Tree Structure Based Analyses on Compressive Sensing for Binary Sparse Sources

Jingjing Fu[1,2], Zhouchen Lin[2], Bing Zeng[1], Feng Wu[2]

1. Hong Kong University of Science and Technology, Kowloon, Hong Kong
2. Microsoft Research Asia, Beijing, China

Email: *jjfu@ust.hk, zhoulin@microsoft.com, eezeng@ust.hk, fengwu@microsoft.com*

Abstract

This paper proposes a new approach to theoretically analyze compressive sensing directly from the randomly sampling matrix $\boldsymbol{\Phi}$ *instead of a certain recovery algorithm. For simplifying our analyses, we assume both input source and random sampling matrix as binary. Taking anyone of source bits, we can constitute a tree by parsing the randomly sampling matrix, where the selected source bit as the root. In the rest of tree, measurement nodes and source nodes are connected alternatively according to* $\boldsymbol{\Phi}$*. With the tree, we can formulate the probability if one source bit can be recovered from randomly sampling measurements. The further analyses upon the tree structure reveal the relation between the un-recovery probability with random measurements and the un-recovery probability with source sparsity. The conditions of successful recovery are proven on the parameter S-M plane. Then the results of the tree structure based analyses are compared with the actual recovery process.*

I. INTRODUCTION

In the recent years, a new theory on randomly sampling is proposed, called compressive sensing theory. It asserts that the sparse signals can be recovered from a small amount of measurements under incoherent sampling [1]-[2]. To specify, let $\boldsymbol{x} = \{x_1, x_2, \ldots, x_N\} \in \mathbb{R}^N$ with $\boldsymbol{x} = \boldsymbol{\psi u}$, where $\boldsymbol{u}$ has only K non-zero elements and $K \ll N$. $\boldsymbol{x}$ is called as K-sparse with respect to the transform $\boldsymbol{\psi}$. Random measurements $\boldsymbol{y} = \{y_1, y_2, \ldots, y_M\} \in \mathbb{R}^M$ are generated by

$$\boldsymbol{y} = \boldsymbol{\Phi x}, \quad \text{where } \boldsymbol{\Phi} \in \mathbb{R}^{M \times N}. \tag{1}$$

Φ is a randomly sampling matrix, and the number of measurement $M \ll N$. It is an ill-posed problem recovering x from y. But, the compressive sensing theory testifies that K-sparse signal x can be recovered by $O(Klog(N/K))$ measurements as long as $\boldsymbol{\Phi}$ satisfies the restricted isometry property (RIP), and the recovery can be achieved with probability close to one by solving the following convex optimization

$$\hat{\boldsymbol{u}} = argmin\|\boldsymbol{u}\|_1, \text{ subject to } \boldsymbol{y} = \boldsymbol{\Phi\psi u}. \tag{2}$$

$\|\boldsymbol{u}\|_1$ denotes the $l1$-norm of the vector $\boldsymbol{u}$.

The l1-norm minimization, namely, basis pursuit (BP) [3], is solved by linear programming (LP), whose computational complexity is impractically high for large N. Accordingly, many recovery algorithms are proposed aiming to reduce the measurements needed for recovery and enhance the computational efficiency, such as orthogonal marching pursuit (OMP) [4], matching pursuit (MP), chaining pursuit (CP) [5], and so on. Most of these works realize the recovery by iteratively adjusting the vector u in terms of all the measurements. Some recent works treat the randomly sampling $\boldsymbol{\Phi}$ as one kind of channel codes and then realize the recovery like channel decoding. The work categorized into this line includes Reed-Solomon (RS) code [6], Low-Density-Parity-Check (LDPC) code [7], Sudocode [8], expander graph [9][10] and bipartite graph [11].

