Available online at www.sciencedirect.com
ScienceDirect
Procedia Computer Science 00 (2011) 000–000
Procedia Computer Science
www.elsevier.com/locate/procedia

Distributed compressive sensing technique for data gathering in ireless Sensor Networks

Alireza Masoum*, Nirvana Meratnia, Paul J.M. Havinga

Pervasive Systems Group, University of Twente, Enschede, Netherlands

Abstract

Compressive sensing is a new technique utilized for energy efficient data gathering in wireless sensor networks. It is characterized by its simple encoding and complex decoding. The strength of compressive sensing is its ability to reconstruct sparse or compressible signals from small number of measurements without requiring any apriori knowledge about the signal structure. Considering the fact that wireless sensor nodes are often deployed densely, the correlation among them can be utilized for further compression. By utilizing this spatial correlation, we propose a joint sparsity-based compressive sensing technique in this paper. Our approach employs Bayesian inference to build probabilistic model of the signals and thereafter applies belief propagation algorithm as a decoding method to recover the common sparse signal. The simulation results show significant gain in terms of signal reconstruction accuracy and energy consumption of our approach compared with existing approaches.

Keywords: Wireless sensor networks; Compressive sensing; Sparsity; Belief propagation.

1. Introduction

Traditional data gathering approaches in wireless sensor networks (WSNs) transmit all measurements to the base station at the cost of energy and bandwidth. Considering the fact that transmitting all measurements in case of high sampling frequency is neither feasible in terms of resource consumption not useful in terms of data provision as environment does not change very frequently, in recent years adaptive sampling approaches have been utilized. Adaptive sampling approaches aim to only send those measurements, which indicate significant change in the environment. By doing so they aim to reduce communication and consequently to increase network lifetime. Some of these approaches also benefit from redundancy and spatial and temporal correlation of sensor nodes' readings to reduce number of transmitted measurements and to lower down the communication cost even further. Another approach to tackle the problem of high communication cost to transfer sensor measurements from the WSN to a base station is data compression. However, this technique suffers from a restriction imposed by Niquist-Shannon theory, which states that in order to accurately recover the compressed signal, signal must be sampled with frequency *(N)* higher or equal to twice of its maximum frequency of signal [2]. In most cases these number of samples are still too high for limited resources of wireless sensor networks. In addition, compression techniques require identifying the location of large coefficients. To overcome these restrictions, compressive sensing has been put forward.

Compressive sensing (CS) is a concept coming from signal processing field. The strength of compressive sensing is its ability to reconstruct sparse or compressible signal from small number of measurements without requiring any a priori knowledge about the signal structure. CS is advantageous whenever signals are sparse in a

* Corresponding author. Tel.: +31- 53- 489- 3629
E-mail address: A.Masoum@utwente.nl

known basis, measurements (computation at the sensor end) are expensive and computations at the receiver end are cheap [3]. These characteristics completely match WSNs. Compared with data compression, applying CS in WSNs offers promising improvements as low power sensor nodes are not generally suitable for implementing encoding of data compression techniques [4]. CS has a simple encoding procedure to be executed by sensor nodes while a complex decoding procedure will be executed by the base station. The number of measurements required by compressive sensing is much lower than Nyquist-Shannon rate and depends on the sparsity level of the signal. In addition, there is no need to post-process the samples to reduce data size. These properties have made CS an energy efficient data gathering technique which has low processing and transmission costs [5].

There are a number of prior works which investigate the effectiveness of conventional CS techniques in WSNs. Some of these works discuss on sensor readings projection techniques [6]. Authors in [7] build a data aggregation tree to acquire these projections while an adaptive compressive sensing technique proposed in [6]. Most of these techniques consider only temporal correlation among individual signals measured by single sensor node. In addition, some others address scenarios where there is either temporal and spatial correlation among different signals as in a WSNs. The most well-known CS technique proposed for correlated signals is the distributed compressive sensing technique (DCS) [8]. DCS introduces a greedy algorithm based joint signal recovery method which reconstruct different signals acquired by different sensor nodes in a WSN where these signals are assumed to satisfy some joint sparsity models. DCS technique in [9] employ Frechet mean of the signals to discover the common support of the sparse signal. Then it utilizes a new greedy algorithm, called precognition matching pursuit (PMP), to minimize the number of measurements.

In other hand, previous review articles in CS limit their base recovery algorithms to Linear programming and Greedy algorithm [10][11]. These techniques suffer from complexity, accuracy and speed problems. Bayesian CS (BCS) [12] is a technique which utilizes statistical characterization of the signal to complement the conventional methods. It can provide better performance in terms of accurate data reconstruction or reduced number of measurements. However, there is a few works in WSNs area which benefits from this technique to outperform their performance [13][14]. TC-CSBP [14] is a belief propagation (BP) based BCS technique which employ temporal correlation among sensor node readings to reconstruct the signal. However, this technique ignore spatial correlation among sensor nodes' readings.

With the present paper, we provide spatial correlation based distributed BCS method which exploit belief propagation algorithm[15][16] to reconstruct the original signal. Considering the spatial correlation among sensor nodes, the common sparsity profile for all sensor nodes' readings has been assumed. This common sparsity profile allows our approach to reconstruct the signal and support set together. To do so belief propagation (BP) technique is implemented on bipartite graph which utilize iterative message passing among the graph nodes to find the solution with high accuracy. These messages are the Gaussian probability density functions which provide posterior distribution of signals.

