Thesis for Master's Degree

Belief Propagation based Compressive Sensing Recovery Method in the Sparse Weighted Graphs

Haeung Choi

School of Information and Communications

Gwangju Institute of Science and Technology

2015

석사학위논문

희소한 가중 그래프 상에서의 신뢰전파 기반 압축센싱 복구방법

최 해 웅

정 보 통 신 공 학 부

광 주 과 학 기 술 원

2015

Belief Propagation based Compressive Sensing Recovery Method in the Sparse Weighted Graphs

Advisor : Prof. Heung-No Lee

by

Haeung Choi

Graduate: School of Information and Communications

Gwangju Institute of Science and Technology

A thesis submitted to the faculty of the Gwangju Institute of Science and Technology in partial fulfillment of the requirements for the degree of Master of Science in the School of Information and Communications

Gwangju, Republic of Korea

2015 . 06. 10.

Approved by

Professor Heung-No Lee

Committee Chair

Belief Propagation based Compressive Sensing Recovery Method in the Sparse Weighted Graphs

Haeung Choi

Accepted in partial fulfillment of the requirements

for the degree of Master of Science

June 10th 2015

Committee Chair ________________
Prof. Heung-No Lee

Committee Member ________________
Prof. Jong Won Shin

Committee Member ________________
Prof. Hyo Sung Ahn

MS/IC Haeung Choi(최해웅). Belief Propagatioin based Compressive Sensing Recovery
20131217 Method in the Sparse Weighted Graph (희소한 가중 그래프 상에서의 신뢰전파 기반 압축센싱 복구 방법). School of Information and Communication. 2015. 53p. Advisor: Prof. Heung-No Lee.

Abstract

Our study is related on Compressive Sensing(CS), which is a famous signal processing framework used in many fields like image processing and machine learning. By CS, we can get compressed measurement of signal and reconstruct the original signal from it in polynomial time. This can be used as a blind sample-and-compression combined procedure or super-resolution method of the original signals.

Belief Propagation, which is a message passing algorithm that works on graph model, evaluates marginal probability density function by propagating messages from a node to other nodes. If we consider set of variable nodes in bipartite graph as a sparse signal and set of the other nodes as a compressed measurement, we can use the Belief Propagation as a compressive sensing recovery algorithm. Furthermore, unlike other recovery algorithms, we can directly use prior distribution and can get the probabilistic distribution of original signal instead of just estimated values.

In this study, a new compressive sensing recovery algorithm based on Belief Propagation, called CS-WBP is proposed and empirically compared with other Belief Propagation based recovery algorithms such as Approximate Message Passing(AMP) and canonical Compressive Sensing via Belief Propagation(CS-BP). This new algorithm can be used on specific sensing matrix which CS-BP cannot cover and show better performance than AMP for certain conditions.

Contents

List of Tables

List of Figures

1 Introduction

1. 1. Compressive Sensing Overview

1. 1. 1. Sampling Theorem to Compressive Sensing

With developments of computer and digital signal processing, many signal processing has been done in digital domain. However, many natural signals that we would like to handle are *analog* signals. These analog signals must go through a converting stage called analog-to-digital-converting (ADC) to yield digital signals which can be processed in digital circuit. *Nyquist-Shannon sampling theorem*, or simply, *sampling theorem* has been used as a basic principle of ADC for a century. This theorem says that:

> If a function $f(t)$ contains no frequencies higher than W, it is completely determined by giving its ordinates at a series of points spaced $W/2$ seconds apart. [1]

i.e., we need to sample with sampling rate which is higher than two times of maximum frequency of target signal. Then, we can reconstruct original analog signal from the sampled signals. This limit of sampling rate, called *Nyquist rate*, is known to be a fundamental limit of ADC.

However, the sampled signals are often large in volume. Therefore, before for them to be stored or transmitted, they must go through a compression stage to reduce the size of the sampled signals. Then, there are natural question ariased by Donoho in [2]

> "Everyone" now knows that most of the data we acquire "can be thrown away" with almost no perceptual loss (...) why go to so much effort to acquire all the data when most of what we get will be thrown away? Can we not just directly measure the part that will not end up being thrown away? [2]

In order to answer to the above question, compressed sensing (CS) [2] is developed and studied. CS starts from the inverse problem of underdetermined linear system,

$$\mathbf{y} = \mathbf{A}\mathbf{x}$$

where $\mathbf{y}$ is $M \times 1$ measurement vector, $\mathbf{A}$ is $M \times N$ sensing matrix, $\mathbf{x}$ is $N \times 1$ signal and $M < N$.

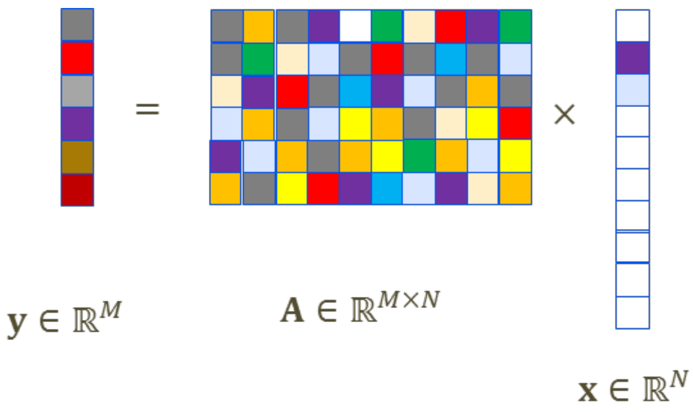

Figure 1 Compressive Sensing

This type of ill-posed problem has infinitely many solutions so that we cannot recover $\mathbf{x}$ from $\mathbf{y}$ given $\mathbf{A}$ in general. But in CS, we can separate a specific solution using *sparsity* constraint.

The terminology “sparse” means that almost elements are zero and only a few elements can have non-zero values. For example, sparse signal is a signal which has a few picks and remained values are all zero. The Compressive Sensing theory says that if a signal is sparse enough, information can be compressively acquired by the random projection in measurement without loss of information. Since a sparse signal has small amount of information compared with the size of signal, the compressive measurement by linear projection can store all signal information without loss. This statement can be expressed mathematically using information theory,

$$H(\mathbf{x}) \leq H(\mathbf{y}) \qquad (1)$$

where $H(\cdot)$ operator means information theoretic entropy of random variable, which is the amount of information. (1) means that the amount of information that measurement vector $\mathbf{y}$ can hold is more than that of sparse signal vector $\mathbf{x}$. Thus, measurement vector $\mathbf{y}$ can save all information of $\mathbf{x}$ without loss of information.

Even if the signal is not sparse, we can transform the signal to another domain which makes the signal sparse. For example, sinusoidal wave in time domain is not sparse, but it has unique nonzero element in frequency domain and the other elements are all zero. In the cases like this, we call the domain "sparsifying domain" and the signal "compressible" signal. Using the orthonormal transform basis $\mathbf{\Psi}$, we can express the $N \times 1$ non-sparse signal vector $\mathbf{z}$ as a sparse vector $\mathbf{x}'$ and establish the CS problem.

$$\begin{gathered}\mathbf{\Psi z} = \mathbf{x}' \\ \mathbf{y} = \mathbf{Az} = \mathbf{A}'\mathbf{\Psi z} = \mathbf{A}'\mathbf{x}'\end{gathered}$$

By assuming that the signal is sparse enough, we can distinguish unique solution for $\mathbf{x}$. The sparsest vector among the feasible solutions will be the original signal. Here, we can establish the CS recovery problem as optimization problem like this:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_0 \;\; s.t.\; \mathbf{y} = \mathbf{Ax} \qquad (2)$$

where $\|\cdot\|_0$ is L0 norm, which is defined as number of nonzero elements in a vector. However, in order to solve this L0 minimization problem, we have to search all possible nonzero pattern of signal.

Even if we know K which is the number of nonzero elements in $\mathbf{x}$, checking all $\binom{N}{K}$ possible nonzero pattern is NP-hard problem which cannot be solved in polynomial time.

Instead, let's consider minimum L1 norm reconstruction like this.

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \;\; s.t.\ \mathbf{y} = \mathbf{A}\mathbf{x} \quad (3)$$

As you can see in Figure 2, L1 optimization is likely to give the sparse solution. Figure 2 shows the L1 and L2 balls with unit radius. Because of diamond-shape of L1 ball, minimum L1 norm solution is sparse in many cases while minimum L2 norm solution is hardly sparse.

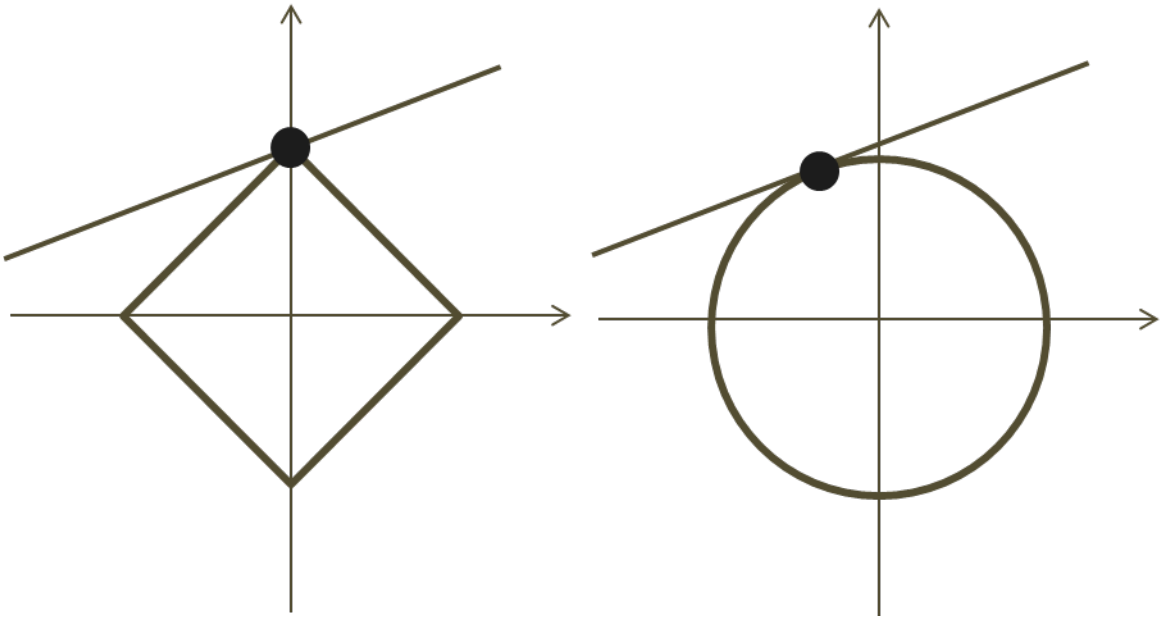

Figure 2. L1 and L2 balls with unit radius

Using minimum l1 norm reconstruction, sparse signal with K nonzero elements can be recovered using only $M \geq cK\log(N/K)$ length of measurement vector [3][4]. This L1 minimization problem in (3) is also called Basis Pursuit and it can be solve by linear programming called Basis Pursuit Denoising [5] with $O(N^3)$ computation. If $\mathbf{y}$ is noisy measurement, we can reformulate (3) as like:

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \;\; s.t. \;\; \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \le \varepsilon$$

Or, equivalently,

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \;\; s.t. \;\; \|\mathbf{x}\|_1 \le q \qquad (4)$$

(4) also called LASSO[6]. By bringing constraint to the objective, (4) becomes

$$\hat{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 ,$$

which is well known convex optimization problem and there are many algorithms to solve this problem.

