

RECOVERY OF SPARSE SIGNALS VIA BRANCH AND BOUND LEAST-SQUARES

Abolfazl Hashemi and Haris Vikalo

Department of Electrical and Computer Engineering
University of Texas at Austin, Austin, TX, USA

ABSTRACT

We present an algorithm, referred to as Branch and Bound Least-Squares (BBLS), for the recovery of sparse signals from a few linear combinations of their entries. Sparse signal reconstruction is readily cast as the problem of finding a sparse solution to an underdetermined system of linear equations. To solve it, BBLS employs an efficient search strategy of traversing a tree whose nodes represent the columns of the coefficient matrix and selects a subset of those columns by relying on Orthogonal Least-Squares (OLS) procedure. We state sufficient conditions under which in noise-free settings BBLS with high probability constructs a tree path which corresponds to the true support of the unknown sparse signal. Moreover, we empirically demonstrate that BBLS provides performance superior to that of existing algorithms in terms of accuracy, running time, or both. In the scenarios where the columns of the coefficient matrix are characterized by high correlation, BBLS is particularly beneficial and significantly outperforms existing methods.

Index Terms— compressed sensing, sparse signal recovery, branch-and-bound algorithm, accelerated orthogonal least-squares

1. INTRODUCTION

Reconstruction of a sparse signal from a relatively small number of its linear measurements is typically formalized as the problem of finding a sparse solution to an underdetermined system of linear equations. Such problems arise in a number of practical scenarios including estimation of sparse channels in communication systems [1], sparse subspace clustering [2], compressive DNA microarrays [3], and a number of other applications in signal processing [4–6]. Consider the linear measurement model

$$\mathbf{y} = \mathbf{A}\mathbf{x} + \boldsymbol{\nu}, \quad (1)$$

where $\mathbf{y} \in \mathbb{R}^n$ denotes the vector of observations, $\mathbf{A} \in \mathbb{R}^{n \times m}$ is the coefficient matrix assumed to be full rank, $\boldsymbol{\nu} \in \mathbb{R}^n$ is the additive measurement noise vector, and $\mathbf{x} \in \mathbb{R}^m$ is a k -sparse unknown vector, i.e., a vector with at most k non-zero components. Formally, the search for a sparse approximation to

\mathbf{x} leads to the so-called cardinality-constrained least-squares problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_2^2 \quad \text{subject to} \quad \|\mathbf{x}\|_0 \leq k \quad (2)$$

which is known to be NP-hard; here $\|\mathbf{x}\|_0$ denotes the ℓ_0 -norm of \mathbf{x} , i.e., the number of non-zero components of \mathbf{x} . The fact that optimization problem (2) is computationally challenging has motivated search for efficient heuristics which explore trade-off between accuracy and speed. To enable computationally efficient search for sparse \mathbf{x} approximating (2), techniques such as Basis Pursuit (BP) [7, 8] replace the non-convex ℓ_0 -norm-constrained optimization (2) by a sparsity-promoting ℓ_1 -norm optimization. Another approach has been taken by iterative heuristics including orthogonal matching pursuit (OMP) [9] and orthogonal least-squares [10] algorithms which attempt to solve (2) greedily; specifically, those methods rely on locally optimal decisions to identify columns of \mathbf{A} which correspond to non-zero components of \mathbf{x} . The performance of greedy heuristics has been studied in various settings for both OMP [11–15] and OLS [16–24]. In recent years, heuristics that exploit low complexity of OMP and rely on it to traverse a search tree along paths that represent promising candidates for the support of \mathbf{x} have been proposed. The Tree-search Based OMP (TB-OMP) [25], revisited as Multipath Matching Pursuit (MMP) and analyzed in [26], combines the so-called breadth-first and depth-first search strategies with OMP to conduct a suboptimal search through the tree of all possible subsets (paths) where each node is required to have a fixed and limited number of children. A similar breath-first search method is proposed in [27] with application to MIMO radar where multiple measurement vectors are available. A*OMP [28] performs A* search to look for the best solution among all possible subsets (paths). Although these algorithms improve the performance of OMP, their complexity is prohibitive for large k . In addition, when the columns of \mathbf{A} are correlated – which often arises in applications such as [1, 2] – their performance significantly deteriorates. Therefore, improving on the performance of greedy algorithms while retaining practical feasibility and exhibiting robustness in various scenarios remains a challenge.