The rest of this paper is organized as follows. In Section 2 the fundamental of CS is introduced. Section 3 and 4 presents system model and our approach respectively. Section 5 presents the simulation scenarios and evaluation results. Finally in section 6 we end up with some conclusion.

2. Fundamental of compressive sensing

2.1. Compressive sensing theory

Compressive sensing states that sparse or compressible signals can be accurately or approximately recovered from a number of linear projections [3][17]. Sparse signal is a signal which naturally exhibits sparsity while compressible signal can be well approximated with sparse representation through transforming to another space, where a small number of the coefficients represent most of the power of the signals [17]. In what follows mathematical description of CS as presented in [3][17] is given:

Let us assume that a discrete signal $X \in R^N$ which is presented by $N \times 1$ column vector, has sparse representation in some basis such as Fourier or Wavelet. Considering this sparsity concept, this signal can be expressed in term of the basis as

$$X = \sum_{k=1}^{N} a_k \, \psi_k = \Psi a \qquad (1)$$

where Ψ is an $N \times N$ orthonormal basis matrix $\Psi = [\psi_1, \psi_2, \dots, \psi_N]$, ψ_k, $k = 1,2,\dots,N$ is a $N \times 1$ vector, and $a = [a_1, a_2, \dots, a_N]$ is the $N \times 1$ column vector of the coefficient sequence of X in Ψ domain.

Signal X is compressible or sparse in Ψ basis, if its coefficient vectors have a few large elements and many small or zero elements. In other words, most of the elements in a are zero. CS states that if signal X is *K-sparse* on Ψ basis, it can be captured and recovered from M non-adaptive, linear measurements ($K < M \ll N$) with a certain restriction. The sampled signal via CS is described as:

$$Y = \Phi X \qquad (2)$$

where $Y = [y_1, y_2, \dots, y_M]$ is $M \times 1$ measurement matrix, $\Phi = [\Phi_1, \Phi_2, \dots, \Phi_M]$ epresents a $M \times N$ sensing matrix and each $\Phi_i, i = 1,2,\dots,N$ is a $N \times 1$ vector. It must be mentioned that Φ is a random matrix which can be assumed as second basis. Each element $y_i$in measurement matrix is a product of vector X and a vector $\Phi_i$from sensing matrix. We can substitute Xwith $\Psi\mathbf{a}$ then we can rewrite y as:

$$Y = \Phi X = \Phi\Psi a = \Theta a \qquad (3)$$

where $\Theta = \Phi\Psi$ is a $M \times N$ matrix.

CS demonstrates that sparse signal can be recovered from M measurements if it can satisfy restricted isometric property (RIP). RIP states that Φ and Ψ must be incoherent, which means that the rows of Φ must not sparsely represent the columns of Ψ (and vice versa). Formally speaking, a matrix Θ of size M × N satisfies the RIP of order K if it can be the minimum number such that

$$(1 - \delta_k)\|a\|_2^2 \le \|\Theta a\|_2^2 \le (1 + \delta_k)\|a\|_2^2 \qquad (4)$$

where $\delta_k \in (0,1)$ is a restricted isometric constant (RIC). Equation (4) must be hold for all a with $\|a\|_0 \le K$, and $\|a\|_0$ is ℓ_0 norm which shows number of non-zero elements in a. ℓ_p norm of vector a is defined as:

$$\|a\|_p^p = \sum_{i=1}^{N} |a_i|^p \qquad (5)$$

RIP guarantees the exact recovery of x with high probability if $M \ge cK\, log\,\frac{N}{K}$ (6).

However, the recovery of the signal X form Y is an NP hard problem but it can be achieved through optimization. To do so, $\ell_1$minimization is widely used for CS signal reconstruction, while $\ell_0$minimization is computationally intractable. We can recover the coefficients of sparse signal a by solving ℓ_1 norm minimization as follows:

$$\acute{X} = \Psi\acute{a}\,; \quad \acute{a} = arg \min_{a\in R^N} \|a\|_{\ell_1} \quad s.t. \quad Y = \Phi X \qquad (7)$$

2.2. Distributed compressive sensing

Distributed compressive sensing is a technique which utilizes joint sparse signal recovery method to reconstruct the sparse signal. According to this technique, sparse representation of each signal consists of a common part and an innovative part [11][13]. In this model, all signals share a common sparse component while each individual signal contains a sparse innovation component.

$$X^i = Z_C^i + Z_{In}^i \qquad (8)$$

where the Z_C^i is common to the all of the X^i and its sparsity level is the minimum sparsity level of all signals in basis Ψ. The signals Z_{In}^i are the unique portions of the X^i and have its sparsity level in the same basis. In this technique, recovery process focus on reconstructing the common part as much as possible in order to do reconstruction more precisely. When the proportion of common part is far more than individual part, the reconstruction error decreases.