These CS recovery algorithms can be divided into two broad groups.

(i) Greedy approximation algorithms

Basically, greedy algorithms try to solve L0 norm regulation problem (2).However, finding the global minima of (2) is NP-hard problem. Greedy algorithms iteratively find local minima each iteration and the final result becomes an approximation of solution of (2). Matching Pursuit (MP) [8] and Orthogonal Matching Pursuit (OMP) [9] are typical greedy algorithms. Also, there are many variants of them and other related greedy algorithms such as Compressive Sampling Matching Pursuit (CoSaMP) [10], Subspace Pursuit (SP) [11] and Stage-wise Orthogonal Matching Pursuit (StOMP) [12].

(ii) L1 Optimization based algorithms

Algorithms of this type operate for solving BP problem (3) or LASSO problem (4). Since there were many studies about convex optimization, there are various methods for these problems. Traditional interior point method is the most basic one. For example, Kim *et al*. suggested a successful truncated Newton based interior point method for L1 regularized least squares [13]. Most of these algorithms use additional techniques for efficiency. Application of Alternative Direction Method (ADM) is a good example [14] while Iterative Shrinkage Thresholding Algorithm (ISTA) [15] and its improvement, Fast ISTA (FISTA) [16] are also efficient L1 optimization based algorithms.

1. 2. Belief Propagation and Compressive Sensing

1. 2. 1. Graph Model of Compressive Sensing

Typical CS problem can be modeled as a graphical model. If we interpret the sensing matrix $\mathbf{A}$ as a representation of relationship between nodes, we can say $\mathbf{A}$ is a bipartite graph.

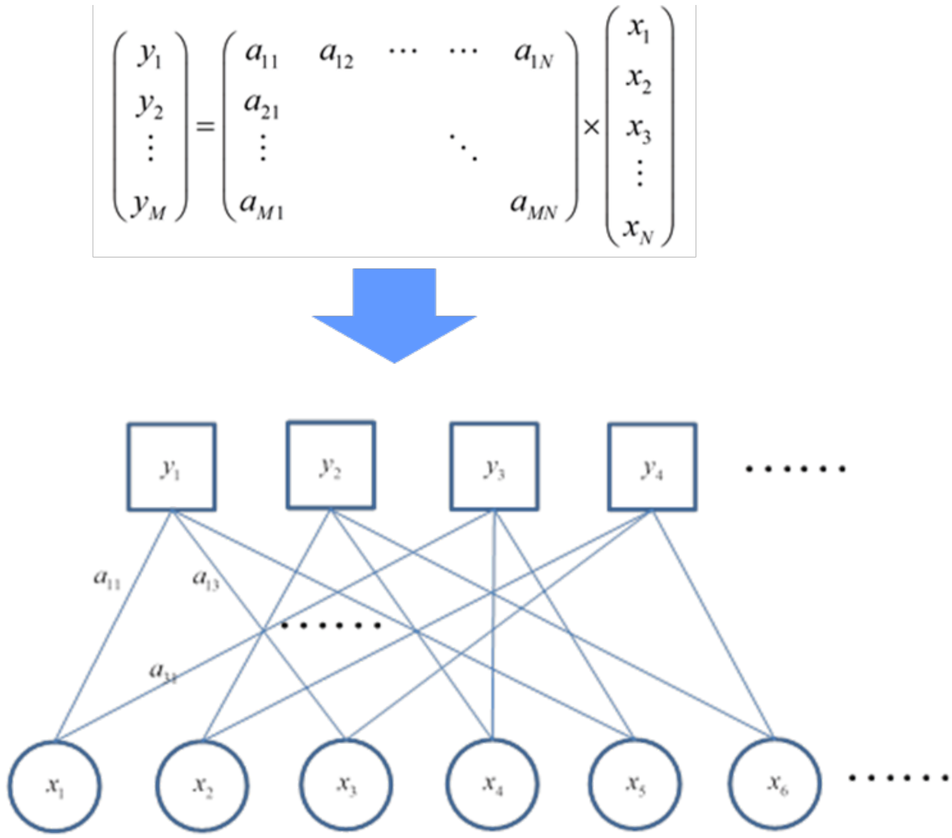

Figure 3 Factor graph model of compressive sensing

In this graph, the weight of each branch is nonzero elements of sensing matrix. On the other hand, branches corresponding to zero elements are not displayed in the graph. A set of variable nodes means sparse signal vector $\mathbf{x}$ and set of factor node means measurement vector $\mathbf{y}$. If the sensing matrix is sparse, then each element of measurement vector $\mathbf{y}$ will become a linear sum of just a few elements in signal vector $\mathbf{x}$ and also there will be a few edges between a measurement node and signal nodes. We call this sparse and bipartite graph *factor graph*.

In the factor graphs, an element of $\mathbf{y}$ can be expressed as a function of several elements of $\mathbf{x}$. Thus, we do not need to consider global function of all elements of $\mathbf{x}$ to calculate each element of $\mathbf{y}$. Instead, calculating local functions and sharing the result will be enough because elements of $\mathbf{y}$ are independent with each other. For example, in the Figure 3, it is difficult to calculate global function,

$$\mathbf{y} = f(\mathbf{x}) = f(x_1, x_2, x_3, ...)$$

but calculating of local functions,

$$\mathbf{y} = f(x_1, x_3, x_5) \times f(x_2, x_4, x_6) \times f(x_1, x_4, x_5) \times \cdots$$

is much easier. Thus, we can consider using the sum-product message passing algorithm—which is a Belief Propagation (BP) suitable for the factor graph model[17]—for reconstruction of sparse signal *x* from measurement *y*.

1. 2. 2. Belief Propagation

Belief propagation (BP) is an efficient methodology for sharing statistical information over graphical models. In the BP, local result, called *messages*, are passed to other local nodes so that local nodes can use the result of other nodes to their operation. This procedure is called message passing.

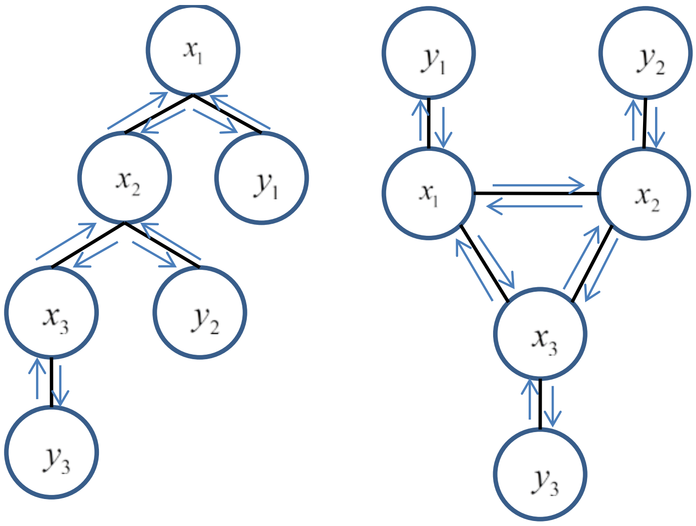

Figure 4 Tree-structure and loopy structure

Originally, BP was proposed to find exact solution of tree-structured graph problem [18], but it also gives approximate solution of general graph with loops[19]. Although it's exact conditions for convergence are not well known, BP has been used to solve general bipartite graph problem such as decoding of Low Density Parity Check (LDPC) code[20]. And many other existing algorithms developed in the artificial intelligence, signal processing, and digital communications community use based on the BP. [17] In conclusion, BP is known to be exact on trees and accurate for locally tree like graphs. [20][21][22]

BP has many message passing rules such as max-sum message passing and sum-product message passing. In my work, we are going to focus on sum-product type BP because of the reasons which will be presented in Section 2. Unless otherwise stated, all BP in this work is the sum-product type BP. In the BP, we use the probability density distributions as messages. This algorithm calculates marginal probability distributions of variable nodes in factor graph. Before explain the detail, we are going to use the subscript i to direct unspecified variable node, j for unspecified factor node, v for a specific variable node, and f for a specific factor node. For example, $\mu_{v \to f}$ is a message from variable node x_v to factor node y_f.

The BP consists of these three steps:

(i) Variable to factor(V2F) update

At first, V2F message are updated. These messages are the probability distribution of x_v conditional on all factor nodes except a target factor node y_f which means

$$f_{X_v}(x_v \mid \mathbf{y}_{n(v)\setminus\{f\}})$$

where $n(v)$ is a set of indices of factor nodes which are connected to the variable node x_v.

Since every factor node which is connected with x_v locally calculates the probability distribution of x_v and sends it to x_v as a factor to variable(F2V) message, the V2F messages become the product of these F2V messages except from target factor node. Figure 5 shows the calculation of V2F message from x_1 to y_1.

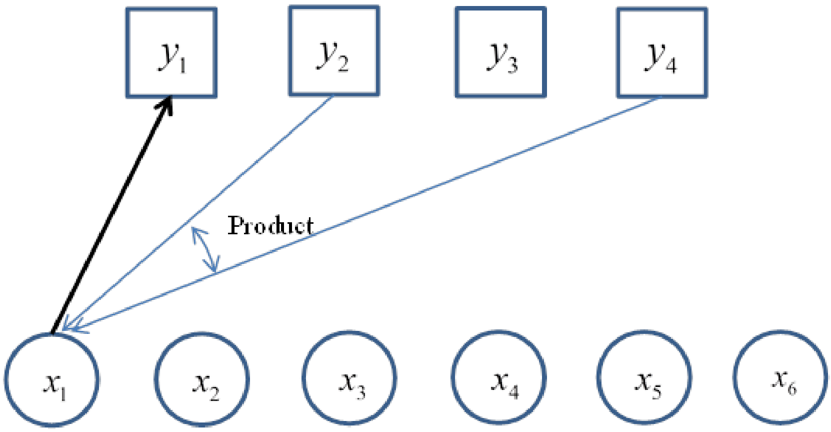

Figure 5 Calculation of V2F message

This calculation can be expressed like

$$\mu_{v \to f} := \prod_{j \in n(v) \setminus \{f\}} \mu_{j \to v}$$

where $n(v)$ is the set of factor nodes' indices which are connected with x_v node.

