INFONET, GIST
Journal Club (2013. 05. 09)

Turbo Reconstruction of Structured Sparse Signals

Authors: Philip Schniter

Publication: Information Sciences and System (CISS), 2010

Speaker: Hyeong-ho Baek

Short summary: **This paper considers the reconstruction of structured-sparse signals from noisy linear observations. In particular, the support of the signal coefficients is parameterized by hidden binary pattern, and a structured probabilistic prior (e.g., Markov random chain/field/tree) is assumed on the pattern. Exact inference is discussed and an approximate inference scheme, based on loopy belief propagation (BP), is proposed. The proposed scheme iterates between exploitation of the observation-structure and exploitation of the pattern-structure, and is closely related to noncoherent turbo equalization, as used in digital communication receivers. An algorithm that exploits the observation structure is then detailed based on approximate message passing ideas.**

I. INTRODUCTION

The main objective is to estimate the sparse signal $\mathbf{x} \in \mathbb{C}^N$ from the noisy linear measurements $\mathbf{y} \in \mathbb{C}^M$,

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{w} \tag{1}$$

where $\mathbf{A} \in \mathbb{C}^{M \times N}$ is a known matrix and $\mathbf{w} \in \mathbb{C}^M$ is additive noise, often modeled as circular white Gaussian, i.e., $\mathbf{w} \sim CN(0, \sigma^2 \mathbf{I})$. By "sparse," we mean that the signal has only a few (say K, where $K << N$) non-zero coefficients.

In many cases of interest, the system of equations in (1) is underdetermined, i.e., $M << N$, so that, even in the noiseless case, there is no unique inverse. However, when $\mathbf{x}$ is known to be sparse, it is possible to accurately reconstruct $\mathbf{x}$ from $\mathbf{y}$ if the columns of A are sufficiently incoherent. For various sparse reconstruction algorithms, including convex- optimization-based, greedy, and iterative thresholding algorithms, there exist elegant bounds on reconstruction error that hold when $\mathbf{A}$ satisfies a certain restricted isometry property (RIP). In many applications, however, the signal $\mathbf{x}$ has structure beyond simple sparsity. For example, the wavelet transform coefficients of natural scenes are not only approximately sparse, but also exhibit persistence across scales, which manifests as correlation within the sparsity pattern. Many other forms of structure in the sparsity pattern are also possible, and so we desire a powerful and flexible approach to modeling and exploiting such structure.

In this paper, we take a probabilistic approach to modeling sparsity structure, allowing the use of, e.g., Markov chain (MC), Markov random field (MRF), and Markov tree (MT) models [2]. Such models have been previously exploited for sparse reconstruction, but only to a limited extent. For example, [3] and [4] proposed Monte-Carlo-based [5] sparse reconstruction algorithms using MRF and MT models, respectively, and [6] and [7] proposed to iterate matching- pursuit with MAP pattern detection based on MRF and MT models, respectively. Monte-Carlo algorithms, while flexible, are typically regarded as computationally too expensive for many problems of interest. Matching-pursuit algorithms are typically much faster, but the schemes in [6], [7] are ad hoc. We attack the problem of reconstructing structured-sparse signals through the framework of belief propagation (BP) [8]. While BP has been successfully used to recover unstructured sparse signals (e.g., [9], [10]), we believe that its application to structured sparse signals is novel. As we shall see, the BP framework suggests an iterative approach, where sparsity pattern beliefs are exchanged between two blocks, one exploiting observation structure and the other exploiting pattern structure. In this regard, our scheme resembles turbo equalization from digital communications [11], where bit beliefs are exchanged between a soft equalizer and a soft decoder. Our two blocks are themselves naturally implemented using BP, and we detail a particularly efficient algorithm based on the approximate message passing (AMP) framework recently proposed by Donoho, Maleki, and Montanari [10].

