A Generalized Framework for Learning and Recovery of Structured Sparse Signals

Justin Ziniel

Abstract

We report on a framework for recovering single- or multi-timestep sparse signals that can learn and exploit a variety of probabilistic forms of structure. Message passing-based inference and empirical Bayesian parameter learning form the backbone of the recovery procedure. We further describe an object-oriented software paradigm for implementing our framework, which consists of assembling modular software components that collectively define a desired statistical signal model. Lastly, numerical results for an example structured sparse signal model are provided.

Index Terms

Compressed sensing, structured sparse signal recovery, multiple measurement vectors, structured sparsity, dynamic compressed sensing

I. Introduction

For many decades, digital signal processing has been guided by the celebrated Shannon–Nyquist sampling theorem [1], which dictates that a bandlimited analog signal can be uniquely determined from equi-spaced digital samples collected at a rate that is twice the analog signal's bandwidth. The implications of this theorem have been far-reaching, motivating an engineering demand for hardware with ever-increasing sampling rates as signals of increasingly high dimensionality are being sampled. The exponential growth in pixel density on digital cameras is but one indicator of this trend.

The author is with the Department of Electrical and Computer Engineering (ECE), The Ohio State University, Columbus, Ohio. This research was conducted collaboratively with Philip Schniter, ECE Department, The Ohio State University, Columbus, Ohio, and Sundeep Rangan, ECE Department, Polytechnic Institute of New York University, Brooklyn, New York.

Work supported in part by NSF grant CCF-1018368 and DARPA/ONR grant N66001-10-1-4090.

While Moore's law has allowed for steady increases in the sampling rate of hardware, the desire to recover high-resolution, large-scale data has often outstripped the ability to effectively sample the underlying phenomenon for a variety of reasons. For example, the speed of image acquisition in MRI scanners is inherently limited by certain physical and physiological constraints [2]. Acquiring higher resolution images without requiring unacceptably long scan times will mean violating the Shannon-Nyquist sampling theorem by sub-sampling the images. Examples abound in many other disciplines where, for one reason or another, sub-Nyquist sampling is an unavoidable reality. Coping with, or ideally counteracting, the degradation in quality introduced by such sub-sampling has become an important and active area of research.

Within the past decade, there has been a tremendous amount of literature emerging from the signal processing, statistics, and applied mathematics communities on new sampling theory that enables sub-Nyquist sampling of high-dimensional signals while still guaranteeing the ability to accurately reconstruct such signals [3], [4]. This new approach, which has been termed *compressed sensing* (CS), replaces the objective of recovering *any* bandlimited signal with the objective of recovering signals which possess a sparse representation. In this context, the concept of a sparse representation means that the signal, although possessing high ambient dimensionality, can be succinctly described with just a small amount of information. Many real-world signals possess such a sparse representation; for instance, it's the reason why digital cameras are able to compress large raw images into compact JPEG files.

In recent years, a great deal of effort by the CS community has been directed at developing ways to incorporate additional signal structure beyond simple sparsity into recovery techniques [5]. Recent work into Bayesian approaches aimed at exploiting the low-dimensional structure inherent in many real-world signals has demonstrated that significant performance gains can be achieved, even when the structure must be learned (e.g., [6]–[10]).

In this work we present a flexible framework for performing empirical Bayesian estimation of structured sparse signals. Our approach follows the "turbo CS" [11] principle of breaking apart an intractable global inference problem into smaller sub-problems for which efficient and accurate inference is possible. By exchanging information between the sub-problems, we obtain a high-quality approximation of the solution to the global problem. Additionally, as a byproduct of solving these sub-problems, we are often able to learn model parameters iteratively from the data using an expectation-maximization (EM) algorithm.

II. A Structured CS Signal Model

We consider the task of recovering a collection of sparse vectors $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$ from a collection of measurement vectors $\{\boldsymbol{y}^{(t)}\}_{t=1}^{T}$, where $\boldsymbol{x}^{(t)} \in \mathbb{C}^N$, $\boldsymbol{y}^{(t)} \in \mathbb{C}^M$, and (typically) $M < N$. The relationship between $\boldsymbol{x}^{(t)}$ and $\boldsymbol{y}^{(t)}$ is determined as follows: Each $\boldsymbol{x}^{(t)}$ is transformed through the linear process

$$\boldsymbol{z}^{(t)} = \boldsymbol{A}^{(t)}\boldsymbol{x}^{(t)}, \qquad t = 1, \ldots, T \tag{1}$$