(ii) Factor to variable(F2V) update

Secondly, F2V messages are updated. These messages are the conditional distribution of a variable node x_v, given one factor node y_f and other variable nodes which means

$$f_{X_v}(x_v \,|\, \mathbf{x}_{n(f) \setminus \{v\}}, y_f)$$

where $n(f)$ is a set of indices of variable nodes which are connected to the factor node y_f.

Since y_f has the probability distributions from all connected variable nodes $\mu_{i\in n(f)\to f}$, we can calculate F2V message $\mu_{f\to v}$ by marginalizing the joint distribution with respect to the other variable nodes except x_v. Figure 6 shows this calculation of F2V message from y_f to x_1.

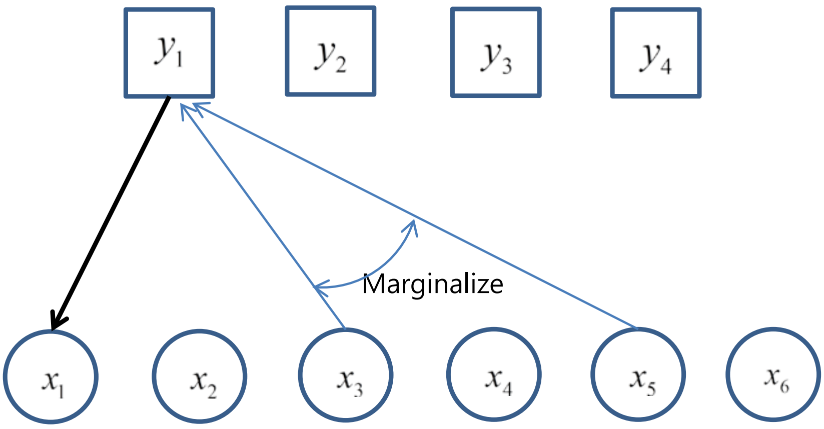

Figure 6 Calculation of F2V message

This calculation can be expressed like

$$\mu_{f\to v} = \int_{\mathbf{x}_{n(f)\backslash x_v}} f(\mathbf{x}_{n(f)}) \prod_{i\in n(f)\backslash v} \mu_{i\to f} d\mathbf{x}_{n(f)\backslash x_v}$$

where $n(f)$ is the set of variable nodes' indices which are connected with y_f and $f(\mathbf{x})$ is a joint distribution of variable nodes. From the relationship between the factor node and variable nodes,

$$y_f = \sum_{i\in n(f)} x_i$$

and

$$x_v = y_f - \sum_{i\in n(f)\backslash v} x_i \qquad (5)$$

We can find the F2V message from y_f to x_v. Let the random variable $R = \sum_{i \in n(f) \backslash v} X_i$. Then (5) becomes

$$X_v = y_f - R$$

and finally the probability distribution of X_v will be $f_{X_v}(x_v) = f_R(y_f - r)$, where the probability distribution of R can be expressed as a convolution,

$$f_R(r) = \bigotimes_{i \in n(f) \backslash v} f_{X_i}(x_i).$$

(iii) Posterior update

By repeating (i) and (ii), messages are exchanged among all nodes. After all messages are converse, we can get the posterior distribution for each variable node by multiplying all F2V messages. Figure 7 shows the calculation of x_1's posterior distribution. The posterior distribution is proportional to the product of F2V messages,

$$f(x_v) = \frac{1}{Z} \prod_{j \in n(v)} \mu_{j \to v}$$

where Z is a normalization constant.

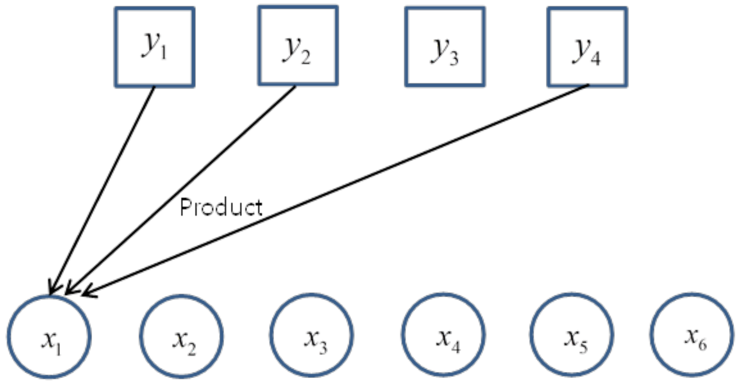

Figure 7 Calculation of posterior distribution

1. 2. 3. Using Belief Propagation on Compressive Sensing Recovery

As mentioned in the section 1.1.1, there are many fast and precise CS recovery algorithms without using BP. However, there are some advantages of BP based algorithms. These advantages are mainly due to the suitability for Bayesian approach.

(i) Direct use of prior information

In many cases, probability distributions of signals are known in prior. Simply putting the initial messages of BP to prior distributions allow to enjoying the benefit of prior information—the better estimation. In many other recovery algorithms, we often have to customize many parts to apply the prior information.

(ii) Output in the form of posterior distribution

We can get posterior distribution while other algorithms give us just estimation values. Using these Bayesian inferences, we can know more about signals. For example, we can find the next-best estimation as a precaution of failure. Or, we can present the statistical characteristics like confidence interval of estimation.

But there are some difficulties to implement BP based algorithms. At first, sampling problem makes implementation difficult. In order to implement algorithm, it should be processed by Digital Signal Processing (DSP). Thus, continuous function such as probability density function (PDF) cannot be processed directly; it have to be sampled. If the graphs are weighted graphs, the expanded and shrunk version of PDF should be evaluated. For example, let $f_{x_i}(x_i)$ is the prior PDF of x_i which is used as initial message from x_i node to factor nodes. Let's remind (5).

$$x_v = y_f - \sum_{i \in n(f)\backslash v} x_i$$

If the edges have weight, then (5) becomes

$$x_v = y_f - \sum_{i \in n(f)\backslash v} a_{fi} x_i$$

where a_{ji} is weight of edge between a factor node y_j and a variable node x_i. i.e., the (j,i) th element of sensing matrix $\mathbf{A}$. Then the random variable R is defined as

$$R = \sum_{i \in n(f)\backslash v} a_{fi} X_i$$

because

$$y_f = a_{fv} X_v + R$$

and

$$a_{fv} X_v = y_f - R.$$

Let $a_{fi} X_i = E_i$. To evaluate $f_R(r) = \underset{i \in n(f) \backslash v}{\otimes} f_{E_i}(e_i)$, we have to calculate $f_{E_i}(e_i) = f_{X_i}\left(\frac{x_i}{a_{fi}}\right)$ which is $a_{fi}-$times expanded version of $f_{X_i}\left(x_i\right)$. Also, we have to calculate the probability distribution of $X_v = \frac{y_f - R}{a_{fv}}$ to update F2V message, which is $a_{fi}-$times shrunk version of $f_R(y_f - r)$. Here, we face with interpolation issue to maintain the sampling rate. Since we store samples of $f_{X_i}\left(x_i\right)$, calculated $f_{E_i}(e_i) = f_{X_i}\left(\frac{x_i}{a_{fi}}\right)$ will be lost the value between two adjacent sample points. Because we have to know the values of these points to evaluate the exact messages, the interpolation of these points is needed—which causes interpolation errors. Without solution to these problems, we cannot implement BP based recovery algorithms.

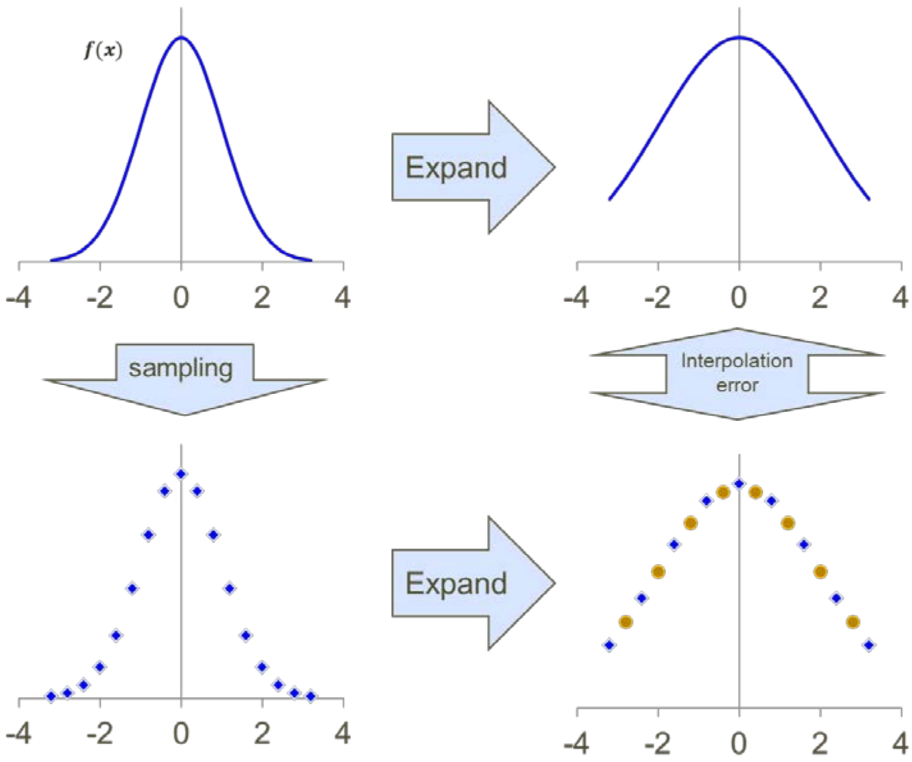

Figure 8 Interpolation error

Computational complexity is also a big problem. In belief propagation, we have to deal messages which are functions, not a few variables. It needs much computational cost to multiplying and marginalizing the messages. Specifically, if we store the probability distribution in l-length samples, the computational cost of convolution of these distributions which is needed to update message is $O(l \log l)$. Moreover, we have to calculate messages for each edge. If the graph is not sparse enough, the computational cost will increase more and more.

2. Existing Belief Propagation-Based Algorithms

2. 1. Examples of BP-Based Algorithms

In the previous section I, we introduced some issues that BP-based algorithms have to solve. Here are some examples of BP-based algorithms which solve the issues successfully.

