When in Doubt, SWAP: High-Dimensional Sparse Recovery from Correlated Measurements

Authored by:

Richard Baraniuk
Divyanshu Vats

Abstract

We consider the problem of accurately estimating a high-dimensional sparse vector using a small number of linear measurements that are contaminated by noise. It is well known that standard computationally tractable sparse recovery algorithms, such as the Lasso, OMP, and their various extensions, perform poorly when the measurement matrix contains highly correlated columns. We develop a simple greedy algorithm, called SWAP, that iteratively swaps variables until a desired loss function cannot be decreased any further. SWAP is surprisingly effective in handling measurement matrices with high correlations. We prove that SWAP can be easily used as a wrapper around standard sparse recovery algorithms for improved performance. We theoretically quantify the statistical guarantees of SWAP and complement our analysis with numerical results on synthetic and real data.

1 Paper Body

An important problem that arises in many applications is that of recovering a high-dimensional sparse (or approximately sparse) vector given a small number of linear measurements. Depending on the problem of interest, the unknown sparse vector can encode relationships between genes [1], power line failures in massive power grid networks [2], sparse representations of signals [3, 4], or edges in a graphical model [5,6], to name just a few applications. The simplest, but still very useful, setting is when the observations can be approximated as a sparse linear combination of the columns in a measurement matrix X weighted by the non-zero entries of the unknown sparse vector. In this paper, we study the problem of recovering the location of the non-zero entries, say S ? , in the unknown vector, which is equivalent to recovering the columns of X that y depends on. In the literature, this problem is often to referred to as the sparse recovery or the support recovery problem. Although several tractable sparse recovery algorithms have been proposed in the literature, statistical guarantees for accurately estimating S ? can only be provided under conditions that limit

how correlated the columns of X can be. For example, if there exists a column, say Xi , that is nearly linearly dependent on the columns indexed by S ? , some sparse recovery algorithms may falsely select Xi . In certain applications, where X can be specified a priori, correlations can easily be avoided by appropriately choosing X. However, in many applications, X cannot be specified by a practitioner, and correlated measurement matrices are inevitable. For example, when the columns in X correspond to gene expression values, it has been observed that genes in the same pathway produce correlated values [1]. Additionally, it has been observed that regions in the brain that are in close proximity produce correlated signals as measured using an MRI [7]. In this paper, we develop new sparse recovery algorithms that can accurately recover S ? for measurement matrices that exhibit strong correlations. We propose a greedy algorithm, called SWAP, that iteratively swaps variables starting from an initial estimate of S ? until a desired loss function cannot be decreased any further. We prove that SWAP can accurately identify the true signal support 1

under relatively mild conditions on the restricted eigenvalues of the matrix X T X and under certain conditions on the correlations between the columns of X. A novel aspect of our theory is that the conditions we derive are only needed when conventional sparse recovery algorithms fail to recover S ? . This motivates the use of SWAP as a wrapper around sparse recovery algorithms for improved performance. Finally, using numerical simulations, we show that SWAP consistently outperforms many state of the art algorithms on both synthetic and real data corresponding to gene expression values. As alluded to earlier, several algorithms now exist in the literature for accurately estimating S ? . The theoretical properties of such algorithms either depend on the irrepresentability condition [5, 8?10] or various forms of the restricted eigenvalue conditions [11,12]. See [13] for a comprehensive review of such algorithms and the related conditions. SWAP is a greedy algorithm with novel guarantees for sparse recovery and we make appropriate comparisons in the text. Another line of research when dealing with correlated measurements is to estimate a superset of S ? ; see [14?18] for examples. The rest of the paper is organized as follows. Section 2 formally defines the sparse recovery problem. Section 3 introduces SWAP. Section 4 presents theoretical results on the conditions needed for provably correct sparse recovery. Section 5 discusses numerical simulations. Section 6 summarizes the paper and discusses future work.

