MONTE CARLO METHODS FOR COMPRESSED SENSING

T. Blumensath

ISVR Signal Processing and Control Group
The University of Southampton
Southampton, SO17 1BJ

ABSTRACT

In this paper we study Monte Carlo type approaches to Bayesian sparse inference under a squared error loss. This problem arises in Compressed Sensing, where sparse signals are to be estimated and where recovery performance is measured in terms of the expected sum of squared error. In this setting, it is common knowledge that the mean over the posterior is the optimal estimator. The problem is however that the posterior distribution has to be estimated, which is extremely difficult. We here contrast approaches that use a Monte Carlo estimate for the posterior mean. The randomised Iterative Hard Thresholding algorithm is compared to a new approach that is inspired by sequential importance sampling and uses a bootstrap re-sampling step based on importance weights.

***Index Terms*—** Compressed Sensing, Iterative Hard Thresholding, Sparse Inverse Problem, Bayesian methods, Importance Sampling

1. INTRODUCTION

Sparse signal representations and Compressed Sensing [1], [2], [3], [4] have over the last ten years developed into standard tools in signal processing and sampling. Compressed Sensing allows us to sample and reconstruct finite dimensional sparse signals using fewer samples than predicted by the Nyquist rate. Let $\mathbf{x}$ be an N dimensional vector which is sampled using M linear measurements $\{\langle \mathbf{x}, \phi_n \rangle\}$, where $\langle \cdot, \cdot \rangle$ is the inner product and ϕ_n a measurement vector. Importantly, assume that $\mathbf{x}$ is sparse or approximately sparse, that is, many of the elements x_i of $\mathbf{x}$ are zero (or have a small magnitude). Let $\Phi \in \mathbb{R}^{M \times N}$ be the matrix with rows ϕ_n so that the vector containing all the measurements is

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{e}, \tag{1}$$

where $\mathbf{e}$ is observation noise.

If we take M measurements, then a challenging case is one in which $M < N$ so that the recovery problem: "estimate $\mathbf{x}$ from $\mathbf{y}$," becomes ill posed unless we further constrain $\mathbf{x}$. In traditional compressed sensing, the constraint is the assumption that $\mathbf{x}$ is sparse, which might lead to the following optimisation problem

$$\hat{\mathbf{x}} = \arg \min_{\mathbf{x}:\|\mathbf{x}\|_0 \le K} \|\mathbf{y} - \mathbf{\Phi x}\|_2, \tag{2}$$

where $\|\mathbf{x}\|_0$ measures the number of non-zero elements in the vector $\mathbf{x}$. This combinatorial problem is one formulation of the Compressed Sensing recovery problem in which we are searching through all vectors $\mathbf{x}$ with no more than K non-zero elements in order to minimises the squared error $\|\mathbf{y} - \mathbf{\Phi x}\|_2$. Unfortunately, the problem is one of the NP-hard optimisation problems and much of the Compressed Sensing literature has focussed on the study of mathematical conditions that guarantee that more efficient computational methods are able to provide good estimates [1]. Of particular relevance for this paper is the Iterative Hard Thresholding algorithm [7].

This research was supported by EPSRC grant EP/K037102/1.

2. BAYESIAN ESTIMATION AND IMPORTANCE SAMPLING

For a given set of observed data $\mathbf{y}$, let $\{\hat{\sigma}_i(\mathbf{y})\}$ be the set of estimates of several parameters of interest. The optimal Bayesian estimate for σ_i that minimises the cost

$$E[(\sum_i (\hat{\sigma}_i(\mathbf{y}) - \sigma_i)^2)] \tag{3}$$

is known to be the posterior mean

$$\hat{\sigma}_i(\mathbf{y}) = \int \sigma_i p(\{\sigma_i\}|\mathbf{y}) \prod_i d\sigma_i, \tag{4}$$

where E is the expectation with respect to $\mathbf{y}$ and $\{\sigma_i\}$ and where $p(\{\sigma_i\}|\mathbf{y})$ is the posterior distribution of $\{\sigma_i\}$ given $\mathbf{y}$ (see e.g. [5]).

