Generalized Binary Search

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

***Abstract*—This paper studies a generalization of the classic binary search problem of locating a desired value within a sorted list. The classic problem can be viewed as determining the correct one-dimensional, binary-valued threshold function from a finite class of such functions based on queries taking the form of point samples of the function. The classic problem is also equivalent to a simple binary encoding of the threshold location. This paper extends binary search to learning more general binary-valued functions. Specifically, if the set of target functions and queries satisfy certain geometrical relationships, then an algorithm, based on selecting a query that is maximally discriminating at each step, will determine the correct function in a number of steps that is logarithmic in the number of functions under consideration. Examples of classes satisfying the geometrical relationships include linear separators in multiple dimensions. Extensions to handle noise are also discussed. Possible applications include machine learning, channel coding, and sequential experimental design.**

I. Problem Specification

Binary search can be viewed as a simple guessing game in which one is given an ordered list and asked to determine an unknown target value by making queries of the form "Is the target value greater than x?" For example, consider the integer guessing game in which the list is the set of integers from 1 to 100. The optimal strategy, which is familiar to most people, is to first ask if the number is larger than 50, and then ask similar "bisecting" questions of the intervals that result from this and subsequent queries. At each step of this process, the uncertainty about the location of the unknown target is halved, and thus after j steps the number of remaining possibilities is no larger than $100 \cdot 2^{-(j+1)}$.

The binary search problem can also be cast as learning a one-dimensional threshold function from queries in the form of point samples. Consider the threshold function $f(x) = \mathbf{1}_{\{x \leq t\}}$ on the interval $[0,1]$, where $t \in [0,1)$ is the threshold location and $\mathbf{1}_{\{x \leq t\}}$ is 1 if $x \leq t$ and 0 otherwise. Suppose that t belongs to the set $\{0, \frac{1}{n}, \ldots, \frac{n-1}{n}\}$. The location of t can then determined from $O(\log n)$ point samples using a bisection procedure analogous to the process above. In fact, if $n = 2^m$ for some integer m, then each point sample provides one bit in the m-bit binary expansion of t.

Very similar strategies can be employed even if the answers to the queries are unreliable [1], [2], [3], the so-called noisy binary search problem. The first result that we are aware of here was due to [1], based on maintaining a probability distribution on the target value (initially uniform), querying/sampling at the median of the distribution at each step, and then adjusting the distribution based on the response/observation according to a quasi-Bayes update. The method is based on a binary symmetric channel coding scheme that employs noiseless feedback [4]. Alternative approaches to the noisy binary search problem are essentially based on repeating each query in the classic binary search several times in order to be confident about the "correct" answer [2], [3].

This paper considers a generalized form of binary search based on the notion of maximally discriminative queries. We consider an abstract setting in which queries are selected from a set $\mathcal{X}$. The correct response to a query $x \in \mathcal{X}$ is either 'yes' (+1) or 'no' (-1), and is revealed by an *oracle* only after the query is selected. The queries are also put to a finite collection of hypotheses $\mathcal{H}$ with cardinality $|\mathcal{H}|$. Each hypothesis $h \in \mathcal{H}$ is a mapping from $\mathcal{X}$ to $\{-1, 1\}$. We assume that $\mathcal{H}$ contains the unknown oracle (i.e., the correct hypothesis) and that no two hypotheses agree on all possible queries (i.e., the hypotheses are unique with respect to $\mathcal{X}$). The goal is to find the correct hypothesis as quickly as possible through a sequence of carefully chosen queries. In particular, we study the following algorithm, which selects the maximally discriminating query at each step.

```
Generalized Binary Search (GBS)
initialize: $i = 1$, $\mathcal{H}_1 = \mathcal{H}$.
while $|\mathcal{H}_i| > 1$
1) Select $x_i = \arg\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_i} h(x)|$.
2) Query oracle with $x_i$ to obtain response $y_i$.
3) Set $\mathcal{H}_{i+1} = \{h \in \mathcal{H}_i : h(x_i) = y_i\}$, $i = i + 1$.
```

The query selection criterion picks a query that is maximally discriminative at each step (e.g., if $\min_{x \in \mathcal{X}} |\sum_{h \in \mathcal{H}_i} h(x)| = 0$ then there exists a query for which half of the hypotheses predict $+1$ and the other half predict -1). There may be more than one query that achieves the minimum, and in that case any minimizer is acceptable. Since the hypotheses are unique with respect to $\mathcal{X}$, it is clear that the algorithm above terminates in at most $|\mathcal{H}|$ queries (since it is always possible to find query that eliminates at least one hypothesis at each step). Note that exhaustive linear search also requires $O(|\mathcal{H}|)$ queries.

