Sample Complexity for 1-bit Compressed Sensing and Sparse Classification

Ankit Gupta
Samsung Telecommunications America
Richardson TX 75082
agupta2@sta.samsung.com

Robert Nowak
Electrical and Computer Engineering
University of Wisconsin-Madison
nowak@ece.wisc.edu

Benjamin Recht
Computer Sciences Department
University of Wisconsin-Madison
brecht@cs.wisc.edu

***Abstract*—This paper considers the problem of identifying the support set of a high-dimensional sparse vector, from noise-corrupted 1-bit measurements. We present passive and adaptive algorithms for this problem, both requiring no more than $O(d\log(D))$ measurements to recover the unknown support. The adaptive algorithm has the additional benefit of robustness to the dynamic range of the unknown signal.**

I. Introduction

Identifying a sparse collection of discriminative features in high-dimensional classification problems has received considerable attention from the machine learning and statistics communities [1], [2], [3], [4]. Additionally, several authors have considered the problem of quantized *compressed sensing* [5], [6], [7] where the aim is to recover a sparse vector from highly quantized, noisy measurements. In both of these scenarios, standard algorithms for sparse signal recovery are not applicable because the only available information is the sign of a given measurement. Instead, algorithms based on logistic regression or support vector machines are used to identify the unknown support. The hope is that, as in the case of compressed sensing, the number of measurements required to successfully identify the unknown signal support scales proportional to the signal's sparsity.

In this paper, we show that this is intuition is indeed correct and achievable with relatively simple algorithms. Specifically, this paper addresses the problem of identifying the support set of a sparse high-dimensional vector from highly quantized 1-bit measurements. The number of measurements required will be referred to as the *sample complexity* of the problem. In particular, we are interested on the dependence of the achievable sample complexity on the dimension of the unknown vector, D, its sparsity, d, and the signal to noise ratio.

We first present a passive algorithm that recovers the support by thresholding a correlation function. The sample complexity of this algorithm scales as $O(d\log D)$, but has an unfortunate quadratic dependence on the dynamic range of the signal to be recovered. We then describe an adaptive algorithm that eliminates the dependency on dynamic range while maintaining the $O(d\log D)$ sample complexity. Note that for both algorithms the number of samples required to a sparse support set obeys the same asymptotic scaling as the L_1 minimization based approaches for the canonical compressed sensing problem [8], [9]. To the best of our knowledge, our results provide the first bounds on the sample complexity of support recovery (i.e., feature selection) from binary-valued data.

II. Main Results

The unknown sparse vector will be denoted by α. By sparse, we mean that only $d \ll D$ of the elements of α are non-zero. We assume that we are only allowed to make measurements of the following form:

$$c(\mathbf{x}) \;=\; \text{sgn}(\alpha_1 x_1 + \cdots + \alpha_D x_D + n)\;, \tag{1}$$

where $\mathbf{x} \in \mathbb{R}^D$ is a 'sensing' vector (which we may be able to select or design) and n is Gaussian noise with unit variance. In words, we observe the sign of the inner product of α and $\mathbf{x}$, possibly corrupted by Gaussian noise.

We let K denote the *dynamic range* of α its support set.

$$K = \frac{|\max_{i=1}^{D} \alpha_i|}{|\min_{i:\alpha_i\neq 0} \alpha_i|}. \tag{2}$$

We also denote the signal to noise ratio by SNR and define it to be the square of the minimum nonzero α (since noise variance is assumed to be one).

In Section III, we present a passive algorithm such that with

$$m \geq 2\pi^2(K^2 d + \text{SNR}^{-1})\log(D/\delta) \tag{3}$$

randomly sampled measurements, we can recover the true signal support with probability at least $1-\delta$. In Section IV we present the adaptive procedure such that

$$m \geq \frac{1536d}{\Big(1/2 - Q(\sqrt{\text{SNR}}/70)\Big)^2}\log\frac{D}{\delta} \tag{4}$$

sequentially selected measurements suffice to recover the support of α, again with probability at least $1-\delta$. Here $Q(x)$ denotes the cumulative distribution of a standard normal random variable. Note that the number of measurements in (4) is independent of the dynamic range K, but the measurement vectors at each step of the algorithm have an adaptively chosen support (rather than global support).