$$p(\mathbf{x}|\mathbf{y}) = \int \frac{p(\mathbf{y}|\mathbf{x},\mathbf{u})p(\mathbf{x}|\mathbf{u})p(\mathbf{u})}{p(\mathbf{y})}\, d\mathbf{u} \tag{5}$$

$$E[(\sum_i (\hat{\mathbf{x}}_i(\mathbf{y}_i) - \mathbf{x}_i)^2)] \tag{6}$$

is known to be the posterior mean

$$\hat{\mathbf{x}}_i(\mathbf{y}_i) = \int \mathbf{x} p(\mathbf{x}|\mathbf{y}_i) d\mathbf{x}, \tag{7}$$

In most cases of interest, the integration over the posterior required to evaluate the posterior mean cannot be evaluated and so approximate estimates have to be used. A common approach here are Monte Carlo methods [6].

A Monte Carlo estimate approximates the integral by a sum over samples x_i drawn from the distribution $p(x)$

$$\int f(x)p(x)\, dx \approx \sum_{i=1}^{N} \frac{1}{N} f(x_i), \tag{8}$$

where $p(x)$ is a probability distribution and $f(x)$ a function (e.g. using $f(x) = x$ would give us the mean of the distribution $p(x)$). If we were able to draw samples from the distribution $p(x)$, then the approximation becomes exact in the limit if we take more and more samples [6] [1].

If we cannot draw the samples from the distribution $p(x)$, then we can instead draw samples from an alternative proposal distribution $q(x)$. If we choose $q(x)$ such that the ratio $p(x)/q(x)$ remains bounded for all values of x for which $p(x)$ is defined, then we can write a Monte Carlo estimate as

$$\int f(x)p(x)\,dx = \int \frac{p(x)}{q(x)} f(x)q(x)\,dx \approx \sum_{i=1}^{N} \frac{1}{N}\frac{p(x_i)}{q(x_i)} f(x_i), \tag{9}$$

where now a weighted Monte Carlo estimate is used with proposal distribution $q(\cdot)$ and weights $w_i = \frac{p(x_i)}{q(x_i)}$.

In many cases, we can only evaluate the distributions $p(\mathbf{x}_i)$ and $q(\mathbf{x}_i)$ up to some unknown constant. In this case, it is customary to normalise the wights w_i such that $\sum_i w_i = 1$.

3. BAYESIAN HARD THRESHOLDING (BHT)

In the setting of Compressed Sensing, the idea to use an average estimate, rather than a single point estimate, has been promoted recently in [8] and [9]. Inspired by the realisation that the posterior mean is the optimal estimate to minimise expected squared error, both of these publications introduce random element selection into two well known sparse recovery algorithms, Orthogonal Matching Pursuit [10] and Iterative Hard Thresholding [7]. However, the randomisation introduced into these approaches does not guarantee that the algorithms produce samples drawn from the relevant posterior distribution and so, the average estimate remains sub-optimal.

We here build on this idea and contrast them to a new Monte Carlo based approach. We are motivated by the observation that, if we can only sample from an approximation to the posterior distribution, then computational Bayesian reasoning suggests the use of importance weights, which should then be used for an Importance estimate. We here concentrate on approaches that are similar to those in [9] and that do not add a significant increase in computational cost to the method presented there. Similar ideas that extend the approach in [8] follow similar arguments.

3.1. The Bayesian Model

There are many different approaches to formalise a sparse bayesian model. We here use a formulation based on indicator variables $u_i \in \{0,1\}$. Let $\mathbf{u}$ be the vector of variables u_i. The individual entries in the sparse vector $\mathbf{x}$ are then assumed to have the following conditional distribution $p(\mathbf{x}|\mathbf{u}) = \prod_i p(x_i|u_i)$, where

$$p(x_i|u_i) = \begin{cases} \mathcal{N}(0,\sigma_x) & \text{if } u_i = 1 \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

with $\mathcal{N}(0,\sigma_x)$ being a zero mean normal distribution with variance σ_x.

This prior distribution is complimented by a normal likelihood defied through the observation equation

$$\mathbf{y} = \mathbf{\Phi x} + \mathbf{e}, \tag{11}$$

where $\mathbf{e}$ is a multivariate i.i.d. normal variable with variance $\sigma_\mathbf{e}$.

$$p(\mathbf{y}|\mathbf{x},\mathbf{u}) = \mathcal{N}(\mathbf{\Phi x}, \sigma_e \mathbf{I}), \tag{12}$$

[1]See the reference for the exact conditions on the samples required for this to hold.