Although all these algorithms are dedicated to solving the same optimization problem, different models and tools are applied. Thus the final recovery inevitably relies on the properties of algorithms as well as the desired computational time and storage. However, no matter what recovery algorithm is taken, the core problem keeps the same, namely, the final recovery should be only determined by the sparsity of x and $\boldsymbol{\Phi}$. Therefore, one question arises here. For a given x, can we directly analyze $\boldsymbol{\Phi}$ to judge whether the source can be recovered without really solving Eq. (1)?

To achieve it, this paper proposes a tree-based approach to analyze the compressive sensing. Taking any single source element in $\boldsymbol{x}$ as the root, a tree structure can be built by parsing $\boldsymbol{\Phi}$. The source elements and measurements, which directly or indirectly connect to the source element, constitute the nodes of the tree. The branches of the tree are the connections radiating from source elements to measurements and then from measurements to source elements. In the tree structure, source element nodes increase the uncertainty whereas the measurement nodes provide information to decrease the uncertainty. Therefore, with the tree, we can formulate the probability that the source element at the root cannot be recovered. From the formulation, we can further derive the necessary conditions for successful recovery.

The proposed approach not only enables us to analyze the problem (1) without recovery but also provides a way to design better $\boldsymbol{\Phi}$. Since the tree structure completely depends on the randomly sampling matrix $\boldsymbol{\Phi}$, the different designs of the matrix can be evaluated by the analysis of un-recovery probability upon the corresponding tree structures. Therefore, we can design the randomly sampling matrix of compressive sensing to achieve more efficient recovery. But in this paper, we will focus on the first aspect.

The rest of the article is arranged as follows. In section 2, we discuss how to build a tree for a given source element by parsing $\boldsymbol{\Phi}$. In section 3, the un-recovery probability is deduced based on the tree structure and the necessary conditions for successful recovery are derived and the recovery performance of compressive sensing is analyzed in terms of the un-recovery probability. Section 4 compares theoretical analysis results with the actual recovery. Finally, Section 5 concludes this paper.

II. GENERATION OF TREE STURCTURE

In the compressive sensing, the number of measurements is far fewer than the size of the source, which causes the optimization problem (1), which can be written in the following form

$$\begin{bmatrix} y_1 \\ \cdots \\ y_j \\ \cdots \\ y_M \end{bmatrix} = \begin{bmatrix} \Phi_{11} & \cdots & \Phi_{1i} & \Phi_{1(i+1)} & \cdots & \Phi_{1N} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Phi_{j1} & \cdots & \Phi_{ji} & \Phi_{j(i+1)} & \cdots & \Phi_{jN} \\ \cdots & \cdots & \cdots & \cdots & \cdots & \cdots \\ \Phi_{M1} & \cdots & \Phi_{Mi} & \Phi_{M(i+1)} & \cdots & \Phi_{MN} \end{bmatrix} \begin{bmatrix} x_1 \\ \cdots \\ x_i \\ x_{i+1} \\ \cdots \\ x_N \end{bmatrix} \tag{3}$$

The recovery of the source completely depends on relations embedded in the random sampling matrix $\boldsymbol{\Phi}$. Taking the i-th element of $\boldsymbol{x}$ as an example, it is contained in y_j with weight Φ_{ji}, for $j = 1,2,\dots,M$. These weights compose the i-th column vector (circled by dashed line) of the random sampling matrix $\boldsymbol{\Phi}$. In order to describe the recovery of the source element x_i, we constitute a tree-like structure by taking x_i as the root and the measurements $\{y_j\}$ as the son nodes. The relation between x_i and $\{y_j\}$ are represented by branches radiating from the root to the measurement nodes.

Considering any measurement y_j, it is the weighted sum of the source $\boldsymbol{x}$, where the weights originate from the j-th row vector (circled by dashed line) of $\boldsymbol{\Phi}$. This relation is represented by a set of branches starting from the measurement node y_j and ending at element nodes $\{x_j\}$ in the third layer. Following this way, the tree structure of x_i can be spanned layer by layer. Figure 1 shows a tree structure, where small rectangles indicate the source element nodes and small circles are the measurement nodes.

