FORWARD-BACKWARD SEARCH FOR COMPRESSED SENSING SIGNAL RECOVERY

Nazim Burak Karahanoglu[1,2], Hakan Erdogan[1]

[1]Department of Electronics Engineering, Sabanci University, Istanbul, Turkey
[2]Information Technologies Institute, TUBITAK-BILGEM, Kocaeli, Turkey

ABSTRACT

Reconstruction of sparse signals from reduced dimensions requires the solution of an l_0 norm minimization, which is unpractical. A number of algorithms have appeared in literature, including ℓ_1 minimization, greedy pursuit algorithms, Bayesian methods and nonconvex optimization. This manuscript introduces a greedy approach, called the Forward-Backward Pursuit (FBP), which iteratively enlarges the support by consecutive forward and backward steps. At each iteration, the forward step first expands the support, while the following backward step prunes it. The number of atoms selected by the forward step is selected higher than the number of removals, hence the support is expanded at the end of each iteration. The recovery performance of the proposed method is demonstrated via simulations including different nonzero coefficient distributions in noisy and noise-free scenarios.

***Index Terms*—** Compressed sensing, forward-backward search, sparse signal reconstruction, greedy algorithms

1. INTRODUCTION

The fundamental problem of Compressed sensing (CS) is the reconstruction of the sparse signals from reduced sets of observations. Let $\mathbf{x}$ be a K-sparse signal of length N and $\mathbf{\Phi}$ the $M \times N$ observation matrix where $M < N$. We are interested in recovering $\mathbf{x}$ back from the observation

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}. \tag{1}$$

The dimensionality reduction via $\mathbf{\Phi}$ makes the analytical solution of (1) ill-posed. Hence, the recovery problem is reformulated as an l_0 norm minimization due to sparsity:

$$\mathbf{x} = \arg\min \|\mathbf{x}\|_0 \quad s.t. \quad \mathbf{y} = \mathbf{\Phi}\mathbf{x}. \tag{2}$$

As direct solution of (2) is computationally intractable, a number of alternative solutions have emerged. An overview of major algorithmic classes can be found in [1].

Among CS reconstruction methods, convex relaxation [2, 3, 4, 5, 6] replaces the l_0 minimization in (2) with l_1 minimization, making the solution tractable via convex optimization algorithms such as linear programming, as proposed by Basis Pursuit (BP) [6]. Greedy pursuit algorithms, such as Matching Pursuit (MP) [7], Orthogonal MP (OMP) [8], Compressive Sampling MP (CoSaMP) [9], Subspace Pursuit (SP) [10] and Iterative Hard Thresholding (IHT) [11, 12], employ iterative mechanisms. Among these, MP and OMP build up the support of $\mathbf{x}$ iteratively, adding one element per iteration. SP and CoSaMP, on the other hand, apply a two-stage iterative scheme, which at each iteration first expands and then shrinks the support by the same number of elements, keeping the support size fixed between iterations. [13] provides an algorithmic framework called Two-Stage Thresholding (TST), into which algorithms such as SP and CoSaMP fall.

In this paper, we propose a two-stage iterative greedy algorithm, which we call Forward-Backward Pursuit (FBP). FBP possesses forward selection and backward removal steps which iteratively expand and shrink the support estimate of $\mathbf{x}$. Selecting the forward step size higher than the backward step size, the support is iteratively expanded. FBP has some advantages over both SP and OMP. Unlike SP, FBP does not require the a priori estimate of the sparsity, K, which is not easy to provide in practice. On the other hand, the backward step allows FBP to remove some possibly misplaced atoms from the estimate of the support, which is a clear advantage over OMP. Similar to SP and CoSaMP, FBP is also a member of TST-type schemes, while the idea of expanding the support is investigated for the first time in this concept. Moreover, our results show that FBP can perform better than SP in some scenarios, which indicates that SP is not overall the best TST scheme as proposed in [13].

An adaptive forward-backward greedy algorithm, FoBa has been investigated in [14] for the sparse learning problem. This algorithm consists of forward and backward steps with fixed size 1. Additionally, it includes an adaptive criterion for deciding whether or not to take the backward step. FBP, on the other hand, does take forward and backward steps with sizes greater than 1. This allows FBP to terminate in less iterations. In addition, FBP includes no adaptive criterion for taking the backward step (Note that this is not trivial when the backward step size is greater than 1.). Finally, FoBa has been applied for the sparse learning problem, while we propose and evaluate FBP for CS signal recovery.