It thus remains to specify the distribution of the indicator variables u_i. Two potential alternatives here are

$$p(\mathbf{u}) = \begin{cases} \frac{1}{\binom{N}{K}} & \text{if } u_i = 1 \text{ for exactly K indices i} \\ 0 & \text{otherwise,} \end{cases} \tag{13}$$

or

$$p(\mathbf{u}) = \prod_i p(u_i) = \binom{N}{K} p^K (1-p)^{N-K}. \tag{14}$$

The first proposal draws a set of K distinct indices (without replacement) and thus leads to a distribution on the indicator variables that is not independent. The distribution of the indicator variables in the second distribution is, by contrast, i.i.d. binary.

3.2. A Bayesian IHT algorithm

As in [9], we consider a randomised IHT algorithm. However, our aim is to ensure that samples are drawn form the correct distribution. For the Bayesian model above, the true posterior is proportional to

$$p(\mathbf{x}|\mathbf{y}) \propto \sum_{u_i \in \{0,1\}} p(\mathbf{y}|\mathbf{x},\mathbf{u}) p(\mathbf{x}|\mathbf{u}) p(\mathbf{u}), \tag{15}$$

where the summation is over all combinations of $\mathbf{u}$. Due to the combinatorial nature of the problem, we cannot evaluate the above summation and so cannot efficiently sample from this distribution. Instead, we sample $\mathbf{x}$ from a Markov Chain, where each sample in the chain is defined using a randomised iterative hard thresholding iteration. Let $\mathbf{x}^1 = \mathbf{0}$ and define

$$\mathbf{a}^{n+1} = \mathbf{x}^n + \mu^n \mathbf{g}^n, \tag{16}$$

where $\mathbf{g}^n = \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi x}^n)$ and

$$\mu^n = \frac{\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n}}{\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}}, \tag{17}$$

with $\Gamma^n = \{i : x_i \neq 0\}$. We then use a proposal distribution $\pi(\mathbf{u}|\mathbf{a}^{n+1})$ which samples the new indicator variables $\mathbf{u}^{n+1}$ from a distribution that depends on $\mathbf{a}^{n+1}$ and thus on $\mathbf{x}^n$. The idea is that large values in $\mathbf{a}^{n+1}$ should indicate locations in which $\mathbf{x}$ is likely to be non-zero so that for large n the distribution $p(\mathbf{x}|\mathbf{y})$ is, up to some constant c, well approximated by our sampling distribution

$$p(\mathbf{x}|\mathbf{y}) \approx c\pi(\mathbf{y}|\mathbf{x},\mathbf{u})\pi(\mathbf{x}|\mathbf{u})\pi(\mathbf{u}|\mathbf{x}^n). \tag{18}$$

The conditional distributions $\pi(\mathbf{y}|\mathbf{x},\mathbf{u})$ and $\pi(\mathbf{x}|\mathbf{u})$ could be, for example, the true prior and likelihood $p(\mathbf{x}|\mathbf{y},\mathbf{u})$, which are multivariate normal. However, sampling from $p(\mathbf{x}|\mathbf{y},\mathbf{u})$ would require the calculation of a matrix inverse, which will be slow for large problems and an alternative i.i.d. proposal distribution can be used instead.

The Markov Chain is defined through the following model. In the first iteration we sample from

$$\pi(\mathbf{x}^1, \mathbf{u}_1) \propto \pi(\mathbf{x}^1|\mathbf{u}^1)\pi(\mathbf{u}^1), \tag{19}$$

whilst in iteration n we use

$$\pi(\mathbf{x}^n, \mathbf{u}^n|\mathbf{x}^{n-1}) \propto \pi(\mathbf{x}^n|\mathbf{u}^n, \mathbf{x}^{n-1})\pi(\mathbf{u}^n|\mathbf{x}^{n-1}). \tag{20}$$

3.3. Calculating Importance Weights

For any sample $\mathbf{x}_s$ we draw, we calculate Importance weights. To do this, we need to evaluate the distribution $\pi(\mathbf{x}^n, \mathbf{u}^n)$. Unfortunately, to evaluate $\pi(\mathbf{x}^n, \mathbf{u}^n)$, we would have to integrate the distribution $\pi(\mathbf{x}^1, \mathbf{u}^1) \prod_{n>1} \pi(\mathbf{x}^n, \mathbf{u}^n | \mathbf{x}^{n-1})$ over all possible sample paths $\mathbf{x}^1, \dots \mathbf{x}^{n-1}$ which is not feasible.