2 Problem Setup Throughout this paper, we assume that y ? Rn and X ? Rn?p are known and related to each other by the linear model y = X? ? + w , (1) ? p where ? ? R is the unknown sparse vector that we seek to estimate. We assume that the columns of X are normalized, i.e., Xi 22 /n = 1 for all i ? [p], where we use the notation [p] = {1, 2, . . . , p} throughout the paper. In practice, normalization can easily be done by scaling X and ? ? accordingly. We assume that the entries of w are i.i.d. zero-mean sub-Gaussian random variables with parameter ? so that E[exp(twi)] ? exp(t2 ? 2 /2). The sub-Gaussian condition on w is common in the literature and allows for a wide class of noise models, including Gaussian, symmetric Bernoulli, and bounded random variables. We let k be the number of non-zero entries in ? ? , and let

S ? denote the location of the non-zero entries. It is common to refer to S ? as the support of ? ? and we adopt this notation throughout the paper. Once S ? has been estimated, it is relatively straightforward to estimate ? ? . Thus, we mainly focus on the sparse recovery problem of estimating S ? . A classical strategy for sparse recovery is to search for a support of size k that minimizes a suitable loss function. For a support S, we assume the least-squares loss, which is defined as follows:

2 L(S; y, X) := min y ? XS ?22 = ?? [S]y , (2) 2

??R—S—

where XS refers to an n ? —S— matrix that only includes the columns indexed by S and ?? [S] = I ? XS (XST XS)?1 XST is the orthogonal projection onto the null space of the linear operator XS . In this paper, we design a sparse recovery algorithm that provably, and efficiently, finds the true support for a broad class of measurement matrices that includes matrices with high correlations.

3 Overview of SWAP We now describe our proposed greedy algorithm SWAP. Recall that our main goal is to find a support S that minimizes the loss defined in (2). Suppose that we are given an estimate, say S (1) , of the true support and let L(1) be the corresponding least-squares loss (see (2)). We want to transition to another estimate S (2) that is closer (in terms of the number of true variables), or equal, to S ? . Our main idea to transition from S (1) to an appropriate S (2) is to swap variables as follows: (1)

Swap every i ? S (1) with i ? (S (1))c and compute the resulting loss Li,i = L({S (1) i} ?i ; y, X). (1)

If mini,i Li,i ¡ L(1) , there exists a support that has a lower loss than the original one. Sub(1) sequently, we find {i, i } = arg mini,i Li,i and let S (2) = {S (1) i} ? {i }. We repeat the 2

200 100 0 0

0.05

0.1

0.15

0.5

0 3

0.2

(a)

10 Mean # of Iterations

TLasso S?TLasso FoBa S?FoBa CoSaMP S?CoSaMP MaR S?MaR

300

True Positive Rate

1 400

(b)

4

5 6 7 Sparsity Level

8

(c)

5

0 3
4
5 6 7 Sparsity Level
8
(d)

Figure 1: Example of using SWAP on pseudo real data where the design matrix X corresponds to gene expression values and y is simulated. The notation S-Alg refers to the SWAP based algorithms. (a) Histogram of sparse eigenvalues of X over 10, 000 random sets of size 10; (b) legend; (c) mean true positive rate vs. sparsity; (d) mean number of iterations vs. sparsity. Algorithm 1: SWAP(y, X, S) Inputs: Measurements y, design matrix X, and initial support S. Let r = 1, S (1) = S, and L(1) = L(S (1) ; y, X) (r) (r) 2 Swap i ? S with i ? (S (r))c and compute the loss Li,i = L({S (r)i} ? i ; y, X). 1
3 4 5 6
7
(r)

if mini,i Li,i ¡ L(r) then (r) {i, i } = argmini,i Li,i (In case of a tie, choose a pair arbitrarily) Let S (r+1) = {S (r) i} ? i and L(r+1) be the corresponding loss. Let r = r + 1 and repeat steps 2-4. else Return S = S (r) .