However, if it is possible to select queries such that at each step a fixed fraction of the remaining viable hypotheses are eliminated, then the correct hypothesis will be found in $O(\log |\mathcal{H}|)$ steps. The main result of this paper shows that GBS exhibits this property, provided that $\mathcal{X}$ and $\mathcal{H}$ satisfy certain geometrical relationships. Extensions to noisy GBS are also discussed. The emphasis in this paper is on determining the correct hypothesis with the fewest number of queries, and not on the computational complexity of selecting the queries. The motivation for this is that in many applications computational resources might be relatively inexpensive whereas obtaining the correct responses to queries may be very costly.

Sequential strategies similar to GBS are quite common in the machine learning literature. For example, [5] considered a very similar problem, and showed that the expected number of queries required by a similar search algorithm is never too much larger than any other strategy. However, general conditions under which such strategies yield exponential speed-ups over exhaustive linear search were not determined. We mention also the work in [6], which draws parallels between binary search and source coding. That work, however, assumes the possibility of making arbitrary queries (rather than queries restricted to a certain set) and so in the present context the problem considered there essentially reduces to encoding each hypothesis with $\log |\mathcal{H}|$ bits. Here we are interested in the interplay between a specific query space $\mathcal{X}$ and the hypothesis space $\mathcal{H}$. Classic binary search is an instance in which $\mathcal{X}$ and $\mathcal{H}$ are matched so that search and source coding are essentially the same problem, as pointed out above. We identify geometrical conditions on the the pair $(\mathcal{X}, \mathcal{H})$ that guarantee that GBS determines the correct hypothesis in $O(\log |\mathcal{H}|)$ queries.

II. Combinatorial Conditions for GBS

First consider an arbitrary sequential search procedure. Let $i = 1, 2, \ldots$ index the sequential process, x_i denote the query at step i, and y_i denote the correct response revealed by an oracle *after* the query is selected. If $\mathcal{H}_i$ denotes the set of viable hypotheses at step i (i.e., all hypotheses consistent with the queries up to that step), then ideally a query $x_i \in \mathcal{X}$ is selected such that the resulting viable hypothesis space $\mathcal{H}_{i+1}$ satisfies $|\mathcal{H}_{i+1}| \leq a_i \, |\mathcal{H}_i|$, for $0 < a_i < 1$, where $|\mathcal{H}_i|$ denotes the cardinality of $\mathcal{H}_i$. This condition is met if and only if

$$\left| \sum_{h \in \mathcal{H}_i} h(x_i) \right| \;\leq\; c_i \, |\mathcal{H}_i| \qquad (1)$$

for some $0 \leq c_i < 1$, in which case $a_i \leq (1 + c_i)/2$. Condition (1) quantifies the degree of *uncertainty* among the hypotheses in $\mathcal{H}_i$ for the query x_i. The smaller the value of $\left|\sum_{h \in \mathcal{H}_i} h(x_i)\right|$ the greater the uncertainty. Assuming that such an uncertainty condition holds for $i = 1, 2, \ldots,$ then after n steps of the algorithm

$$|\mathcal{H}_n| \;\leq\; |\mathcal{H}| \prod_{i=1}^{n} (1 + c_i)/2$$

and, in particular, the algorithm will terminate with the correct hypothesis as soon as $|\mathcal{H}| \prod_{i=1}^{n} (1 + c_i)/2 \leq 1$. Note that (1) trivially holds with $c_i = 1 - 2|\mathcal{H}_i|^{-1}$ ($a_i = 1 - |H_i|^{-1}|$), since there exists a query that eliminates at least one hypothesis at each step (recall that the hypotheses are assumed to be unique with respect to $\mathcal{X}$). Thus, we are interested in cases in which the c_i are uniformly bounded from above by a constant $0 \leq c < 1$ that does not depend on $|\mathcal{H}|$. In that case,

$$|\mathcal{H}_n| \;\leq\; \left(\frac{1 + c}{2}\right)^n |\mathcal{H}|$$

and the process terminates after at most $\lceil \log |\mathcal{H}| / \log(2/(1 + c)) \rceil = O(\log |\mathcal{H}|)$ steps. Based on the observations above, we can state the following:

Theorem 1. *Let $\mathcal{P}(\mathcal{H})$ denote the power set of $\mathcal{H}$. GBS converges to the correct hypothesis in $O(\log |\mathcal{H}|)$ if there exists a $0 \leq c < 1$ that does not depend on $|\mathcal{H}|$ such that*

$$\max_{\mathcal{G} \in \mathcal{P}(\mathcal{H})} \inf_{x \in \mathcal{X}} |\mathcal{G}|^{-1} \left| \sum_{h \in \mathcal{G}} h(x) \right| \;\leq\; c \qquad (2)$$

Condition (2) is sufficient, but may not be necessary since certain subsets in $\mathcal{P}(\mathcal{H})$ may never result through any sequence of queries. However, note that in the classic binary search setting, the condition does hold with $c = 1/3$, and thus the number of viable hypotheses is reduced by a factor of at least $(1+c)/2 = 2/3$ at each step. The value of $c = 1/3$ is an upper bound that is achieved only in the worst-case situation, when $\mathcal{G}$ consists of three elements; most of the steps in classic binary search reduce the number of viable hypotheses by a factor of roughly $1/2$. Unfortunately, verifying condition (2) in general is combinatorial, and so in the next section we seek conditions that are more easily verifiable.

III. Geometrical Conditions for GBS

Let P denote a probability measure over $\mathcal{X}$, assume that every $h \in \mathcal{H}$ is measurable with respect to P, and define the constant $0 \le c_P \le 1$ by

$$c_P \;=\; \max_{h\in\mathcal{H}} \left| \int_{\mathcal{X}} h(x)\, dP(x) \right| \;. \tag{3}$$

Note that by the triangle inequality, (3) implies

$$\max_{\mathcal{G}\in\mathcal{P}(\mathcal{H})} \; |\mathcal{G}|^{-1} \left| \sum_{h\in\mathcal{G}} \int_{\mathcal{X}} h(x)\, dP(x) \right| \;\le\; c_P \;. \tag{4}$$

Inequality (4) is a sort of relaxation of (2), with the minimization over $\mathcal{X}$ replaced by an average, and its verification requires only the calculation of the first P-moment of each $h \in \mathcal{H}$. Note that the minimal value of c_P is given by

$$c^* \;=\; \min_P \max_{h\in\mathcal{H}} \left| \int_{\mathcal{X}} h(x)\, dP(x) \right| \;, \tag{5}$$

where the minimization is over probability measures on $\mathcal{X}$. It is not hard to see that the minimizer exists because $\mathcal{H}$ is finite. Observe that the query space $\mathcal{X}$ can be partitioned into a finite number of disjoint sets such that every $h \in \mathcal{H}$ is constant for all queries in each such set. Let $\mathcal{A} = \mathcal{A}(\mathcal{X},\mathcal{H})$ denote the collection of these sets, which are at most $2^{|\mathcal{H}|}$ in number. The sets in $\mathcal{A}$ are equivalence classes in the following sense. For every $A \in \mathcal{A}$ and $h \in \mathcal{H}$, the value of $h(x)$ is constant (either $+1$ or -1) for all $x \in A$. Note that $\mathcal{X} = \bigcup_{A\in\mathcal{A}} A$. Therefore, the minimization in (5) can be carried out over a space of finite-dimensional probability mass functions over the elements of $\mathcal{A}$. The value of c^* will play an important role in characterizing the behavior of the GBS, but it does not need to be explicitly determined.

Note that for each $\mathcal{G} \in \mathcal{P}(\mathcal{H})$ one of two situations can occur:

1) $\min_{x\in\mathcal{X}} |\mathcal{G}|^{-1} \left|\sum_{h\in\mathcal{G}} h(x)\right| \;\le\; c^*$
2) $\min_{x\in\mathcal{X}} |\mathcal{G}|^{-1} \left|\sum_{h\in\mathcal{G}} h(x)\right| \;>\; c^*$

If c^* is reasonably small, then the first situation guarantees that a "good" discriminating query exists (i.e., one that will reduce the number of viable hypotheses by a factor of at least $(1+c^*)/2$). In the second situation, a highly discriminating query may not exist. The only guarantee is that there is always a query that eliminates at least one hypothesis, since the hypotheses are assumed to be unique with respect to $\mathcal{X}$. Therefore, a condition is required to ensure that such "bad" situations are not too problematic.

Note that if $\min_{x\in\mathcal{X}} |\mathcal{G}|^{-1} \left|\sum_{h\in\mathcal{G}} h(x)\right| > c^*$, then there exist $x, x' \in \mathcal{X}$ such that $|\mathcal{G}|^{-1} \sum_{h\in\mathcal{G}} h(x) > c^*$ and $|\mathcal{G}|^{-1} \sum_{h\in\mathcal{G}} h(x') < -c^*$. This follows since otherwise (4) cannot be satisfied with c^*. Under a mild condition discussed next, the existence of such an x and x' implies that the cardinality of $\mathcal{G}$ must be rather small. The condition is given in terms of the two following definitions.