Instead, we take our motivation from the field of sequential importance sampling. Let us draw S samples $\mathbf{x}_s^1$ from the distribution $\pi(\mathbf{x}^1) \propto \pi(\mathbf{x}^1|\mathbf{u}^1)\pi(\mathbf{u}_i^1)$. As in sequential Monte Carlo methods, we approximate the distribution $p(\mathbf{x}|\mathbf{y})$ with the following distribution

$$p(\mathbf{x}|\mathbf{y}) \approx \sum_s w_s \delta(\mathbf{x}_s^2), \tag{21}$$

where the weights are the importance weights

$$w_s^1 \propto p(\mathbf{x}_s^1|\mathbf{y})/(\pi(\mathbf{x}_s^1|\mathbf{u}_s^1)\pi(\mathbf{u}_s^1)). \tag{22}$$

We then use boot strapping to generate a new set of samples $\tilde{\mathbf{x}}_s^1$ by drawing samples from $\mathbf{x}_s^1$ with replacement where each $\mathbf{x}_s^1$ is drawn with probability w_s^1. In the limit as S goes to infinity, the samples $\mathbf{x}_s^1$ are effectively drawn form the desired distribution, however, due to the finite S, the approximation is biased. To reduce this bias, we use several sampling steps and thus generate a Markov Chain of samples.

In the n^{th} step of the Markov Chain, we draw the S samples $\mathbf{x}_s^n$ from the distribution

$$\pi(\mathbf{x}^n|\mathbf{u}^n)\pi(\mathbf{u}^n|\tilde{\mathbf{x}}^{n-1}). \tag{23}$$

The marginal distribution for $\mathbf{x}_s^n$ and $\mathbf{u}_s^n$ is then approximately

$$\begin{aligned} \pi(\mathbf{x}_s^n, \mathbf{u}_s^n) &= \int_{\tilde{\mathbf{x}}^{n-1}} \pi(\mathbf{x}^n, \tilde{\mathbf{x}}^{n-1}\mathbf{u}^n)\pi(\tilde{\mathbf{x}}^{n-1}) \\ &\approx \sum_{r=1}^{S} \pi(\mathbf{x}_s^n|\mathbf{u}_s^n)\pi(\mathbf{u}_s^n|\tilde{\mathbf{x}}_r^{n-1})\pi(\tilde{\mathbf{x}}_r^{n-1}). \end{aligned} \tag{24}$$

which can be used to calculated new importance weights

$$w_s^n \propto p(\mathbf{x}_s^n|\mathbf{y})/(\sum_{r=1}^{S} \pi(\mathbf{x}_s^n|\mathbf{u}_s^n)\pi(\mathbf{u}_s^n|\tilde{\mathbf{x}}_r^{n-1})\pi(\tilde{\mathbf{x}}_r^{n-1})). \tag{25}$$

3.4. Calculating and sampling from the proposal distributions

The speed of convergence of the Monte Carlo estimate to the true sample mean depends (to some extend) on the closeness of the sample distribution to the distribution of interest [6].

Depending on the true prior probability $p(\mathbf{u})$, different choices are advisable. If our prior is that in equation (13), then the posterior will be zero for any signal that has more then K non-zero elements. This implies that Importance weights will also be zero whenever the samples $\mathbf{u}^n$ have more than K non-zero elements. Weights that are zero do not contribute to the estimated sample mean leading to an inefficient sampling scheme. To avoid this, the proposal distribution should ideally constrain the samples' sparsity. This leads necessarily to a proposal distribution that is not independent. The proposed sampling in [9] is of this type and is defined implicitly through the following procedure (where $\mathcal{I}_i(\gamma)$ is the indicator function which is 1 if $i = \gamma$ and 0 otherwise).