In this paper, we exploit a recursive relation between the components of the optimal solution to (2) and propose

a branch-and-bound search algorithm, referred to as Branch and Bound Least-Squares (BBLS), to recover the true support of \mathbf{x} . Unlike the existing greedy-search methods, we utilize a scheduling procedure to limit the number of nodes in each level of the search tree. This strategy is motivated by the observation that indices selected in the levels close to the top of the tree are more likely to be in the true support of the signal. Therefore, BBLS examines more promising paths first, while its complexity can be strictly controlled by limiting the maximum number of visited paths. We state sufficient conditions under which in the noiseless setting BBLS successfully recovers support of \mathbf{x} with high probability, and empirically demonstrate its superior performance in several scenarios.

Before describing the algorithm, we briefly discuss the notation used in the paper. The uppercase letters denote matrices and bold lowercase letters represent vectors. We assume the coefficient matrix $\mathbf{A} \in \mathbb{R}^{n \times m}$ has full column rank, i.e., $m > n$; we denote the (i, j) entry of \mathbf{A} by \mathbf{A}_{ij} , and the j^{th} column of \mathbf{A} is denoted by \mathbf{a}_j . Let $\mathcal{I} = \{1, \dots, m\}$ be the set of indices of the entries of \mathbf{x} (or, equivalently, columns of \mathbf{A}), and \mathcal{S}_{true} , be the set of true indices, i.e., indices of the nonzero elements of \mathbf{x} . For the set $\mathcal{T} \subset \mathcal{I}$, $\mathbf{A}_{\mathcal{T}}$ is a submatrix of \mathbf{A} including columns indexed by \mathcal{T} . $\mathbf{P}_{\mathcal{T}}^{\perp} = \mathbf{I} - \mathbf{A}_{\mathcal{T}}\mathbf{A}_{\mathcal{T}}^{\dagger}$ is projection operator onto the orthogonal complement of subspace spanned by the columns of $\mathbf{A}_{\mathcal{T}}$ where $\mathbf{A}_{\mathcal{T}}^{\dagger} = (\mathbf{A}_{\mathcal{T}}^{\top}\mathbf{A}_{\mathcal{T}})^{-1}\mathbf{A}_{\mathcal{T}}^{\top}$ is the Moore-Penrose pseudo-inverse of $\mathbf{A}_{\mathcal{T}}$, and \mathbf{I} is an $n \times n$ identity matrix. For a matrix \mathbf{A} , $\mathbf{A} \sim \mathcal{B}(\frac{1}{2}, \pm \frac{1}{\sqrt{n}})$ means the entries of \mathbf{A} are drawn independently from a Bernoulli distribution and take values $\frac{1}{\sqrt{n}}$ and $-\frac{1}{\sqrt{n}}$ with equal probability. Similar definition holds for $\mathbf{A} \sim \mathcal{N}(0, \frac{1}{n})$.

It will be useful for the derivation of BBLS to briefly review the selection procedure of the OLS algorithm. Define

$$M_{OLS}(\mathcal{S}_{i-1}, j) = \left| \mathbf{r}_{\mathcal{S}_{i-1}}^{\top} \frac{\mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} \mathbf{a}_j}{\|\mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} \mathbf{a}_j\|_2} \right| \quad (3)$$

where \mathcal{S}_i is the subset of indices chosen in the first i iteration of OLS, and $\mathbf{r}_{\mathcal{S}_{i-1}} = \mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} \mathbf{y}$ is the residual vector in the i^{th} iteration. OLS chooses a new index j_s in the i^{th} iteration by finding $j_s = \arg \max_{j \in \mathcal{I} \setminus \mathcal{S}_{i-1}} M_{OLS}(\mathcal{S}_{i-1}, j)$ and adds it to \mathcal{S}_{i-1} to obtain \mathcal{S}_i . Projection matrix needed for the $(i+1)^{\text{st}}$ iteration is related to the current projection matrix according to $\mathbf{P}_{\mathcal{S}_i}^{\perp} = \mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} - \tilde{\mathbf{a}}_{j_s} \tilde{\mathbf{a}}_{j_s}^{\top}$ where $\tilde{\mathbf{a}}_{j_s} = \mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} \mathbf{a}_{j_s} / \|\mathbf{P}_{\mathcal{S}_{i-1}}^{\perp} \mathbf{a}_{j_s}\|_2$.