2.3. Belief propagation

Belief Propagation (BP) is an iterative message passing algorithm which can calculate the marginal distribution or find the estimates such as MAP and MMSE in Bayesian networks and Markov random fields [15][16]. In the sparse signal recovery area, BP runs on factor graphs and considered as fast decoder in Bayesian compressive sensing frameworks. This factor graph is a bipartite graph which provides a graphical representation of sparse signal recovery procedure. It consists of two disjoint nodes: variable nodes and connection (factor) nodes which are connected through undirected edges whenever there is a dependency between these nodes. According to BP, these edges contains probability distribution functions on the variable nodes. There is only two direction for edges between variable and connection nodes [16] as:

- Edges or messages from a variable node to the connection node
  This edges contain probability which is calculated by gathering the all incoming edges (excepts edges coming from node f) and multiplying them, which is described as follows:

$$\mu_{x\to C}(x) = \prod_{v\in N(x)\backslash\{C\}} \mu_{v\to x}(x) \qquad (9)$$

where $\mu_{v\to x}$ shows the edges from node v to node x and $N(x)$ denotes the neighbor nodes of x and $N(x)\backslash\{C\}$ denotes the neighbor nodes of x except for node C. As an example consider the Fig. 1 which shows the part of a factor graph. The message from the variable nodes X to the connection node c_m is given by:

$$\mu_{x\to c_m}(x) = \mu_{c_1\to x}(x) \times \mu_{c_2\to x}(x) \times \mu_{c_3\to x}(x) \times \ldots \times \mu_{c_{m-1}\to x}(x) \qquad (10)$$

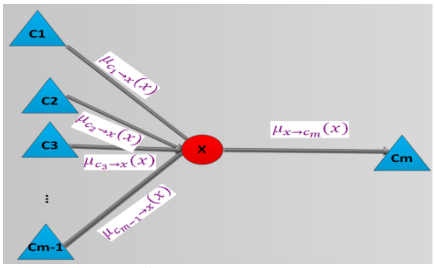

Fig. 1. Message going from variable node to connection node

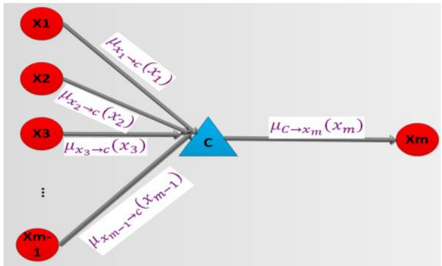

Fig. 2. Message going from connection node to variable node

- Edges from a connection node to the variable node
  This probability computed by obtaining the all incoming edges to the node f except the link from node x, multiplying them by f and finally finding the sum of all connected variable nodes except for the node x. In general, the edges going from the connection nodes to the variable node can be described as follows:

$$\mu_{C\to x}(x) = \sum_{\sim\{x\}} C\big(N(C)\big) \prod_{v\in N(C)\backslash\{x\}} \mu_{v\to C}(v) \qquad (11)$$

  where $N(C)$ is the all variable nodes connected to the f and $\sum$ is the sum over all connected variables except for x. $\mu_{v\to C}(v)$ shows the edges going from variable node v to the connection node C. As an example consider the Fig. 2 which shows the part of a factor graph. The message from the connection nodes C to the variable node x_m is given by:

$$\mu_{C\to x_m}(x_m) = \sum_{x_1,x_2,x_3,\ldots,x_{m-1}} C(x_1,x_2,x_3,\ldots,x_{m-1}) \times \mu_{x_1\to c}(x_1) \times \mu_{x_2\to c}(x_2) \times \mu_{x_3\to c}(x_3) \times \ldots \times \mu_{x_{m-1}\to c}(x_{m-1}) \qquad (12)$$

3. System model

We assume a network consisting of N static homogeneous wireless sensor nodes deployed densely. These nodes are in charge to accurately monitor the area in which they are deployed and to transmit their measurements to the sink node. The sink node is a high performance computation unit which has enough computational and power resources. Upon receiving data from sensor nodes, sink node must reconstruct and present the state of the environment with certain level of accuracy. Since sensor nodes have very limited resources, it is essential to gather and transfer as few data as possible. To do so, we employ compressive sensing technique to accurately reconstruct the state of the environment at the sink from as few as possible measurements.

3.1. Signal model

In order to model signal, we start with single sensor node. We consider the phenomenon to be monitored as a signal which is sampled by sensor nodes. Following formula (2), this phenomenon can be modeled as:

$$Y^i = \boldsymbol{\Phi}^i X^i \qquad (13)$$

where Y^i is the $M \times 1$measurement vector, X^i is the monitored signal coefficients or signal representation vector and $\boldsymbol{\Phi}^i$ is a $M \times N$random measurement matrix. We generate the elements of the sensing matrix using a random Bernouli distribution.

We assume that a set of nodes denoted by P directly transfer their measurement vector Y^i to the sink node. Therefore, sink node gather P measurement vectors. Since we have assumed a dense network, there is spatial correlation among sensor nodes. For all measurement vectors gathered at the sink node, we redefine measurements and signal representation as follows:

$$Y = \begin{bmatrix} y_{11}^1 & \cdots & y_{1P}^P \\ \vdots & \ddots & \vdots \\ y_{M1}^1 & \cdots & y_{MP}^P \end{bmatrix} \quad (14) \qquad X = \begin{bmatrix} x_{11}^1 & \cdots & x_{1P}^P \\ \vdots & \ddots & \vdots \\ x_{N1}^1 & \cdots & x_{NP}^P \end{bmatrix} \quad (15)$$

where each column of these matrixes corresponds to measurements of one sensor node and its signal representation vectors. Each signal X^iconsists of a small number of large elements and a large number of small elements. To distinguish between large and small elements, we define a status vector for each signal to show whether each coefficient of this signal is large or small.