2. 1. 1. Approximate Message Passing (AMP)

Approximate Message Passing (AMP) [23] is a successful approximate BP-based algorithm which has good phase transition and competitive reconstruction speed. In this algorithm, the sensing matrix is a zero-mean i.i.d. Gaussian random matrix. Generally, dense sensing matrix like Gaussian matrix slows down BP process because more edges cause more calculations for messages. Furthermore, dense sensing matrix has too many cycles to become tree-like structure thereby causing failure of BP iteration. However, by Gaussian approximation, amount of messages significantly reduced. With some abuse of Central Limit Theorem (CLT), message becomes Gaussian distribution; so now we don’t have to deal whole PDF. i.e., everything we have to pass is just parameters, mean and variance. Moreover, by 1^{st} order Taylor approximation, AMP much reduced its computational cost. In other hands, there is no sampling problem because we can generate marginal distribution from mean and variance of it because the distribution is known to be a Gaussian and the PDF of Gaussian random variable can be fully expressed by mean and variance.

2. 1. 2. Compressive Sensing via Belief Propagation (CS-BP)

CS-BP[24] is also a good BP-based algorithm that overcomes complexity and sampling issues in different way. Unlike AMP, CS-BP does not use approximation but uses Non-parametric BP (nBP). Because non-parametric BP passes full PDFs as messages, output can be PDF of various shapes. For example, in AMP, output distribution will be always Gaussian or Laplace distribution although real distribution can be different. Another different point is that it works at sparse sensing matrix. Since this sparse sensing matrix has a few edges and hence a few loops so that nBP is suitable in terms of speed and performance. And finally, sampling problem is also solved by restricting nonzero elements of sensing matrix as 1 or -1. Expansion or shrink of messages doesn't happen because of this. But this algorithm has a problem that estimation values can be chosen among sample points of prior PDF hence causing some quantization error. Bayesian Hypothesis Test-Belief Propagation (BHT-BP) [25] is an advanced type of CS-BP which removes quantization error successfully. BHT-BP used CS-BP algorithm to detect the sparse support set from noisy measurement and estimate the nonzero values in terms of MMSE using the detected support.

2. 2. Drawbacks of Existing Algorithms

These BP-based algorithms were successfully implemented and have been used, but there are some limitations. At large size and zero-mean i.i.d. matrix, AMP is known to converse quickly; otherwise may diverse. [26][27]. i.e., AMP cannot work at small size sensing matrix with far from zero mean. And, since AMP uses relaxed BP(r-BP) which is also called parametric BP, it cannot provide exact marginal posterior distributions. The best think it can do is approximation of posterior distribution from mean and variance. So it is difficult to benefit from prior information and posterior distribution on AMP. CS-BP has drawbacks, also. It can work only on matrix which is consist of {-1, 0, 1}. From an implementation point of view, the sensing matrix is compressive sampling device like Spatial Light Modulator (SLM) in single pixel camera. [31] By the way, this type of sampling device is not easy to customize since it uses physical structure and materials. Thus, in many real applications of CS, sensing matrix is not a design factor but defined condition. Although CS-BP is an algorithm has many advantages, it is difficult to be used in various applications because of its restriction on sensing matrix.

3. Proposed Method: Compressive Sensing via Weighted Belief Propagation (CS-WBP)

3. 1. Objective and Approach

Although there are some BP-based algorithms such as AMP and CS-BP, they have limitations. At first, AMP is a good CS recovery algorithm but it lost one of main advantages of BP-based algorithms—Bayesian approach. AMP is difficult to exploit prior information and output is limited to the Gaussian distribution. It is also a problem that mean-removal technique [28] should be used to prevent performance decrease if the sensing matrix is a far-from zero-mean matrix. In the case of CS-BP, a tight limitation on sensing matrix is the biggest problem. Since CS-BP (and BHT-BP, also) can only be used for ternary matrix consist of {-1, 0, 1}, it cannot used for various applications. Thus, we aimed to make a BP-based algorithm which can overcome these limitations. First of all, we used nBP so that we can exploit the prior information easily and get posterior distribution output. Then, our algorithm should cover non-ternary and nonzero-mean sensing matrix while treating computational complexity and sampling issue at the same time.

3. 1. 1. Sparse and weighted sensing matrix

To use nBP, we have to use graph of tree-like structure. At first, to reduce computational complexity holding on to nBP, we consider sparse sensing matrix. If the sensing matrix is sparse, then the graph will be more tree-like. It is already mentioned in Section I. In other hand, we considered non-uniform weights unlike that of CS-BP so that our algorithm becomes the extended version of CS-

BP. i.e., the nonzero values of the sensing matrix are not restricted to 1 or -1. Then the graph becomes "weighted" graph. In fact, the original graph used in CS-BP can be called weighted graph because it can have -1 value, also. But we regarded that the magnitude of the weight which causes expansion or shrinking of PDF is more important than the sign of the value. This is the reason why we called sensing matrices we used are "weighted". However, we cannot accept arbitrary weight value to satisfy the discretized signal and measurement model to implement. Here, we can know that the value of nonzero elements in sensing matrix should be an integer because the product $a_{ji}x_i$ have to be discretized. In conclusion, the sensing matrix becomes sparse and integer valued matrix which is called a sparse weighted graph.

By using matrix of this type, we also can get benefit in terms of hardware implementation. To implement compressive sensing hardware devices, we have to design a physical structure of compressive sampler corresponding to the sensing matrix. But it is very complicated to make random structure of almost infinite degree of freedom which is known to be good. Instead, many studies on real-application choose pseudo-random or well-designed deterministic sensing matrices of low degree of freedom. Famous examples such as single-pixel camera [31] and lensless imaging system [32] even used just binary (0 or 1) pseudo-random sensing matrix. If we don't think such extreme cases, it is more reasonable that the kind of value which an element in the sensing matrix can have should be limited.

3. 1. 2. Quantized of signal and measurement

We assumed quantized signal model. It is mainly because of sampling problem. If the signal and message are quantized, we do not need to worry about interpolation error; because these distributions become probability mass functions (PMF) instead of PDFs. Simple zero-stuffing on empty points is enough to treat the interpolation issue.

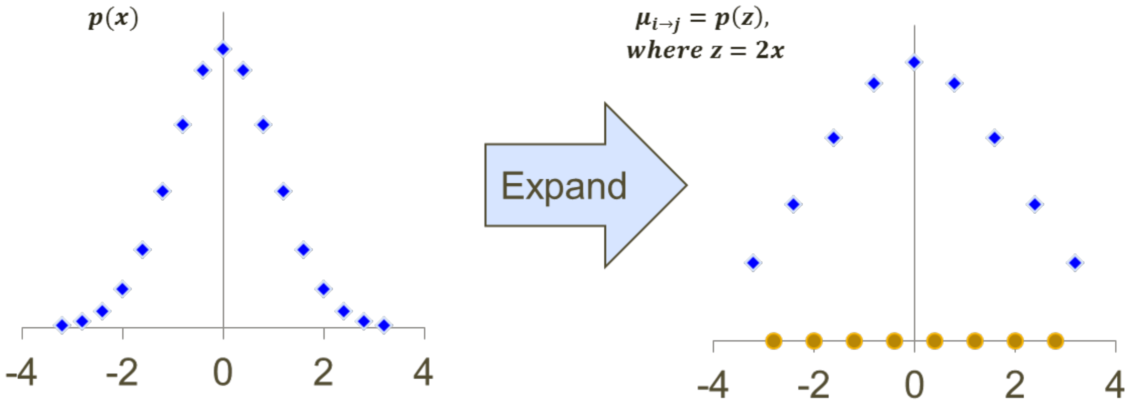

Figure 9 Zero stuffing

We can easily find quantized signals in near. Quantized signal model can bring some advantage for these signals.

(i) Digital samples

First type of quantized signals is sampled digital data. Actually, a major benefit of Compressive Sensing is that we can combine sampling and compressing of natural signals. However, usage of CS is not restricted to ADC. Rather, nowadays it is used for many signal processing fields treat digital date such as machine learning [30] and image processing [29] under the name of Sparse Representation.

(ii) Naturally quantized signals

The other type is naturally quantized signal. We often face naturally quantized signals. For example, counting data, which is sensing data about the numbers certain events are occurred is a typical naturally quantized data. Detecting, which is a process to distinguish whether a certain event is occur or not, also can be a part of counting. Microscopic measurement can be another example. It is well known physical phenomena that energy level of monochromatic light or electrons in an atom is quantized.

In addition, most of these naturally quantized signals are non-negative. Number of events occurring, energy level and intensity are good examples of non-negative signals. In these cases, we can use this prior knowledge to improve the performance of algorithms by applying non-negative constraint.

3. 1. 3. Non-negative constraint

At section 3.1.1, we already mentioned that our most of target signals not only quantized but also non-negative. We aimed at applying our algorithm to this type of non-negative signal applications; our original targets were turbid media imaging[34] and spectral intensity measuring[35].

Sensing matrix is also non-negative in our scenario. Let’s remind section 2.2; a limitation of AMP is that it cannot work well on far from zero mean matrices. Non-negative matrix is a typical example of them.

3. 2. System Model

(i) Sparse signal

Our target signal model is quantized, non-negative, and sparse.

$$\mathbf{x} \in \{0, 1, 2, ..., x_{\max}\}^N$$

Here, $x_{\max}$ is a positive integer since we assume the quantization level of the signal is one for simple calculation. If the quantization level is not one, then we can compensate the signal values by dividing and multiplying the quantization level at the input/output stages. In original CS-BP, $\mathbf{x}$ was driven from i.i.d. two-state Gaussian-mixture model[36]. However, in our scenario, prior knowledge is a little bit different. Let the signal vector

$$\mathbf{x} = \begin{bmatrix} x_1 & x_2 & \cdots & x_N \end{bmatrix}^T$$

is a realization of a series of i.i.d. random variables

$$\mathbf{X} = \begin{bmatrix} X_1 & X_2 & \cdots & X_N \end{bmatrix}^T.$$

Here, $x(i)$ can have two states—support set or not. Let the probability that i th element is support set be

$$\Pr(s_i = 1) = q$$

then the probability that the element is not support set be

$$\Pr(s_i = 0) = 1 - q$$

where $0 \le q = K / N < 1$. After whether support set or not is given, we can define conditional PMF for corresponding state and overall distribution

$$p_{X_i|S_i}(x_i \mid s_i = 1)$$

$$p_{X_i|S_i}(x_i \mid s_i = 0)$$

$$p_{X_i}(x_i) = q \cdot p_{X_i|S_i}(x_i \mid s_i = 1) + (1-q) \cdot p_{X_i|S_i}(x_i \mid s_i = 0)$$

For example, we can consider a two state mixture model that $p_{X_i|S_i}(x_i \mid s_i = 0)$ follows exponential distribution with very small mean and variance and $p_{X_i|S_i}(x_i \mid s_i = 1)$ follows Poisson distribution. Unlike mixture Gaussian model, these conditional distributions do not need to follow Gaussian distribution.