This work is organized as follows: First, we give a brief description of OMP and SP algorithms. The FBP algorithm is introduced in Section 3. The reconstruction performance of FBP is demonstrated in Section 4 in comparison to BP, SP and OMP algorithms. We conclude with a brief summary.

2. GREEDY PURSUITS

In this section we summarize two greedy algorithms, OMP and SP, regarding their similarities with the FBP algorithm. Beforehand, we define the notation that is used throughout the paper: Let $S^{(k)}$ be the estimated support of $\mathbf{x}$ after the k'th iteration, while $\tilde{S}^{(k)}$ stands for the expanded support after the forward selection step of the k'th iteration. $\tilde{\mathbf{y}}^{(k)}$ denotes the approximation of $\mathbf{y}$ after the k'th iteration. $\mathbf{r}^{(k)}$ is the residue of $\mathbf{y}$ after the k'th iteration. $\mathbf{\Phi}_S$ denotes the matrix composed of the columns of $\mathbf{\Phi}$ indexed by S. Similarly, $\mathbf{x}_S$ is the vector consisting of the entries of $\mathbf{x}$ with indices in S.

OMP is an iterative algorithm that searches for the support of $\mathbf{x}$ by identifying one element per iteration. It starts with $S_{(0)} = \emptyset$ and $\mathbf{r}^{(0)} = \mathbf{y}$. At the iteration k, OMP expands the estimated support of the last iteration, $S^{(k-1)}$, with the index of the dictionary atom closest to $\mathbf{r}^{(k-1)}$ (i.e. index of the largest magnitude entry of $\mathbf{\Phi}^{\mathrm{T}}\mathbf{r}^{(k-1)}$). Following, $\tilde{\mathbf{y}}^{(k)}$ is computed via orthogonal projection of $\mathbf{y}$ onto $\mathbf{\Phi}_{S^{(k)}}$ and the residue is updated as $\mathbf{r}^{(k)} = \mathbf{y} - \tilde{\mathbf{y}}^{(k)}$. The iterations are carried out until the termination criterion is met. In this work, we stop OMP when $\|\mathbf{r}^{(k)}\|_2$ falls below the threshold $\varepsilon\|\mathbf{y}\|_2$.

SP and CoSaMP combine selection of multiple columns per iteration with a pruning step, maintaining K element support sets throughout the iterations. At iteration k, SP first expands the selected support with indices of the K largest magnitude entries of $\mathbf{\Phi}^{\mathrm{T}}\mathbf{r}^{(k-1)}$), obtaining an extended support $\tilde{S}^{(k)}$ of size $2K$ (Alternatively, CoSaMP expands the support by $2K$ elements.). In the second step, the orthogonal projection coefficients of $\mathbf{y}$ onto $\mathbf{\Phi}_{\tilde{S}^{(k)}}$ are computed, and the estimated support $S^{(k)}$ is obtained by pruning $\tilde{S}^{(k)}$ to contain only the indices of the K largest magnitude projection coefficients. $\mathbf{r}^{(k)}$ is finally computed using the approximation $\tilde{\mathbf{y}}^{(k)}$ which is obtained by orthogonal projection of $\mathbf{y}$ onto $\mathbf{\Phi}_{S^{(k)}}$. The iterations are stopped when $\|\mathbf{r}^{(k)}\|_2 \geq \|\mathbf{r}^{(k-1)}\|_2$. Once stopped at iteration l, the final estimate of the support is $S^{(l-1)}$, while orthogonal projection of $\mathbf{y}$ onto $\mathbf{\Phi}_{S^{(l-1)}}$ yields the corresponding nonzero values. CoSaMP and SP are provided with Restricted Isometry Property (RIP) [2, 15, 16] based theoretical guarantees, showing that these two-stage schemes reduce the reconstruction error iteratively when some certain RIP conditions are met.