above steps to find a sequence of supports S (1) , S (2) , . . . , S (r) , where S (r) has the property that (r) mini,i Li,i ? L(r) . In other words, we stop SWAP when perturbing S (r) by one variable increases or does not change the resulting loss. These steps are summarized in Algorithm 1. Figure 1 illustrates the performance of SWAP for a matrix X that corresponds to 83 samples of 2308 gene expression values for patients with small round blue cell tumors [19]. Since there is no ground truth available, we simulate the observations y using Gaussian w with ? = 0.5 and randomly chosen sparse vectors with non-zero entries between 1 and 2. Figure 1(a) shows the histogram of the T eigenvalues of 10,000 randomly chosen matrices XA XA /n, where —A— = 10. We clearly see that these eigenvalues are very small. This means that the columns of X are highly correlated with each other. Figure 1(c) shows the mean fraction of variables estimated to be in the true support over 100 different trials. Figure 1(d) shows the mean number of iterations required for SWAP to converge. Remark 3.1. The main input to SWAP is the initial support S. This parameter implicitly specifies the desired sparsity level. Although SWAP can be used with a random initialization S, we recommend using SWAP in combination with another sparse recovery algorithm. For example, in Figure 1(c), we run SWAP using four different types of initializations. The dashed lines represent standard sparse recovery algorithms, while the solid lines with markers represent SWAP algorithms. We clearly see that all SWAP based algorithms outperform standard algorithms. Intuitively, since many sparse recovery algorithms can perform partial support recovery, using such an initialization results in a smaller search space when searching for the true support. Remark 3.2. Since each iteration of SWAP necessarily produces a unique loss, the supports S (1) , . . . , S (r) are all unique. Thus, SWAP clearly converges in a finite number of iterations. The exact convergence rate depends on the correlations in the matrix X. Although we

do not theoretically quantify the convergence rate, in all numerical simulations, and over a broad range of design matrices, we observed that SWAP converged in roughly O(k) iterations. See Figure 1(d) for an example. Remark 3.3. Using the properties of orthogonal projections, we can write Line 2 of SWAP as a difference of two rank one projection matrices. The main computational complexity is in computing 3

this quantity k(p ? k) times for all i ? S (r) and i ? (S (r)c . If the computational complexity of computing a rank k orthogonal projection is Ik , then Line 2 can be implemented in time O(k(Ik + p ? k). When k p is small, then Ik = O(k 3). When k is large, then several computational tricks can be used to significantly reduce the computational time. Remark 3.4. SWAP differs significantly from other greedy algorithms in the literature. When k is known, the main distinctive feature of SWAP is that it always maintains a k-sparse estimate of the support. Note that the same is true for the computationally intractable exhaustive search algorithm [10]. Other competitive algorithms, such as forward-backwards (FoBa) [20] or CoSaMP [21], usually estimate a signal with higher sparsity level and iteratively remove variables until k variables are selected. The same is true for multi-stage algorithms [22?25]. Intuitively, as we shall see in Section 4, by maintaining a support of size k, the performance of SWAP only depends on correlations among the columns of the matrix XA , where A is of size at most 2k and it includes the true support. In contrast, for other sparse recovery algorithms, —A— ? 2k. In Figure 1, we compare SWAP to several state of the art algorithms (see Section 5 for a description of the algorithms). In all cases, SWAP results in superior performance.

4 Theoretical Analysis of SWAP 4.1 Some Important Parameters In this Section, we collect some important parameters that determine the performance of SWAP. First, we define the restricted eigenvalue as

X?22 ? : ?0 ? k + , —S ? supp(?)— = k . (3) ?k+ := inf n?22 The parameter ?k+ is the minimum eigenvalue of certain blocks of the matrix X T X/n of size 2k that includes the blocks XST? XS ? /n. Smaller values of ?k+ correspond to correlated columns in the matrix X. Next, we define the minimum absolute value of the non-zero entries in ? ? as (4) ?min := min? —?i? — . i?S

A smaller ?min will evidently require more number of observations for exact recovery of the support. Finally, we define a parameter that characterizes the correlations between the columns of the matrix XS ? and the columns of the matrix X(S ?)c , where recall that S ? is the true support of the unknown sparse vector ? ? . For a set ?k,d that contains all supports of size k with atleast k ? d active variables from S ? , define ?d as