Definition 1. *Two sets $A, A' \in \mathcal{A}$ are said to be k-*neighbors *if k or fewer hypotheses predict different values on A and A'. For example, A and A' are 1-neighbors if all but one element of $\mathcal{H}$ satisfy $h(x) = h(x')$ for all $x \in A$ and $x' \in A'$.*

Definition 2. *The query and hypothesis space $(\mathcal{X},\mathcal{H})$ are said to be k-*neighborly *if the k-neighborhood graph of $\mathcal{A}$ is connected (i.e., for every pair of sets in $\mathcal{A}$ there exists a sequence of k-neighbor sets that begins at one of the pair and ends with the other).*

Theorem 2. *If $(\mathcal{X},\mathcal{H})$ is k-neighborly, then GBS terminates with the correct hypothesis after at most $\lceil \log|\mathcal{H}|/\log(\alpha^{-1})\rceil$ queries, where $\alpha = \max\{\frac{1+c^*}{2}, \frac{k+1}{k+2}\}$ and $c^* = \min_P \max_{h\in\mathcal{H}} \left|\int_{\mathcal{X}} h(x)\, dP(x)\right|$.*

Remark 1. *Note that GBS requires no knowledge of c^*.*

Proof: Let c be any number satisfying $c^* \le c < 1$ and let x_i denote the query selected according to GBS at step i. If $|\mathcal{H}_i|^{-1} |\sum_{h\in\mathcal{H}_i} h(x_i)| \le c$, then the query x_i reduces the number of viable hypotheses by a factor of at least $(1+c)/2$. Otherwise, there exist $x, x' \in \mathcal{X}$ such that $|\mathcal{H}_i|^{-1} \sum_{h\in\mathcal{H}_i} h(x) > c$ and $|\mathcal{H}_i|^{-1} \sum_{h\in\mathcal{H}_i} h(x') < -c$, since (4) must be satisfied with c according to the definition of c^*.

Let $A, A' \in \mathcal{A}$ denote the subsets containing x and x', respectively. The k-neighborly condition guarantees that there exists a sequence of k-neighbor sets beginning at A and ending at A'. Note that $|\mathcal{H}_i|^{-1} \left|\sum_{h\in\mathcal{H}_i} h(\cdot)\right| > c$ on every set and the sign of $|\mathcal{H}_i|^{-1} \sum_{h\in\mathcal{H}_i} h(\cdot)$ must change at some point in the sequence. It follows that there exist points $x, x' \in \mathcal{X}$ such that $|\mathcal{H}_i|^{-1}\sum_{h\in\mathcal{H}_i} h(x) > c$ and $|\mathcal{H}_i|^{-1}\sum_{h\in\mathcal{H}_i} h(x') < -c$ *and* furthermore, all but at most k of the hypotheses predict the same value for both x and x'.

Two inequalities follow from this observation. First, $\sum_{h\in\mathcal{H}_i} h(x) - \sum_{h\in\mathcal{H}_i} h(x') > 2c|\mathcal{H}_i|$. Second, $|\sum_{h\in\mathcal{H}_i} h(x) - \sum_{h\in\mathcal{H}_i} h(x')| \leq 2k$. Combining these inequalities yields $|\mathcal{H}_i| < k/c$. Furthermore, there exists a query that eliminates at least one hypothesis due to the uniqueness of the hypotheses with respect to $\mathcal{X}$. Thus, at least one hypothesis must respond incorrectly to x_i, and so $|\mathcal{H}_{i+1}| \leq |\mathcal{H}_i| - 1 = |\mathcal{H}_i|(1-|\mathcal{H}_i|^{-1}) < |\mathcal{H}_i|(1-c/k)$.