3. FORWARD-BACKWARD PURSUIT

OMP and other MP variants are forward greedy algorithms, where each iteration consists of a single forward step that expands the support estimate $S^{(k)}$. Whilst this mechanism builds up $S^{(k)}$ iteratively, any element that is inserted into $S^{(k)}$ stays there till termination of the algorithm. In case an incorrect element is chosen, there is no chance to remove it from the support, which may cause the recovery to fail. To illustrate, consider a well-known example: Let $\mathbf{x}$ be the summation of two equal magnitude sinusoids with very close frequencies, f_1 and f_2, and $\mathbf{\Phi}$ be an overcomplete sinusoidal dictionary, containing atoms with frequencies f_1, f_2 and $f_3 = (f_1+f_2)/2$ among others. OMP will first select the component with frequency f_3. Then, during the next iterations, the algorithm will try to cover for this error, choosing components other than the two correct ones, and will finally fail.

Each iteration of SP and CoSaMP, on the other hand, employs two stages, which iteratively expand and shrink the support, allowing both addition and removal of nonzero indices to the support estimate. These algorithms require the sparsity level K a priori, as the support size is fixed. This is an important handicap in most practical cases, where K is unknown or it is not desired to fix it. It is possible to choose K very large, however in that case the nice theoretical guarantees of CoSaMP and SP are nearly lost, as the probability of satisfying the required RIP condition decreases with increasing K.

The Forward-Backward Pursuit algorithm provides an iterative forward-backward scheme that both allows removal of misidentified atoms from the support estimate and requires no a priori knowledge of K. The scheme can be outlined as follows: The support estimate is initialized as an empty set: $S^{(0)} = \emptyset$, and the residue as $\mathbf{r}^{(0)} = \mathbf{y}$. At iteration k, first the forward step expands $S^{(k-1)}$ by indices of α largest magnitude elements in $\mathbf{\Phi}^{\mathrm{T}}\mathbf{r}^{(k-1)}$. This builds up the extended support $\tilde{S}^{(k)}$. Then the projection coefficients are computed by orthogonal projection of $\mathbf{y}$ onto $\mathbf{\Phi}_{\tilde{S}^{(k)}}$. The backward step prunes $\tilde{S}^{(k)}$ by removing the indices of β smallest magnitude projection coefficients, which produces the support estimate $S^{(k)}$. Following, $\tilde{\mathbf{y}}^{(k)}$ is computed via orthogonal projection of $\mathbf{y}$ onto $\mathbf{\Phi}_{S^{(k)}}$ and the residue is updated as $\mathbf{r}^{(k)} = \mathbf{y} - \tilde{\mathbf{y}}^{(k)}$. The iterations are carried on until either $\|\mathbf{r}^{(k)}\|_2$ falls below the threshold $\varepsilon\|\mathbf{y}\|_2$. To avoid the algorithm running for too many iterations in case of a failure, the maximum number of iterations is set as K_{max}. After termination at iteration l, $S^{(l)}$ gives the nonzero indices of the estimate $\tilde{\mathbf{x}}$, and the corresponding nonzero elements are the projection coefficients of $\mathbf{y}$ onto $S^{(l)}$. The pseudo-code of FBP is given in algorithm 1.

Unlike SP, FBP does not require an a priori estimate of the sparsity level K. Selecting the forward step size α greater than the backward step size β, FBP builds up support estimate by $\alpha - \beta$ atoms per iteration. With this choice, the backward pruning step does not require K a priori. In addition, the termination criterion of FBP does not also require the a priori estimate of K as the termination is based on the residual power. The pay-off for this, however, is that the nice theoretical guarantees cannot be provided in a way similar to SP or CoSaMP, which make use of the support size being fixed as K after the backward step. For the time being, we cannot provide a complete theoretical analysis of FBP, and leave this as a future work. Note that, however, most of the theoretical