$$S^i = \{s_1, s_2, \dots, s_N\} \quad (16)$$

For the collection of P correlated signals gathered from P sensor nodes, we have state matrix as follow:

$$S = \begin{bmatrix} s_{11}^1 & \cdots & s_{1P}^P \\ \vdots & \ddots & \vdots \\ s_{N1}^1 & \cdots & s_{NP}^P \end{bmatrix} \quad (17)$$

Each column of this matrix, shows the state variables for each sensor node and each state variable $s_{np}^i$is a binary variable ($s_{np}^i \in \{0,1\}$) which takes the value 0 when the corresponding element has a small magnitude, and value 1 when the element has a large magnitude.

$$s_{np}^i = \begin{cases} 0 \ , & if\ x_{np}^i\ is\ Small \\ 1 \ , & if\ x_{np}^i\ is\ Large \end{cases} \quad (18)$$

We utilize Gaussian distribution to associate the probability function for each coefficient of the signal. For $s_{np}^i = 1$, we select a high variance zero mean Gaussian distribution and for s_{np}^i=0, we choose low variance zero mean Gaussian distribution. Therefore, the conditional probability of z_{ik}on s_{np}^ican be represented as:

$$p\left(x_{np}^i \middle| s_{np}^i\right) = \begin{cases} p\left(x_{np}^i \middle| s_{np}^i = 0\right) \sim N(\mu_0, \sigma_0^2\ ) \\ p\left(x_{np}^i \middle| s_{np}^i = 1\right) \sim N(\mu_1, \sigma_1^2\ ) \end{cases} \quad (19)$$

We refer K and N as the signal sparsity level and the signal dimension respectively and define support set of the signal as a set of the positions where element of the signal are non-zero (large).

$$SSet^i = \{b_1, b_1, \dots, b_k\} \qquad (20)$$

Since we use distributed compressive sensing technique in our approach, we can change multiple measurement representation to the single measurement representation. According (8), the common sparisty part is same among all sensor nodes. Therefore, for this part, our multiple measurement vectors reduce to the single vector and we have a common measurement basis matrix and a single state vector for all sensor nodes readings.

$$\Phi = \Phi^i \quad (21) \qquad S_{N\times P} \to S_{N\times 1} \quad (22)$$

3.2. Signal recovery

We consider sparse signal recovery as a Bayesian inference problem. Therefore the recovery of the signal can be achieved by maximizing the posterior probability (MAP) estimation as follows:

$$\hat{X}, \hat{S} = \arg\max_{z,s} f(X, S|Y) \quad (23)$$

This representation is the signal wise MAP estimator which we resort it to the components wise MAP estimator to be able to solve it through the sum-product algorithm as follows:

$$\hat{X} = \arg\max_{X,S} \sum f(X|Y, Y)\, f(S|Y) \quad (24)$$

Substituting f with probability function, we will have:

$$\hat{X} = \arg\max_{X,S} \sum P(X|Y, S)\, P(S|Y) \quad (25)$$

During recovery process, sink node follows two main goals: estimating the signal elements and detecting the support set of the signal. In order to detect the support set, sink node utilizes posterior distribution of s_{np}^{i} to find the following hypothesis test:

$$\frac{P(s_{np}^{i}=1|X)}{P(s_{np}^{i}=0|X)} \quad (26)$$

4. Our proposed distributed compressive sensing

Given an already densely deployed WSN, sensor nodes regularly utilize compressive sensing technique to encode their readings and send it to the sink node. Upon receiving this information by the sink node, it decodes this information and reconstruct sensor nodes' readings which will be explained in following sections. To do so, sink node needs to utilize recovery algorithm to build sensor nodes readings. Greedy and linear programming are the most common recovery algorithms utilized in WSNs[5][6]. Considering statistical characteristic of the signals, Bayesian based recovery approaches can complement conventional CS methods based on linear programming or greedy algorithms. The recovery method proposed here is a joint sparsity-based compressive sensing technique which consider this property. According to this method, sink node employs Bayesian inference to build probabilistic model of the signals and thereafter applies belief propagation as a decoding method to recover the sparse signal. We organize the network activities into several rounds. This means that the base station runs recovery algorithms at intervals of a round time unit. In what follows, we focus on the recovery algorithms the base station runs to reconstruct the sensor nodes measurements.

Since our signal recovery algorithm utilizes joint sparse signal modeling, we will execute signal recovery process two times. First we will recover the common part of the signal and then individual parts will be reconstructed by taking the result of common recovery part results.

4.1. Common Part Recovery

Proposed belief propagation based approach here, utilizes graphical representation to reconstruct the signal. This graphical representation is based on factor graph representation. Before describing graph representation, we need to find prior probability distribution of signal elements which is considered as an input to the factor graph. Each state variable is supposed as a Bernouli random variable with$P(s_i = 1) = \beta$. Since signal is K sparse, we can assume β=K/N. Therefore, the prior probability distribution of each state variable is stated as follows:

$$p(s_i) = \begin{cases} \beta = \dfrac{K}{N} & s_i = 1 \\ 1-\beta = 1-\dfrac{K}{N} & s_i = 0 \end{cases} \quad (27)$$

Poposed factor graph presented in Fig. 3 is a bipartite graph $G = (VN, CN, E)$cosisting of variable nodes VNs, connection nodes CNs and edges Es. As it mentioned in previous section, this factor graph representation allow us to find marginal probability distribution easily. To build this graph, we consider variable nodes and connection nodes and connect these node through the undirected edges when a connection node depends on a variable node. This representation allow us to calculate the marginal distribution functions by message passing between variable nodes and connection nodes.