Si Si ?1 2 ? ? ? ? ?

i,S

S,S 2 1 ? ?d := max ? min , S = S ? S , (5) Si ? c S??k,d S i?(S) ?S ?i,i where ?B = X T ?? [B]X/n. Popular sparse regression algorithms, such as the Lasso and the OMP, 2 can perform accurate support recovery when ? 2 = maxi?(S ?)c ?i,S ? ??1 S ? ,S ? 1 ¡ 1. We will show in Section 3.2 that SWAP can perform accurate support recovery when ?d ¡ 1. Although the form of ?d is similar to ?, there are several key differences, which we highlight as follows: ?

Since ?k,d contains all supports such that —S ? S— ? d, it is clear that ?d is the 1 norm of a d ? 1 vector, where d ? k. In contrast, ? is the 1 norm of a k ? 1 vector. If indeed ? ¡ 1, i.e., accurate support recovery is possible using the Lasso, then SWAP can be initialized by the output of the Lasso. In this case, ?(?) = 0 and SWAP also outputs the true support as long as S ? minimizes the loss function. We make this statement precise in Theorem 4.1. Thus, it is only when ? ? 1 that the parameter ?d plays a role in the performance of SWAP. ? The parameter ? directly computes correlations between the columns of X. In contrast, ?d computes correlations between the columns of X when projected onto the null space of a matrix XB , where —B— = d ? 1. ? Notice that ?d is computed by taking a maximum over supports in the set ?d S ? and a minimum over inactive variables in each support. The reason that the minimum appears in ?d is because we choose to swap variables that result in the smallest loss. In contrast, ? is computed by taking a maximum over all inactive variables. 4

4.2 Statement of Main Results In this Section, we state the main results that characterize the performance of SWAP. Throughout this Section, we assume the following: (A1) The observations y and the measurement matrix X follow the linear model in (1), where the noise is sub-Gaussian with parameter ?, and the columns of X have been normalized. (A2) SWAP is initialized with a support S (1) of size k and S is the output of SWAP. Since k is typically unknown, a suitable value can be selected using standard model selection algorithms such as cross-validation or stability selection [26]. Our first result for SWAP is as follows. Theorem 4.1. Suppose (A1)-(A2) holds and —S ? S (1) — ? 1. If n ¿ 1/(18? 2), then P(S = S ?) ? 1 as (n, p, k) ? ?.

4+log(k2 (p?k)) , 2 c2 ?min ?2k /2

where 0 ¡ c2 ?

The proof of Theorem 4.1 can be found in the extended version of our paper [27]. Informally, Theorem 4.1 states that if the input to SWAP falsely detects at most one variable, then SWAP is high-dimensional consistent when given a sufficient number of observations n. The condition on n is mainly enforced to guarantee that the true support S ? minimizes the loss function. This condition is weaker than the sufficient conditions required for other computationally tractable sparse recovery algorithms. For example, the method FoBa is known to be superior to other methods 2 such as the Lasso and the OMP. As shown in [20], FoBa requires that n = ?(log(p)/(?3k+ ?min)) for high-dimensional consistent support recovery, where the choice of , which is greater than k, depends on the correlations in the matrix X. In contrast, the condition in (4.1), which reduces 2 to n = ?(log(p ? k)/(?2k ?min)), is weaker since 1/?3k+ ¡ 1/?2k for ¿ k and p ? k ¡ p. This shows that if a sparse recovery algorithm can accurately estimate the true support, then SWAP does not introduce any false positives and also outputs the true support. Furthermore, if a sparse regression algorithm falsely detects one variable, then SWAP can potentially recover the correct support. Thus, using SWAP with other algorithms does not harm the sparse recovery performance of other algorithms. We now consider the more interesting case when SWAP is initialized by a support S (1) that falsely detects more than one variable. In this case, SWAP will clearly needs more than

one iteration to recover the true support. Furthermore, to ensure that the true support can be recovered, we need to impose some additional assumptions on the measurement matrix X. The particular assumption we enforce will depend on the parameter ?k defined in (5). As mentioned in Section 4.1, ?k captures the correlations between the columns of XS ? and the columns of X(S ?)c . To simplify the statement ? ? in the next Theorem, define let g(?, ?, c) = g(?, ?, c) = (? ? 1) + 2c(? + 1/ ?) + 2c2 . Theorem 4.2. Suppose (A1)-(A2) holds and —S ? S (1) — ¿ 1. If for a constant c such that 0 ¡