where $\boldsymbol{A}^{(t)} \in \mathbb{C}^{M \times N}$ is a known linear operator. Each "transform coefficient" $z_m^{(t)}$ is then observed through an independent scalar "observation channel," defined by the conditional distribution $p(y_m^{(t)}|z_m^{(t)})$, to yield a measurement $y_m^{(t)}$. Here we note that the standard noisy CS model $\boldsymbol{y}^{(t)} = \boldsymbol{A}^{(t)}\boldsymbol{x}^{(t)} + \boldsymbol{w}^{(t)}$ is a special case of the aforementioned signal model when the elements of $\boldsymbol{w}^{(t)}$ are independent, as is the matrix CS model $\boldsymbol{Y} = \boldsymbol{A}\boldsymbol{X} + \boldsymbol{W}$. We assume without loss of generality that $\boldsymbol{x}^{(t)}$ is sparse in the canonical basis.

As mentioned in Section I, $\{\boldsymbol{x}^{(t)}\}_{t=1}^{T}$ oftentimes exhibits substantial structure beyond simple sparsity. For convenience, we will refer to the structure present within a single vector $\boldsymbol{x}^{(t)}$ as spatial structure, and structure across multiple such vectors as temporal structure. Also, we will use the overbar notation $\bar{\boldsymbol{x}} \triangleq \text{vec}([\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}])$ to denote the vectorization of all signal timesteps (with other overbar variables defined analogously). In order to model the spatial and temporal structure probabilistically, we introduce a set $\mathcal{H}$ of hidden random variables such that $p(\bar{\boldsymbol{x}}|\mathcal{H})$ becomes separable, i.e., $p(\bar{\boldsymbol{x}}|\mathcal{H}) = \prod_{t=1}^{T}\prod_{n=1}^{N} p(x_n^{(t)}|\mathcal{H})$.

The utility of using a separable signal prior is twofold: it oftentimes simplifies the task of describing the structure probabilistically, and allows us to apply a powerful inference algorithm known as GAMP [12] within a turbo inference framework. For ease of explanation, we will henceforth focus on one particular choice of hidden variables, namely, where each signal coefficient $x_n^{(t)}$ can be expressed as the product of two hidden variables: $x_n^{(t)} = s_n^{(t)} \cdot \gamma_n^{(t)}$, where $s_n^{(t)} \in \{0, 1\}$ is an indicator of support set membership, and $\gamma_n^{(t)} \in \mathbb{C}$ expresses the amplitude of a non-zero $x_n^{(t)}$. This decomposition results in the following collection of hidden random variables:

$$\mathcal{H} \triangleq \{\boldsymbol{s}^{(t)}, \boldsymbol{\gamma}^{(t)}\}_{t=1}^{T}. \tag{2}$$

We stress that this is simply one of many choices of $\mathcal{H}$ and $p(\bar{\boldsymbol{x}}|\mathcal{H})$; many others could be considered within the framework we propose. (See Section V for another example.)

Given the hidden variables in (2), we can model a wide variety of signal structures by choosing appropriate priors for $\bar{\boldsymbol{s}}$ and $\bar{\boldsymbol{\gamma}}$. Specifically, $p(\bar{\boldsymbol{s}})$ can be used to model spatio-temporal structure in the support of $\bar{\boldsymbol{x}}$, such as block-, tree-, or clustered-sparsity, while $p(\bar{\boldsymbol{\gamma}})$ can be used to model spatio-temporal correlations in the amplitudes of $\bar{\boldsymbol{x}}$.

III. Turbo Inference and Parameter Learning

For the signal model of Section II, the joint posterior distribution of all of the random variables, given the measurements, can be expressed using Bayes' rule as:

$$p(\bar{\boldsymbol{x}}, \mathcal{H} | \bar{\boldsymbol{y}}; \mathcal{P}) \propto p(\bar{\boldsymbol{y}} | \bar{\boldsymbol{x}}; \mathcal{P}) p(\bar{\boldsymbol{x}} | \mathcal{H}; \mathcal{P}) p(\mathcal{H}; \mathcal{P}), \tag{3}$$

where $\propto$ indicates equality up to a normalizing constant and $\mathcal{P}$ denotes a set of model parameters that are used to parameterize the signal model, e.g., prior means, variances, etc., which we treat as deterministic unknowns. The objective of our inference procedure will be to obtain marginal posterior distributions of each random variable, e.g., $p(x_n^{(t)} | \bar{\boldsymbol{y}}; \mathcal{P})$, which can be used to produce an MMSE estimate of the signal. At the same time, we would like to learn the model parameters $\mathcal{P}$ from the data in a principled fashion.