(ii) Sensing matrix

Sensing matrix is also quantized, non-negative and sparse in our scenario.

$$\mathbf{A} \in \{0,1,2,...,a_{\max}\}^{M \times N}$$

Here, $a_{\max}$ is a positive integer. If $a_{\max} = 1$, our sensing matrix will be same with that of CS-BP with nonnegative constraint. Sensing matrix of CS-BP become sparse and binary matrix under the nonnegative constraint. Figure 10 shows the difference between the sensing matrix of CS-BP and that of our scenario. (a) is the sensing matrix of CS-BP, which is a binary matrix. On the other hand, (b) is the sensing matrix where $a_{\max} = 64$.


(a)

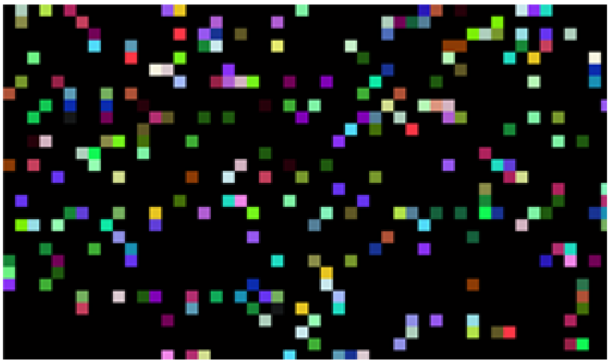
(b)

Figure 10. Sensing matrices according to $a_{\max}$.

Finally, the quantized noisy measurement is here:

$$\mathbf{y} = round(\mathbf{A}\mathbf{x} + \mathbf{n})$$

where $\mathbf{n} \in \mathcal{R}^M$ is a white Gaussian noise vector.

3. 3. Proposed Algorithms

3. 3. 1. Algorithm description

Finally, we introduce our recovery algorithm, Compressive Sensing via Weighted Belief Propagation (CS-WBP). Basically, this algorithm is non-ternary extension of CS-BP. CS-WBP operate alike CS-BP on {-1,0,1} matrices while it also can work on non-ternary matrices. Although we developed and tested this algorithm on non-negative signal and matrix condition, note that CS-WBP can be used regardless of non-negativeness.

Here is the flow diagram of CS-WBP algorithm. Figure 11 shows that the process that CS-WBP estimate sparse signal from measurement. First, BP iteration with expansion/shrinking function is performed for the quantized measurement and input prior PMF. (If we don't have any prior information, the prior PMF is initialized by uniform distribution.) As a result, the marginal posterior PMF of signal is obtained and used for Maximum a posteriori estimation of the signal. Note that we also can get marginal posterior PMF, not only the estimation value. This can be used for other propose; for example, we can apply the support detection and MMSE estimation scheme of BHT-BP in here. Any other Bayesian approach which can be done with posterior PMF can be applied easily.

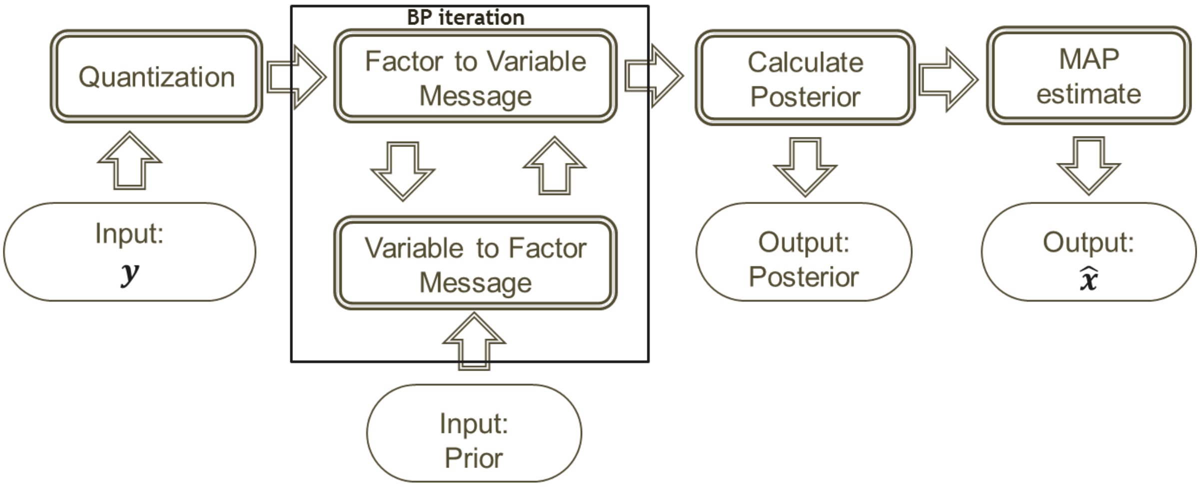

Figure 11 Flow diagram of CS-WBP.

For explaining the detail iteration stage of CS-WBP algorithm, let's remind original BP in section 1.

The CS-WBP consists of these three steps:

(i) Variable to factor(V2F) update

At first, V2F message are updated. These messages are the PMF of x_v conditional on all factor nodes except a target factor node y_f which means

$$P_{X_v}(x_v \mid \mathbf{y}_{n(v)\setminus\{f\}})$$

where $n(v)$ is a set of indices of factor nodes which are connected to the variable node x_v.

Since every factor node which is connected with x_v locally calculates the probability distribution of x_v and sends it to x_v as a factor to variable(F2V) message, the V2F messages become the product of these F2V messages except from target factor node. This calculation can be expressed like

$$\mu_{v\to f} := \frac{1}{Z}\prod_{j\in n(v)\setminus\{f\}} \mu_{j\to v}$$

where $n(v)$ is the set of factor nodes' indices which are connected with x_v node and Z is a constant called normalization constant which is used for normalizing the sum of V2F message become 1 so that it can be PMF.

(ii) Factor to variable(F2V) update

Secondly, F2V messages are updated. These messages are the conditional PMF of a variable node x_v, given one factor node y_f and other variable nodes. i.e.,

$$P_{X_v}(x_v \mid \mathbf{x}_{n(f)\setminus\{v\}}, y_f)$$

where $n(f)$ is a set of indices of variable nodes which are connected to the factor node y_f.

Since y_f knows the PMF from all connected variable nodes $\mu_{i\in n(f)\to f}$ (i.e., V2F messages), we can calculate F2V message $\mu_{f\to v}$ by marginalizing the joint PMF with respect to the other variable nodes except x_v. This calculation can be expressed like

$$\mu_{f\to v} = \frac{1}{Z}\sum_{\mathbf{x}_{n(f)\backslash x_v}} P(\mathbf{x}_{n(f)}) \prod_{i\in n(f)\backslash v} \mu_{i\to f} d\mathbf{x}_{n(f)\backslash x_v}$$

where $n(f)$ is the set of variable nodes' indices which are connected with y_f and $f(\mathbf{x})$ is a joint distribution of variable nodes and Z is normalization constant. From the relationship between the factor node and variable nodes,

$$y_f = \sum_{i\in n(f)} a_{fi} x_i$$

and

$$a_{fv} x_v = y_f - \sum_{i\in n(f)\backslash v} a_{fi} x_i \qquad (6)$$

we can find the F2V message from y_f to x_v. Let the random variable $R = \sum_{i\in n(f)\backslash v} a_{fi} X_i$. Then (6) becomes

$$a_{fv} X_v = y_f - R$$

Let $a_{fi} X_i = E_i$. Then $P_{E_i}(e_i) = P_{a_{fi} X_i}\left(a_{fi} x_i\right)$ which is $a_{fi}-$times expanded version of $P_{X_i}\left(x_i\right)$. Then we can calculate

$$P_R(r) = \underset{i\in n(f)\backslash v}{\otimes} P_{E_i}(e_i)$$

where $\otimes$ is the convolution operator. After get $P_R(r)$, finally we can evaluate the F2V message,

$$P_{X_v}(x_v) = P_{\frac{y_f - R}{a_{fv}}}\left(\frac{y_f - r}{a_{fv}}\right)$$

which is a_{fi} – times shrunk version of $P_R(y_f - r)$.

Here, we need a function to treat expansion and shrinking of PMFs. The expansion and shrinking function can be defined like

$$\varepsilon(P(x), c) = \begin{cases} P\left(\frac{x}{c}\right), & where \quad \left\lceil \frac{x}{c} \right\rceil = \frac{x}{c} \\ 0, & where \quad \left\lceil \frac{x}{c} \right\rceil \neq \frac{x}{c} \end{cases}$$

where c is a scaling factor. If $c \geq 1$, $\varepsilon(P(x), c)$ acts as expansion function and if $c < 1$, $\varepsilon(P(x), c)$ acts as shrinking function. When it works as expansion function, it doesn't interpolate but zero-stuff at the new points because $P(x)$ is a discrete function. For example, let

$$P(x) = \begin{cases} 0.5 & x = 0 \\ 0.3 & x = 1 \\ 0.2 & x = 2 \end{cases},$$

and

$$P(z) = \varepsilon(P(x), c).$$

Then $P(z)$ will become

$$P(z)=\begin{cases}0.5 & z=0\\ 0 & z=1\\ 0.3 & z=2\\ 0 & z=3\\ 0.2 & z=4\end{cases}.$$

Since $P(x)$ is PMF, $P(x)$ has zero value at the non-integer points. At this example,

$$P(x=0.5)=0,$$

therefore,

$$P(z=2x=1)=P(x=0.5)=0.$$

(iii) Posterior update

By repeating (i) and (ii), messages are exchanged among all nodes. After all messages are converse, we can get the posterior PMF for each variable node by multiplying all F2V messages. The posterior PMF is proportional to the product of F2V messages. Finally, the posterior PMF becomes

$$P(x_v)=\frac{1}{Z}\prod_{j\in n(v)}\mu_{j\to v}$$

where Z is a normalization constant.

However, we have to consider the convergence issue at here. As mentioned in Section 1.2.2, we cannot guarantee the convergence of BP in tree-like structure with cycles. Furthermore, even if it converges, the speed of convergence is also can be a problem. The convergence speed of BP iteration directly effects on the running time of CS-WBP algorithm. Thus, we have to use termination condition to guarantee the completion of BP iteration and to control the convergence speed. And damping technique also can be used to improve convergence in the practical implementation. We used termination condition which limits the maximum iteration of BP and allow finishing the iteration in early step when the variation of messages is ignorable. Also we used damping technique at the update of F2V messages.

Here is the pseudo code of CS-WBP.