2 log (p) c2 ¡ 1/(18? 2), g(?k , ?k,1 , c?) ¡ 0, log kp ¿ 4 + log(k 2 (p ? k)), and n ¿ c2 ? 2 ?k2 , then min 2k P(S = S ?) ? 1 as (n, p, k) ? ?. Theorem 4.2 says that if SWAP is initialized with any support of size k, and ?k satisfies the condition stated in the theorem, then SWAP will output the true support when given a sufficient number of observations. In the noiseless case, i.e., when ? = 0, the condition required for accurate support recovery reduces to ?k ¡ 1. The proof of Theorem 4.2, outlined in [27], relies on imposing conditions on each support of size k such that that there exists a swap so that the loss can be necessarily decreased. Clearly, if such a property holds for each support, except S ? , then SWAP will output the true support since (i) there are only a finite number of possible supports, and (ii) each iteration of SWAP results in a different support. The dependence on kp in the expression for the number of observations n arises from applying the union bound over all supports of size k. The condition in Theorem 4.2 is independent of the initialization S (1) . This is why the sample complexity, i.e., the number of observations n required for consistent support recovery, scales as

log kp . To reduce the sample complexity, we can impose additional conditions on the support S (1) that is used to initialize SWAP. Under such assumptions, assuming that —S ? S (1) — ¿ d, the 5

performance of SWAP will depend on ?d , which is less than ?k , and n will scale as log refer to [27] for more details.

p d . We

5 Numerical Simulations In this section, we show how SWAP compares to other sparse recovery algorithms. Section 5.1 presents results for synthetic data and Section 5.2 presents results for real data. 5.1 Synthetic Data To illustrate the advantages of SWAP, we use the following examples: (A1) We sample the rows of X from a Gaussian distribution with mean zero and covariance ?. The ? are specicovariance ? is block-diagonal with blocks of size 10. The entries in each block ? ? ? fied as follows: ?ii = 1 for i ? [10] and ?ij = a for i = j. This construction of the design matrix is motivated from [18]. The true support is chosen so that each variable in the support is assigned to a different block. The non-zero entries in ? ? are chosen uniformly between 1 and 2. We let ? = 1, p = 500, n = 100, 200, k = 20, and a = 0.5, 0.55, . . . , 0.9, 0.95. (A2) We sample X from the same distribution as described in (A1). The only difference is that the true support is chosen so that five different blocks contain active variables and each chosen block contains four active variables. The rest of the parameters are also the same. In both (A1) and (A2), as a increases, the strength of correlations between the columns increases. Further, the restricted eigenvalue