2. BRANCH AND BOUND LEAST-SQUARES

BBLS relies on a branch-and-bound search procedure to traverse in a depth-first manner a tree whose nodes represent columns of \mathbf{A} . We use a schedule $\mathbf{L} = [L_1, \dots, L_k]$ to control the size of the search space explored by the algorithm; in particular, for each node visited by BBLS at

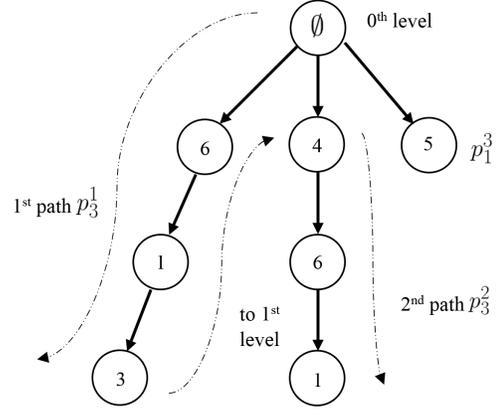


Fig. 1: An illustration of the BBLS tree search in the noiseless scenario with $k = 3$ and $\mathcal{S}_{true} = \{4, 6, 1\}$. The parameters of BBLS are $\mathbf{L} = [3, 1, 1]$ and $N_p = 4$. The first examined path, $p_1^3 = \{6, 1, 3\}$, is identical to the output of OLS but has non-zero residual norm. Hence BBLS proceeds and finds that for the second path, $p_2^3 = \{4, 6, 1\}$, $\|\mathbf{r}_{p_2^3}\|_2 = 0$ (i.e., p_2^3 is the optimal solution).

the i^{th} level of the tree, L_{i+1} denotes the number of its descendants at the $(i+1)^{\text{st}}$ level that are also visited. Let $p_i^\ell = \{s_1^\ell, \dots, s_i^\ell\}$ denote the set of indices chosen as the first i steps of the ℓ^{th} path constructed by BBLS. The algorithm then selects $\{j_{s_1}, \dots, j_{s_{L_{i+1}}}\}$ that result in L_{i+1} largest values of $M_{OLS}(p_i^\ell, j)$ in (3) for $j \in \mathcal{I} \setminus p_i^\ell$ (bounding step). Next, we update $p_{i+1}^\ell = p_i^\ell \cup \{j_{s_1}\}$ (branching step). BBLS performs these steps until it reaches the k^{th} level (bottom of the search tree) thus completing the current path, and computes the corresponding value of the objective $\|\mathbf{r}_{p_k^\ell}\|_2$, i.e., the Euclidean norm of the residual vector associated with the ℓ^{th} path. If $\|\mathbf{r}_{p_k^\ell}\|_2$ is smaller than a predetermined threshold ϵ , BBLS stops and returns p_k^ℓ as the estimated support.¹ Otherwise, BBLS keeps traversing up and down the tree until it either finds a path with sufficiently small objective value, or it visits N_p paths, or it reaches the top level and visits all $\prod_{i=1}^k L_i$ possible paths (if $N_p > \prod_{i=1}^k L_i$). A simple illustration of the BBLS search is shown in Fig. 1.

The scheduling procedure employed by BBLS is motivated by the simple observation that the indices selected at the top levels of the search tree are more likely to be from \mathcal{S}_{true} . Therefore, it is meaningful for $\{L_i\}_{i=1}^k$ to be a non-increasing sequence of positive integers. The simulation results in Section 3 verify that BBLS with scheduling is an efficient scheme for sparse recovery. Note that for an efficient implementation of (3) in BBLS, we employ the accelerated OLS recursions originally introduced in [22]. Specifically, the selection procedure of OLS can be rephrased as $j_s = \arg \max_{j \in \mathcal{I} \setminus \mathcal{S}_i} \|\mathbf{q}_j\|_2$, where $\mathbf{q}_j = (\mathbf{a}_j^{\top} \mathbf{r}_i / \mathbf{a}_j^{\top} \mathbf{t}) \mathbf{t}$ and $\mathbf{t} = \mathbf{a}_j - \sum_{l=1}^i \frac{\mathbf{a}_j^{\top} \mathbf{u}_l}{\|\mathbf{u}_l\|_2^2} \mathbf{u}_l$. Furthermore, the residual vector $\mathbf{r}_{\mathcal{S}_i}$