This shows that if $|\mathcal{H}_i|^{-1}|\sum_{h\in\mathcal{H}_i} h(x_i)| > c$, then the query x_i reduces the number of viable hypotheses by a factor of at least $(1-c/k)$. Also, recall that if $|\mathcal{H}_i|^{-1}|\sum_{h\in\mathcal{H}_i} h(x_i)| \leq c$, then the query x_i reduces the number of viable hypotheses by a factor of at least $(1+c)/2$. It follows that the each GBS query reduces the number of viable hypotheses by a factor of at least

$$\min_{c\geq c^*}\max\left\{\frac{1+c}{2}, 1-c/k\right\} = \max\left\{\frac{1+c^*}{2}, \frac{k+1}{k+2}\right\}.$$

■

IV. Applications

For a given pair $(\mathcal{X},\mathcal{H})$, the effectiveness of GBS hinges on determining (or bounding) c^* and establishing that $(\mathcal{X},\mathcal{H})$ are neighborly. Recall the definition of the bound c^* from (5). A *trivial* bound is

$$\max_{h\in\mathcal{H}}\left|\int_{\mathcal{X}} h(x)\,dP(x)\right| \;\leq\; 1-2|\mathcal{H}|^{-1}\ ,$$

since this bound simply produces the convergence factor $1-|\mathcal{H}|^{-1}$, which is achieved by an exhaustive linear search. Non-trivial moment bounds are those for which

$$\max_{h\in\mathcal{H}}\left|\int_{\mathcal{X}} h(x)\,dP(x)\right| \;\leq\; c\ ,$$

for a $0 \leq c < 1$ that does not depend unfavorably on $|\mathcal{H}|$. In this section we consider several illustrative applications of GBS, calculating/bounding c^* and verifying neighborliness of $(\mathcal{X},\mathcal{H})$ in each case.

A. Classic Binary Search

Classic binary search can be viewed as the problem of determining a threshold value $t \in (0,1)$. Let $\mathcal{H}$ be a set of hypotheses of the form $h_v(x) = 2\,\mathbf{1}_{\{x>v\}} - 1$, where $v \in V$ and V is a finite set of points in $(0,1)$ and $\mathbf{1}_B$ denotes the indicator of the event B. Each query $x \in \mathcal{X} \subset [0,1]$ receives a correct response $y = 2\,\mathbf{1}_{\{x>t\}} - 1$ from the oracle. Assume that $t \in V$ (i.e., the oracle is contained in $\mathcal{H}$) and assume that $V \subset \mathcal{X}$.

First consider c^*. Assume that $\mathcal{X}$ contains the points 0 and 1. Then taking P to be two point masses at $x = 0$ and $x = 1$ of probability $1/2$ each yields $\left|\int_{\mathcal{X}} h(x)\,dP(x)\right| = 0$ for every $h \in \mathcal{H}$, since $h(0) = -1$ and $h(1) = 1$ for every $h \in \mathcal{H}$. Thus, $c^* = 0$.

Now consider the neighborly condition. Recall that $\mathcal{A}$ is the partition on $\mathcal{X}$ induced by $\mathcal{H}$, such that for each set $A \in \mathcal{A}$ every $h \in H$ has a constant response. In this case, each such set is an interval of the form $A_i = (v_{i-1}, v_i]$, $i = 1,\ldots,|V|+1$, where $v_1 < v_2 < \cdots < v_{|V|}$ are the ordered values in V and $v_0 = 0$ and $v_{|V|+1} = 1$. Note that since $V \subset \mathcal{X}$, each set A_i contains at least one query. Furthermore, observe that only a single hypothesis, h_{v_i}, has different responses to queries from A_i and A_{i+1}. Thus, each successive pair of such sets are 1-neighbors. Moreover, the 1-neighborhood graph is connected in this case, and so $(\mathcal{X},\mathcal{H})$ are 1-neighborly.

We conclude that the generalized binary search algorithm of Theorem 2 determines the optimal hypothesis in $O(\log|\mathcal{H}|)$ steps; i.e., the classic binary search result.

B. Interval Classes

Let $\mathcal{X} = [0,1]$ and consider a finite collection of hypotheses of the form $h_{a,b}(x) = 2\,\mathbf{1}_{a\leq x<b} - 1$, with $0 \leq a < b \leq 1$. Assume that the hypotheses do not have endpoints in common, and that one produces the correct prediction at all points in $[0,1]$. The partition $\mathcal{A}$ again consists of intervals, and since there are no common endpoints, the neighborly condition is satisfied with $k = 1$. To bound c^*, note that the minimizing P must place some mass within and outside each such interval. If the intervals all have length at least $\ell > 0$, then taking P to be the uniform measure on $[0,1]$ yields that $c^* \leq |2\ell - 1|$, irrespective of the number of interval hypotheses under consideration. Therefore, in this setting GBS determines the correct hypothesis in $O(\log|\mathcal{H}|)$ steps.