For large SNR, the denominator in (4) is approximately $1/2$. In this case the sample complexity of the adaptive algorithm is $O(d\log\frac{D}{\delta})$. However, since

$$Q(x) \approx 1/2 - x/\sqrt{2\pi}$$

when x is small, the sample complexities of both the passive and active algorithms scale as SNR^{-1} for very low SNR.

III. Passive algorithm

For the passive algorithm, first consider the case without noise. Assume that we are given m noiseless random observations of the form $\{(\mathbf{x}_1, c(\mathbf{x}_1)), \cdots, (\mathbf{x}_m, c(\mathbf{x}_m))\}$, where $\mathbf{x}_i$ are i.i.d. vectors, such that each component of $\mathbf{x}_i$ is an i.i.d. unit variance zero-mean Gaussian random variable.

The algorithm works as follows. For each coordinate $j \in \{1, \cdots, D\}$, we form the empirical quantities l_j:

$$l_j = \frac{1}{m} \sum_{i=1}^{m} c(\mathbf{x}_i) \text{sgn}(\mathbf{x}_{i,j}), \tag{5}$$

where $\mathbf{x}_{i,j}$ denotes the j^{th} coordinate of the i^{th} sample. If $|l_j| > (1/K\sqrt{d}\pi)$, we label the coordinate as being relevant and reject it otherwise (note that the threshold of rejection need not be known a priori, and can be obtained by visually inspecting l_j).

With this criterion, we have

Lemma 1. *The probability of accepting an incorrect coordinate or rejecting a correct coordinate is upper bounded by* $\exp(-m/2K^2 d\pi^2)$.

Proof: The random variables $c(\mathbf{x}_i)\text{sgn}(\mathbf{x}_{i,j})$ only take on the values ± 1. Moreover,

$$\mathbb{E}[c(\mathbf{x}_i)\text{sgn}(\mathbf{x}_{i,j})] = \frac{2}{\pi} \arcsin \rho_j \tag{6}$$

where

$$\rho_j = \frac{\alpha_j}{\sqrt{\sum_{i=1}^{D} \alpha_i^2}}. \tag{7}$$

This identity for the covariance of the signs of two Gaussian random variables is due to Grothendieck and follows from Lemma 2.2 in [10]. Now by (2), we have $|\rho_j| \geq \frac{1}{K\sqrt{d}}$ for all j. The Lemma now follows by applying Hoeffding's inequality. ∎

If we fix the error rate (false detection and false alarm) probability as equal to δ, we require $2\pi^2 K^2 d \log(D/\delta)$ samples to achieve this desired error rate.

In the noisy case, $c(\mathbf{x}_i) = \text{sgn}(\tilde{y}_i')$, where $\tilde{y}_i' \triangleq \alpha_1 \mathbf{x}_{i,1} + \cdots + \alpha_D \mathbf{x}_{i,D} + n$, where n is unit variance Gaussian. In this case, we can use

$$\rho_j = \frac{\alpha_j}{\sqrt{\sum_{i=1}^{D} \alpha_i^2 + 1}}. \tag{8}$$

in Lemma 1, and the analysis is otherwise identical, albeit with the modification that the minimum absolute value of the covariance between a relevant coordinate and $\tilde{y}_i'$ is now at least $1/(\sqrt{K^2 d + \text{SNR}^{-1}})$. Repeating the argument after Lemma 1, we require $2\pi^2(K^2 d + \text{SNR}^{-1}) \log(D/\delta)$ measurements, to recover the relevant coordinates with probability $1 - \delta$.

IV. Adaptive algorithm

The drawback of the passive algorithm is the dependence on the dynamic range, K. This dependence arises because strong components can mask weaker components. The 1-bit nature of our set-up compounds this problem. If we were measuring direct inner products, rather than just their signs, then conventional iterative schemes, such as orthogonal matching pursuit, can discover components one by one, remove them and focus on the residual. This is possible because of the linear form of the measurement model in that setting. In the 1-bit situation, however, the non-invertible nonlinearity of the sign function makes it impossible to remove components from the measurements themselves. The one way to eliminate components is to, in effect, remove them before measurements are taken by placing zeros in the corresponding coordinates of the measurement vector. This observation is at the heart of the adaptive algorithm, and the dependence on K may be unavoidable without adaptation of the measurement process.