parameter for (A1) is greater than the restricted eigenvalue parameter of (A2). We use the following sparse recovery algorithms to initialize SWAP: (i) Lasso, (ii) Thresholded Lasso (TLasso) [25], (iii) Forward-Backward (FoBa) [20], (iv) CoSaMP [21], (v) Marginal Regression (MaR), and (vi) Random. TLasso first applies Lasso to select a superset of the support and then selects the largest k as the estimated support. In our implementation, we used Lasso to select 2k variables and then selected the largest k variables after least-squares. This algorithm is known to have better performance that the Lasso. FoBa uses a combination of a forward and a backwards algorithm. CoSaMP is an iterative greedy algorithm. MaR selects the support by choosing the largest k variables in —X T y—. Finally, Random selects a random subset of size k. We use the notation STLasso to refer to the algorithm that uses TLasso as an initialization for SWAP. A similar notation follows for other algorithms. Our results are shown in Figure 2. We use two metrics to assess the performance of SWAP. The first metric is the true positive rate (TPR), i.e., the number of active variables in the estimate divided by the total number of active variables. The second metric is the the number of iterations needed for SWAP to converge. Since all the results are over supports of size k, the false postive rate (FPR) is simply 1 ? TPR. All results for SWAP based algorithms have markers, while all results for non SWAP based algorithms are represented in dashed lines. From the TPR performance, we clearly see the advantages of using SWAP in practice. For different choices the algorithm Alg, when n = 100, the performance of S-Alg is always better than the performance of Alg. When the number of observations increase to n = 200, we observe that all SWAP based algorithms perform better than standard sparse recovery algorithms. For (A1), we have exact support recovery for SWAP when a ? 0.9. For (A2), we have exact support recovery when a ¡ 0.8. The reason for this difference is because of the differences in the placement of the non-zero entries. Figures 2(a) and 2(b) shows the mean number of iterations required by SWAP based algorithms as the correlations in the matrix X increase. We clearly see that the number of iterations increase with the degree of correlations. For algorithms that estimate a large fraction of the true support (TLasso, FoBa, and CoSaMP), the number of iterations is generally very small. For MaR and Random, the number of iterations is larger, but still comparable to the sparsity level of k = 20. 6

0.6 0.4 0.2 0.6 0.7 0.8 Degree of Correlation
0.8
0.8
0.4 0.9
0.6
0.5
(d) Example (A2), n = 100
0.9
25
0.4 0.6 0.7 0.8 Degree of Correlation
0.6 0.7 0.8 Degree of Correlation
(c) Example (A1), n = 100

Mean # of Iterations
1
Mean TPR
1
0.5
0.4
0 0.5
0.9
(b) Example (A1), n = 100
0.6
0.6
0.2
0 0.5
(a) Legend
Mean TPR
1 0.8 Mean TPR
Mean TPR
Lasso S?Lasso TLasso S?TLasso FoBa S?FoBa CoSaMP S?CoSaMP MaR S?MaR S?Random
1 0.8
0.6 0.7 0.8 Degree of Correlation
20 15 10 5 0 0.5
0.9
(e) Example (A2), n = 100
0.6 0.7 0.8 Degree of Correlation
0.9
(f) Example (A2), n = 100
30 0.8 0.6 0.4 0.2 0 0.5
0.6 0.7 0.8 Degree of Correlation
0.9
(g) Example (A1), n = 200
1 20
Mean TPR
Mean # of Iterations
Mean TPR
1
10
0.8 0.6 0.4 0.2
0 0.5
0.6 0.7 0.8 Degree of Correlation
0.9
(h) Example (A1), n = 200
0 0.5
0.6 0.7 0.8 Degree of Correlation
0.9
(i) Example (A2), n = 200

Figure 2: Empirical true positive rate (TPR) and number of iterations required by SWAP.

5.2 Gene Expression Data We now present results on two gene expression cancer datasets. The first dataset1 contains expression values from patients with two different types cancers related to leukemia. The second dataset2 contains expression levels from patients with and without prostate cancer. The matrix X contains the gene expression values and the vector y is an indictor of the type of cancer a patient has. Although this is a classification problem, we treat it as a recovery problem. For the leukemia data, p = 5147 and n = 72. For the prostate cancer data, p = 12533 and n = 102. This is clearly a high-dimensional dataset, and the goal is to identify a small set of genes that are predictive of the cancer type. Figure 3 shows the performance of standard algorithms vs. SWAP. We use leave-one-out crossvalidation and apply the sparse recovery algorithms described in Section 5.1 using multiple different choices of the sparsity level. For each level of sparsity, we choose the sparse recovery algorithm (labeled as standard) and the SWAP based algorithm that results in the minimum least-squares loss over the training data. This allows us to compare the performance of using SWAP vs. not using SWAP. For both datasets, we clearly see that the training and testing error is lower for SWAP based algorithms. This means that SWAP is able to choose a subset of genes that has better predictive performance than that of standard algorithms for each level of sparsity.