By taking advantage of the fact that $p(\bar{\boldsymbol{y}} | \bar{\boldsymbol{x}}; \mathcal{P})$ $\big(\equiv p(\bar{\boldsymbol{y}} | \bar{\boldsymbol{z}}; \mathcal{P})\big)$ is separable due to the independent observation assumption, and that $p(\bar{\boldsymbol{x}} | \mathcal{H}; \mathcal{P})$ is separable by definition of $\mathcal{H}$, we can conveniently describe (3) using a graphical model known as a *factor graph*. The sample factor graph shown in Fig. 1 expresses the probabilistic structure of the model (2), where $p(\mathcal{H}) = p(\bar{\boldsymbol{s}}) p(\bar{\boldsymbol{\gamma}})$ and $p(x_n^{(t)} | \mathcal{H}) = p(x_n^{(t)} | s_n^{(t)}, \gamma_n^{(t)})$, using circles to denote random variables (here $\{x_n\}$, $\{s_n\}$, and $\{\gamma_n\}$) and squares to denote posterior factors. Although the subsequent discussion will focus on this example factor graph, our technique generalizes to any factor graph that describes separable signal and observation-channel priors.

A popular means of performing inference on probabilistic factor graphs is via belief propagation (BP) [13], whose objective is to compute the posterior marginals of all unobserved random variables. Due to the numerous loops in the factor graph of Fig. 1, BP is not guaranteed to produce exact marginals. However, for inference within each of the planes on the left of Fig. 1, there exists an attractive approximate message passing algorithm known as GAMP [12]. GAMP's appeal for our problem stems from several considerations: *(i)* GAMP supports arbitrary separable signal and observation-channel priors, *(ii)* inference is rapid and highly accurate, and

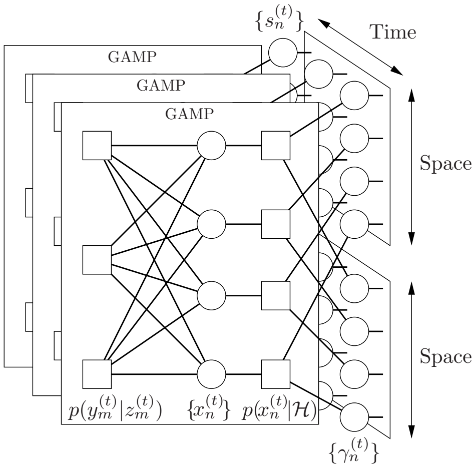


Fig. 1: A factor graph representation of the joint posterior distribution of (3). In this example, $M = T = 3$, $N = 4$, and $\mathcal{H} = \{\boldsymbol{s}^{(t)}, \boldsymbol{\gamma}^{(t)}\}_{t=1}^{T}$.

(iii) theoretical analyses demonstrate that the behavior of GAMP can be accurately predicted by a set of state evolution equations [12].

To perform inference on the complete factor graph of Fig. 1, we employ the “turbo CS” framework of [11], which alternates[1] between exploiting the measurement structure (using GAMP) and exploiting the signal structure specified by $p(\mathcal{H}; \mathcal{P})$. This approach is reminiscent of modern turbo communications receivers, which alternate between channel equalization and decoding. The messages leaving the GAMP planes in Fig. 1 constitute beliefs about the hidden variables $\mathcal{H}$, given the measurements $\bar{\boldsymbol{y}}$. These messages act as inputs to the $\{s_n^{(t)}\}$ and $\{\gamma_n^{(t)}\}$ nodes in the rightmost portion of the factor graph. There, inference can be performed using any technique (e.g., the forward-backward algorithm [13]) that provides extrinsic likelihoods of the hidden variables. These likelihoods act as updated beliefs about the hidden variables, given the underlying structure, and are fed back to GAMP, after which the entire process is repeated.