¹In the noise-free scenario, $\epsilon = 0$. In the noisy measurements case, ϵ^2 is a scaled variance of the measurement noise, i.e., $\epsilon^2 \propto k\sigma_v^2$.

required for the next iteration is formed as $\mathbf{r}_{S_i} = \mathbf{r}_{S_{i-1}} - \mathbf{u}_{i+1}$ where $\mathbf{u}_{i+1} = \mathbf{q}_{j_s}$. The BBLs algorithm is formalized as Algorithm 1.

Remark: Note that BBLs employs ϵ at the bottom of the tree. We can generalize the idea of discarding paths that violate a predetermined threshold and build a framework that uses a schedule $E = [\epsilon_1, \dots, \epsilon_k]$ to reduce the search space by pruning less promising paths at all levels of the tree. The schedule E can be determined based on the statistical properties of \mathbf{A} , $\boldsymbol{\nu}$, and the non-zero entries of \mathbf{x} . Further discussion on statistically scheduled pruning is left to future work.

2.1. Exact recovery conditions for BBLs

When \mathbf{A} is drawn at random from either $\mathcal{N}(0, 1/n)$ or $\mathcal{B}(\frac{1}{2}, \pm \frac{1}{\sqrt{n}})$, singular values of \mathbf{A} are with high probability concentrated around 1, which can be used to establish probabilistic performance guarantees for BBLs. In particular, in combination with a simple inductive argument, Theorem 2.1 below implies that for such matrices and in the noise-free setting, BBLs with $N_p = \prod_{i=1}^k L_i$ construct a path which corresponds to the true support of \mathbf{x} . The proof of Theorem 2.1 is omitted for brevity.

Theorem 2.1. *Let ϵ and δ be arbitrary constants such that $0 < \epsilon < 1$ and $0 < \delta < 1$. Assume that $\mathbf{x} \in \mathbb{R}^m$ is a k -sparse arbitrary vector, $\mathbf{A} \sim \mathcal{N}(0, 1/n)$ or $\mathbf{A} \sim \mathcal{B}(\frac{1}{2}, \pm \frac{1}{\sqrt{n}})$, and that noiseless measurements $\mathbf{y} = \mathbf{A}\mathbf{x}$ are given. Suppose that for the ℓ^{th} path, p_k^ℓ , BBLs has chosen indices from $\mathcal{S}_{\text{true}}$ in the first i levels (i.e., $p_i^\ell = \{s_1^\ell, \dots, s_i^\ell\} \subset \mathcal{S}_{\text{true}}$). Then, at least one among L_{i+1} children of s_i^ℓ is in $\mathcal{S}_{\text{true}}$ with probability exceeding*

$$\left(1 - 2e^{-(n-i)c_0(\epsilon)}\right)^2 \left(1 - 2\left(\frac{12}{\delta}\right)^k e^{-nc_0(\frac{\delta}{2})}\right) \left(1 - 2e^{-\frac{n}{k-i} \frac{1-\epsilon}{1+\epsilon} (1-\delta)^2}\right)^{m-k-L_{i+1}+1}, \quad (4)$$

where $c_0(\epsilon) = \frac{\epsilon^2}{6}(1-\epsilon)$.

2.2. Computational complexity

The worst case complexity of the BBLs algorithm is analyzed next. In step 2 of Algorithm 1, \mathbf{t} requires $\mathcal{O}(nk)$ operations, \mathbf{q}_j costs $\mathcal{O}(n)$, and they need to be computed for at most m columns. Therefore, the aggregate complexity of step 2 is $\mathcal{O}(mnk)$. Step 3 requires additional $\mathcal{O}(n)$ operations. Computing the ℓ_2 norm of the residual vector in step 6 requires $\mathcal{O}(n)$. Since each path includes k indices, the total cost to find a path is $\mathcal{O}(mnk^2)$. Since there are at most N_p of such paths, the worst case complexity of Algorithm 1 is $\mathcal{O}(N_p mnk^2)$.