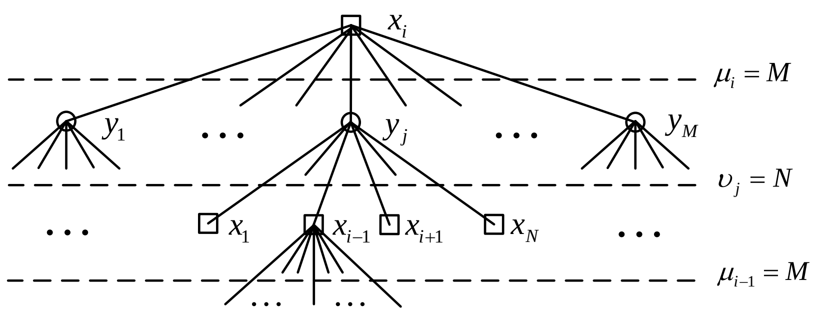

Figure 1: Illustration of the tree structure

For clearly describing the tree structure, we define the degree of a node equivalent to the branches connecting to it. Then let μ_i denote the degree of the element node x_i, and υ_j denote the degree of the measurement node y_i. Notice that if Φ_{ji} is zero, the corresponding branch connecting between x_i and y_j is omitted from the tree structure. But in the common sense, the entries of random sampling matrix $\boldsymbol{\Phi}$ are non-zero, like Gaussian random matrix, and thus none of the branches can be omitted from the tree structure. Therefore, $\mu_i = M$ and $\upsilon_j = N$, for any i and j. Here N is the size of source and M is the number of measurements.

Although the tree structure is built up from $\boldsymbol{\Phi}$, its bulky size and complex numeric relations greatly complicate the analysis on the recovery of compressive sensing. To simplify the tree structure, we consider $\boldsymbol{x}$ as a binary sparse source with independent and identical distribution P_x, where the transform $\boldsymbol{\psi}$ is omitted as an identity matrix. We set the probability $P_x\{x_i = 1\} = p_w$ and $P_x\{x_i = 0\} = p_b$. As $\boldsymbol{x}$ is a sparse source, p_w should be much smaller than p_b. For a binary source, it is reasonable to assume the random sampling matrix $\boldsymbol{\Phi}$ as binary too, namely, the element in $\boldsymbol{\Phi}$ is one or zero.

We assume one measurement always randomly samples S elements in $\boldsymbol{x}$, that is, only S entries are non-zero in every row vector of $\boldsymbol{\Phi}$. To satisfy the restricted isometry property (RIP), S is selected far smaller than the source size N. Then the considerable entries in $\boldsymbol{\Phi}$ are zeros, so that the corresponding branches are removed from the tree structure. The tree structure is dramatically simplified. An example is given in Figure 2. Since in the i-th column vector of $\boldsymbol{\Phi}$, only three entries are ones at position j_1, $j_2$and j_3, then the degree of node x_i reduces from M to 3. And in the j_2-th row vector of $\boldsymbol{\Phi}$, only four entries are ones at position i_1, i_2, i_3 and i_4. Then the degree of node y_{j_2} reduces from N to 4.

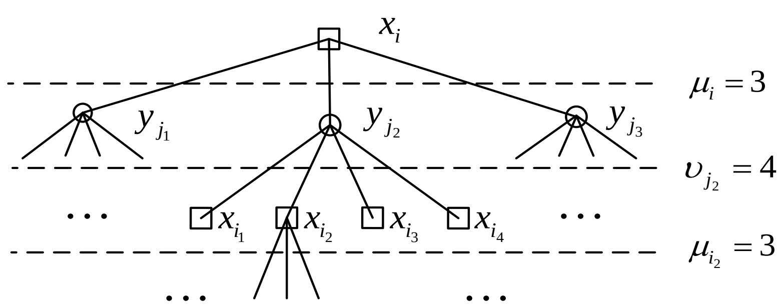

Figure 2: Illustration of the simplified tree structure

Thereby, one measurement equals the sum of S selected elements and is evaluated in the range [0, S]. Since both the source and the measurement are binary, the numeric relation complexity of the tree structure is reduced and furthermore the recovery of an element node can be deduced from its related measurements. Based on this idea, we classify the measurements to two sets and discussed separately:

a) The measurment result is zero or S. The values of the sampled elements can be judged immediately as zero or one.