At the same time that turbo inference is being performed, the estimates of the unknown model parameters $\mathcal{P}$ can be updated through an expectation-maximization (EM) [15] learning procedure. Oftentimes it makes sense to use the hidden variables $\mathcal{H}$ as the “missing data,” in which case the EM update for $\mathcal{P}$, at iteration $k+1$, can be expressed as the optimization problem

$$\mathcal{P}^{k+1} = \underset{\mathcal{P}}{\operatorname{argmax}} \; \mathrm{E}_{\mathcal{H}|\bar{\boldsymbol{y}}}\big[\log p(\bar{\boldsymbol{y}}, \mathcal{H}; \mathcal{P}) | \bar{\boldsymbol{y}}; \mathcal{P}^k\big],$$

[1] Another way to exploit GAMP for message passing on the complete factor graph of Fig. 1 is through “hybrid-GAMP” [14]. Unlike the “turbo” message-passing schedule adopted in this work, hybrid-GAMP employs a flooding schedule.

where $\mathcal{P}^k$ is the estimated value of all model parameters as of iteration k. For many signal models, $\log p(\bar{\boldsymbol{y}}, \mathcal{H}; \mathcal{P})$ will decouple into a sum of many terms that depend only on small subsets of hidden variables and parameters. Therefore, it is often possible to obtain closed-form EM updates of the model parameters using only marginal or pairwise joint posteriors, e.g., $p(s_n^{(t)}|\bar{\boldsymbol{y}})$ and $p(\gamma_n^{(t)}, \gamma_n^{(t-1)}|\bar{\boldsymbol{y}})$ in [16]. Since belief propagation provides these posteriors, it is feasible to perform these EM updates as an auxiliary procedure to the main turbo inference process with very little additional cost.

IV. Object-Oriented Software Implementation

One of the defining features of the framework we propose in Sections II and III is that it decouples the global inference problem of marginalizing (3) into smaller sub-problems that require only local, not global, information to complete their tasks. Furthermore, these sub-problems roughly correspond to inference on different regions of the factor graph. Consequently, the object-oriented programming (OOP) paradigm is useful for building a powerful and flexible software implementation of our approach, which we call EMturboGAMP.

We will now describe a software implementation[2] of EMturboGAMP that uses OOP principles to allow one to solve a variety of structured CS problems. At its core, our implementation relies on assembling objects of different classes in a modular fashion to specify a particular signal model. To date, we have defined four abstract classes: `Signal`, `Observation`, `SupportStruct`, and `AmplitudeStruct`, and a container class called `TurboOpt` that holds objects derived from these four classes.

At the highest level, EMturboGAMP consists of two steps performed repeatedly: in the first step, GAMP is run for a particular choice of “local” signal and observation-channel priors[3]. In the second step, the final state of GAMP messages is given to the `TurboOpt` object, which works together with the `Signal` and `Observation` objects to provide updated local signal and observation-channel priors to GAMP for the next turbo iteration.

Concrete implementations of these four abstract classes are responsible for overseeing specific tasks. A `Signal` class object defines the marginal prior distribution $p(x_n^{(t)}; \mathcal{P})$ and hidden

[2] Available at www.ece.osu.edu/~schniter/EMturboGAMP

[3] The GAMP algorithm is conventionally run with a fixed choice of signal and observation-channel priors. In the turbo framework, these priors (from GAMP’s perspective) are updated every iteration, thus we refer to them as “local priors”.