3. SIMULATION RESULTS

To evaluate performance of the BBLs algorithm, we compare it to that of five other sparse recovery schemes for var-

Algorithm 1 Branch and Bound Least-Squares (BBLs)

Input: \mathbf{y} , \mathbf{A} , sparsity level k , threshold ϵ , schedule \mathbf{L} , max number of paths N_p

Output: recovered support $\hat{\mathcal{S}}$, estimated signal $\hat{\mathbf{x}}$

1. (Initialize) $\mathcal{S} = \emptyset$, $\mathbf{r}_{p_0^\ell} = \mathbf{y}$, $r_{\ell_2} = \|\mathbf{y}\|_2$, $i = 1$, $\ell = 1$.
 2. (Bounding) Let $\mathbf{S}_i = []$ and $l_i = 0$,
for $j \in \mathcal{I} \setminus \mathcal{S}$ **do**
 $\mathbf{t} = \mathbf{a}_j - \sum_{l=1}^{i-1} \frac{\mathbf{a}_j^\top \mathbf{u}_l}{\|\mathbf{u}_l\|_2} \mathbf{u}_l$, $\mathbf{q}_j = \frac{\mathbf{a}_j^\top \mathbf{r}_{i-1}}{\mathbf{a}_j^\top \mathbf{t}} \mathbf{t}$
end for
Select $\mathbf{S}_i = [j_{s_1}, \dots, j_{s_{L_i}}]$ corresponding to L_i largest terms $\|\mathbf{q}_j\|_2$
 3. (Branching) $l_i = l_i + 1$. If $l_i > L_i$ go to 4, else $\mathcal{S} = \mathcal{S} \cup \{\mathbf{S}_i(l_i)\}$, $\mathbf{u}_i = \mathbf{q}_{j_{s_{l_i}}}$, $\mathbf{r}_{p_i^\ell} = \mathbf{r}_{p_{i-1}^\ell} - \mathbf{u}_i$, go to 5.
 4. (Decrease i) If $i = 1$ go to 7, else $\mathcal{S} = \mathcal{S} \setminus \{\mathbf{S}_i(l_i)\}$, $i = i - 1$, and go to 2.
 5. (Increase i) If $i = k$ go to 6, else $i = i + 1$ and go to 2.
 6. (Solution found) Save the ℓ^{th} path $p_k^\ell = \mathcal{S}$ and its objective value $\|\mathbf{r}_{p_k^\ell}\|_2$. If $\|\mathbf{r}_{p_k^\ell}\|_2 < r_{\ell_2}$ update $r_{\ell_2} = \|\mathbf{r}_{p_k^\ell}\|_2$. $\ell = \ell + 1$, if $\ell > N_p$ or $r_{\ell_2} < \epsilon$ go to 7, else go to 3.
 7. Terminate the algorithm. Return the path $p_k^{\ell^*}$ with minimum residual norm as $\hat{\mathcal{S}}$, and the estimate $\hat{\mathbf{x}} = \mathbf{A}_{\hat{\mathcal{S}}}^\dagger \mathbf{y}$.
-

ious sparsity levels k ; we limit the study to noise-free settings. In particular, we considered OMP [9], Accelerated OLS (AOLS) [22], MMP with breadth-first (MMP-BF) and depth-first (MMP-DF) implementations², and BP with the so-called LASSO formulation for a fast implementation with regularization parameter $\lambda = 0.0001$. The stopping threshold for BBLs, MMP-DF, OMP, AOLs, and LASSO was set to 10^{-13} (MMP-BF, a breadth-first algorithm, does not use a stopping threshold). For BBLs, MMP-BF, and MMP-DF we set the maximum number of paths to $N_p = 50$. The schedule used for BBLs is $L = [6, 6, 3, 1, \dots, 1]$. We set $n = 64$ and $m = 128$; k changes from 3 to 39. The non-zero elements of \mathbf{x} – whose locations are chosen uniformly – are independent and identically distributed normal random variables. In order to construct \mathbf{A} , we consider the so-called hybrid scenario [16] to simulate both correlated and uncorrelated dictionaries. Specifically, we set $\mathbf{A}_j = \frac{\mathbf{b}_j + t_j \mathbf{1}}{\|\mathbf{b}_j + t_j \mathbf{1}\|_2}$ where $\mathbf{b}_j \sim \mathcal{N}(0, \frac{1}{n})$, $t_j \sim \mathcal{U}(0, T)$ with $T \geq 0$, and $\mathbf{1} \in \mathbb{R}^n$ is the all-one vector. In addition, $\{\mathbf{b}_j\}_{j=1}^m$ and $\{t_j\}_{j=1}^m$ are statistically independent. Notice that as T increases, the so-called mutual coherence parameter of \mathbf{A} increases, resulting in a more correlated coefficient matrix; $T = 0$ corresponds to an incoherent \mathbf{A} .