II. Signal Model

Our structured-sparse signal model uses hidden binary indicators $\{s_n\}_{n=1}^{N}$, where $s_n \in \{0,1\}$. In particular, $s_n = 1$ indicates that the signal coefficient x_n is active while $s_n = 0$ indicates that

x_n is inactive. Assuming that the active signal coefficients are independently but non-identically distributed, we can write

$$p(x_n \mid s_n) = s_n q_n(x_n) + (1 - s_n)\delta(x_n) \quad (2),$$

Where $q_n(\cdot)$ denotes the pdf of x_n, when active, and $\delta(\cdot)$ denotes the Dirac delta. We refer to $\mathbf{s} = [s_1, s_2, \cdots, s_N]^T \in \{0,1\}^N$ as the sparsity pattern, and model structure in $\mathbf{s}$ through an assumed prior pmf $p(\mathbf{s})$.

III. TURBO INFERENCE

Our primary goal is estimating the structured-sparse signal $\mathbf{x}$ given the observations $\mathbf{y} = \mathbf{y}_0$ in model (1). In particular, we are interested in computing minimum mean-squared error(MMSE) estimates of $\{x_n\}$.

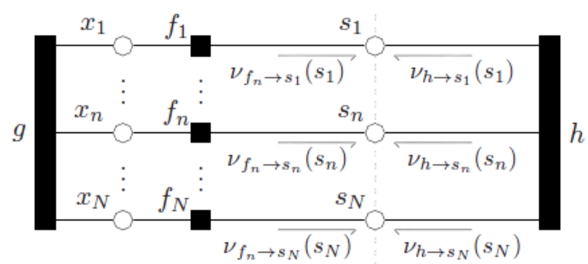


Figure 1 Factor graph of posterior $p(\mathbf{x}, \mathbf{s} \mid \mathbf{y} = \mathbf{y}_0)$. The boxes represent factor nodes and the circles represent variable nodes. Dashed line partitions the factor graph into two sub-graphs

A. Exact inference

The estimation task is facilitated by the following factorization of the posterior pdf shown by the factor graph in Fig. 1.

$$p(\mathbf{x}, \mathbf{s} \mid \mathbf{y} = \mathbf{y}_0) \propto p(\mathbf{y} = \mathbf{y}_0 \mid \mathbf{x}, \mathbf{s}) p(\mathbf{x}, \mathbf{s}) = \underbrace{p(\mathbf{s})}_{\triangleq h(\mathbf{s})} \underbrace{p(\mathbf{y} = \mathbf{y}_0 \mid \mathbf{x})}_{\triangleq g(\mathbf{x})} \prod_{n=1}^{N} \underbrace{p(x_n \mid s_n)}_{\triangleq f_n(x_n, s_n)} \quad (3)$$

We use $\propto$ to denote equality after scaling to unit area.

The MMSE estimate of x_n is given by the mean of the marginal posterior $p(x_n \mid \mathbf{y} = \mathbf{y}_0)$, which can be written as

$$p(x_n \mid \mathbf{y}=\mathbf{y}_0) = \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} p(\mathbf{x},\mathbf{s} \mid \mathbf{y}=\mathbf{y}_0) = \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} \frac{p(\mathbf{x},\mathbf{s},\mathbf{y}=\mathbf{y}_0)}{p(\mathbf{y}=\mathbf{y}_0)} \tag{4}$$

$$\propto \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} p(\mathbf{x},\mathbf{s},\mathbf{y}=\mathbf{y}_0) = \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} p(\mathbf{s} \mid \mathbf{x},\mathbf{y}=\mathbf{y}_0)\, p(\mathbf{y}=\mathbf{y}_0 \mid \mathbf{x})\, p(\mathbf{x}) \tag{5}$$

$$= \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} p(\mathbf{s} \mid \mathbf{x})\, p(\mathbf{y}=\mathbf{y}_0 \mid \mathbf{x})\, p(\mathbf{x}) = \sum_{\mathbf{s}\in\{0,1\}^N} \int_{\mathbf{x}_{-n}} g(\mathbf{x})\, p(\mathbf{x} \mid \mathbf{s})\, p(\mathbf{s}) \tag{6}$$

$$= \sum_{s_n=0}^{1} f_n(x_n,s_n)\, p(s_n) \int_{\mathbf{x}_{-n}} g(\mathbf{x}) \prod_{q\neq n} \sum_{s_q=0}^{1} f_q(x_q,s_q) \sum_{\mathbf{s}_{-n,q}\in\{0,1\}^{N-2}} p(\mathbf{s}_{-n} \mid s_n) \tag{7}$$