Algorithm 1 FORWARD-BACKWARD PURSUIT

Input: $\mathbf{\Phi}$, $\mathbf{y}$
Define: α, β, $K_{\max}$, ε
Initialize: $S^{(0)} = \emptyset$, $\mathbf{r}^{(0)} = \mathbf{y}$
$k = 0$
while true **do**
  $k = k + 1$
  $T_f = \{$indices corresponding to α largest magnitude elements in $\mathbf{\Phi}^{\mathrm{T}}\mathbf{r}^{(k-1)}\}$
  $\tilde{S}^{(k)} = S^{(k-1)} \cup T_f$
  $\tilde{\mathbf{w}} = \arg\min_{\mathbf{z}} \|\mathbf{y} - \mathbf{\Phi}_{\tilde{S}^{(k)}}\mathbf{z}\|$
  $T_b = \{$indices corresponding to β smallest magnitude elements in $\tilde{\mathbf{w}}\}$
  $S^{(k)} = \tilde{S}^{(k)} \setminus T_b$
  $\mathbf{w} = \arg\min_{\mathbf{z}} \|\mathbf{y} - \mathbf{\Phi}_{S^{(k)}}\mathbf{z}\|$
  $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{\Phi}_{S^{(k)}}\mathbf{w}$
  if $\|\mathbf{r}^{(k)}\|_2 \le \varepsilon\|\mathbf{y}\|_2$ or $|S^{(k)}| \ge K_{\max}$ **then**
    break
  end if
end while
$\tilde{\mathbf{x}}_{S^{(k)}} = \mathbf{w}$
$\tilde{\mathbf{x}}_{\{1,2,\ldots,N\}\setminus S^{(k)}} = 0$
return $\tilde{\mathbf{x}}$

analysis steps of SP or CoSaMP also hold for FBP.

As a clear advantage over OMP, the backward step of FBP gives the algorithm the ability to remove atoms from the support. Let's once again consider the example with sinusoids. Assume we run FBP with $\alpha = 3$ and $\beta = 1$. During the forward step of the first iteration, FBP will select the three components with frequencies f_1, f_2 and f_3. Following orthogonal projection, the backward step will eliminate f_3, and the recovery will be successful after the first iteration.

4. EXPERIMENTAL EVALUATION

In this section, reconstruction performance of FBP is demonstrated in comparison to BP, SP and OMP. The experiments cover both noise-free and noisy scenarios where reconstruction performance and run times are compared. Each test is repeated over 500 randomly generated samples of length $N = 256$. Nonzero entries of the test samples are modeled as iid random variables either from standard Gaussian distribution or uniform distribution in $[-1, 1]$. We refer to these as Gaussian or uniform sparse signals. $M = 100$ observations were taken from each vector. For each vector, a different observation matrix is drawn from Gaussian distribution with mean 0 and standard deviation $1/N$. ε is set to 9×10^{-7} in the noiseless case, while it is selected with respect to the noise level in noisy scenarios. $K_{\max}$ is selected as 55. Note that, in this work, ε and $K_{\max}$ are also valid for OMP, that is, OMP is run until $\|\mathbf{r}\|_2 \le \varepsilon\|\mathbf{y}\|_2$, with a maximum of $K_{\max}$ iterations allowed. For the noiseless cases, the results are stated in terms of the average normalized mean-squared-error (ANMSE) and exact reconstruction rates. ANMSE is defined as

$$ANMSE = \frac{1}{500}\sum_{i=1}^{500}\frac{\|\mathbf{x}_i - \hat{\mathbf{x}}_i\|_2^2}{\|\mathbf{x}_i\|_2^2} \tag{3}$$

where $\hat{\mathbf{x}}_i$ is the recovery of the i'th test vector $\mathbf{x}_i$. For noisy observations, we state the ANMSE in the decibel (dB) scale, calling it the distortion ratio.

Figure 1 depicts the reconstruction performance for Gaussian sparse vectors. A range of α values are employed for FBP, where $\beta = \alpha - 1$, expanding the support by one element per iteration. In general, FBP provides better recovery than the others, and its performance is improved with α. FBP outperforms the other algorithms in exact reconstruction rates. BP, SP and OMP start to fail at around $K = 25$, while for $\alpha \ge 20$, the FBP failures begin only when $K > 30$. As for the ANMSE, FBP is the best performer when $\alpha \ge 20$. With this setting, BP can beat FBP only when $K = 45$. In this example, we also observe that OMP performs better than SP. This is due to the chosen termination criterion, $\|\mathbf{r}\|_2 \le \varepsilon\|\mathbf{y}\|_2$, which improves OMP performance over the conventional scheme that runs for exactly K iterations (see [17] for a discussion of the two termination criteria.)