Before delving into the details of the adaptive algorithm, we first provide a high-level description. Each step of the algorithm involves finding one significant component. This is accomplished using a binary search tree. The root of the tree consists of the set of all indices under consideration and each leaf corresponds to one of the indices. Pairs of intermediate nodes are comprised of (roughly) even-sized, disjoint subsets of the indices of their parent node, above. The tree is illustrated for a set of four indices in Figure 1. The algorithm identifies a significant component by generating a path from the root to a leaf, such that each node along the path contains the index of at least one significant component. Once a significant component is found, it is removed from further consideration, and the procedure is repeated with a new tree with a root consisting of only the remaining components (i.e., with the discovered component(s) removed). The key ingredient in this process is a statistical test for significant components at each node, which is accomplished using adaptive measurements.

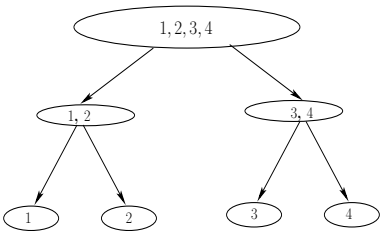

Fig. 1. Binary search tree for the adaptive algorithm for the case $D = 4$.

A. Adaptive Measurements

Measurements of the following form are collected at each node encountered in the search process. Suppose the node consists of a set $S \subset \{1, \ldots, D\}$ of indices. For a $D \times 1$ measurement vector, $\mathbf{x} = (x_1, \ldots, x_D)$, as follows.

$$x_i \overset{iid}{\sim} \begin{cases} N(0,1), & i \in S \\ 0, & i \notin S \end{cases}$$

Using $\mathbf{x}$, collect k iid measurements, $y_i = \text{sgn}(\langle \alpha, \mathbf{x} \rangle + w_i)$, $i = 1, \ldots, k$. Note that $\mathbf{x}$ is held fixed, but the noises $\{w_i\}$ are independent across measurements. If the empirical mean of the observations thus obtained is close to zero, then α and

$\mathbf{x}$ are probably orthogonal, and since $\mathbf{x}$ was chosen randomly, this suggests that none of $\{\alpha_i\}_{i\in S}$ are significant. This is formalized as follows.

Lemma 2. *Define* $\alpha_{\min} := \min_i |\alpha_i|$ *and let* $S \subset \{1,\ldots,D\}$ *and*

$$x_i \overset{iid}{\sim} \begin{cases} N(0,1), & i \in S \\ 0, & i \notin S \end{cases}$$

If $\min_{i\in S} |\alpha_i| \geq \alpha_{\min} > 0$*, then*

$$\mathbb{P}(|\langle \alpha, \mathbf{x}\rangle| > \alpha_{\min}/70) > 0.8 \tag{9}$$

Proof: Consider the orthonormal projection of the Gaussian vector $\mathbf{x}$ in the subspace spanned by the common coordinates of α and $\mathbf{x}$. The magnitude of this projection is a Gaussian variable with variance equal to the number of common coordinates (i.e., at least 1). Further the angle this projection makes with the projection of α is uniformly distributed over $[0,\pi]$. The condition in (9) is satisfied if the magnitude of the projection is greater than 1/10, and the angle between the projection and the projection of α is between $[0, 0.45\pi] \cup [0.55\pi, \pi]$ (because $\cos(0.45\pi) > 1/7$). The first event occurs with probability 0.92 (by using the cdf of the normal distribution) and the second event occurs with probability 0.9. The result follows since $0.9 \times 0.92 > 0.8$. ■

B. Testing for Significant Components

Let Q denote the standard normal cumulative distribution, $Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-x^2/2} dx$, and let A denote the event on which $|\langle \alpha, \mathbf{x}\rangle| > \alpha_{\min}/70$. Then the random variables $y_i = \text{sgn}(\langle \mathbf{x}, \alpha\rangle + w_i)$, where $w_i \overset{iid}{\sim} N(0,1)$, satisfy

$$|\mathbb{E}[y_i|A]| > 1 - 2Q(\alpha_{\min}/70).$$

To show this, consider the case when $\langle \mathbf{x}, \alpha\rangle$ is positive; the proof for the case case when it is negative follows by symmetry. If $\langle \mathbf{x}, \alpha\rangle > \alpha_{\min}/70$, then $y_i = -1$ implies that the magnitude of the noise must be at least $\alpha_{\min}/70$. Therefore $\mathbb{P}(y_i = -1) < Q(\alpha_{\min}/70)$, and $\mathbb{E}[y_i] = 1 - 2\mathbb{P}(y_i = -1) > 1 - 2Q(\alpha_{\min}/70)$, where here probabilities and expectations are with respect to the noise w_i. On the other hand, if $\alpha_i = 0$ for all $i \in S$, then $E[y_i] = 0$.