Where $\mathbf{s}_{-n}$ denotes vector $\mathbf{s}$ with the n^{th} element omitted, and $\mathbf{s}_{-n,q}$ denotes $\mathbf{s}$ with both the n^{th} and q^{th} elements omitted. Writing $p(\mathbf{s}_{-n} \mid s_n) = p(\mathbf{s}_{-n,q} \mid s_q, s_n)\, p(s_q \mid s_n)$, the last summation in (7) reduces to $p(s_q \mid s_n)$, giving

$$p(x_n \mid \mathbf{y}=\mathbf{y}_0) \propto v_{f_n\to x_n}(x_n)\, v_{g\to x_n}(x_n) \tag{8}$$

$$v_{f_n\to x_n}(x_n) \triangleq \sum_{s_n=0}^{1} f_n(x_n,s_n)\, p(s_n) \tag{9}$$

$$v_{g\to x_n}(x_n) \triangleq \int_{\mathbf{x}_{-n}} g(\mathbf{x}) \prod_{q\neq n} \sum_{s_q=0}^{1} f_q(x_q,s_q)\, p(s_q \mid s_n) \tag{10}$$

B. Implementing the Message Passes

Whereas exact posterior calculation via (8)-(10) is computationally prohibitive for typical problem sizes, approximate calculation can be efficiently accomplished using message passing. Using the framework of BP, the functions $v_{f_n\to x_n}(\cdot)$ and $v_{g\to x_n}(\cdot)$ can be approximated.

$$v^{(t)}_{f_n\to x_n}(x_n) \propto \sum_{s_n=0}^{1} f_n(x_n,s_n)\, v^{(t)}_{s_n\to f_n}(s_n) \tag{11}$$

$$v^{(t)}_{g\to x_n}(x_n) \propto \int_{\mathbf{x}_{-n}} g(\mathbf{x}) \prod_{q\neq n} \underbrace{\sum_{s_q=0}^{1} f_q(x_q,s_q)\, v^{(t)}_{s_q\to f_q}(s_q)}_{v_{f_q\to x_q}(x_n)=v_{x_q\to g}(x_n)} \tag{12}$$

Which depend on the other messages

$$v^{(t)}_{s_n\to f_n}(s_n) = v^{(t)}_{h\to s_n}(s_n) \propto \sum_{\mathbf{s}_{-n}\in\{0,1\}^{N-1}} h(\mathbf{s}) \prod_{q\neq n} \underbrace{v^{(t-1)}_{s_q\to h}(s_q)}_{=v^{(t-1)}_{f_q\to s_q}(s_q)} \tag{13}$$

$$\nu^{(t)}_{f_n \to s_n}(s_n) \propto \int_{x_n} f_n(x_n, s_n) \underbrace{\nu^{(t)}_{x_n \to f_n}(x_n)}_{=\nu^{(t)}_{g \to f_n}(x_n)} \tag{14}$$

We use the superscript-(t) to denote iteration. These messages can then be combined for marginal inference:

$$p^{(t)}(x_n \mid \mathbf{y} = \mathbf{y}_0) \propto \nu^{(t)}_{f_n \to x_n}(x_n) \nu^{(t)}_{g \to x_n}(x_n) \tag{15}$$

$$p^{(t)}(s_n \mid \mathbf{y} = \mathbf{y}_0) \propto \nu^{(t)}_{f_n \to s_n}(s_n) \nu^{(t)}_{h \to s_n}(s_n) \tag{16}$$

Where $p^{(t)}$ denotes the iteration-t approximation to the pdf.