To investigate the performance of FBP with respect to β, we repeat the Gaussian sparse reconstruction experiment using different β values for $\alpha = 20$. Figure 2 illustrates the reconstruction results for this case. We observe that the exact reconstruction rates are improved when β is increased. On the other hand, the choice of β does not significantly affect the reconstruction error. This indicates that for smaller β values, failures start occurring at smaller magnitude nonzero elements, which do not significantly change the error. The results indicate that FBP is the best performer when $\beta > 15$, where it provides higher exact reconstruction rates than the others. As for the previous example, BP can produce lower ANMSE only for $K = 45$.

Figure 3 compares the speed of the FBP, SP and OMP. Here, BP is excluded as its run time is incomparably higher. Expectedly, increasing α while $\beta = \alpha - 1$ slows down FBP, following the increase in the dimensions of the orthogonal projection step. On the other hand, increasing $\alpha - \beta$ decreases the number of necessary iterations and FBP terminates faster. Most important, when $\alpha = 20$ and $\beta \le \alpha - 2$, the run time of FBP, SP and OMP are very close. In case $\alpha = 20$ and $\beta = 17$, the speed of FBP and OMP are almost the same, while the reconstruction performance of FBP is significantly better than the other algorithms involved.

Next, we simulate recovery of uniform sparse signals. The results are shown in Figure 4 for α in [2, 30] and $\beta = \alpha - 1$. We observe that FBP yields higher exact reconstruction rates than the other algorithms. In case $\alpha \ge 20$, FBP failures begin when $K > 30$, while the other algorithms start to fail at around $K = 25$. On the other hand, convex

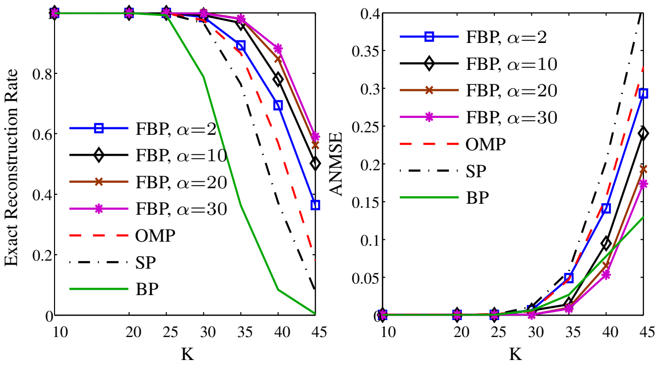

Fig. 1. Reconstruction results over sparsity for Gaussian sparse vectors. For FBP, $\beta = \alpha - 1$.

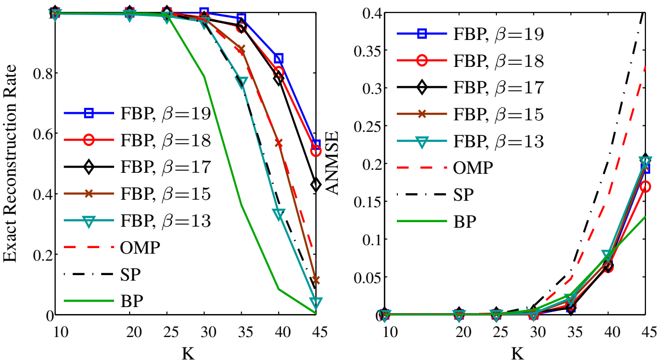

Fig. 2. Reconstruction results over sparsity for Gaussian sparse vectors. For FBP, $\alpha = 20$.

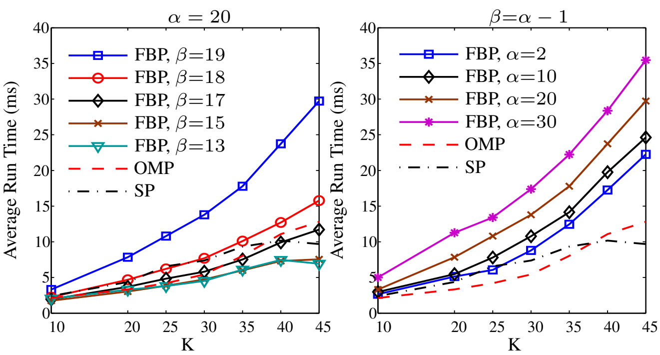

Fig. 3. Average run time per test sample for Gaussian sparse vectors. For FBP, $\alpha = 20$.

relaxation introduces less ANMSE than the greedy competitors, especially for $K > 30$. ANMSE of FBP and SP are close, with FBP being slightly better when $\alpha \geq 20$.