CS-WBP Algorithm
This algorithm iteratively calculates marginal posterior distributions of variable nodes, given graph, values of factor nodes, and prior distribution of each variable node. **Input:** Graph: $\mathbf{A} = \begin{pmatrix} a_{11} & \cdots & a_{1N} \\ \vdots & \ddots & \vdots \\ a_{N1} & \cdots & a_{MN} \end{pmatrix}$ Factor nodes: $\mathbf{y} = [y_1 \quad y_2 \quad \cdots \quad y_M]^T$ Prior distributions: $P_{X_i,prior}(x_i)$ **Output:** Posterior distributions of variable nodes: $P_{X_i}(x_i)$ **Definitions:**

$$\varepsilon(P(x),c) \triangleq \begin{cases} P\left(\frac{x}{c}\right), & \left\lceil \frac{x}{c} \right\rceil = \frac{x}{c} \\ 0, & \left\lceil \frac{x}{c} \right\rceil \neq \frac{x}{c} \end{cases}$$

$$termination \triangleq \{updated\ \ posterior \simeq old\ \ posterior \,\|\, iteration = \max\_iteration\}$$

$\alpha \triangleq$ damping coefficient, $0 < \alpha < 1$.

Initialize:

$$P_{X_i}(x_i) \leftarrow P_{X_i,prior}(x_i)$$

For all $v \in \{1,2,\ldots,N\}$ and $f \in \{1,2,\ldots,M\}$, $\mu_{v\to f} \leftarrow P_{X_v,prior}(x_v)$

Main Loop:

Repeat

For all $v \in \{1,2,\ldots,N\}$ and $f \in \{1,2,\ldots,M\}$,

$$P_{Z_{fv}}(z_{fv}) \leftarrow \underset{i\in n(f)\backslash v}{*} \varepsilon(\mu_{v\to f}, a_{fv})$$

$$\mu_{f\to v} \leftarrow \varepsilon\left(P_{Z_{fv}}(y_f - z_{fv}), \frac{1}{a_{fv}}\right)$$

$$\mu_{v\to f} \leftarrow \alpha\mu_{v\to f} + (1-\alpha)\prod_{j\in n(v)\backslash f} \mu_{j\to v}$$

while$\{\sim termination\}$

$$P_{X_i}(x_i) \leftarrow \prod_{j\in n(v)} \mu_{j\to v}$$

3. 3. 2. Computational complexity

The computational complexity of BP recovery is

$$O\left(N \log N l \log l\right)$$

where N is length of signal and l is length of messages. We need $O(l)$ computations to calculate a V2F message. In other hands, we need $O(l \log l)$ computations if we use fast convolution for F2V update. V2f and F2V update also have to be done for $O(N)$ times per one iteration, so overall complexity at an iteration becomes $O(Nl \log l)$. This iteration will be finished at $O(\log N)$ iterations in case of tree-like structure. [19] Overall, complexity of CS-WBP becomes $O(N \log N l \log l)$. If we consider the l as a constant, than it becomes $O(N \log N)$.

4. Simulation Results and Analysis

4.1. Simulation setup

To verify the performance of CS-WBP we conducted simulation experience with the algorithm. We used the non-negative integer valued signal with length $N = 500$ and the number of nonzero elements $K = 50$. i.e., we used signal of sparsity 0.1. The sensing matrix is generated by the product of irregular LDPC matrix with column degree of 5 and uniform discrete random mask in the interval $[1, a_{max}]$. i.e., The sensing matrix has randomly selected 5 nonzero values in a column and the nonzero values are uniformly distributed over integers within $[1, a_{max}]$. Note that we used a minor modification to prevent generations of matrix which contains “all-zero-rows”. If the sensing matrix has all-zero-rows, the net number of measurement decreased thereby causing the distortion of the results. The simulation was done for different $a_{\max}$ ’s, 1, 3, and 7. We use AWGN noisy measurements with Signal-to-Noise Ratio (SNR) conditions of 30dB and 50dB, and the simulation was done 100 times for each experimental point. As a parameter setting, we limited the maximum iteration by 30 times and used damping coefficient $\alpha = 0.5$. We calculated the needed number of measurements for perfect recovery using the simulation result, while considering the cases that Mean Squared Error (MSE) is smaller than 0.01 as the cases of perfect recovery. For reference, Non-negative Generalized AMP (NNGAMP) [Vila13] is compared. Actually, the NNGAMP is not a perfect partner for comparison because we didn’t use any mean-removal technique for the algorithm, just scaled the whole system so that the effective mean value of sensing matrix can be maintained while $a_{\max}$ is varying. The mean-removal technique or other technique for stable operation of AMP is a research issue which is receiving many attentions from many other researches like [33] and we also have interest. Thus, this performance comparison is not quite fair but it is worth observing the performance variation according to $a_{\max}$.

4.2. Simulation Result

Here is the simulation result at the SNR of 50dB. Figure 12 shows the MSE versus the number of measurements in a ratio to the length of signal according to the values of $a_{\max}$. As we can see, the MSE performance of CS-WBP gets better while $a_{\max}$ is increase. In the other hands, the performance of NNGAMP is stationary. This phenomenon is well displayed in the Table 1. The table shows the minimum measurements ratio for perfect recovery. The needed measurement ratio for perfect recovery by CS-WBP algorithm decreases from 0.28 to 0.26, and 0.24 while the $a_{\max}$ increases to 1, 3, and 7. This result means the performance of CS-WBP is improved by increasing $a_{\max}$ because smaller measurement ratio means more tough condition for recovery. i.e., CS-WBP algorithm with $a_{\max}=1$ cannot perfectly recovery the signal where the number of measurements is 120, while CS-WBP with $a_{\max}=7$ can. In other hand, the performance of NNGAMP is almost same regardless of $a_{\max}$.

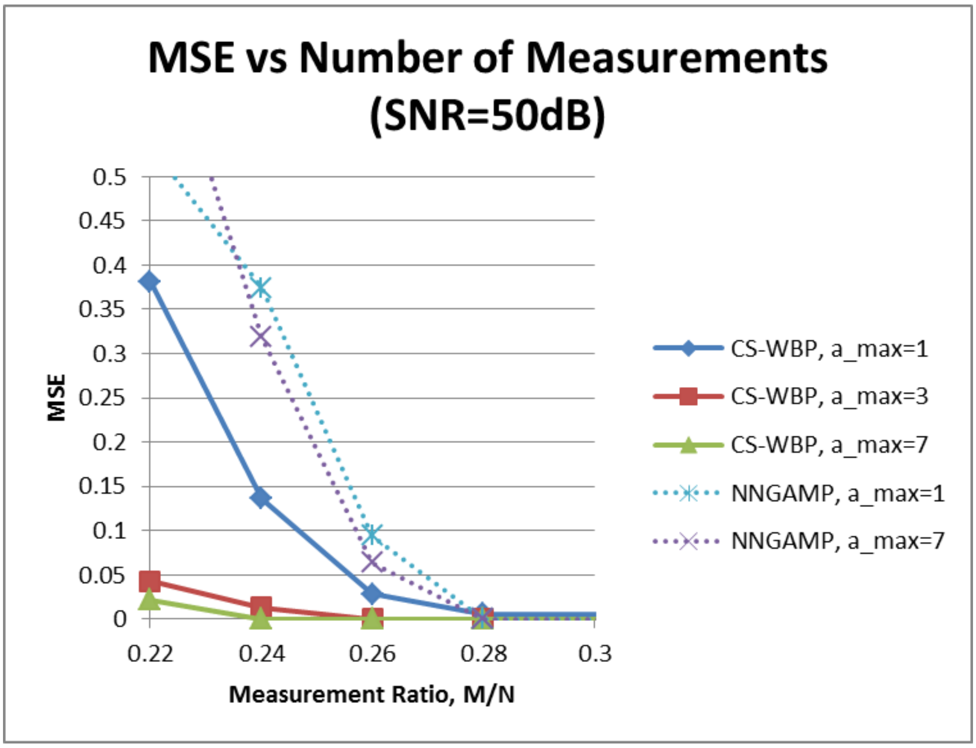

Figure 12 MSE versus number of measurements at SNR=50dB.

SNR=50dB	Measurements ratio for perfect recovery
CS-WBP ($a_{max}=1$)	0.28
CS-WBP ($a_{max}=3$)	0.26
CS-WBP ($a_{max}=7$)	0.24
NNGAMP ($a_{max}=1$)	0.28
NNGAMP ($a_{max}=7$)	0.28

Table 1 Measurements ratio for perfect recovery at SNR=50dB.

Here is the same simulation result but it is at the SNR of 30dB. Figure 13 shows the MSE versus the number of measurements in a ratio to the length of signal according to the values of a_{max}. As we can see, the MSE performance of CS-WBP gets worse while a_{max} is increase, which is completely opposite against the case of SNR=50dB. In the other hands, the performance of NNGAMP is stationary. It also can be seen in Table 2. The table shows the minimum measurements ratio for perfect recovery. The needed measurement ratio for perfect recovery by CS-WBP algorithm increases from 0.4 to 0.5, and 0.7 while the a_{max} increases to 1, 3, and 7. This result means the performance of CS-WBP degenerate by increasing a_{max}. In other hand, the performance of NNGAMP is almost same regardless of a_{max}.

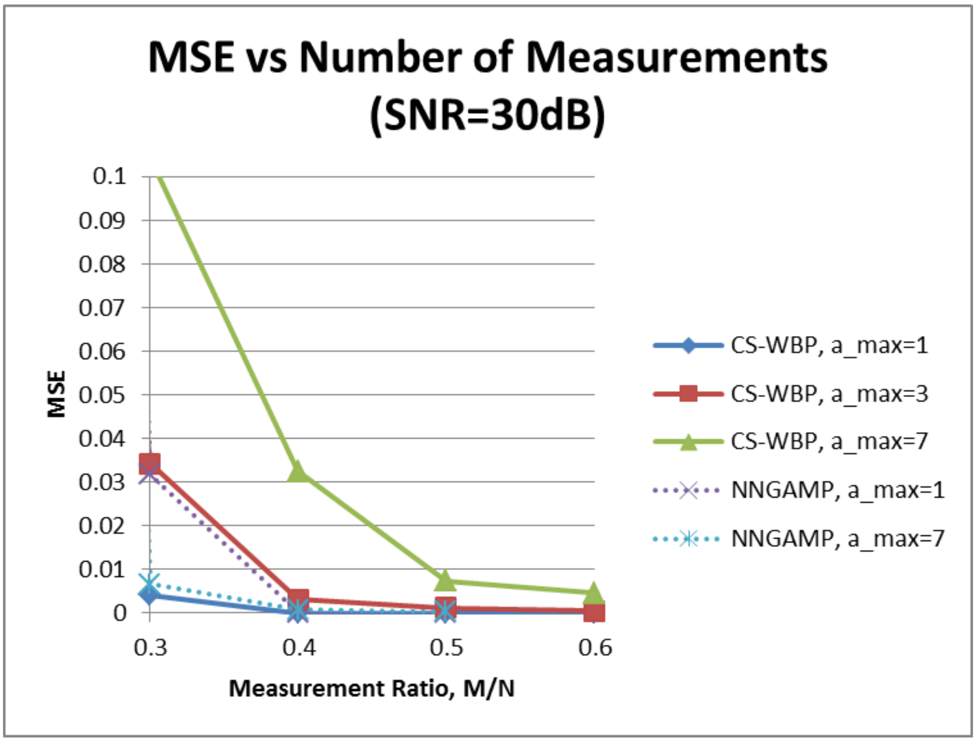

Figure 13 MSE versus number of measurements at SNR=30dB

SNR=30dB	Measurements ratio for perfect recovery
CS-WBP ($a_{\max} = 1$)	0.4
CS-WBP ($a_{\max} = 3$)	0.5
CS-WBP ($a_{\max} = 7$)	0.7
NNGAMP ($a_{\max} = 1$)	0.4
NNGAMP ($a_{\max} = 7$)	0.4

Table 2 Measurement ratio for perfect recovery at SNR=30dB.