We now partition our factor graph into the two sub-graphs separated by the dashed line in Fig.1. The message $\left\{\nu^{(t)}_{f_n \to s_n}(\cdot)\right\}_{n=1}^{N}$ form the outputs of the left sub-graph and the inputs to the right one, while the messages $\left\{\nu^{(t)}_{h \to s_n}(\cdot)\right\}_{n=1}^{N}$ form the outputs of the right sub-graph and the inputs to the left one. From this, we can interpret the BP scheme as iterationg between two blocks, one which performs inference on the left sub-graph (which models structure in the observation) and the other which performs inference on the right sub-graph (which models structure in the sparsity pattern), with message-passing between blocks.

We will henceforth refer to inference on the left sub-graph of Fig.1 as "sparsity pattern equalization" (SPE) and inference on the right sub-graph as "sparsity pattern decoding" (SPD). We now formally decouple these subtasks and represent each of them using a separate factor graph, as in Fig. 2. For this, we define two additional t^{th} iteration constraint functions,

$$h^{(t)}_n(s_n) \triangleq \nu^{(t)}_{h \to s_n}(s_n) \tag{17}$$

$$d^{(t)}_n(s_n) \triangleq \nu^{(t-1)}_{f_n \to s_n}(s_n) \tag{18}$$

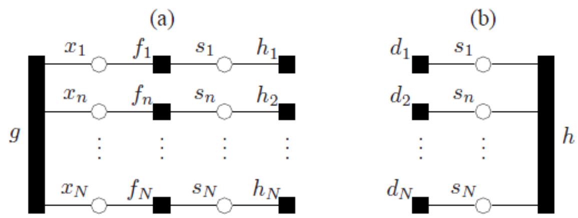


Figure 2 Decoupling of partitioned factor graph from Fig. 1 into
(a) sparsity pattern equalization and (b) sparsity pattern decoding.

IV. Sparsity Pattern Equalization

Below we outline a BP-based technique that follows the "approximate message passing" (AMP) framework recently proposed by Donoho, Maleki, and Montanari. Since we focus on a single iteration t, we suppress the superscript-(t) notation on messages in this section.

For BP-based SPE, we expand the g node in Fig. 2(a), yielding the loopy factor graph in Fig.3, with constraints

$$g_m(x) \triangleq CN(y_m; a_m^H \mathbf{x}, \sigma^2) \tag{19}$$

Where a_m^H denotes the m^{th} row of $\mathbf{A}$. Noting that SPE will require several iterations of message passing between nodes $\{g_m\}$ and $\{x_n\}$, we will henceforth use $v^i_{x_n \to g_m}$ and $v^i_{g_m \to x_n}$ to denote the SPE-iteration-i messages. In addition, we will assume Gaussian active-coefficients, i.e.,

$$q_n(x_n) = CN(x_n; 0, \sigma_n^2) \tag{20}$$

We use λ_n to abbreviate $h_n(1)$, the prior probability of $s_n = 1$ assumed by SPE. Thus, the coefficient is Bernoulli-Gaussian, with the form

$$v_{f_n \to x_n}(x_n) = \lambda_n CN(x_n; 0, \sigma_n^2) + (1 - \lambda_n)\delta(x_n) \tag{21}$$

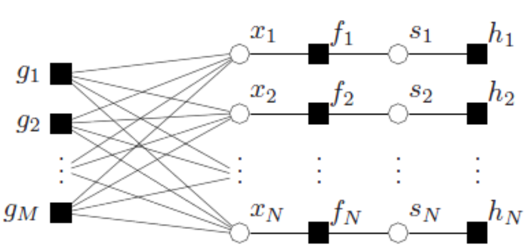


Figure 3 Factor graph for BP-based implementation of SPE

A. BP approximation via the large-system limit

Exact calculation of $v^i_{g_m \to x_n}(x_n)$ would involve the iteration of 2^{N-1} terms, which is cleary impractical. However, in the large system limit(i.e., $M, N \to \infty$ with M/N fixed), the central limit theorem motivates the treatment of $v^i_{g_m \to x_n}(x_n)$ as Gaussian. In this case, it is sufficient to parameterize the inputs to g_m via

$$\mu^i_{nm} \triangleq \int_{x_n} x_n v^i_{x_n \to g_m}(x_n) \tag{22}$$

$$v^i_{nm} \triangleq \int_{x_n} (x_n - \mu^i_{nm})^2 v^i_{x_n \to g_m}(x_n) \tag{23}$$