Finally, we simulate recovery from inaccurate observations, where observed vectors are contaminated with white

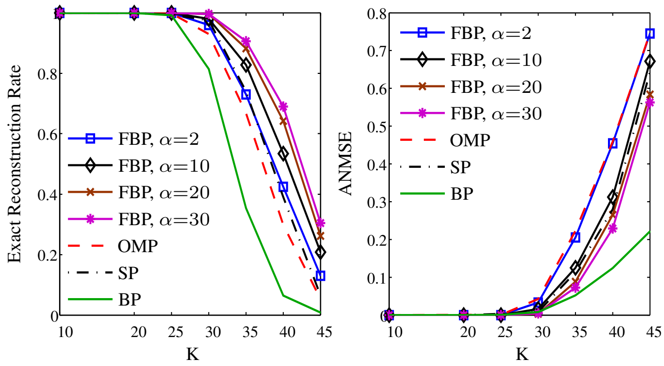

Fig. 4. Reconstruction results over sparsity for uniform sparse signals. For FBP, $\beta = \alpha - 1$.

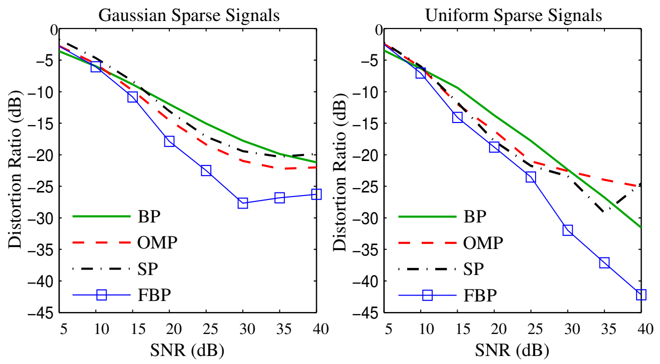

Fig. 5. Average recovery distortion over SNR in case of noise contaminated observations. For FBP, $\alpha = 20$ and $\beta = 17$.

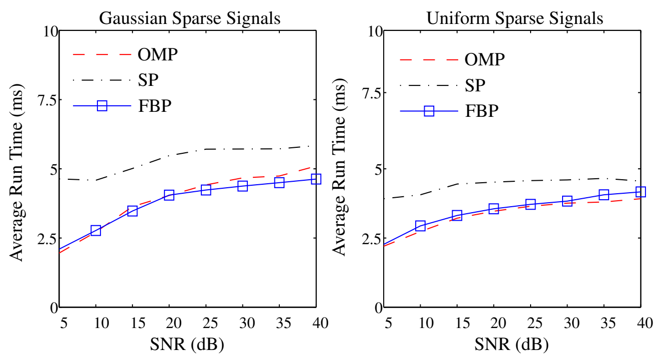

Fig. 6. Average run time per test sample in case of noise contaminated observations. For FBP, $\alpha = 20$ and $\beta = 17$.

Gaussian noise at signal-to-noise ratios (SNR) varying from 5 to 40 dB. FBP is run with $\alpha = 20$ and $\beta = 17$, as we have seen above that OMP and FBP require similar run times for this choice. Figure 5 shows the recovery results for noisy Gaussian and uniform sparse signals, while the run times are compared in Figure 6. Clearly, FBP yields the most accurate recovery in both cases, while BP can do slightly better than FBP only when SNR is 5 dB. The run times reveal that FBP is not only the most accurate algorithm, but is also as fast as OMP with $\alpha = 20$ and $\beta = 17$. Note that, increasing β beyond 17 would improve FBP recovery, while the algorithm would require a longer run time than OMP.

Recovery of binary nonzero signals are also of interest for us. We skip the details because of space limitations, however, we observed that SP generally performs better than FBP in binary examples, while BP yiels the best recovery.