Performance of each algorithm is characterized by two metrics: (i) Exact Recovery Rate (ERR), defined as the percentage of instances where the support of \mathbf{x} is recovered exactly, and (ii) the running time of the algorithms in seconds.

²For MMP-BF and MMP-DF, we used the code provided by the authors of [26].

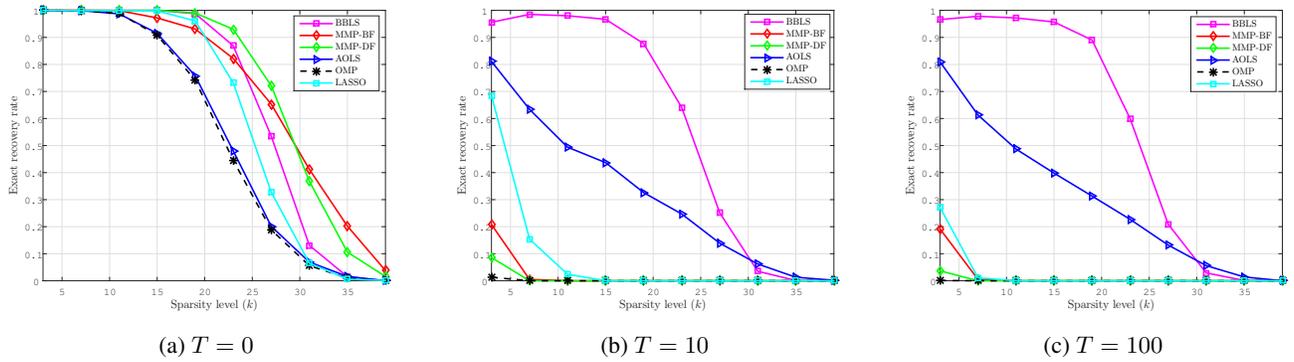


Fig. 2: Exact recovery rate comparison of BBLs, MMP-BF, MMP-DF, AOLS, OMP, and LASSO for $n = 64$, $m = 128$, \mathbf{A} with hybrid columns, and the k non-zero components of \mathbf{x} drawn from $\mathcal{N}(0, 1)$.

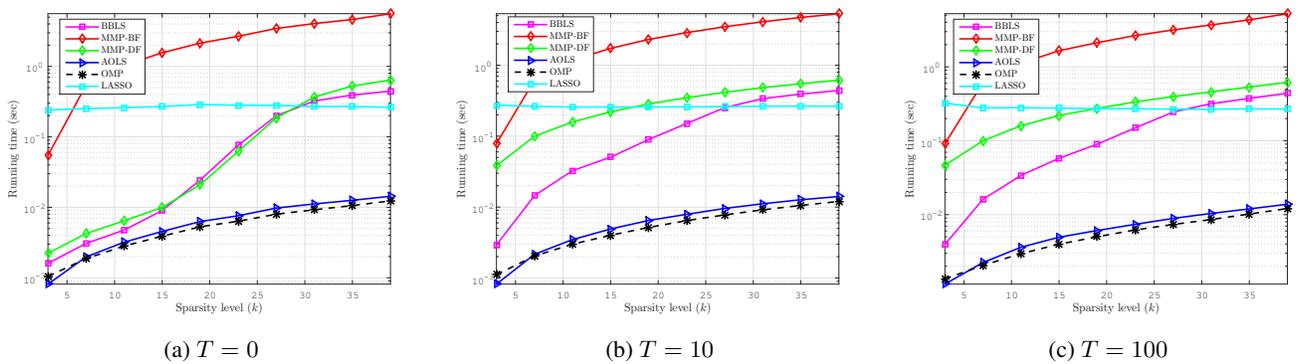


Fig. 3: Running time (seconds) comparison of BBLs, MMP-BF, MMP-DF, AOLS, OMP, and LASSO for $n = 64$, $m = 128$, \mathbf{A} with hybrid columns, and the k non-zero components of \mathbf{x} drawn from $\mathcal{N}(0, 1)$.

