary symmetric channel for large q ," in *Global Telecommunications Conference, 2007. GLOBECOM '07. IEEE*, nov. 2007, pp. 283–287.
- [13] Fan Zhang and H.D. Pfister, "Verification decoding of high-rate ldpc codes with applications in compressed sensing," *Information Theory, IEEE Transactions on*, vol. 58, no. 8, pp. 5042–5058, aug. 2012.
- [14] Y. Eftekhari, A. Heidarzadeh, A.H. Banihashemi, and I. Lambadaris, "Density evolution analysis of node-based verification-based algorithms in compressed sensing," *Information Theory, IEEE Transactions on*, vol. 58, no. 10, pp. 6616–6645, oct. 2012.
- [15] Tom Richardson and Ruediger Urbanke, *Modern Coding Theory*, Cambridge University Press, New York, NY, USA, 2008.