b) The measurment result is in the range [1, *S-1*]. The recovery of the element node depends on the recvoery situation of the other (*S-1*) element nodes, called neighboring nodes. If any neighboring node is unable be recovered, the current element node fails too. If all the neighboring nodes have been recovered, the element node can be recovered sequentially.

In order to evaluate the recovery of one source element, the un-recovery probability P_u of a source element x_i is defined as the probability that the value of x_i cannot be recovered from the tree structure.

Thereby, the un-recovery probability of x_i is evaluated as zero, in the case that it is directly recovered from the related measurements. Otherwise more assistant information is required to judge the recovery of the element, and the availability of assistant information exhibits as the un-recovery probability of the neighboring nodes. In Figure 2, if none of $\{y_{j_1}\ y_{j_2}\ y_{j_3}\}$ is zero or three, the un-recovery probability of x_i depends on the un-recovery probability of the neighboring nodes in the third layer (e.g. x_{i_1}, x_{i_2}, $x_{i_3}$and x_{i_4}), and the un-recovery of these nodes is derived from the sub-tree structure in the same way. Due to the random property of the source and $\boldsymbol{\Phi}$, two cases on recovery happen with certain probabilities, and this will be analyzed in the next section.

III. Un-recovery probabiliy upon tree structure

Since tree structure of any element node possesses the same statistical property, and therefore the un-recovery probability P_u of any element node is the same on expectation. For the entire source, it is recovered with probability $1 - P_u$. Aiming to deduce P_u from tree structures, we study the characteristics of tree structure first.

Unlike the measurement node, the degree of element nodes varies from zero to M. The probability that an element node is of degree l , equals to the probability that it is sampled l times.

$$P(x_i|\mu_i = l) = N(p_s)^l(1-p_s)^{M-l}\binom{M}{l}/N_S. \tag{4}$$

where $p_s = S/N$, is the probability that one source element is sampled for each measurement, and N_s is the total number of the source elements sampled.

$$N_S = N(1-(1-p_s)^M). \tag{5}$$

Considering the conditional probability in (4), the average degree α of element nodes is,

$$\alpha = \sum_{l=1}^{M} l \cdot P(x_i|\mu_i = l) = SM/N_s. \tag{6}$$

For convenient analysis, we reform the tree structure in a statistical way to yield a regular tree structure. Simply speaking, we assume each element is sampled the same times α, and hence each element node is of the same degree α. An example of regular tree structure is depicted in Figure 3, $\alpha = 3$ and $S = 4$.

Given the sparsity of the source, we can deduce the distribution of measurements. According to the classification of the measurement in last section, the probability of two sets is,

$$\rho_j = \begin{cases} p_w^S + p_b^S & j = 1 \\ 1 - p_w^S - p_b^S & j = 2 \end{cases}. \tag{7}$$

Where ρ_1 denotes the probability of value as one or S; ρ_2 is the probability of values from one to *S -1*. In words, from one measurement x_i can be directly recovered with probability ρ_1, the other case happens with probability ρ_2.