Which yields outputs from g_m that take the form

$$v^i_{g_m \to x_n}(x_n) \propto CN(A_{mn} x_n; z^i_{mn}, c^i_{mn}) \tag{24}$$

$$z^i_{mn} \triangleq y_m - \sum_{q \neq n} A_{mq} \mu^i_{qm} \tag{25}$$

$$c^i_{mn} \triangleq \sigma^2 + \sum_{q \neq n} | A_{mq} |^2 \ \mu^i_{qm} \tag{26}$$

From (22), (23), we see that μ^{i+1}_{nm} and v^{i+1}_{nm} are then determined by the mean and variance, respectively of the pdf

$$v^{i+1}_{x_n \to g_m}(x_n) \propto v_{f_n \to x_n}(x_n) \prod_{l \neq m} v^i_{g_l \to x_n}(x_n) \tag{27}$$

Using following equation

$$\prod_q CN(x; \mu_q, v_q) \propto CN\left(x; \frac{\sum_q \frac{\mu_q}{v_q}}{\sum_q \frac{1}{v_q}}, \frac{1}{\sum_q \frac{1}{v_q}} \right) \tag{28}$$

The product term in (27) reduces to

$$CN\left(x_n;\frac{\sum_{l\neq m}A_{\ln}^{*}z_{\ln}^{i}/c_{\ln}^{i}}{\sum_{l\neq m}|A_{\ln}|^2/c_{\ln}^{i}},\frac{1}{\sum_{l\neq m}|A_{\ln}|^2/c_{\ln}^{i}}\right) \tag{29}$$

And so, under the large-system-limit approximations

$$c_{\ln}^{i}\approx c_n^i \triangleq \frac{1}{M}\sum_{m=1}^{M}c_{mn}^{i} \tag{30}$$

And $\sum_{l\neq m}|A_{\ln}|^2\approx\sum_{l=1}^{M}|A_{\ln}|=1$, (27) simplifies to

$$v_{x_n\to g_m}^{i+1}(x_n)\propto(\lambda_n CN(x_n;0,\sigma_n^2)+(1-\lambda_n)\delta(x_n))\times CN(x_n;\sum_{l\neq m}A_{\ln}^{*}z_{\ln}^{i},c_n^i) \tag{31}$$

Applying (28) to (31), we find, after some algebra, that

$$\mu_{nm}^{i+1}=\alpha_n(c_n^i)\theta_{nm}^i/(1+\gamma_{nm}^i) \tag{32}$$

$$v_{nm}^{i+1}=\gamma_{nm}^i\,|\,\mu_{nm}^{i+1}|^2+\mu_{nm}^{i+1}c_n^i/\theta_{nm}^i \tag{33}$$

$$\theta_{nm}^i\triangleq\sum_{l\neq m}A_{\ln}^{*}z_{\ln}^{i} \tag{34}$$

$$\gamma_{nm}^i\triangleq\beta_n(c_n^i)\exp(-\zeta_n(c_n^i)\,|\,\theta_{nm}^i|^2) \tag{35}$$

Where $\alpha_n(c)\triangleq\frac{\sigma_n^2}{c+\sigma_n^2}$, $\beta_n(c)\triangleq\frac{1-\lambda_n}{\lambda_n}\frac{c+\sigma_n^2}{c}$, $\zeta_n(c)\triangleq\frac{\sigma_n^2}{c(c+\sigma_n^2)}$.

The i^{th} SPE iteration yields the x_n-posterior approximation

$$p^i(x_n\,|\,\mathbf{y}=\mathbf{y}_0)\propto v_{f_n\to x_n}(x_n)\prod_{l=1}^{M}v_{g_l\to x_n}^{i-1}(x_n) \tag{36}$$

The mean and variance of (36) constitute the MMSE estimate of x_n and its MSE. Nothing that (35) differs from (27) only in the inclusion of the m^{th} product term.

B. Approximate message passing

The approximate BP algorithm outlined updates $O(NM)$ variables per iteration. When N and M are large, the resulting complexity may be undesirably high, motivating us to find a simpler scheme.