We will call this approach **Proposal Distribution 1**.

- $\Gamma^0 = \emptyset$
- Iterate for $k = 0, k++, n \leq K$
  - $c = \sum_{i \notin \Gamma^0} a_i(\mathbf{x}^n)$
  - $\{\tilde{a}_i(\mathbf{x}^n)\} = \{a_i(\mathbf{x}^n)/c : i \notin \Gamma^0\}$
  - $\gamma^k \sim p(\gamma|\{\tilde{a}_i(\mathbf{x}^n)\}) = \prod_{i \notin \Gamma^0} \mathcal{I}_i(\gamma)\tilde{a}_i(\mathbf{x}^n)$
  - $\Gamma^k = \Gamma^{k-1} \cup \gamma^k$,

The other approach, tailored to the prior in equation (14), would be to sample from an independent distribution.

We will call this approach **Proposal Distribution 2**.

$$\pi(\mathbf{u}|\mathbf{a}) = \prod_i f(a_i)^{u_i}(1 - f(a_i))^{1-u_i}, \tag{26}$$

where $f(\cdot)$ is a function such that $f(\mathbf{a}_i) \leq 1$.

Both of these proposal distributions are easy to sample from, however, to calculate the weights, we also need to be able to evaluate $\pi(\mathbf{u}^n|\mathbf{a}^n)$ for given $\mathbf{u}^n$ and $\mathbf{a}$. This is easy for equation (26), which is given in closed form, but the dependent distribution proposed in [9] is only defined implicitly through the recursion used for sampling. The problem is the following. The distribution from which γ is sampled in each recursion changes, because samples that are selected in previous iterations are removed so that the probabilities for element selection changes. After K elements have been selected, there are $K!$ different permutations in the order in which the elements could have been selected and the probability for each of these is different. Thus, we will have to sum the probabilities over $K!$ permutations, which is computationally expensive. In order to use this proposal distribution, we thus need to approximate the value of this probability. In our experiments, we use an approximation that assumes that each permutation has roughly the same probability as the probability for the order we have actually sampled. This probability is easy to keep track of during sampling.

4. THE ALGORITHM

This leads to the following algorithm to draws S different Importance samples

- Initialise $\tilde{\mathbf{x}}_s^0 = \mathbf{0}$ for $s \in \{1, 2, \dots, S\}$
- Iterate for $n = 0, n++, n \leq N_{max}$
  - Iterate for $s = 1, s++, s \leq S$
    - $\mathbf{g}^n = \mathbf{\Phi}^T(\mathbf{y} - \mathbf{\Phi}\tilde{\mathbf{x}}_s^n)$
    - $\mu^n = \frac{\mathbf{g}_{\Gamma^n}^T \mathbf{g}_{\Gamma^n}}{\mathbf{g}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n}^T \mathbf{\Phi}_{\Gamma^n} \mathbf{g}_{\Gamma^n}}$
    - $\mathbf{a}^{n+1} = (\mathbf{x}^n + \mu^n \mathbf{g}^n)$,
    - $\mathbf{u} \sim \pi(\mathbf{u}|\mathbf{a}^{n+1})$
    - $\mathbf{x}_s^{n+1} \sim \pi(\mathbf{a}^{n+1}|\mathbf{x}_s^n, \mathbf{u})$
    - calculate the weight w_s^{n+1} using equation (25)
  - $\hat{w}_s^{n+1} = w_s^{n+1}/\sum_s w_s^{n+1}$
  - draw $\{\tilde{\mathbf{x}}_s^{n+1}\}_s$ from $\{\mathbf{x}_s^{n+1}\}_s$, with probability $\hat{w}_s^{n+1}$
- $\mathbf{x} = 1/S \sum_s \tilde{\mathbf{x}}_s$

As with all Monte Carlo estimates, it is difficult to decide how many samples to draw. In addition, we here also have the added complexity of the Markov Chain, which we can, again, terminate at any iteration. This will not have any influence on the convergence of the estimate in the limit of infinitely many samples, but will influence the variance and bias of any finite sample estimate.