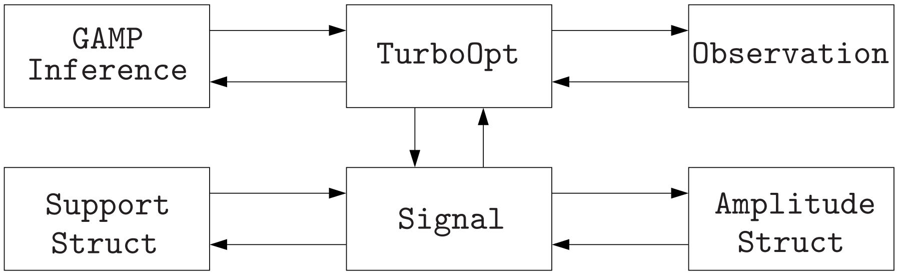


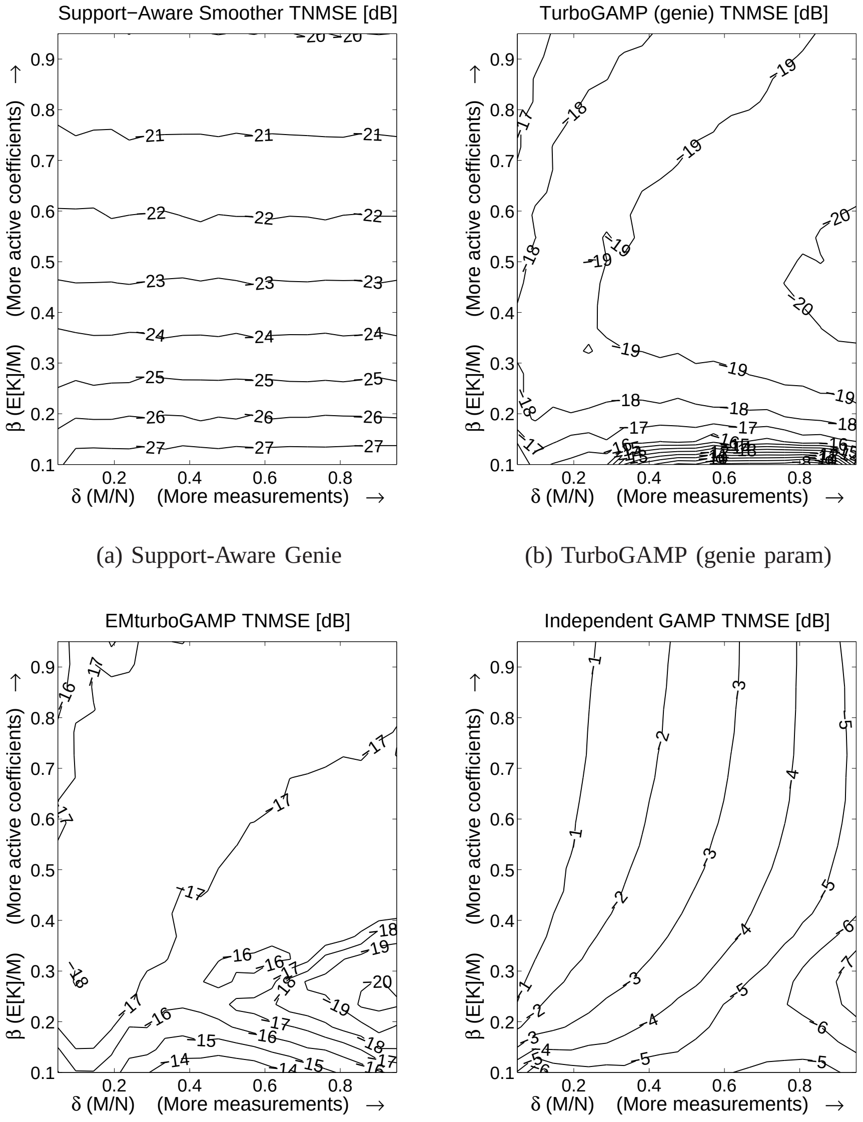
Fig. 2: Information flow between EMturboGAMP classes and GAMP.

variables $\mathcal{H}$. It delegates the task of exploiting the signal support structure to a `SupportStruct` object, which defines $p(\bar{\boldsymbol{s}};\mathcal{P})$, and the task of exploiting the signal amplitude structure to an `AmplitudeStruct` object, which defines $p(\bar{\boldsymbol{\gamma}};\mathcal{P})$. The `Observation` class object specifies the observation-channel prior $p(y_m^{(t)}|z_m^{(t)};\mathcal{P})$. Each class is also responsible for performing EM updates of any model parameters $\in \mathcal{P}$ which they define. In Fig. 2 we summarize the relationship between the various classes and GAMP, illustrating how information flows between them.

One nice property of the OOP approach is that it makes it easy to incorporate new signal or observation-channel models without requiring one to code up an entirely new algorithm from scratch. A new marginal signal prior, for example, can be specified by creating a new sub-class of the `Signal` class. Each new sub-class must implement the handful of methods (functions) specified by its abstract super-class, but the manner in which they are implemented is entirely up to the programmer. This allows EMturboGAMP to be assured of a common interface without mandating the way in which the inference is performed. For further details, we invite the reader to explore the MATLAB® code we have made publicly available (and to contribute additional classes as well!).