Our graph consists of two sub-graphs which have common nodes at VN_2 which contain signal coefficients $\{z_i\}$: The first sub-graph (sub-graph 1) computes the approximate posterior marginal distribution $\{z_i\}$. Selecting appropriate probability distribution functions in this section, guarantees low sparsity solution. The second sub-graph (sub-graph 2), provides approximation for the signal elements and estimates signal sparsity level. For the sub-graph 1, two variable nodes and one connection node has been defined: state variables VN_1 and signal coefficients VN_2 , are the variable nodes, while $CN_1$is the connection node which provides a link between state variables S and signal elements Z.

By running belief propagation technique over this graph, the edges going from the variable node to the connection node provides belief about the current estimation of the signal coefficients. This belief will be used to update the probability about the signal sparsity level later. Therefore, VN_1sends the distribution of each state variable $p(s_i)$ to CN_1. Then CN_1 finds the Gaussian distribution of z_{ik} by marginalizing signal elements based on state variables as follows:

$$\mathrm{CN}_1(z_{ik}, s_i) \sim p(z_{ik}|s_i) \quad (28)$$

Now it is necessary to find the edges going from the connection nodes to the variable node. According to (11), this edge can be calculated as follows:

$$\mu_{CN_1 \to Z}(z) = \sum_{\sim(z)} CN_1\big(N(CN_1)\big) \prod_{v \epsilon N(CN_1)\backslash\{Z\}} \mu_{v \to CN_1}(v) \quad (29)$$

where $N(CN_1)$is the all variable nodes connected to the CN_1 and $\sum$ is the sum over all connected variables except for z. $\mu_{v\to CN_1}(v)$shows the edges going from variable node v to the connection node CN_1.

BP states that edges going from connection nodes to the signal elements nodes represent the belief about sparsity level. In fact, this edge carry two Gaussian distributions for zero and non-zero coefficient of signal. Therefore, CN_1 calculates the mixture Gaussian distribution of signal elements and send the parameters of this distribution to the VN_2. Upon receiving these parameters, VN_2 calculates the prior probability distribution of z_{ik} as follows:

$$p(z_{ik}|s_i) = \beta \times p(z_{ik}|s_i = 0) + (1-\beta) \times p(z_{ik}|s_i = 1) = \ \beta \times N(\mu_1, \sigma_1^2\ ) + (1-\beta) \times N(\mu_0, \sigma_0^2\ )$$
$$= \ \beta \times N(\mu_1, \sigma_1^2\ ) + (1-\beta) \times \delta(z_{ik}) \qquad (30)$$

where $\delta(x)$ is a Dirac distribution function and $\int \delta(x)\, dx = 1$.

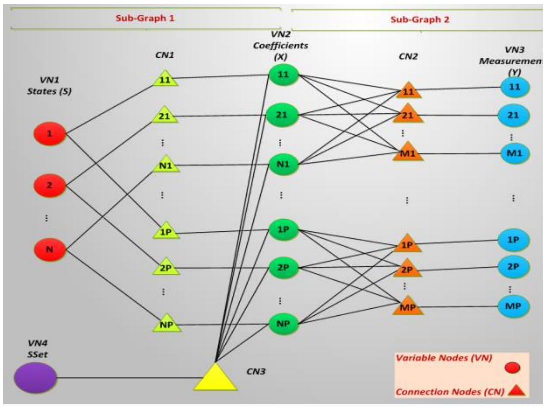

Fig. 3. Factor graph representation

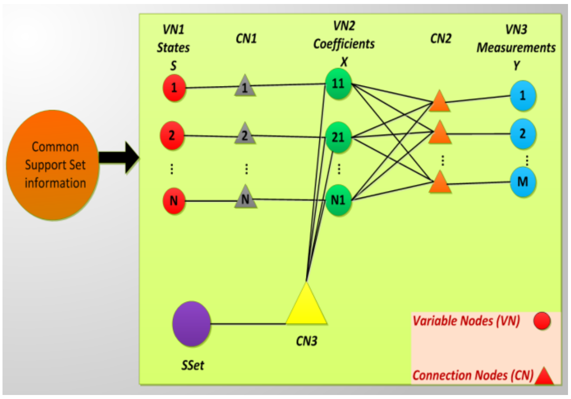

Fig. 4. Factor graph representation for individual part

The second sub-graph located in the right side of this factor graph has two variable nodes and one connection node: VN_2 which is common with sub-graph 1, measurement variables VN_3 are the variable nodes while $CN_2$is the connection node which provides a link between $VN_2$and VN_3. This sub-graph is in charge of calculating marginal distribution of signal elements. According to the formula (9), this distribution can be calculated by multiplying all incoming message to the variable nodes as follow:

$$CN_2(z) = \prod_{u \in N(z)} \mu_{u \to z}(z) \quad (31)$$

The connection node CN_2 receives all these incoming edges from variable node VN_2. For this connection node, the edge coming from VN_2 provides belief about the current state of signal elements. The incoming edges are the mixture of Gaussian densities. In fact, each member of VN_2broadcasts its Gaussian density to all connection nodes involved in its measurement. $\mathrm{CN}_2$is a delta function node which provides relationship between signal observation variable node VN_3 and signal coefficient variable nodes VN_2. This connection node is defined as follow:

$$CN_2(Z^k, y_{ik}) = \delta(y_{ik} - \sum_i a_{ik} z_{jk}), \qquad z_{ik} \in Z^k \quad (32)$$