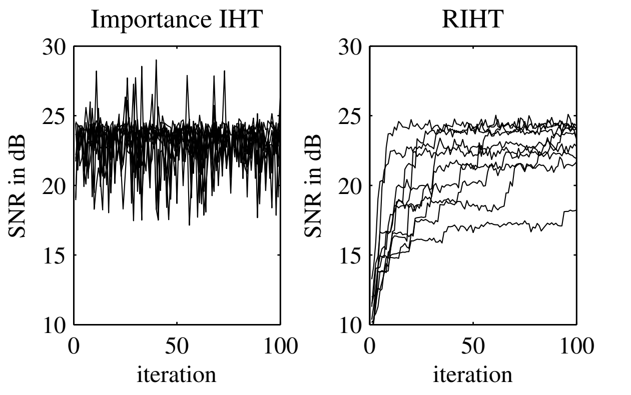

Fig. 1. Fixed sparsity prior. Convergence of the Importance sampling algorithm with iteration by iteration bootstrap re-sampling (left) and the Randomised IHT algorithm (RIHT - right).

5. INITIAL RESULTS

In this section we simulate a small compressed sensing recovery problem. Each instance is constructed by generating a real matrix $\mathbf{\Phi} \in \mathbb{R}^{64\times 128}$ with i.i.d zero mean and unit variance normal entries, followed by a normalisation step in which the columns of $\mathbf{\Phi}$ were normalised to unit length. We either use a fixed sparsity level of $K = 10$ or draw indicator variables from a Bernoulli distribution. Non-zero $\mathbf{x}$ are drawn from a zero mean unit variance normal. Gaussian noise is added to the observations $\mathbf{y}$, again with zero mean, but with standard deviation of 0.1. Results are reported in terms of the normalised mean squared error (as well as in terms of SNR in dB).

$$\text{NMSE} = \frac{\|\mathbf{x}-\hat{\mathbf{x}}\|^2}{\|\mathbf{x}\|^2}, \tag{27}$$

where $\mathbf{x}$ is the true sparse signal and $\hat{\mathbf{x}}$ its estimate.

Where used, the IHT algorithm was stopped as soon as

$$\big|\|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^{n-1}\|/\|\mathbf{y}\| - \|\mathbf{y}-\mathbf{\Phi}\mathbf{x}^{n}\|/\|\mathbf{y}\|\big| \leq 10^{-6}. \tag{28}$$

5.1. Known sparsity with uniform prior

Using *Proposal Distribution 1*, we compared the Randomised IHT algorithm of [9] to an Importance estimate in which the same sampling scheme is used, but where we also calculated and used an approximation to the importance weights to resample the "particles" in each iteration. For a single instance of the measurement system and sparse signal, we run the algorithms 10 times, each for 100 iterations, drawing 20 sample in each run. In each iteration, we calculated the SNR value between the sample mean and the true value. The SNR per iteration is plotted in Figure 1.

It is clear from this figure that the re-sampling step based on the importance weights introduced in our method leads to a significantly faster convergence of the method. The Randomised IHT algorithm on the other hand converges much slower and, for the example shown here, does not seem to reach the stationary distribution after 100 iterations. This convergence increase however also comes at the cost of an increase in the variance in the estimate. The normalised RMS values achieved after 100 iterations is shown in Figure 2.

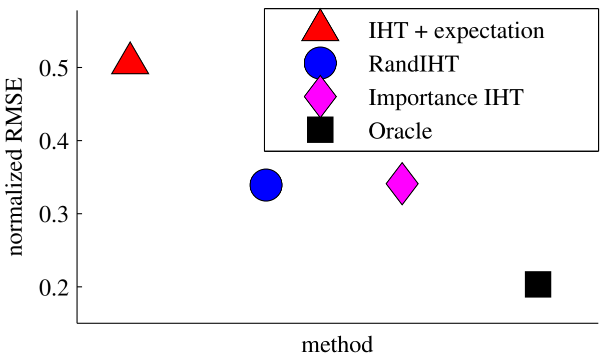

Fig. 2. Fixed sparsity prior. Normalised RMS error for the Importance IHT algorithm (magenta diamond), the Randomised IHT algorithm (blue circle), the traditional IHT algorithm (red triangle) and an oracle estimate that assumes knowledge of the sparse support (black square). Results are averaged over 100 problem realisations.