However, consider the special case in which the intervals are disjoint. Then it is not hard to see that the best allocation of mass is to place $1/|\mathcal{H}|$ mass in each subinterval, resulting in $c^* = 1-2|\mathcal{H}|^{-1}$. And so, GBS is not guaranteed to terminate in fewer than $|\mathcal{H}|$ steps (the number of steps required by exhaustive linear search). In this case, however, note that if queries of a different form were allowed, then much better performance is possible. For example, if queries in the form of dyadic subinterval tests were allowed (e.g., tests that indicate whether or not the correct hypothesis is $+1$-valued anywhere on a dyadic subinterval of choice), then the correct hypothesis can be identified through $O(\log|\mathcal{H}|)$ queries (essentially a binary encoding of the correct hypothesis). This emphasizes the importance of the geometrical relationship between $\mathcal{X}$ and $\mathcal{H}$ embodied in the neighborly condition and the value of c^*. Optimizing the query space to the structure of $\mathcal{H}$ is somewhat related to the ideas in [6] and to the theory of compressed sensing [7], [8].

C. Linear Separators in $[-1,1]^d$

Multi-dimensional threshold functions are particularly relevant in machine learning and pattern classification. Learning binary classifiers based on hyperplanes in $d > 1$ dimensions is thus an important generalization of classic binary search. Let $\mathcal{X} = [-1,1]^d$, $d \geq 1$, and consider a finite collection of hyperplanes of the form $\langle a,x\rangle + b = 0$, where $a \in \mathbb{R}^d$, $b \in \mathbb{R}$, and $\langle a,x\rangle$ is the inner product between a and x. Assume that every hyperplane in the collection is distinct and intersects the set $(-1,1)^d$. Two d-dimensional threshold functions are associated with each hyperplane: $h_{a,b}(x) = 2\mathbf{1}_{\{\langle a,x\rangle+b>0\}} - 1$ and $-h_{a,b}(x)$. Let $\mathcal{H}$ denote the set of threshold functions formed from the finite collection of hyperplanes in this fashion. Assume that the correct label at each point $x \in \mathcal{X}$ is given by one function in $\mathcal{H}$.

To bound c^*, let P be point masses of probability 2^{-d} at each of the 2^d vertices of the cube $[-1,1]^d$. Then $\left|\int_{\mathcal{X}} h(x)\, dP(x)\right| \leq 1 - 2^{-d+1}$ for every $h \in \mathcal{H}$, since for each h there is at least one vertex on where it predicts $+1$ and one where it predicts -1. Thus, $c^* \leq 1 - 2^{-d+1}$.

To verify the neighborly condition, note that in this case every set in the partition $\mathcal{A}$ is a polytope delineated by a subset of the hyperplanes. And, since the hyperplanes are distinct, two sets which share a common face are 2-neighbors (only the two hypotheses associated with the hyperplane that defines that face predict differently on queries from the two sets). Clearly, since the sets in $\mathcal{A}$ tesselate $\mathcal{X}$, the 2-neighborhood graph is connected and so $(\mathcal{X},\mathcal{H})$ is 2-neighborly. We conclude that the GBS determines the optimal hypothesis in $O(2^{d-1}\log|\mathcal{H}|)$ steps. This appears to be a new result.

A noteworthy case is the collection of hypotheses formed by threshold functions based on hyperplanes of the form $\langle a,x\rangle = 0$, i.e., hyperplanes passing through the origin. In this case, with P as specified above, $c^* = 0$, since each hypothesis responds with $+1$ at half of the vertices and -1 on the other half. Therefore, GBS determines the optimal hypothesis in no more than $O(\log|\mathcal{H}|)$ steps, independent of the dimension. Related results for this special case have been previously reported; see [9] and the references therein. Note that even if the hyperplanes do not pass through the origin ($b \neq 0$), $O(\log|\mathcal{H}|)$ convergence is still attained so long as $|b|$ is not too large. This generalizes earlier results.

In the case of general linear separators, the dependence on dimension d can also be eliminated with an additional assumption. Suppose that for a certain P on $\mathcal{X}$ the P-moment of the optimal hypothesis is known to be upper bounded by a constant $\rho < 1$ that does not depend on $|\mathcal{H}|$. Then all hypotheses that violate the bound can be eliminated from consideration and GBS applied to the set of remaining hypotheses will determine the correct hypothesis in $O(\log|\mathcal{H}|)$ steps. Situations like this can arise, for example, in binary classification problems with side/prior knowledge that the marginal probabilities of the two classes are somewhat balanced. Then the moment of the correct hypothesis, with respect to the marginal probability distribution of features, is bounded far away from 1 and -1. This provides another generalization of earlier results.