4.3. Analysis

Overall, the recovery performance of CS-WBP increased where SNR=50dB and it decreased where SNR=30dB according to the $a_{\max}$. Then why these completely opposite results came out? First, let think about 50dB case. Actually, this case is almost same with noiseless case because we use the quantized measurements. If the intensity of all elements of noise vector are smaller than 0.5, the round-off version of noisy measurement and noiseless measurement will be same. We can calculate the probability that the event like this happens. Put the event E. Then the probability will become:

$$\Pr\{E\} = \Pr\{|w| < 0.5\}^M = erf\left(\frac{1}{2\sigma_w\sqrt{2}}\right)^M$$

where $erf(\cdot)$ is the error function which is defined as

$$erf(x) = \frac{2}{\sqrt{\pi}}\int_0^x e^{-t^2}dt\,.$$

When $x \geq 0$, $erf\left(\frac{1}{x}\right)$ is almost 1 until x reaches some threshold value. After x exceeds the threshold value, $erf\left(\frac{1}{x}\right)$ start to decrease. Figure 14 shows it well.

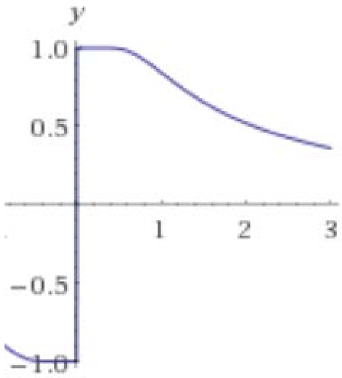

Figure 14 Plot of erf(1/x)

We decided this threshold value as a point that $erf\left(\frac{1}{x}\right) = 0.999$. Since $1/erf^{-1}(0.999) \approx 0.4$, the threshold value for σ_w become $\frac{\sqrt{2}}{10}$. i.e., if $\sigma_w \leq \frac{\sqrt{2}}{10}$, the system will works like noiseless system because

$$\Pr\{E\} = erf\left(\frac{1}{2\sigma_w\sqrt{2}}\right)^M \approx 1.$$

We can express the threshold value in terms of SNR. Since we defined the SNR as $E[(Y-w)^2]/\sigma_w^2$, we have to calculate $E[(Y-w)^2]$ first.

From the signal model

$$y_j = a_{j1}x_1 + a_{j2}x_2 + \cdots + a_{jN}x_N + w_j,$$

we can derive the random variable Y like as

$$
\begin{aligned}
(Y-w) &= B_1U_1X_1 + B_2U_2X_2 + \cdots + B_NU_NX_N \\
&= \sum_{k=1}^{Bi} U_k X_k
\end{aligned},
$$

where B is a Bernoulli random variable with success probability $\frac{L}{M}$, U is an uniform discrete random variable from the interval $[1, a_{\max}]$, and Bi is a binomial random variable with N times trials and success probability $\frac{L}{M}$. Then we can find $E[(Y-w)^2]$.

$$
\begin{aligned}
E[(Y-w)^2] &= E[N(BUX)^2] + 2E[B]^2E[U]^2E[X]^2\sum_{k=1}^{N-1}k] \\
&= NE[(BUX)^2] + 2\left(\frac{L}{M}\right)^2\left(\frac{a_{\max}}{2}\right)^2 E[X]^2\sum_{k=1}^{N-1}k) \\
&= \frac{LN}{M}\frac{(a_{\max}+1)(2a_{\max}+1)}{6}E[X^2] + \frac{a_{\max}^2L^2}{4M^2}E[X]^2N(N-1) \\
&= \frac{LN\lambda}{M}\frac{(a_{\max}+1)(2a_{\max}+1)}{6} + \frac{a_{\max}^2L^2\lambda^2N(N-1)}{4M^2} \\
&\approx \frac{a_{\max}^2L^2\lambda^2N^2}{4M^2}
\end{aligned}
$$

Then we can calculate the threshold SNR by

$$
SNR_{th} = \frac{E[(Y-w)^2]}{\sigma_{w,th}^2} \approx \frac{100a_{\max}^2L^2\lambda^2N^2}{8M^2}.
$$

For example, in our simulation setting,

$$
SNR_{th,a_{\max}=1} \approx 5000 = 37\text{dB}
$$

$$
SNR_{th,a_{\max}=3} \approx 50000 = 47\text{dB}
$$

$$SNR_{th,a_{\max}=7} \approx 250000 = 54\text{dB}$$

In conclusion, our system will operate like noiseless at higher SNR than this threshold; otherwise it will operate with noisy measurement. According to this threshold, the 50dB is almost noiseless case and 30dB is a clear noisy case. Then, let's think these two cases separately.

4.3.1. Noiseless case

In the noiseless case, we can present an information theoretic necessary condition for perfect recovery.

Condition 1. $H(X) \le \frac{M}{N} H(Y)$. i.e, the entropy of signal should be less than the entropy of measurement, where $\mathbf{x}$ and $\mathbf{y}$ are drawn from i.i.d. random distribution X and Y, respectively.

proof) Let's think about *typical decoder*. For the perfect recovery, the number of possible $\mathbf{y}$ should be larger or same with the number of possible set of $\mathbf{x}$. i.e., $|\mathcal{X}| \le |\mathcal{Y}|$. The cardinality of typical sets become asymptotically

$$|\mathcal{X}| = 2^{NH(X)}$$

and

$$|\mathcal{Y}| = 2^{MH(Y)}$$

where $N \to \infty$. Here, we can easily derive **Condition 1**.

Let's come back to our signal model. We can express the random variable Y like as:

$$Y = \sum_{k=1}^{B} U_k X_k$$

where U_k is an uniform discrete random variable from the interval $[1, a_{\max}]$ and B is a binomial random variable with N time of trials and success probability $\frac{L}{M}$. Here, if $a_{\max}$ increases, then the number of values Y can have also increases so the entropy of Y, $H(Y)$ increases. As a result, the sensing matrices with larger $a_{\max}$ is more likely to satisfy the **Condition 1** than the sensing matrices with smaller $a_{\max}$. i.e., the sensing matrices with larger $a_{\max}$ is better matrices than those with smaller $a_{\max}$. Thus, the upper bound of recovery performance will be higher at the larger $a_{\max}$. This inference is quite suitable to our observation for CS-WBP.

4.3.2. Noisy case

In the noisy case, we can explain the observed simulation result by Cramer-Rao Bound.

Cramer-Rao Lower Bound:

$$\mathrm{var}(\hat{x}) \geq \frac{1}{I(x)}$$

where $I(\cdot)$ is fisher information which is defined as

$$I(x) \triangleq -E\left[\frac{\partial^2 l(y;x)}{\partial x^2}\right]$$

and $l(y;x)$ is a log-likelihood function. Since

$$l(y;x) = -\frac{1}{2\sigma_w^2}(\mathbf{y} - \mathbf{Ax})^T(\mathbf{y} - \mathbf{Ax}) + c\,,$$

$$
\begin{aligned}
I(x) &= -E\left[\frac{\partial^2 l(y;x)}{\partial x^2}\right] = E\left(\frac{tr(\mathbf{A}^T\mathbf{A})}{\sigma_w^2}\right) \\
&= \frac{tr(E[\mathbf{A}^T\mathbf{A}])}{\sigma_w^2}
\end{aligned}
$$

Remind that

$$
SNR = \frac{E[(Y-w)^2]}{\sigma_w^2} \approx \frac{100a_{\max}^2 L^2 \lambda^2 N^2}{8M^2}.
$$

From here,

$$
\sigma_w^2 = \frac{100a_{\max}^2 L^2 \lambda^2 N^2}{8M^2 SNR}.
$$

And $tr(E[\mathbf{A}^T\mathbf{A}])$ is the sum of column powers in $\mathbf{A}$, which is

$$
tr(E[\mathbf{A}^T\mathbf{A}]) = LN\frac{(a_{\max}+1)(2a_{\max}+1)}{6}.
$$

Finally we can calculate the fisher information.

$$
\begin{aligned}
I(x) &= \frac{8LNM^2(a_{\max}+1)(2a_{\max}+1)SNR}{600L^2N^2\lambda^2 a_{\max}^2} \\
&= \frac{M^2(a_{\max}+1)(2a_{\max}+1)}{75LN\lambda^2 a_{\max}^2} SNR
\end{aligned}
$$

Then the Cramer-Rao Lower Bound becomes

$$
\begin{aligned}
\operatorname{var}(\hat{x}) &\geq \frac{1}{I(x)} \\
&\geq c\frac{a_{\max}^2}{(a_{\max}+1)(2a_{\max}+1)}
\end{aligned},
$$

which is an increasing function of $a_{\max}$. It means that at the noisy case, the minimum value of MSE grows with increasing $a_{\max}$.