Now let $\bar{y} = k^{-1} \sum_{i=1}^k y_i$, the empirical average of the $\{y_i\}$. The statistic $\bar{y}$ concentrates about its mean, which is the basis for detecting significant components. As discussed above, the magnitude of the mean is either zero (in the case when no non-zero components are present) or strictly larger than $1-2Q(\alpha_{\min}/70)$ with probability at least 0.8 with respect to the random draw of $\mathbf{x}$. This suggests thresholding $\bar{y}$ at

$$\tau := 1/2 - Q(\alpha_{\min}/70). \tag{10}$$

The following lemmas characterize the action of the threshold.

Lemma 3. *If* $\alpha_i = 0$ *for all* $i \in S$*, then*

$$\mathbb{P}(\bar{y} > \tau) \quad < \quad \exp(-k\,\tau^2/2), \quad \textit{by Hoeffding's inequality.}$$

Lemma 4. *If* $\min_{i\in S} |\alpha_i| \geq \alpha_{\min}$*, then*

$$\mathbb{P}(\bar{y} < \tau) \quad < \quad 0.2 + \exp(-k\,\tau^2/2)$$

Proof: From Lemma 2, it follows that $|\langle \alpha, \mathbf{x}\rangle|$ is greater than $\alpha_{\min}/70$ with probability at least 0.80. From the discussion after Lemma 2 in this case the mean of the random variables y_i is at least $1 - 2Q(\alpha_{\min}/70)$. The result follows by Hoeffding's inequality and the union bound. ■

Note that if $k \geq \frac{6}{\tau^2}$, then $\exp(-k\,\tau^2/2) \leq 0.05$. Let us assume that k satisifies this condition, and define the indicator variable $\mathbf{1}_{\bar{y}>\tau}$, which takes the value 1 if $\bar{y}$ exceeds the threshold and 0 otherwise. If $\min_{i\in S} |\alpha_i| \geq \alpha_{\min}$, then this indicator is probably 1 (i.e., with probability at least 0.75). On the other hand, if $\alpha_i = 0$ for all $i \in S$, then the indicator is probably 0 (i.e., with probability at least 0.95).

We can boost these probabilities by repeating the procedure m times. That is, generate m independent random $\mathbf{x}$ according to the specificiations of Lemma 2, and for each such $\mathbf{x}$ average k repeated measurements of the $\text{sgn}(\langle \mathbf{x}, \alpha\rangle + w_i)$ and construct the indicator variable described above. Finally, the resulting indicator variables can be averaged and tested to see if the average is above or below $1/2$. We can then apply Chernoff's bound to obtain a decision about whether $\min_{i\in S} |\alpha_i| \geq \alpha_{\min}$ or $\max_{i\in S} |\alpha_i| = 0$ that is correct with probability at least $1-\delta$, for any $\delta > 0$ we desire, by choosing $m = O(\log 1/\delta)$. Thus, we have a test for significant components at a given node of the tree that is incorrect with probability at most δ.

C. Top-Down Tree Search Algorithm

Now we can build a path from the root to a leaf, as discussed above, and by the union bound the overall probability of error is at most $O(\delta \log D)$ since the depth of the tree is $\log D$. The procedure is summarized in Figure 2. If we carryout this entire process to d times in order to recover the d non-zero components of α, then the total probability of error is $O(\delta d \log D)$, which follows from another application of the union bound. By a simple recalibration, we see that the total probability of error can be controlled to be at most δ, for any $\delta > 0$, using a total of $O(d \log D \log(d \log D/\delta))$ measurements. The constant suppressed by the big-O notation depends only on $\alpha_{\min}$, not on K, as desired. However, this algorithm incurs an extra logarithmic factor (e.g., $\log(d \log D)$). This factor can be removed using a more sophisticated tree search based on a random walk on the tree, similar to the so-called *comparison tree* algorithm proposed in [11].