V. A Numerical Example

To demonstrate both the flexibility of our proposed framework, as well as the convenience of its OOP-based software implementation, we undertook a numerical comparison of EMturboGAMP against several structure-aware and structure-agnostic algorithmic variants. To understand the average performance of EMturboGAMP under a variety of test conditions, we empirically evaluated mean-squared error (MSE) performance on the sparsity-undersampling plane, calculating MSE at various combinations of the normalized sparsity ratio, β (i.e., the ratio of non-zero coefficients-to-measurements, K/M), and undersampling ratio, δ (i.e., the ratio of measurements-



(a) Support-Aware Genie

(b) TurboGAMP (genie param)

(c) EMturboGAMP

(d) Independent GAMP

Fig. 3: Median TNMSE performance (in dB) across the sparsity-undersampling plane for four different recovery methods

to-unknowns, M/N). In particular, for each (δ, β) pair, multiple independent signal realizations were recovered, and for each realization the time-averaged normalized MSE (TNMSE) was computed, where $\text{TNMSE}(\bar{\boldsymbol{x}}, \hat{\bar{\boldsymbol{x}}}) \triangleq \frac{1}{T}\sum_{t=1}^{T} \|\boldsymbol{x}^{(t)} - \hat{\boldsymbol{x}}^{(t)}\|_2^2 / \|\boldsymbol{x}^{(t)}\|_2^2$, and $\hat{\boldsymbol{x}}^{(t)}$ is an estimate of $\boldsymbol{x}^{(t)}$.

In our experiment, we considered a multiple measurement vector (MMV) problem in which each coefficient was marginally a Bernoulli-Gaussian-mixture of the form

$$p(x_n^{(t)}) = (1-\lambda)\delta(x_n^{(t)}) + \tfrac{\lambda}{2}\mathcal{N}(x_n^{(t)}; +1, \tfrac{1}{4}) + \tfrac{\lambda}{2}\mathcal{N}(x_n^{(t)}; -1, \tfrac{1}{4}). \tag{4}$$

We wished to impose smooth variations in the amplitudes over time, in addition to the time-invariant support constraint inherent in the classical MMV model. To accomplish this, we defined our hidden variables as $\mathcal{H} \triangleq \{\boldsymbol{s}^{(t)}, \boldsymbol{\gamma}_1^{(t)}, \boldsymbol{\gamma}_2^{(t)}\}_{t=1}^T$, where $\gamma_{1,n}^{(t)}$ was marginally distributed $\mathcal{N}(+1, \frac{1}{4})$, $\gamma_{2,n}^{(t)}$ was marginally $\mathcal{N}(-1, \frac{1}{4})$, and $s_n^{(t)} \in \{0, 1, 2\}$ specified whether $x_n^{(t)}$ was non-zero, and if so, which component Gaussian it was drawn from. Independent Gauss-Markov processes with correlation ρ, (i.e., $\gamma_{d,n}^{(t)} = \rho\gamma_{d,n}^{(t-1)} + (1-\rho)e_{d,n}^{(t)}$, $d = 1, 2$, where $e_{d,n}^{(t)}$ is a Gaussian perturbation process), were used to model amplitude correlations for both $\bar{\boldsymbol{\gamma}}_1$ and $\bar{\boldsymbol{\gamma}}_2$, while $p(\bar{\boldsymbol{s}})$ was chosen to enforce joint sparsity.

As an observation-channel model, we considered additive noise with a heavy-tailed distribution. Specifically, the observation-channel prior was

$$p(y_m^{(t)}|z_m^{(t)}) = (1-\pi)\mathcal{N}(z_m^{(t)}; 0, \nu_0) + \pi\mathcal{N}(z_m^{(t)}; 0, \nu_1), \tag{5}$$

where $\nu_1 = 1000 \cdot \nu_0$, and $\pi = 0.10$.

Iso-dB contours of median TNMSE performance for a signal model in which $N = 1024$, $T = 6$, $\rho = 0.95$, and $\text{SNR} = 25\text{dB}$ are plotted in Fig. 3 for four different recovery methods. In Fig. 3a, we show the performance of a support-aware genie smoother that had perfect knowledge of the support, and perfect knowledge of the signal model and its parameters, $\mathcal{P}$. In Fig. 3b, we show the performance of TurboGAMP with perfect signal model knowledge (i.e., no need for EM learning) but no support knowledge. Then, in Fig. 3c, we plot the performance of EMturboGAMP with EM learning of the model parameters. Finally, Fig. 3d shows the performance of a structure-agnostic GAMP recovery method, which had knowledge of $p(x_n^{(t)}; \mathcal{P})$ and $p(y_m^{(t)}|z_m^{(t)}; \mathcal{P})$, but no knowledge of $p(\bar{\boldsymbol{\gamma}}_1)$, $p(\bar{\boldsymbol{\gamma}}_2)$, or $p(\bar{\boldsymbol{s}})$. The structure-agnostic GAMP method was allowed to refine model parameters using EM learning to compensate for model mismatch. Despite this, the advantages of exploiting the additional signal structure are clearly evident.