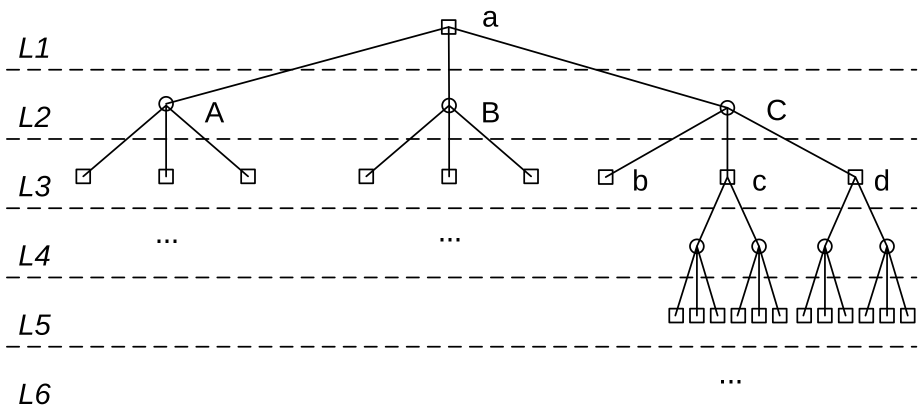

Figure 3: Illustration of a regular tree structure

A. Un-recovery probability of each layer

In order to deduce P_u via tree structures, it is necessary to find how the recovery is triggered and how the un-recovery probability of nodes are propagated through layers.

Taking the regular tree in Figure 3 for illustration, we denote the root of the tree as “a”. The node “C” connects to three other element nodes denoted as “b”, “c”, and “d”, respectively. The un-recovery probability of “a” can be analyzed from this sub-tree only, because the characteristic of other two sub-trees is the same statistically. If the value at “C” is equal to 0 or *S*, all element nodes connected to it can be recovered. The un-recovery conditions of “a” should read: (i) all measurement nodes connected to “a” have value neither 0 nor *S*; (ii) at least one of other neighboring nodes (e.g., “b”, “c”, or “d”) cannot be recovered from its sub-tree. Thus, the un-recovery probability of “a” can be calculated by

$$P_u = P_u^1 = (\rho_2)^{\alpha}[1 - (1 - P_u^3)^{S-1}]^{\alpha} \tag{8}$$

where P_u^3 is the un-recovery probability of a third-layer node, ρ_2 is the probability that measurement nodes are of set two, $1 - (1 - P_u^3)^{S-1}$ is the probability that at least one out of these $(S - 1)$ neighboring nodes cannot be recovered. This relation can be applied to other layers, so that a general formula is generated as

$$P_u^{2k-1} = (\rho_2)^{\alpha-1}[1 - (1 - P_u^{2k+1})^{S-1}]^{\alpha-1}, k = 2,3, \ldots \tag{9}$$

P_u^{2k+1} is the un-recovery probability of a $(2k + 1)$-th-layer node. Distinct from tree root “a”, only $(\alpha - 1)$ measurement nodes are taken into account for each sub-tree root (“b”, “c” or “d”).

Since each element is of the same independent identical distribution, for the tree structure of any source element, the un-recovery probability P_u of the root node is deduced from P_u^3. And the un-recovery probability of element node in (*2k-1*)-th layer can be derived from that in (*2k+1*)-th layer.

B. Conditions of successful recovery

When P_u is small, the sampled elements are recovered with high probability. And we define the case that $P_u = 0$ as successful recovery, otherwise as unsuccessful recovery. With respect to Eq. (8) and (9), one can compute the un-recovery probability of an element node P_u through a simple iteration, starting from some initial value P_u^{2k+1} at a selected k. let $\beta = \alpha - 1$ and define $g(x)$ as follow

$$g(x) = \rho_2^{\beta}[1-(1-x)^{S-1}]^{\beta}, \quad x \in [0,1] \tag{10}$$

Whose graphs are shown by the solid curves in Figure 4. Note that $g(x)$ increases monotonically and has exactly one inflexion point (a point where $g''(x) = 0$). As $\beta \geq 1$, the stationary points of (10) are the intersections of $g(x)$ and the linear function $h(x) = x$. By observing Figure 4, there exist two cases with the distinct stationary point distribution.