D. Random Walk Tree Search Algorithm

The random walk algorithm operates as follows. First, form a binary tree such that the root of the tree consists of all the D coordinates and the left (right) child node contains the lower (upper) half of the (lexicographically ordered) coordinates in their parent node (cf. Figure 1). The tree is constructed so that the leaves contain exactly one coordinate. We next extend each leaf to infinity, and the root node upwards to infinity, so that each leaf has infinite (grand)children with the same coordinate as itself, and the root has infinite (grand)parents, each with all

Top-Down Search Tree Algorithm

initialize: set of discovered components $A = \emptyset$ and set of components to be tested $S = \{1, \ldots, D\} \backslash A$.

while $\texttt{test(S)} = 1$
  while $|S| > 1$
    split S into two (near)
    equisized subsets, S_1 and S_2.
    if $\texttt{test}(\texttt{S}_1) = 1$, then set $S = S_1$
     elseif $\texttt{test}(\texttt{S}_2) = 1$, then set $S = S_2$
     else $S = \emptyset$.
  $A = A \bigcup S$.
  $S = \{1, \ldots, D\} \backslash A$.

subroutine `test`
  generate m random $\mathbf{x}$ with support on S
  for each $\mathbf{x}$
    collect k measurements and form $\mathbf{1}_{\bar{y} > \tau}$
  if average$(\mathbf{1}_{\bar{y} > \tau}) \geq 1/2$, then $\texttt{test(S)} = 1$
  else $\texttt{test(S)} = 0$

Fig. 2. Top-Down Search Tree algorithm

the coordinates. The random walk algorithm proceeds in a fashion similar to the top-down algorithm above, except that it moves down (or up) according to the value of $\mathbf{1}_{\bar{y}}$ for a single measurement vector $\mathbf{x}$, rather than average$(\mathbf{1}_{\bar{y} > \tau})$ which requires m random measurement vectors. This eliminates an extra $\log(d \log D)$ factor from the sample complexity.

As above, let A denote the set of discovered components, and initialize $A = \emptyset$. Denote the components to be tested as $S = \{1, \ldots, D\} \backslash A$. Beginning at the root node consising of all components in S, and moving toward the leaves, we test for non-zero support at each node as follows. At a given node, for each of the two children generate a random measurement vector $\mathbf{x}$ with non-zero support only on the subset of coordinates associated with that child. The values of the non-zero coordinates are i.i.d. realizations of a unit variance Gaussian. We then collect k measurements using $\mathbf{x}$ and compute the average $\bar{y}$. We move down to a child if $\bar{y} > \tau$ (or choose one out of the two randomly if both satisfy this property). If there is neither child satisfies this condition, then we backtrack to the parent. After M such steps (where M is specified below in Lemma 6), if we are at a node with a single coordinate, then we add it to the list of discovered coordinates A. Then run the decision tree again with the new set of coordinates $S = \{1, \ldots, D\} \backslash A$. If instead, after M steps, we are at the root, or one of its infinite (grand)parent(s), then we terminate the algorithm and report the set A.

To quantify the sample complexity of this the algorithm we will keep a running counter. The counter is initialized at zero and is incremented by one for correct moves and decremented by one for incorrect moves. More precisely, add $+1$ to the counter if we move down to a node that has at least one relevant coordinate or if we move back to the parent of a node with no relevant coordinates. Otherwise, add -1 to the counter.

Now assume that $k \geq 6/\delta^2$ and define $p := 0.2 + \exp(-k\,\tau^2/2) < 0.25$. Recall that p is the probability of an incorrect decision at a given node. Denote Δ_i as the random change in the counter at the i^{th} step. The following result can be shown for Δ_i.

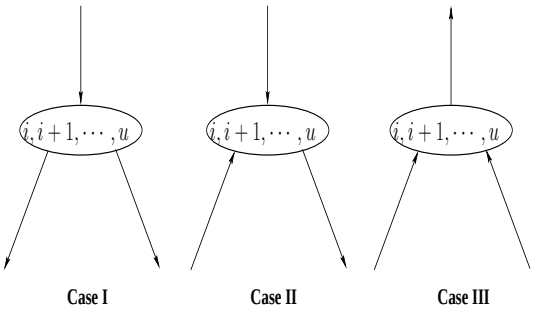

Fig. 3. Three cases for Lemma 5

Lemma 5. *The random variables Δ_i, $i = \{1, 2, \cdots, \}$, are independent and for each i*

$$\mathbb{P}[\Delta_i = 1] \geq (1-p)^2. \tag{11}$$

Proof: We can have three possible cases at each node of the tree as shown in Figure 3, where the arrows indicate the (correct) paths along which Δ_i increases by 1. We now analyze each of these cases.