D. Discrete Query Spaces

In many situations both the hypothesis and query spaces may be discrete. A machine learning application, for example, may have access to a large (but finite) pool of unlabeled examples, any of which may be queried for a label. Because obtaining labels can be costly, "active" learning algorithms select only those examples that are predicted to be highly informative for labeling. Theorem 2 applies equally well to continuous or discrete query spaces. For example, consider the linear separator case, but instead of the query space $[-1,1]^d$ suppose that $\mathcal{X}$ is a finite subset of points in $[-1,1]^d$. The hypotheses again induce a partition of $\mathcal{X}$ into subsets $\mathcal{A}(\mathcal{X},\mathcal{H})$, but the number of subsets in the partition may be less than the number in $\mathcal{A}([-1,1]^d,\mathcal{H})$. Consequently, the 2-neighborhood graph of $\mathcal{A}(\mathcal{X},\mathcal{H})$ depends on the specific points that are included in $\mathcal{X}$ and may or may not be connected.

Consider two illustrative examples. Let $\mathcal{H}$ be a collection of linear separators as in Section IV-C above and first reconsider the partition $\mathcal{A}([-1,1]^d,\mathcal{H})$. Recall that each set in $\mathcal{A}([-1,1]^d,\mathcal{H})$ is a polytope. Suppose that a discrete set $\mathcal{X}$ contains at least one point inside each of the polytopes in $\mathcal{A}([-1,1]^d,\mathcal{H})$. Then it follows from the results above that $(\mathcal{X},\mathcal{H})$ is 2-neighborly. Second, consider a very simple case in $d = 2$ dimensions. Suppose $\mathcal{X}$ consists of just three non-colinear points $\{x_1,x_2,x_3\}$ and suppose that $\mathcal{H}$ is comprised of six classifers, $\{h_1^+,h_1^-,h_2^+,h_2^-,h_3^+,h_3^-\}$, satisfying $h_i^+(x_i) =$

+1, $h_i^+(x_j) = -1, j \neq i$, $i = 1, 2, 3$, and $h_i^- = -h_i^+$, $i = 1, 2, 3$. In this case, $\mathcal{A}(\mathcal{X}, \mathcal{H}) = \{\{x_1\}, \{x_2\}, \{x_3\}\}$ and the responses to each pair of queries differ for four of the six hypotheses. Thus, the 4-neighborhood graph of $\mathcal{A}(\mathcal{X}, \mathcal{H})$ is connected, but the 2-neighborhood is not.

V. Extensions to Noisy Search

We now turn attention to the so-called noisy binary search problem. The situation considered here is that the oracle no longer returns the correct answer to every query, but instead responds correctly with probability at least $1 - p$ and incorrectly with probability at most p, for an unknown $0 < p < 1/2$. This is equivalent to the situation in which the oracle (sender) communicates answers to the learner (receiver) over a binary symmetric channel with crossover probability p, but the feedback channel (query channel) is noiseless. The goal remains to identify the correct hypothesis in $\mathcal{H}$, despite the fact that the oracle may respond incorrectly. We will assume that the erroneous responses are determined by a random coin toss. Therefore, since the oracle is probably correct, one can decide the correct response to a given query (with very high confidence) by repeating it several times. This observation is the basis for most noisy binary search procedures, although optimal methods require a fairly delicate and subtle application of this basic intuition.

To the best of our knowledge, a version of the classic binary search problem in noise was first considered by Horstein [4] in the context of channel coding with noiseless feedback. The first rigorous analysis, motivated by the work of Horstein, was developed in [1], where the information-theoretic optimality of a multiplicative weighting algorithm was established. A closely related set of results was recently reported in [3], which also includes results similar in spirit to [2]. We also mention the works of [10], [11], which consider adversarial situations in which the total number of erroneous oracle responses is fixed in advance.

Based on an approach similar to that used in many of the papers above, we have the following result. Recall that under the assumptions of Theorem 2, GBS terminates after at most $n_o = O(\log|\mathcal{H}|)$ queries.

Theorem 3. *If the oracle error probability is less than or equal to p, for some unknown $0 < p < 1/2$, and the assumptions of Theorem 2 hold, then there exists a noise-tolerant variant of GBS in the following sense: If GBS terminates in at most n_o queries in the noiseless setting ($p = 0$), then there exists a modified search strategy that, with probability at least $1-\delta$, terminates with the correct hypothesis in at most*

$$O\left(n_o \log(n_o/\delta) \log\log(n_o/\delta)/\epsilon^2\right) \quad \textit{steps,}$$

where $\epsilon = |p - 1/2|$.