Finally, our findings may appear contradictory to [13], however they are not. First, we run OMP until $\|\mathbf{r}\|_2 \leq \varepsilon\|\mathbf{y}\|_2$, by which OMP performance is significantly improved over going exactly for K iterations as in [13]. Second, we run the simulations for different nonzero element distributions than in [13]. We acknowledge that SP performs better than FBP in the binary scenario, which is in parallel to the findings of [13], where SP is named as the optimum TST scheme for this case. In combination, our results indicate that FBP provides better recovery than SP and BP in some distributions where the magnitudes of nonzero elements are not close. For comparable magnitudes, SP seems to be the optimal TST scheme, and BP the best performer. These findings do not contradict with that of [13].

5. SUMMARY

This work proposes a forward-backward scheme for CS recovery of sparse signals. This two-stage scheme iteratively expands the support estimate for the sparse signal, without requiring K a priori. Over traditional MP type algorithms, FBP provides a backward step for removing possibly misidentified atoms from the solution at each iteration. Simulations show that FBP improves the reconstruction for Gaussian and uniform distributions of nonzero elements in both noisy and noise-free scenarios, in run times close to that of OMP.

6. REFERENCES

[1] J. A. Tropp and S. J. Wright, "Computational methods for sparse solution of linear inverse problems," *Proc. IEEE*, vol. 98, no. 6, pp. 948–958, June 2010.

[2] E. Candès and T. Tao, "Decoding by linear programming," *IEEE Trans. Inf. Theory*, vol. 51, no. 12, pp. 4203–4215, Dec. 2005.

[3] E. Candes and J. Romberg, "Practical signal recovery from random projections," in *Proc. SPIE Computational Imaging*, 2005, vol. 5674, pp. 76–86.

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

[5] D. L. Donoho and M. Elad, "Optimally sparse representation in general (nonorthogonal) dictionaries via ℓ_1 minimization," in *Proc. Natl. Acad. Sci.*, Los Alamitos, CA, Mar. 2003, vol. 100, p. 21972202.

[6] S. Chen, D. Donoho, and M. Saunders, "Atomic decomposition by basis pursuit," *SIAM J. on Sci. Comp.*, vol. 20, no. 1, pp. 33–61, 1998.

[7] S. Mallat and Z. Zhang, "Matching pursuit in a time-frequency dictionary," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

[8] Y. C. Pati, R. Rezaiifar, and P. S. Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conference on Signals, Systems and Computers*, Los Alamitos, CA, 1993, vol. 1, pp. 40–44.

[9] D. Needell and J. A. Tropp, "CoSaMP: Iterative signal recovery from incomplete and inaccurate samples," *Appl. Comp. Harmonic Anal.*, vol. 26, pp. 301–321, 2008.

[10] W. Dai and O. Milenkovic, "Subspace pursuit for compressive sensing signal reconstruction," *IEEE Trans. Inf. Theory*, vol. 55, no. 5, pp. 2230–2249, May 2009.

[11] T. Blumensath and M.E. Davies, "Iterative thresholding for sparse approximations," *J. Fourier Anal. Appl.*, vol. 14, pp. 629–654, 2008.

[12] T. Blumensath and M.E. Davies, "Iterative hard thresholding for compressed sensing," *Appl. Comp. Harmonic Anal.*, vol. 27, no. 3, pp. 265–274, 2009.

[13] A. Maleki and D.L. Donoho, "Optimally tuned iterative reconstruction algorithms for compressed sensing," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 4, no. 2, pp. 330 –341, Apr. 2010.

[14] T. Zhang, "Adaptive forward-backward greedy algorithm for learning sparse representations," *IEEE Trans. Inf. Theory*, vol. 57, no. 7, pp. 4689 –4708, July 2011.

[15] E. Candès and T. Tao, "Near-optimal signal recovery from random projections: universal encoding strategies?," *IEEE Trans. Inf. Theory*, vol. 52, no. 12, pp. 5406–5425, 2006.

[16] E. Candès, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathematique*, vol. 346, no. 9-10, pp. 589–592, May 2008.

[17] N. B. Karahanoglu and H. Erdogan, "A comparison of termination criteria for A*OMP," in *European Signal Processing Conference*, Bucharest, Romenia, 2012.