Each experiment is repeated 1000 times. Fig. 2 illustrates the ERR comparison. For the case of $T = 0$, it can be observed from Fig. 2 (a) that all algorithms perform similarly, with MMP-DF taking the lead for smaller values of k and MMP-BF for larger values of k . Next, consider the more practical case of $T = 10$ shown in Fig. 2 (b) where the columns of \mathbf{A} are correlated. Performance of MMP-DF, MMP-BP, OMP, and LASSO deteriorates severely, and these algorithms fail to recover support of \mathbf{x} when $k \geq 6$. However, BBLs maintains its competitive performance with just a small decay in the exact recovery rate. Fig. 2 (c) demonstrates that when $T = 100$, BBLs maintains its performance while the exact recovery rate of remaining algorithms other than AOLS deteriorates even further (This setting can be used to model arbitrary close data points in the sparse subspace clustering problem [2]). Running time of the algorithms is compared in Fig. 3. We observe that BBLs is fairly computationally efficient, especially when compared with MMP-BF. Moreover, it is evident that the speeds of AOLS, OMP, LASSO, and MMP-BF do not change significantly as T increases. Running time of BBLs increases marginally for smaller values of k for $T > 0$. Finally, running time of MMP-DF noticeably increases when $T > 0$, especially for small k . The presented results sug-

gest that BBLs offers a desirable trade-off between speed and accuracy. Moreover, unlike the other schemes, BBLs maintains highly accurate performance in the situations where the columns (atoms) of the coefficient matrix are correlated.

4. CONCLUSION

We presented the Branch and Bound Least-Squares (BBLs) algorithm, a new scheme for sparse recovery that constructs and traverses a search tree by selecting multiple signal support indices at each level using Orthogonal Least-Squares. Since the indices at the top tree levels are more likely to be in the true support, we employed a schedule to visit multiple promising paths while maintaining low computational complexity. Moreover, we provided sufficient conditions for the exact sparse recovery with BBLs in noise-free settings. Simulation studies demonstrated efficacy of BBLs as compared to popular sparse reconstruction algorithms. In particular, while the performance of almost all other algorithms suffer when the dictionaries are correlated, BBLs remains capable of highly accurate recovery. As part of the future work, it would be of interest to analytically study the observed robust performance of BBLs for correlated coefficient matrices.