1 2

see http://www.biolab.si/supp/bi-cancer/projections/info/leukemia.htm see http://www.biolab.si/supp/bi-cancer/projections/info/prostata.htm

7

1

0.32 0.3 0.28

3

4

6 Sparsity Level

8

10

(a) Training Error

0.24

0.4 0.35 0.3

2.5 0.25

0.26 2

SWAP Standard

0.45

3.5

0.34

CV?Train Error

CV?Test Error

CV?Train Error

2

0.5

SWAP Standard
SWAP Standard
0.36
CV?Test Error
SWAP Standard
1.5
0.5
4
0.38
3 2.5
2
4
6 Sparsity Level
8
2
10
(b) Testing Error
2
3
4 Sparsity Level
5
(c) Training Error
6
0.2
2
3
4 Sparsity Level
5
6
(d) Testing Error

Figure 3: (a)-(b) Leukemia dataset with p = 5147 and n = 72. (c)-(d) Prostate cancer dataset with p = 12533 and n = 102.

6 Summary and Future Work We studied the sparse recovery problem of estimating the support of a high-dimensional sparse vector when given a measurement matrix that contains correlated columns. We presented a simple algorithm, called SWAP, that iteratively swaps variables starting from an initial estimate of the support until an appropriate loss function can no longer be decreased further. We showed that SWAP is surprising effective in situations where the measurement matrix contains correlated columns. We theoretically quantified the conditions on the measurement matrix that guarantee accurate support recovery. Our theoretical results show that if SWAP is initialized with a support that contains some active variables, then SWAP can tolerate even higher correlations in the measurement matrix. Using numerical simulations on synthetic and real data, we showed how SWAP outperformed several sparse recovery algorithms. Our work in this paper sets up a platform to study the following interesting extensions of SWAP. The first is a generalization of SWAP so

that a group of variables can be swapped in a sequential manner. The second is a detailed analysis of SWAP when used with other sparse recovery algorithms. The third is an extension of SWAP to high-dimensional vectors that admit structured sparse representations.

Acknowledgement The authors would like to thank Aswin Sankaranarayanan and Christoph Studer for feedback and discussions. The work of D. Vats was partly supported by an Institute for Mathematics and Applications (IMA) Postdoctoral Fellowship.

2 References

[1] M. Segal, K. Dahlquist, and B. Conklin, ?Regression approaches for microarray data analysis,? Journal of Computational Biology, vol. 10, no. 6, pp. 961?980, 2003. [2] H. Zhu and G. Giannakis, ?Sparse overcomplete representations for efficient identification of power line outages,? IEEE Transactions on Power Systems, vol. 27, no. 4, pp. 2215 ?2224, nov. 2012. [3] E. J. Cand‘es, J. Romberg, and T. Tao, ?Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,? IEEE Trans. Information Theory, vol. 52, no. 2, pp. 489?509, 2006. [4] M. F. Duarte, M. A. Davenport, D. Takhar, J. N. Laska, T. Sun, K. F. Kelly, and R. G. Baraniuk, ?Single-pixel imaging via compressive sampling,? IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 83?91, Mar. 2008. [5] N. Meinshausen and P. B?uhlmann, ?High-dimensional graphs and variable selection with the Lasso,? Annals of Statistics, vol. 34, no. 3, pp. 1436, 2006. [6] P. Ravikumar, M. Wainwright, and J. Lafferty, ?High-dimensional Ising model selection using 1 -egularized logistic regression,? Annals of Statistics, vol. 38, no. 3, pp. 1287?1319, 2010. 8