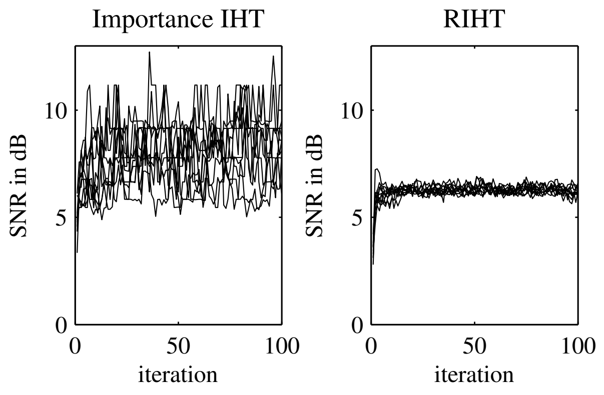

Fig. 3. Bernoulli prior. Convergence of the Importance sampling algorithm with iteration by iteration bootstrap re-sampling (left) and the Randomised IHT algorithm (RIHT - right).

5.2. Unknown sparsity with i.i.d. Bernoulli prior

In our second experiment, we draw the true sparse vector using *Proposal Distribution 2*, so that the expected number of non-zero elements is equal to that in the previous experiment (i.e. 10). We then repeated the experiment above, with the variation that we now used *Proposal Distribution 2* in the algorithm, that is, the indicator variables were drawn from Bernoulli distributions, with the probability of u_i^{n+1} depending on a_i^n.

The convergence of the Importance sampling based IHT algorithm (i.e. the algorithm that uses bootstrapping based on importance weights) and the Randomised IHT algorithm are shown in Figure 3. In this example, there is no appreciable difference in the convergence speed of the two methods, but a clear improvement is evident in the mean SNR for the Importance sampling scheme, even though this has again a significantly higher variance.

6. CONCLUSION

Randomisation of the sparse selection step in several greedy sparse approximation/recovery problems can be used to find better estimates in terms of expected mean squared error. We have here evaluated two possible approaches to calculate samples and appropriate importance weights and compared the results to those obtained without importance weighting.

From our experience here, for the small number of samples feasible here, the choice of the proposal distribution seems significantly more important than the use of the appropriate importance weights.

As with other full Bayesian approaches, a clear drawback is the increase in computing resources required. Instead of running a single instance of the Iterative Hard Thresholding algorithm, one instance has to be run for each sample, so that the computational burden increases linearly with the sample number. However, individual iterations are paralelisable and so, the use of highly parallel computing architectures, such as those of GPU based processors, can make randomised algorithms feasible alternatives in certain applications.

7. REFERENCES

[1] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information.," *IEEE Transactions on Information Theory*, vol. 52, no. 2, pp. 489–509, Feb 2006.

[2] E. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, no. 8, pp. 1207–1223, 2006.

[3] E. Candès and J. Romberg, "Quantitative robust uncertainty principles and optimally sparse decompositions," *Foundations of Comput. Math*, vol. 6, no. 2, pp. 227–254, 2006.

[4] D. Donoho, "Compressed sensing," *IEEE Trans. on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[5] J.M. Bernado and A.F.M. Smith *Bayesian Theory,* Wiley, 2000.

[6] C.P. Robert and G. Casella *Monte Carlo Statistical Methods* Springer-Verlag, 1999.

[7] T. Blumensath and M. Davies, "Iterative hard thresholding for compressed sensing," *to appear in Applied and Computational Harmonic Analysis*, 2009.

[8] M. Elad and I. Yavneh, "A plurality of sparse representations is better than the sparsest one alone," IEEE Transactions on Information Theory, vol. 55, no. 10, pp. 47014714, 2009.

[9] R. Crandall, B. Dong and A. Bilgin "Randomized Iterative Hard Thresholding: A Fast Approximate MMSE Estimator for Sparse Approximations," preprint http://math.arizona.edu/∼dongbin/Publications/RandIHT_June2013.pdf, accessed 15:17:2013.

[10] G. Davis, S. Mallat and M. Avellaneda "Greedy adaptive approximation" J. Constr. Approx., vol 13, pp. 57–98, 1997.