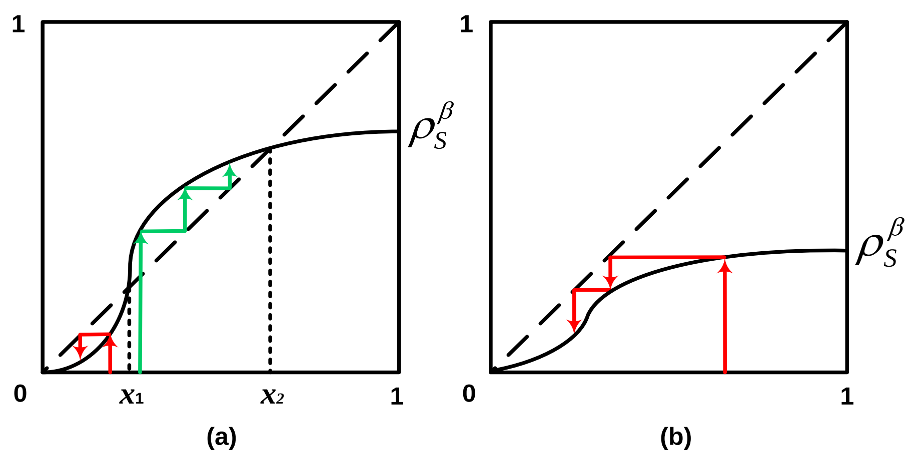

Figure 4: The graphs of two possible cases for $\boldsymbol{g(x)}$

Case 1: Three stationary points $x = 0$, x_1, x_2 $(0 < x_1 < x_2 < 1)$, see Figure 4(a). If the initial point is larger than x_1 (green arrows), the iteration will converge to x_2; if the initial point is smaller than x_1 (red arrows), the iteration will converge to 0. If the initial point is x_1, the iteration will stop at x_1.

Case 2: One stationary point $x = 0$, see Figure 4(b). The iteration will always converge to $x = 0$.

Generally speaking, the initial value of P_u^{2k+1} can be any value with in the range $(0, 1]$, but small un-recovery probability is impossible to achieve at the beginning of the recovery process. Therefore, the convergence point is x_2 for case 1. In words, after enough iterations the un-recovery probability P_u^3 converges to the stationary point x_2, which is the larger root of $g(x) = x$ in the range $(0, 1]$, and the corresponding P_u^1 via (8) is the un-recovery probability of the root element.

In the case 2, to achieve the zero un-recovery probability it is necessary and sufficient that

$$\rho_2^{\alpha-1}[1-(1-x)^{S-1}]^{\alpha-1} < x, \ \forall\, x \in (0,1]. \tag{11}$$

It is guaranteed that, when $\alpha \geq 2$ and the tree is long enough, the iteration converges to 0 and accordingly ends up with the zero un-recovery probability P_u^3. Based on (8), P_u^1 goes to zero with P_u^3 approaching to zero.

The recovery performance of compressive sensing for binary sparse source can be evaluated by referring the un-recovery probability P_u. In terms of the deduction, the P_u of a given source mainly determined by the random sampling parameters *S* and *M*. Herein, we plot the contour graphs of P_u on *S*-*M* plane to explore their relations. For the sources with $N = 10^4$ and different p_w, the corresponding behavior of P_u on the *S*-*M* plane is shown in Figure 5. The white counter curves divide the *S*-*M* plan to several parts according to the values of P_u. For the pairs (*S*, *M*) lay on the curves, the corresponding P_u is just the decimal fractions annotated on the curves. And in the region on the right of the counter curve P_u is smaller than the annotation value. Therefore, $P_u < 10^{-5}$ in the white regions, which are taken for zero un-recovery probability. The black line denotes that the (*S*, *M*) pairs satisfy $\alpha = 2$ via equations (5) and (6), and the upright region of the black line obeys the condition $\alpha > 2$.