VI. Conclusion

In this work we have provided an overview of a new framework for recovering structured sparse signals that possess a variety of possible forms of structure. Our approach, EMturboGAMP, is probabilistic in nature, and allows one to obtain soft signal estimates while learning the probabilistic model parameters adaptively from the data. EMturboGAMP leverages the “turbo CS” concept to break apart a challenging inference problem into easier to manage sub-problems.

One of these sub-problems is efficiently solved using recently proposed approximate message passing techniques.

We further described an object oriented programming (OOP) implementation of EMturboGAMP. The advantage of the OOP paradigm is in the flexibility that it provides the user, allowing one to develop complex probabilistic models by mixing and matching different sub-components, instead of requiring a completely new algorithm be coded from scratch for each model. We demonstrated this flexibility in a numerical example, showcasing the ability of EMturboGAMP to accurately recover an underlying signal that possesses a rich form of probabilistic structure.

References

[1] C. E. Shannon, “Communication in the presence of noise,” *Proc. Institute of Radio Engineers*, vol. 37, no. 1, pp. 1021, Jan. 1949.

[2] M. Lustig, D. L. Donoho, and J. M. Pauly, “Sparse MRI: The application of compressed sensing for rapid MR imaging,” *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[3] E. J. Candes, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489 – 509, Feb. 2006.

[4] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[5] M. F. Duarte and Y. C. Eldar, “Structured compressed sensing: From theory to applications,” *IEEE Trans. Signal Process.*, vol. 59, no. 9, pp. 4053–4085, 2011.

[6] J. Ziniel, L. C. Potter, and P. Schniter, “Tracking and smoothing of time-varying sparse signals via approximate belief propagation,” in *Proc. 44th Asilomar Conf. Sig., Sys., & Comput. (SS&C)*, Pacific Grove, CA, Nov. 2010.

[7] T. Faktor, Y. C. Eldar, and M. Elad, “Modeling statistical dependencies in sparse representations,” in *Wkshp. on Signal Process. w/ Adaptive Sparse Struct. Rep. (SPARS ’11)*, Edinburgh, UK, June 2011.

[8] L. Yu, H. Sun, J. P. Barbot, and G. Zheng, “Compressive sensing for clustered sparse signals,” in *Intl. Conf. Acoust., Speech, Signal Process. (ICASSP)*, Prague, Czech Republic, May 2011.

[9] Z. Zhang and B. D. Rao, “Sparse signal recovery with temporally correlated source vectors using Sparse Bayesian Learning,” *IEEE J. Selected Topics Signal Process.*, vol. 5, no. 5, pp. 912–926, Sept. 2011.

[10] D. Baron and M. F. Duarte, “Universal MAP estimation in compressed sensing,” in *Proc. 49th Allerton Conf. Comm., Control, & Comput.*, Monticello, IL, Sept. 2011.

[11] P. Schniter, “Turbo reconstruction of structured sparse signals,” in *Conf. on Information Sciences and Systems (CISS)*, Princeton, NJ, Mar. 2010, pp. 1 –6.

[12] S. Rangan, “Generalized approximate message passing for estimation with random linear mixing,” in *Proc. IEEE Int. Symp. Inform. Theory*, St. Petersburg, Russia, Aug. 2011, pp. 2168–2172.

[13] F. R. Kschischang, B. J. Frey, and H. A. Loeliger, “Factor graphs and the sum-product algorithm,” *IEEE Trans. Inform. Theory*, vol. 47, no. 2, pp. 498–519, Feb. 2001.

[14] S. Rangan, A. K. Fletcher, V. K. Goyal, and P. Schniter, “Hybrid approximate message passing with applications to structured sparsity,” arXiv:1111.2581 [cs.IT], Nov. 2011.

[15] A. P. Dempster, N. M. Laird, and D. B Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *J. Roy. Statist. Soc., B*, vol. 39, pp. 1–38, 1977.

[16] J. Ziniel and P. Schniter, “Efficient high-dimensional inference in the multiple measurement vector problem,” arXiv:1111.5272 [cs.IT], Nov. 2011.