5. REFERENCES

- [1] Cecilia Carbonelli, Satish Vedantam, and Urbashi Mitra, "Sparse channel estimation with zero tap detection," *IEEE Transactions on Wireless Communications*, vol. 6, no. 5, pp. 1743–1763, 2007.
- [2] Ehsan Elhamifar and René Vidal, "Sparse subspace clustering," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*. IEEE, 2009, pp. 2790–2797.
- [3] Farzad Parvaresh, Haris Vikalo, Sidhant Misra, and Babak Hassibi, "Recovering sparse signals using sparse measurement matrices in compressed DNA microarrays," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 2, no. 3, pp. 275–285, 2008.
- [4] Michael Lustig, David Donoho, and John M Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magnetic resonance in medicine*, vol. 58, no. 6, pp. 1182–1195, 2007.
- [5] Michael Elad, Mario AT Figueiredo, and Yi Ma, "On the role of sparse and redundant representations in image processing," *Proceedings of the IEEE*, vol. 98, no. 6, pp. 972–982, 2010.
- [6] David L Donoho, "Compressed sensing," *Information Theory, IEEE Transactions on*, vol. 52, no. 4, pp. 1289–1306, 2006.
- [7] Emmanuel J Candes and Terence Tao, "Decoding by linear programming," *IEEE transactions on information theory*, vol. 51, no. 12, pp. 4203–4215, 2005.
- [8] Emmanuel J Candes, "The restricted isometry property and its implications for compressed sensing," *Comptes Rendus Mathématique*, vol. 346, no. 9, pp. 589–592, 2008.
- [9] Yagyensh Chandra Pati, Ramin Rezaifar, and PS Krishnaprasad, "Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition," in *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*. IEEE, 1993, pp. 40–44.
- [10] Sheng Chen, Stephen A Billings, and Wan Luo, "Orthogonal least squares methods and their application to non-linear system identification," *International Journal of control*, vol. 50, no. 5, pp. 1873–1896, 1989.
- [11] Qun Mo and Yi Shen, "A remark on the restricted isometry property in orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 58, no. 6, pp. 3654–3656, 2012.
- [12] Joel A Tropp, "Greed is good: Algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, 2004.
- [13] T Tony Cai and Lie Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *Information Theory, IEEE Transactions on*, vol. 57, no. 7, pp. 4680–4688, 2011.
- [14] Joel A Tropp and Anna C Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *Information Theory, IEEE Transactions on*, vol. 53, no. 12, pp. 4655–4666, 2007.
- [15] Alyson K Fletcher and Sundeep Rangan, "Orthogonal matching pursuit: A Brownian motion analysis," *IEEE Transactions on Signal Processing*, vol. 60, no. 3, pp. 1010–1021, 2012.
- [16] Charles Soussen, Rémi Gribonval, Jérôme Idier, and Cédric Herzet, "Joint k-step analysis of orthogonal matching pursuit and orthogonal least squares," *Information Theory, IEEE Transactions on*, vol. 59, no. 5, pp. 3158–3174, 2013.
- [17] Shay Maymon and Yonina C Eldar, "The Viterbi algorithm for subset selection," *IEEE Signal Processing Letters*, vol. 22, no. 5, pp. 524–528, 2015.
- [18] Abolfazl Hashemi and Haris Vikalo, "Sparse linear regression via generalized orthogonal least-squares," *arXiv preprint arXiv:1602.06916*, 2016.
- [19] Abhimanyu Das and David Kempe, "Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection," in *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, 2011, pp. 1057–1064.
- [20] Cédric Herzet, Charles Soussen, Jérôme Idier, and Rémi Gribonval, "Exact recovery conditions for sparse representations with partial support information," *Information Theory, IEEE Transactions on*, vol. 59, no. 11, pp. 7509–7524, 2013.
- [21] Cédric Herzet, Angélique Drémeau, and Charles Soussen, "Relaxed recovery conditions for OMP/OLS by exploiting both coherence and decay," *Information Theory, IEEE Transactions on*, vol. 62, no. 1, pp. 459–470, 2016.
- [22] Abolfazl Hashemi and Haris Vikalo, "Sampling requirements and accelerated schemes for sparse linear regression with orthogonal least-squares," *arXiv preprint arXiv:1608.02549*, 2016.
- [23] Jian Wang and Ping Li, "Recovery of sparse signals using multiple orthogonal least squares," *arXiv preprint arXiv:1410.2505*, 2014.
- [24] Abolfazl Hashemi and Haris Vikalo, "Performance guarantees for sparse reconstruction from noisy measurements via orthogonal least-squares," *arXiv preprint arXiv:1608.02554*, 2016.
- [25] Shane F Cotter and Bhaskar D Rao, "Application of tree-based searches to matching pursuit," in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*. IEEE, 2001, vol. 6, pp. 3933–3936.
- [26] Suhnyuk Kwon, Jian Wang, and Byonghyo Shim, "Multipath matching pursuit," *IEEE Transactions on Information Theory*, vol. 60, no. 5, pp. 2986–3001, 2014.
- [27] Marco Rossi, Alexander M Haimovich, and Yonina C Eldar, "Spatial compressive sensing in mimo radar with random arrays," in *Information Sciences and Systems (CISS), 2012 46th Annual Conference on*. IEEE, 2012, pp. 1–6.
- [28] Nazim Burak Karahanoglu and Hakan Erdogan, "A* orthogonal matching pursuit: Best-first search for compressed sensing signal recovery," *Digital Signal Processing*, vol. 22, no. 4, pp. 555–568, 2012.