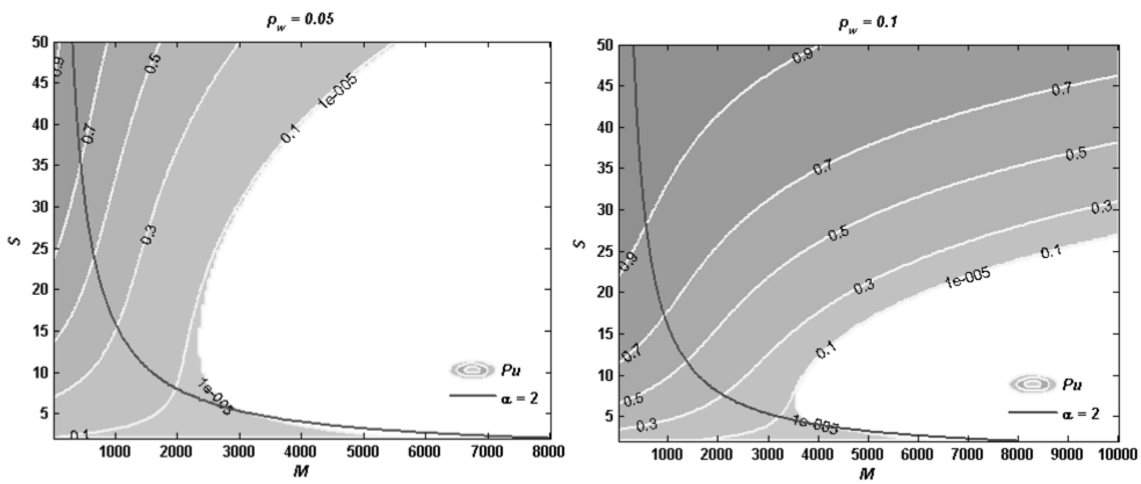

Figure 5: The contour plots of $\boldsymbol{P_u}$ on S-M plane

Following the necessary conditions of successful recovery, the valid region for successful recovery is the white part on the upright of the black curve. It is observed that P_u is monotonically decreases as the number of measurement increases. And by selection optimal S, the number of measurement can be dramatically reduced. In comparison with (a) and (b) in Figure 5, the sparser the source is, the larger valid region for successful recovery. The minimal M for successful recovery decreases with the source sparsity and the optimal S increases with the source sparsity. The random sampling can be efficiently designed with respect to the inherent property of source.

IV. NUMERIC RESULTS

The recovery of compressive sensing can be measured by the unrecovered ratio, which equals to the number of unrecovered elements over the total number of element sampled. Referring the definition of the un-recovery probability, the unrecovered ratio is equivalent to P_u in terms of our analyses, and its values can be figured out by the formula deduced by the corresponding tree structures.

Since the recovery process of the tree structure is similar to that of the graph-based recovery algorithms described in [8][11]. To testify our analyses, we compare the tree structure based analyses results with the practical recovery for the sources with different sparsity, see Figure 6. The source size N = 10^4, and the sparsity of two source is p_w = 0.01 and p_w = 0.05, respectively.

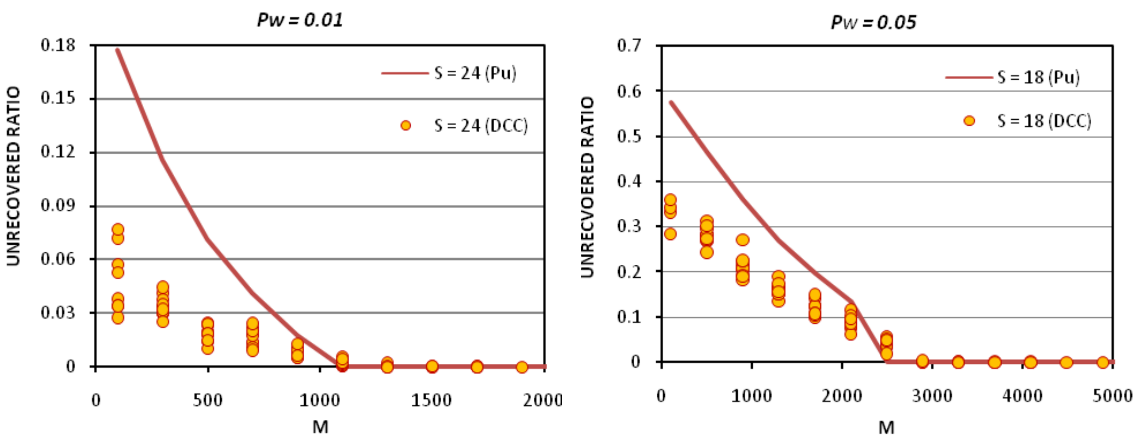

Figure 6: The comparison between the tree structures based analyses results and the actual recovery process for the sources with different sparsity.