Case I: In this case $\mathbb{P}[\Delta_i = -1]$ is given as p^2, that is both tests (for both children) give the wrong answer, thus

$$\mathbb{P}[\Delta_i = 1] = 1 - p^2. \tag{12}$$

Case II: In this case $\mathbb{P}[\Delta_i = -1]$ is given as p^2 (when both tests give wrong answers, and we go down the left child) plus $p(1-p)$ (when the left test gives the right answer and the right test gives the wrong answer and we backtrack), plus $p(1-p)/2$, when the left child gives the wrong answer and we choose it instead of the right child. Thus

$$\mathbb{P}[\Delta_i = 1] = 1 - p^2 - \frac{3}{2}p(1-p) = (1-p)\left(1 - \frac{p}{2}\right). \tag{13}$$

Case III: In this case $\mathbb{P}[\Delta_i = -1]$ is given as $2p - p^2$, i.e. when either test fails, thus

$$\mathbb{P}[\Delta_i = 1] = (1-p)^2 \tag{14}$$

Combining the three cases we find that the minimum value of $\mathbb{P}[\Delta_i = 1]$ is given as $(1-p)^2$. ∎

Since we assumed that $p \leq 0.25$, this implies that $\mathbb{P}[\Delta_i = 1] > 1/2$. Next we calculate the number of trials required to achieve a desired error probability for the tree algorithm. Let $L_i \in \{-1, 1\}$ be i.i.d. random variables such that $P[L_i = 1] = (1-p)^2$. It is obvious that $P[\sum_{i=1}^m \Delta_i > \ell] \geq P[\sum_{i=1}^m L_i > \ell]$, for any ℓ and m. Thus if chose M sufficiently large so that $\mathbb{P}[\sum_{i=1}^M L_i > \log D] > 1 - \delta'$, then because the depth (from root to leafs) is bounded by $\log D$, repeating M steps of the random walk guarantees that we will reach a relevant

coordinate with probability greater than $1-\delta'$, or if there is no such coordinate we are guaranteed to be at one of the grandparents of the root node with probability greater than $1-\delta'$. The following lemma provides an upper bound on M.

Lemma 6. *If*

$$M > (4/(2(1-p)^2-1)^2)\log(D/\delta'), \tag{15}$$

then

$$P\left[\sum_{j=1}^{M} L_j > \log D\right] > 1-\delta'. \tag{16}$$

Proof: We have $\mathbb{E}[L_i] = 2(1-p)^2-1$, using Hoeffding's inequality

$$P\left[\sum_{i=1}^{M} L_i < \ell\right] < \exp(-M(\ell/M - \mathbb{E}[L_i])^2). \tag{17}$$

Thus by setting $-M(\ell/M-\mathbb{E}[L_i])^2 < \log\delta'$, and solving for M, we get

$$M > \frac{2\ell\mathbb{E}[L_i] + \log(1/\delta')}{\mathbb{E}^2[L_i]} + \frac{\sqrt{2\ell\mathbb{E}[L_i] + \log 1/\delta' - 4\ell^2\mathbb{E}^2[L_i]}}{\mathbb{E}^2[L_i]}. \tag{18}$$

Setting $\ell = \log D$ we get

$$P\left[\sum_{i=1}^{M} L_i > \log D\right] > 1-\delta, \tag{19}$$

For M greater than the value specified in (18).