[7] G. Varoquaux, A. Gramfort, and B. Thirion, ?Small-sample brain mapping: sparse recovery on spatially correlated designs with randomization and clustering,? in Proceedings of the 29th International Conference on Machine Learning (ICML-12), 2012, pp. 1375?1382. [8] P. Zhao and B. Yu, ?On model selection consistency of Lasso,? Journal of Machine Learning Research, vol. 7, pp. 2541?2563, 2006. [9] J. A. Tropp and A. C. Gilbert, ?Signal recovery from random measurements via orthogonal matching pursuit,? IEEE Transactions Information Theory, vol. 53, no. 12, pp. 4655?4666, 2007. [10] M. J. Wainwright, ?Sharp thresholds for noisy and high-dimensional recovery of sparsity using 1 -constrained quadratic programming (Lasso),? IEEE Transactions Information Theory, vol. 55, no. 5, May 2009. [11] N. Meinshausen and B. Yu, ?Lasso-type recovery of sparse representations for highdimensional data,? Annals of Statistics, vol. 37, no. 1, pp. 246?270, 2009. [12] P. J. Bickel, Y. Ritov, and A. B. Tsybakov, ?Simultaneous analysis of Lasso and Dantzig selector,? Annals of Statistics, vol. 37, no. 4, pp. 1705?1732, 2009. [13] P. B?uhlmann and S. Van De Geer, Statistics for High-Dimensional Data: Methods, Theory and Applications, Springer-Verlag New York Inc, 2011. [14] H. Zou and T. Hastie, ?Regularization and variable selection via the elastic net,? Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 67, no.

2, pp. 301?320, 2005. [15] Y. She, ?Sparse regression with exact clustering,? Electronic Journal Statistics, vol. 4, pp. 1055?1096, 2010. [16] E. Grave, G. R. Obozinski, and F. R. Bach, ?Trace Lasso: A trace norm regularization for correlated designs,? in Advances in Neural Information Processing Systems 24, J. Shawetaylor, R. Zemel, P. Bartlett, F. Pereira, and K. Weinberger, Eds., 2011, pp. 2187?2195. [17] J. Huang, S. Ma, H. Li, and C. Zhang, ?The sparse laplacian shrinkage estimator for highdimensional regression,? Annals of Statistics, vol. 39, no. 4, pp. 2021, 2011. [18] P. B?uhlmann, P. R?utimann, S. van de Geer, and C.-H. Zhang, ?Correlated variables in regression: clustering and sparse estimation,? Journal of Statistical Planning and Inference, vol. 143, pp. 1835?1858, Nov. 2013. [19] J. Khan, J. S. Wei, M. Ringner, L. H. Saal, M. Ladanyi, F. Westermann, F. Berthold, M. Schwab, C. R. Antonescu, C. Peterson, et al., ?Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks,? Nature medicine, vol. 7, no. 6, pp. 673?679, 2001. [20] T. Zhang, ?Adaptive forward-backward greedy algorithm for learning sparse representations,? IEEE Transactions Information Theory, vol. 57, no. 7, pp. 4689?4708, 2011. [21] D. Needell and J. A. Tropp, ?CoSaMP: Iterative signal recovery from incomplete and inaccurate samples,? Applied and Computational Harmonic Analysis, vol. 26, no. 3, pp. 301?321, 2009. [22] T. Zhang, ?Some sharp performance bounds for least squares regression with l1 regularization,? The Annals of Statistics, vol. 37, no. 5A, pp. 2109?2144, 2009. [23] L. Wasserman and K. Roeder, ?High dimensional variable selection,? Annals of statistics, vol. 37, no. 5A, pp. 2178, 2009. [24] T. Zhang, ?Analysis of multi-stage convex relaxation for sparse regularization,? Journal of Machine Learning Research, vol. 11, pp. 1081?1107, Mar. 2010. [25] S. van de Geer, P. B?uhlmann, and S. Zhou, ?The adaptive and the thresholded lasso for potentially misspecified models (and a lower bound for the lasso),? Electronic Journal of Statistics, vol. 5, pp. 688?749, 2011. [26] N. Meinshausen and P. B?uhlmann, ?Stability selection,? Journal of the Royal Statistical Society: Series B (Statistical Methodology), vol. 72, no. 4, pp. 417?473, 2010. [27] D. Vats and R. G. Baraniuk, ?Swapping variables for high-dimensional sparse regression with correlated measurements,? arXiv:1312.1706, 2013.

9