In Figure 6, the horizontal axis stands for the number of the measurements, whereas the vertical axis represents unrecovered ratio. The unrecovered ratio vary with different number of measurement. For each number of measurements, we decode ten different random graphs by the way described in [11]. The corresponding pairs of unrecovered ratio and M are drawn as points in Figure 6. The unrecovered ratio calculated by (12) is drawn as curve. One can observe that the unrecovered ratio approach to zero almost under the same number of measurement (e.g., M = 1200 when p_w = 0.01 and M = 2500 when p_w = 0.05). Our tree structure based recovery bears an analogy to LDPC, that is, the irregular structure may achieve better recovery performance than the regular structure. Since the structure of the actual recovery is irregular and ours is regular, there always exist gaps between our analyses results and the actual recovery. Therefore, we believe that by well-designing the tree structure instead of simple regularization, the recovery performance can be further improved.

V. CONCLUSIONS

In this paper, we utilize tree structure to represent the random sampling relation among the source and the measurement. Based on the tree structure, the recovery of the compressive sensing on binary sparse source is analyzed by introducing un-recovery probability. The un-recovery probability analyses upon the tree structure present the connection between the random sampling and the recovery performance. The recovery performance estimated is shown and compared with the actual recovery in the compressive sensing. The results show that tree structure based analyses provides a potential way to describe the actual recovery process.

REFERENCE

[1] E. Candes and T. Tao, "Near optimal signal recovery from random projects: Universal encoding strategies?" *IEEE Trans. Information Theory*, vol. 52, no. 12, pp. 5406-5425, 2006.

[2] D. L. Donoho, "Compressive sensing," *IEEE Trans. Information Theory*, vol. 52, no 4, pp. 1289-1306, 2006.

[3] S. Chen, D.Donoho, and M. Saunders, "Atomic decompostion by basisi pursuit," SIAM J. on Sci.Comp., vol 20, no. 1, pp. 33-61, 1998

[4] J. Tropp and A. C. Gilbert, "Signal recovery from partial information via orthogonal matching pursuit," Apr. 2005, Preprint.

[5] A. C. Gilbert, M. J. Strauss, J. Tropp, and R. Vershynin, "Algorithmic linear dimension reduction in the l_1 norm for sparse vectors," Apr. 2006, Submitted.

[6] M. Akcakaya and V. Tarokh, "A frame construction and a universal distortion bound for sparse representations", *IEEE Trans. on Signal Processing*, vol. 56, no. 6, pp 2443-2450, 2008.

[7] S. Sarvotham, D. Baron and R. G. Baraniuk, "Compressed sensing reconstruction via belief propagation," preprint.

[8] S. Sarvotham, D. Baron and R. G. Baraniuk, "Sudocodes – fast measurement and reconstruction of sparse signals", *Proc. International Symposium Information Theory*, 2006.

[9] W. Xu and B. Hassibi, "Efficient compressive sensing with deterministic guarantees using expander graphs," *IEEE Workshop on Information Theory*, pp. 414-419, 2007.

[10] S. Jafarpour, W. Xu, B. Hassibi and R. Calderbank, "Efficient and robust compressed sensing using high-quality expander graphs," preprint.

[11] F. Wu, J. J. Fu, Z. C. Lin, and B. Zeng, "Analysis on rate-distortion performance of compressive sensing for binary sparse source", *Data Compression Conference*, 2009.