For each variable node $\mathrm{z_{ik}}$, variable node $\mathrm{y_{ik}}$ multiply delta function of connection node with other density functions received from variable nodes $\mathrm{y_{ik}}$ to find the marginal distribution of each variable as follow:

$$P(z_{ik}|Y) = CN_{ik}(Z^k, y_{ik}) \times \prod_{i \neq j} \mu_{z_{jk} \to ZY_{ik}}(z_{jk}) \quad (33)$$

This marginal distribution will be calculated in each iteration of algorithm till it converges. Since the signal coefficients are independent, the joint distribution of the signal Z can be calculated as:

$$p(Z^k|Y) = \prod_{i=1}^{N} p(z_{ik}|Y) \qquad (34)$$

Therefore, each connection node gathers all incoming edges (Gaussian densities) which are simply the multiplication of all incoming links to the connection node.

4.1.1. Support set detection

The main goal of the second part of the graph is finding the support set of signal. To do so, the sink node requires finding the posterior distribution of state variables from sensor observations by calculating the following probability ratio:

$$SSet = \left\{ i: \quad \frac{p(s_i = 1|Y)}{p(s_i = 0|Y)} \geq 1 \right\} \quad (35)$$

Factorizing over z_{ik} , this probability ratio becomes

$$\frac{p(s_i = 1|Y)}{p(s_i = 0|Y)} = \frac{\int p(s_i = 1|Y, z_{ik})p(z_{ik}|Y)}{\int p(s_i = 0|Y, z_{ik})p(z_{ik}|Y)} \qquad (36)$$

This part of graph consists of one connection node $CN_3$and two variable nodes VN_3, VN_4 . In each iteration of the algorithm, variable nodes $VN_3 = Y$ pass their marginal distribution values to the variable nodes $VN_2 = Z$. As it mentioned in (30), variable node VN_2 calculates the Gaussian density of all incoming edges through multiplying these messages. Then, the link between variable node z_{ik} and connection node CN_1 pass the parameters of this function to the connection node. These edges provide the second parameter of the aforementioned probability ratio.

The first part of this ratio has been calculated through connection node CN_3. Considering the Bayesian rules and the prior probability distribution, the connection node that links signal elements to the support set model of the signal is described as follow:

$$CN_3(|SSet|, Z) = p(|SSet|\ |Z) = \frac{P(Z|\ |SSet|) \times P(|SSet|)}{P(Z)} \quad (37)$$

where $|SSet|$ shows the cardinality of the recovered support detection set. Since over measurement ratio is more than sparity level K, we need to minimize this cardinality in our calculations. In (30), we already calculated $p(|SSet|\ |Z)$ but in this section there is $R = \binom{N}{K}$ possible sets which can play role of the support set, where K is the cardinality of support set. Since support set has K non-zero elements, there will be $\binom{N}{K}$ possibility for selecting support set. We define $\mathrm{RSet} = \{\mathrm{PSSet}_1, \mathrm{PSSet}_2, \dots, \mathrm{PSSet}_R\}$ as all possible support sets with K cardinality, while $\overline{\mathrm{RSet}}$ is the all possible sets of non-support set. Considering these sets, in order to find $P(Z \mid |SSet|)$, we need to calculate two marginal distributions: One for the elements of each support set PSSet_i (non-zero elements) and another for the elements of $\overline{\mathrm{PSSet}_l}$ (zero elements). Marginal distribution for the elements of PSSet_i is the Gaussian distribution defined as $p(Z_{ik}|s_i = 1)$ while for the elements of $\overline{\mathrm{PSSet}_l}$ this distribution is defined as $p\left(Z_{jk}\middle|s_j = 0\right)$. For each candidate support set, connection node CN_3 multiply the marginal distribution of these elements to find following probability:

$$p(Z^k|\ |PSSet_l|) = \prod_{i \in PSSet_l} p(Z_{ik}|s_i = 1) \times \prod_{j \in \overline{PSSet_l}} p\left(Z_{jk}\middle|s_j = 0\right) \quad (38)$$

Finally for the all member of RSet, connection node CN_3 adds all these marginal distribution as follow:

$$p(Z^k|\ |SSet|) = \sum_{l=1}^{\binom{N}{K}} p(Z^k|\ |PSSet_l|) \quad (39)$$

Since all candidate support sets has same probability to act as a main support set, therefore, $\mathrm{P}(|\mathrm{SSet}| = 1/N$. Finally, $\mathrm{P}(\mathrm{Z}^{\mathrm{k}})$ can be calculated through (29). At the end, connection node CN_3 send $\mathrm{p}\left(\mathrm{Z}^{\mathrm{k}}\middle|\ |\mathrm{SSet}|\right)$ and $\mathrm{p}\left(\mathrm{Z}^{\mathrm{k}}\middle|\mathrm{Y}\right)$ to the variable node SSet. According to(11), this variable node calculates the approximate posterior distribution as follow:

$$p(|SSet||Y) = \sum_{z_{1k}} \sum_{z_{2k}} \cdots \sum_{z_{Nk}} (p(\ |SSet||Z^k) \times \prod_{i=1}^{N} p(z_{ik}\ |Y)) \ \ (40)$$

These probabilities exchange for a certain number of iterations. At the end, sink utilizes these marginal distributions to find the best support set according to (35). After finding support set, base station re-initializes the algorithm according to the support set. To do so, SSest reinitialize the prior distribution of signal coefficients through detected support set. Then, VN_2 and VN_3 iteratively exchange probability functions through CN_2 to find marginal posteriors which led to reconstruct the signal Z.