Recently, Donoho, Maleki proposed AMP algorithms that greatly simplify BP algorithms of the form outlined by tracking only $O(N)$ variables. Using AMP, we find that

$$\theta_n^i = \sum_{m=1}^{M} A_{mn}^* z_m^i + \mu_n^i \tag{37}$$

$$\mu_n^{i+1} = F_n(\theta_n^i; c^i) \tag{38}$$

$$v_n^{i+1} = G_n(\theta_n^i; c^i) \tag{39}$$

$$c^{i+1} = \sigma^2 + \frac{1}{M}\sum_{n=1}^{N} v_n^{i+1} \tag{40}$$

$$z_m^{i+1} = y_m - \sum_{n=1}^{N} A_{mn}\mu_n^{i+1} + \frac{z_m^i}{M}\sum_{n=1}^{N} F_n^{'}(\theta_n^i; c^i) \tag{41}$$

Above, $F_n(.;.)$, $G_n(.;.)$, and $F_n^{'}(.;.)$ are nonlinear functions that depend on the coefficient prior. We chose the Bernoulli-Gaussian prior. Thus, the nonlinear functions take the following form

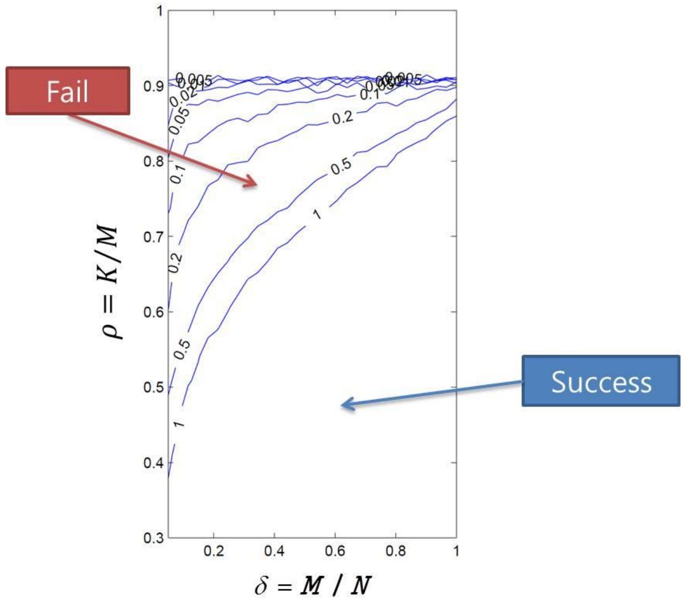
$$F_n(\theta;c) = \frac{\alpha_n(c)}{1+\beta_n(c)e^{-\zeta_n(c)|\theta|^2}}\theta \tag{42}$$

$$G_n(\theta;c) = \beta_n(c)e^{-\zeta_n(c)|\theta|^2} \,|\, F_n(\theta;c)\,|^2 + \frac{c}{\theta}F_n(\theta;c) \tag{43}$$

$$F_n^{'}(\theta;c) = \frac{\alpha_n(c)}{1+\beta_n(c)e^{-\zeta_n(c)|\theta|^2}} \times \left[1 + \frac{\zeta_n(c)\,|\,\theta\,|^2}{1+(\beta_n(c)e^{-\zeta_n(c)|\theta|^2})^{-1}}\right] \tag{44}$$

V. NUMERICAL RESULTS

Numerical experiments were conducted for the observation model (1), where the elements of $\mathbf{A}$ were independently drawn from a $CN\left(0, \frac{1}{M}\right)$ distribution and where the signal coefficients were generated via $p(x_n \mid s_n) = s_n CN(x_n; 0,1) + (1-s_n)\delta(x_n)$ using Markov chain-generated binary sprsity pattern $\{s_n\}$. We set $\gamma \in (0,1]$ called the Markov independence parameter. Note that, as γ increases, the pattern becomes less correlated, with $\gamma = 1$ corresponding to an i.i.d pattern.



VI. DISCUSSION

After meeting, please write discussion in the meeting and update your presentation file.

Appendix

Reference