5. Conclusion

In this thesis, we introduced a new nonparametric Belief Propagation based Compressive Sensing Recovery algorithm called Compressive Sensing via Weighted Belief Propagation (CS-WBP). Compressive Sensing (CS) is an emerging signal processing framework, which is consist of two parts, compressive signal acquisition and reconstruction of signal from compressive measurement. The main point of CS is that if a signal is sparse enough, the signal information can be compressively acquired by the random projection in measurement without loss of information. It is known that L1 norm minimization can recover the signal form measurement in polynomial time. Belief Propagation or Sum-Product Message Passing is an efficient methodology for sharing statistical information over graphical models. BP-based CS recovery algorithms have some advantages mainly due to the suitability with Bayesian approach. However, there are also some difficulty to implement BP-Based algorithms, which are sampling problem and computational complexity. Several BP-based algorithms, such as Approximate Message Passing (AMP), Compressive Sensing via Belief Propagation (CS-BP), and Bayesian Hypothesis Test using nonparametric Belief Propagation (BHT-BP). But they have limitations, AMP is hard to exploit Bayesian approach; in other hand CS-BP and BHT-BP have restriction about sensing matrix. Our proposed method, CS-WBP is overcome these limitations by extending the CS-BP algorithm so that it can work on more general sensing matrices than those of CS-BP. As a result, CS-WBP algorithm can exploit the prior information and the output of algorithm is form of probability distribution, which contains more information than just estimation value. Furthermore, CS-WBP can work on the sparse and weighted sensing matrices, which CS-BP cannot use. Performance of our algorithm is closely connected with SNR and $a_{\max}$, which defines the possible number of nonzero values that sensing matrix can have. If SNR is higher than threshold mentioned in Section 4, the recovery performance gets better with increasing $a_{\max}$. In other hand, if SNR is lower than threshold, the performance gets worse with increasing $a_{\max}$.

6. REFERENCES

[1] C. E. Shannon, “Communication in the Presence of Noise”, *Proc. Institure of Radio Engineers*, vol. 37, no. 1, pp.10-21, 1949.

[2] D. L. Donoho, “Compressed Sensing”, *IEEE Trans. on Information Theory*, vol 52, no. 4, 2006.

[3] E. J. Candes, J. Romberg and T. Tao, “Robust Uncertainty Principles: Exact Signal Reconstruction From Highly Incomplete Frequency Information”, *IEEE Trans. on Information Theory*, vol 52, no 2, 2006.

[4] E. J. Candes and T. Tao, “Near Optimal Signal Recovery from Random Projections: Universal Encoding Strategies?”, *IEEE Trans. on Information Theory*, vol 52, no.12, 2006

[5] S. S. Chen, D. L. Donoho and M. A. Saunders, “Atomic Decomposition by Basis Pursuit”, *SIAM J. Sci. Comput.*, vol. 20, no. 1, 1998.

[6] R. Tibshirani, “Regression Shrinkage and Selection via the Lasso”, *Journal of the Royal Statistical Society*, vol. 58, no. 1, 1996.

[7] Z. Zhang, Y. Xu, J. Yang, X. Li and D. Zhang, “A Survey of Sparse Representation: Algorithms and Applications”, *IEEE Access*, vol. 3, 2015.

[8] S. G. Mallat and Z. Zhang, “Matching Pursuits with Time-frequency Dictionaries”, *IEEE Trans. Signal Processing*, vol. 41, no. 12, 1993.

[9] J. A. Tropp and A. C. Gilbert, “Signal Recovery from Random Measurements via Orthogonal Matching Pursuit”, *IEEE Trans. on Information Theory*, vol. 53, no. 12, 2007.

[10] D. Needell and J. A. Tropp, “CoSaMP: Iterative Signal Recovery from Incomplete and Inaccurate Samples”, *Applied and Computational Harmonic Analysis*, vol. 26, no. 3, 2009.

[11] W. Dai and O. Milenkovic, “Subspace Pursuit for Compressive Sensing Signal Reconstruction”, *IEEE Trans. on Information Theory*, vol. 55, no. 5, 2009.

[12] D. L. Donoho, Y. Tsaig, I. Drori and J.-L. Starck, “Sparse Solution of Underdetermined Systems of Linear Equations by Stagewise Orthogonal Matching Pursuit”, *IEEE Trans. on Information Theory*, vol. 58, no. 2, 2012.

[13] S.-J. Kim, K. Koh, M. Lustig, S. Boyd and D. Gorinevsky, “An Interior-Point Method for Large-Scale L1-Regularized Least Squares”, *IEEE Journal of Selected Topics on Signal Processing*, vol. 1, no. 4, 2007.

[14] J. Yang and Y. Zhang, “Alternating Direction Algorithms for L1-Problems in Compressive Sensing”, *SIAM Journal on Scientific Computing*, vol. 33, no. 1, 2011.

[15] M. A. T. Figueiredo and R. D. Nowak, “A Bound Optimization Approach to Wavelet-Based Image Deconvolution”, *Proc. IEEE Int. Conf. on Image Processing*, vol. 2, 2005.

[16] A. Beck and M. Teboulle, “A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems”, *SIAM Journal on Image Science*, vol. 2, no. 1, 2009.

[17] F. R. Kschischang, B. J. Frey and H.-A. Loeliger, “Factor graphs and the sum-product algorithm”, *IEEE Trans. on Information Theory*, vol. 47, no. 2, 2001

[18] J. Pearl, "Reverend Bayes on inference engines: A distributed hierarchical approach" *Proceedings of the Second National Conference on Artificial Intelligence. Pittsburgh, PA.* pp. 133–136, 1982.

[19] B. J. Frey, and D. J. MacKay, “A revolution: Belief propagation in graphs with cycles,” *Proc. of the 11th Annual Conference on Neural Information Processing Systems (NIPS)*, pp. 479-485, Dec. 1997.

[20] R. G. Gallager, *“Low-Density Parity Check Codes”*, MIT Press: Cambridge, MA, 1963.

[21] M. G. Luby, M. Mitzenmacher, M. A. Shokrollahi, and D. A. Spielman, “Efficient Erasure Correcting Codes”, *IEEE Transactions on Information Theory*, vol. 47, no. 2, Feb. 2001.

[22] T. J. Richardson and R. L. Urbanke, “The Capacity of Low-Density Parity-Check Codes Under Message-Passing Decoding”, *IEEE Transactions on Information Theory*, vol. 47, no. 2, Feb. 2001.

[23] D. L. Donoho, A. Maleki, and A. Montanari, “Message passing algorithms for compressed sensing,” *Proc. National Academy Sciences*, vol. 106, no. 45, pp. 18914–18915, Sep. 2009.

[24] D. Baron, S. Sarvotham and R. G. Baraniuk, “Bayesian Compressive Sensing Via Belief Propagation”, *IEEE Trans. on Signal Processing*, Vol. 58, No. 1, pp.269-280, 2010.

[25] J. Kang, H.-N. Lee and K. Kim, "Bayesian Hypothesis Test using Nonparametric Belief Propagation for Noisy Sparse Recovery," *IEEE Transactions on Signal Processing*, vol. 63, no. 4, Jan. 2015.

[26] F. Caltagirone, L. Zdeborova and F. Krzakala, "On Convergence of Approximate Message Passing", *Proc. IEEE Int. Sym. on Information Theory*, 2014.

[27] S. Ragan, P. Schniter and A. Fletcher, "On the Convergence of Approximate Message Passing with Arbitrary Matrices", *Proc. IEEE Int. Sym. on Information Theory*, 2014.

[28] J. Vila, P. Schniter, S. Rangan, F. Krzakala, and L. Zdeborova, "Adaptive Damping and Mean Removal for the Generalized Approximate Message Passing Algorithm", *40th IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2015)*, 2015.

[29] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image Super-Resolution Via Sparse Representation", *IEEE Transactions on Image Processing*, vol. 19, no. 11, 2010.

[30] R. He, W.-S. Zheng, B.-G. Hu and X.-W. Kong, "Two-Stage Nonnegative Sparse Representation for Large-Scale Face Recognition", IEEE Transactions on Neural Networks and Learning Systems, vol. 24, no. 1, 2013.

[31] D. Takhar, J. N. Laska, M. B. Wakin, M. F. Duarte, D. Baron, S. Sarvotham, K. F. Kelly and R. G. Baraniuk, "A New Compressive Imaging Camera Architecture using Optical-Domain Compression", *Proc. IS&T/SPIE Computational Imaging IV*, 2006.

[32] G. Huang, H. Jiang, K. Matthews and P. Wilford, “Lensless Imaging By Compressive Sensing”, *IEEE Int. Conf. on Image Processing*, 2013.

[33] S. Rangan, P. Schniter, and A. Fletcher, “On the Convergence of Approximate Message Passing with Arbitrary Matrices”, *Proceeding on IEEE International Symposium on Information Theory*, 2014

[34] H. Jang, C. Yoon, E. Chung, W. Choi, and H. Lee, “Holistic random encoding for imaging through multimode fibers,” *Optics Express*, vol. 23, no. 5, 2015.

[35] J. Oliver, W. Lee, and H. Lee, “Filters with random transmittance for improving resolution in filter-array-based spectrometers,” *Optics Express*, vol. 21, no. 4, pp. 3969–3989, Feb. 2013.

[36] J. Pesquet, H. Krim, D. Leporini and E. Hamman, “Bayesian Approach to Best Basis Selection”, *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 5, pp 2634-2637, 1996.

7. Acknowledgement

아직도 광주가 낯설고, 기분은 신입생인데 벌써 2년이 지나 졸업의 때가 다가왔습니다. 붙임성도 부족하고 성격도 모난 제가 무사히 석사과정을 마칠 수 있었던 것은 많은 분들의 도움이 있어서였던 것 같습니다. 우선 많은 것을 가르쳐 주신 지도교수님이신 이흥노 교수님께 감사의 마음을 전하고 싶습니다. 운동시간이나 행사 때에도 함께하며 학문은 물론, 그 외적인 것들도 많이 배웠습니다. 또한 인포넷 연구실의 많은 식구들께도 많은 도움을 받았습니다. 환철이형, 진택이형, 웅비형, 영학이형, 상준이형, 승찬이형, 종목이형, 수길이형, 정민이형, 주성이형, 종목이형, 승윤이형, 재혁이형, 대영이, 그리고 제가 영어가 모자라 많은 대화를 하지 못했던 Oliver, Asif, Zafar, Pavel, Nitin, Yaseen, Rohit, 연구를 비롯해 많은 부분에서 도와주신 CSNL 연구실의 재욱이형, 연구실 일을 많이 덜어주신 현주씨, 수정씨, 흔쾌히 논문 심사를 맡아주신 안효성 교수님, 신종원 교수님, 그리고 그 외 광주과기원에서 스쳐가셨던 모든 분들께 이 지면을 빌려 감사 드립니다. 꼭 바쁠 때만 놀자고 불러내는 성산고등학교 동기녀석들, 학교는 다르지만 같이 대학원 생활 열심히 하자더니 혼자 자퇴해버린 진영이놈, 광주와서 크게 쏜다면서 광주 근처에는 오지도 않는 재호놈도 아주약간은 고맙다.

마지막으로 항상 저를 믿고 지지해주신 아버지, 어머니, 그리고 내동생 자영이 정말 고맙고 사랑합니다.