4.2. Individual part Recovery

After finding the common part, now sink node has to find the individual parts. To do so, it executes the previous model but only run the simplified version of belief propagation approach which is only for single sensor measurements. This simplified version has some input from the common parts which provide information about the common support set. The other details are as same as common recovery part. This simplified version is described in Fig. 4.
Fig. 5 shows the pseudo-code of the proposed algorithm.

1. For all state variables S find prior distribution of each state variable through (28)
2. Compute edges going from state variables S to the coefficients Z from (30)
3. Find prior distribution of signal coefficients Z $p(z_{ik}|s_i)$ through 31
4. Do
   - 4.1 for all measurements Y: compute messages $\mu_{CN_2 \to z}(x)$ using (11)
   - 4.2 for all coefficients: compute messages $\mu_{x_i \to c}(x_i)$using (10)
   - 4.3 for all coefficients find the marginal distributions $P(z_{ik}|Y)$using (34)
   - 4.4 Compute the conditional distribution for SSet $p(|SSet||Y)$ using (41)
   - 4.5 Update the signal's elements priors

   While it converges
5. Produce the estimated sparsity level and Z

Fig. 5. pseudo-code

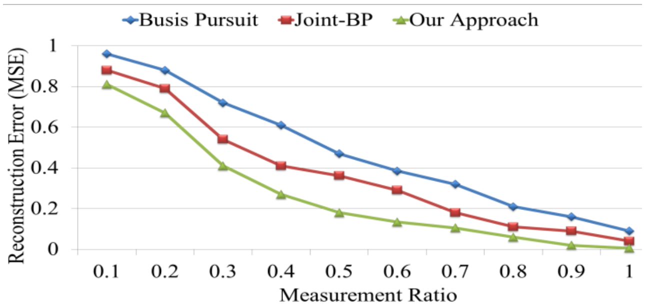

Fig. 6. Reconstruction error versus measurement ratio

5. Simulation

In this section, we discuss the performance of our algorithm in terms of reconstruction error, support set detection accuracy and energy consumption. We consider a network of 10 sensor nodes arranged in a star topology. In this network, sink node is located in the center of area and each sensor node has direct access to the sink. We compare performance of our algorithm with Basis Pursuit [18] and Joint-BP[19].

5.1. Reconstruction Error

In order to measure the accuracy, we calculate minimum square error (MSE). Fig. 6 shows our reconstruction error as a function of measurement ratio $\frac{m}{N}$ for all approaches. As it can clearly be seen, Joint BP and our approach outperforms Basis Pursuit reconstruction algorithm. For small to moderate measurement ratios, our approach clearly outperforms. The difference with the other two approaches becomes less significant as the measurement rate decreases.

5.2. Support detection set accuracy

We define support error rate as $SER = \frac{\sum_{i=1}^{N}((\hat{s_i} \notin \widehat{SSet} | s_l \in SSet) \, || (\hat{s_i} \in | s_l \notin SSet))}{N}$ where $\hat{s_i}$ is the member of detected support set $\widehat{SSet}$ and s_l is the member of original support set $SSet$. It measures the error rate between detected support set $\widehat{SSet}$ and original support set $SSet$. Considering SER , support detection accuracy parameter $SDA = 1 - SER$ and utilized to evaluate the accuracy of different algorithms. Fig. 7 depicts support detection accuracy level as a function of measurements ratio. Our approach provides the best accuracy compared with the other two approaches. There are two main reasons for this: firstly, our approach shares common sparse set among different sensor nodes and attempts to recover this set utilizing spatial correlation among sensor nodes. In addition, it utilizes statistical parameters to find accurate posterior distribution functions, which leads to accurate support set detection.

5.3. Energy Consumption

In order to compare proposed methods in terms of energy consumption, we define relative energy consumption as $REC = \frac{\sum_{i=1}^{p} E_p}{\sum_{i=1}^{p} E_t}$ [6] which is the ratio of overall energy consumption among all nodes to run CS over the consumed energy for sending all measurements without applying CS. For same reconstruction error rate, energy consumption among different algorithms has been compared in Fig. 8. Our approach requires fewer measurements compared to others to obtain the same reconstruction error. In addition it has less data to transmit therefore it provides minimum energy consumption rate.