If we substitute $\mathbb{E}[L_i]$ by one in (18), multiply $\log(1/\delta')$ by two in the numerator, and disregard the negative sign term in the square root, we get an upper bound on the right hand side of the inequality in (18). This upper bound is equal to $(4/(2(1-p)^2-1)^2)\log(D/\delta')$. Thus the desired condition in (16) is satisfied for M greater than the value in (15). ∎

To achieve reliability greater than $1-\delta$, we set $\delta = \delta'/D$, in the above result. The average number of false positives is upper bounded by δ. Thus the algorithm runs at most $d+\delta$ times on an average, and the average sample complexity is no worse than

$$\frac{4dk(1+\delta/d)}{(2(1-p)^2-1)^2}\log\frac{D}{\delta}. \tag{20}$$

which by substituting for k and p can be upper bounded by

$$\frac{1536d}{\tau^2}\log\frac{D}{\delta} \tag{21}$$

Recall that $\tau = 1/2 - Q(\alpha_{\min}/70)$ and we have set SNR $:= \alpha_{\min}^2$. Note that the sample complexity of the random walk algorithm does not include the extra log factor and it is independent of dynamic range K.

V. Discussion

In sharp contrast to the results obtained in [12], our adaptive algorithm can at best provide a constant gain in the number of samples required over passive learning. In [12] the gain is unbounded at low SNR. This discrepancy can be explained as a consequence of the different measurement models. In [12] it was assumed that we have a constant energy per sensing vector $\mathbf{x}$, and thus the SNR grows as we focus on the relevant coordinates. In our active algorithm, we assume that we always probe with a vector that has unit variance in each nonzero component. If we were to change the SNR at each node in the tree such that for nodes with fewer coordinates we focus the sensing energy, the adaptive algorithm would require far fewer samples, and there could potentially be an unbounded gain over the passive case for low SNR.

The passive algorithm is similar to the back-projection algorithm used in compressed sensing, and we suspect that this is the reason that it is dependent on the dynamic range, a behavior that is also observed in its compressed sensing counterpart [13]. Perhaps, the algorithms based on L_1 minimization, such as the one in [5] may yield a passive algorithm that does not depend on the dynamic range of the signal. However, the analysis of such heuristics is quite challenging, mainly because the geometrical analysis of compressed sensing does not carry over to the quantized case. It is an interesting future problem to devise a passive algorithm whose sample complexity scales as $O(d\log D)$ yet is independent of dynamic range.

Acknowledgments

This work partially supported by AFOSR grant FA9550-09-1-0140.

References

[1] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, 2003.

[2] D. Donoho and J. Jin, "Higher criticism thresholding: Optimal feature selection when useful features are rare and weak," *Proceedings of the National Academy of Sciences*, vol. 105, no. 39, p. 14790, 2008.

[3] Y. Ingster, C. Pouet, and A. Tsybakov, "Classification of sparse high-dimensional vectors," *Philosophical Transactions A*, vol. 367, no. 1906, p. 4427, 2009.

[4] B. Krishnapuram, L. Carin, M. Figueiredo, and A. Hartemink, "Learning sparse Bayesian classifiers: multi-class formulation, fast algorithms, and generalization bounds," *IEEE Trans. Pattern Anal. Machine Intell*, vol. 27, pp. 957–968, 2005.

[5] P. Boufounos and R. Baraniuk, "1-bit compressive sensing," in *Conf. on Info. Science and Systems (CISS), Princeton, New Jersey*, 2008.

[6] J. Sun and V. Goyal, "Optimal Quantization of Random Measurements in Compressed Sensing," in *IEEE International Symposium on Information Theory, 2009. ISIT 2009*, 2009, pp. 6–10.

[7] W. Dai, H. Pham, and O. Milenkovic, "A Comparative Study of Quantized Compressive Sensing Schemes," in *Proc. Int. Symp. Inform. Theory*, 2009, pp. 6–10.

[8] E. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Info. Th.*, vol. 52, no. 2, pp. 489–509, 2006.

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

[10] M. X. Goemans and D. P. Williamson, "Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming," *Journal of the ACM*, vol. 42, pp. 1115–1145, 1995.

[11] U. Feige, P. Raghavan, D. Peleg, and E. Upfal, "Computing with noisy information," *SIAM Journal on Computing*, vol. 23, no. 5, pp. 1001–1018, 1994.

[12] J. Haupt, R. Baraniuk, R. Castro, and R. Nowak, "Compressive Distilled Sensing: Sparse Recovery Using Adaptivity in Compressive Measurements," in *Proceedings of the Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, Nov. 2009.

[13] A. Fletcher, S. Rangan, and V. Goyal, "Necessary and sufficient conditions on sparsity pattern recovery," *arXiv*, vol. 804.