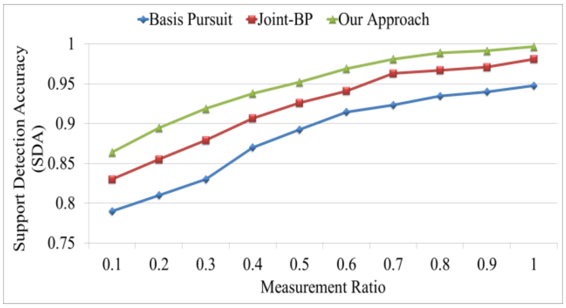

Fig. 7. Support detection accuracy versus measurement ratio

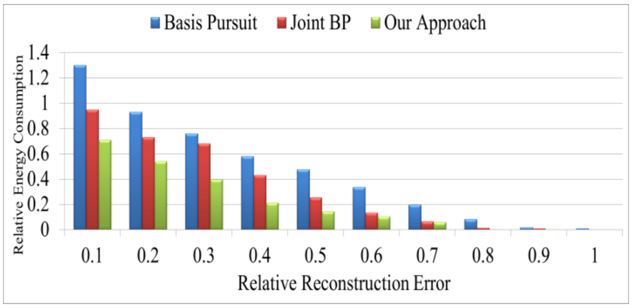

Fig. 8. Relative energy consumption versus reconstruction error rate

6. Conclusion

In this paper a new data gathering approach proposed which collect sensing data from the environment through Joint sparsity based Bayesian compressive sensing taking data accuracy and energy consumption into account. In this Bayesian inference recovery framework, belief propagation algorithm has been employed to compress and reconstruct the spatially correlated signals. This technique is implemented on bipartite graph which utilizes iterative message passing among the graph nodes to find the solution with high accuracy. Simulation results show that our algorithm outperforms Busis Pursuit and Joint-BP in terms of data reconstruction accuracy and energy consumption.

References

[1] G. J. Yick, B. Mukherjee and D. Ghosal, ”Wireless sensor network survey”, Computer Networks, 2008, vol 52, no 12, pp 2292-2330 .

[2] J. T. Srisooksai, K. Keamarungsi, P. Lamsrichan and K. Araki, “Practical data compression in wireless sensor networks: A survey”; Journal of Network and Computer Applications, Volume 35, Issue 1, 2012, Pages 37-59

[3] S. Emmanuel, J. Candès and M. B. Wakin, ”An Introduction To Compressive Sampling” IEEE Signal Processing Magazine, Volume: 25 , Issue: 2, 2008, Pages: 21 – 30.

[4] W. Bajwa, J. Haupt, A. Sayeed and R. Nowak, “Compressive Wireless Sensing” Proceedings of the IPSN, 2006, Page 134-142

[5] Y. G. Quer, R. Masiero, M. Rossi and M. Zorzi, “Sensing, Compression and Recovery for Wireless Sensor Networks: Monitoring Framework Design”,Wireless Communications, IEEE Transactions, Volume: 11, 2012, Page(s): 3447 - 3461.

[6] C. T. Chou, R. Rana, and W. Hu, “Energy efficient information collection in wireless sensor networks using adaptive compressive sensing,” in Proc. IEEE 34th Conf. Local Comput. Netw., Oct. 2009, pp. 443–450.

[7] G. Quer, R. Masiero, D. Munaretto, M. Rossi, J. Widmer, and M. Zorzi, “On the interplay between routing and signal representation for compressive sensing in wireless sensor networks,” in Proc. Inf. Theory Appl.Workshop, 2009, pp. 206–215.

[8] D BARON, M F DUARTE, S SARVOTHAM, M B WAKIN, R G BARANIUK, “Distributed compressed sensing of jointly sparse signals,” in Proc. 39th Asilomar Conf. Signals, Systems and Computers. Pacific Grove, CA, 2005.1537–1541.

[9] W. Chen, M. Rodrigues and I. Wassell, "Distributed Compressive Sensing Reconstruction via Common Support Discovery," in IEEE International Conference on Communications (ICC 2011), Kyoto, Japan, Jun 2011.

[10] D. Needell and J. A. Tropp, “CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,” Appl. Comput. Harmonic Analysis, vol. 26, no. 3, pp. 301–321, 2008.

[11] W. Dai and O. Milenkovic, “Subspace pursuit for compressive sensing: Closing the gap between performance and complexity,” IEEE Trans. Inf. Theory, vol. 55, no. 5, pp. 2230–2249, May 2009.

[12] Sh. Ji, Y. Xue, and L. Carin, *“Bayesian Compressive Sensing”* IEEE Transaction on signal processing, 2008, vol 56, no 6. 2008

[13] R Masiero, G Quer, M Rossi, M Zorzi,“ A Bayesian analysis of compressive sensing data recovery in wireless sensor networks “Ultra Modern Telecommunications & Workshops. ICUMT '09. Page(s): 1 – 6

[14] B. Shahrasbi, A. Talari, N. Rahnavard. TC-CSBP:Compressive sensing for time-correlated data based on belief propagation. In *45st Annual Conference on Information Sciences and Systems, CISS, USA, 2011*. Pp. 1-6, IEEE, 2011

[15] D.L. Donoho, A. Maleki, A. Montanari, “Message passing algorithms for compressed sensing: I. motivation and construction” Information Theory Workshop (ITW), 2010

[16] D. Baron, Sh. Sarvotham and R. G. Baraniuk, “Bayesian Compressive Sensing Via Belief Propagation” IEEE TRANSACTIONS ON SIGNAL PROCESSING, VOL. 58, NO. 1, JANUARY 2010

[17] M.F. Duarte and Y.C. Eldar, ”Structured Compressed Sensing: From Theory to Applications” Vol 59, no 9, 2011, pp. 4053– 4085

[18] J.F.C Mota, J. Xavier, P.M.Q. Aguiar, M. Puschel, “Basis Pursuit in sensor networks” Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference, 22-27 May 2011 Page(s):2916 - 2919

[19] G. Tzagkarakis, J.L. Starck, and P. Tsakalides, “Joint Sparse Signal Ensemble Reconstruction in a WSN Using Decentralized Bayesian Matching Pursuit”, in 19th European Signal Processing Conf., 2011.