Proof: The modified algorithm is based on the simple idea of repeating each query of the GBS several times, in order to overcome the uncertainty introduced by the noise. Since the value of p is unknown in advance, an adaptive procedure is required. Thus, we first recall Lemma 1 from [2].

Lemma 1. *Consider a coin with an unknown probability p of heads. Then for any $\delta' > 0$ there exists an adaptive procedure for tossing the coin such that, with probability at least $1 - \delta'$, the number of coin tosses is at most*

$$m(\delta') = \frac{\log(2/\delta')}{4\epsilon^2} \log\left(\frac{\log(2/\delta')}{4\epsilon^2}\right)$$

and the procedure reports correctly whether heads or tails is more likely.

The proof of the lemma and the procedure itself are based on relatively straightforward, iterated applications of Chernoff's bound; see [2] for further details. For the sake of completeness, we state the procedure here.

```
Adaptive Coin Tossing Procedure
initialize: set $m_o = 1$ and toss the coin once.
for $j = 0, 1, \ldots$ set
    1) $p_j$ = frequency of heads $(+1)$
    2) $I_j = \left[p_j - \sqrt{\frac{(j+1)\log(2/\delta)}{2^j}}, p_j + \sqrt{\frac{(j+1)\log(2/\delta)}{2^j}}\right]$
    3) If $1/2 \in I_j$, then toss coin $m_j$ more times and
       set $m_{j+1} = 2m_j$, otherwise break.
end
If $I_j \subset [-\infty, 1/2]$, output $-1$, otherwise output $+1$.
```

Now consider the n_o queries chosen by GBS in the noiseless case. Repeat each query several times, according to the adaptive procedure above. By the union bound, with probability at least $1 - n_o\delta'$, each query is repeated at most $m(\delta')$ times and the correct responses to all n_o queries are determined. Setting $\delta' = \delta/n_o$ yields the upper bound on the number of queries. ■

Whether or not the bound in Theorem 3 is optimal in the case of noisy GBS is an open question. For classic binary search with noise, more subtle procedures can be used to obtain slight improvements [1], [3].

VI. Conclusions

This paper studied a generalization of the classic binary search problem. In particular, the generalized problem extends binary search techniques to multidimensional threshold functions, which arise in machine learning and pattern classification. If $(\mathcal{X}, \mathcal{H})$ is neighborly (Definition 2) and if c^* does not depend explicity on $|\mathcal{H}|$, then the number of steps required by GBS is $O(\log |\mathcal{H}|)$, exponentially smaller than the number of steps in an exhaustive linear search. The conditions express a geometrical relationship between $\mathcal{X}$ and $\mathcal{H}$ which quantifies how well matched the queries are to the structure of hypotheses.

The GBS problem can also be viewed as a source coding problem in which $\mathcal{X}$ plays the role of a codeset and $\mathcal{H}$ plays the role of a source. In certain cases (e.g., classic binary search) GBS and ideal binary encoding are equivalent, but in general they are not. The neighborly condition and the value of c^* reflect the degree to which $\mathcal{X}$ matches the source $\mathcal{H}$.

Finally, we point out that if the error probability is not bounded away from $1/2$ in the noisy setting, then exponential speed-ups over linear search are no longer achievable by any search strategy. However, appropriate noisy binary search strategies can provide polynomial speed-ups over linear search [12], [13].

References

[1] M. V. Burnashev and K. S. Zigangirov, “An interval estimation problem for controlled observations,” *Problems in Information Transmission*, vol. 10, pp. 223–231, 1974.

[2] M. Kääriäinen, “Active learning in the non-realizable case,” in *Algorithmic Learning Theory*, 2006, pp. 63–77.

[3] R. Karp and R. Kleinberg, “Noisy binary search and its applications,” in *Proceedings of the 18th ACM-SIAM Symposium on Discrete Algorithms (SODA 2007)*, pp. 881–890.

[4] M. Horstein, “Sequential decoding using noiseless feedback,” *IEEE Trans. Info. Theory*, vol. 9, no. 3, pp. 136–143, 1963.

[5] S. Dasgupta, “Analysis of a greedy active learning strategy,” in *Neural Information Processing Systems*, 2004.

[6] S. R. Kulkarni, S. K. Mitter, and J. N. Tsitsiklis, “Active learning using arbitrary binary valued queries,” *Machine Learning*, pp. 23–35, 1993.

[7] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information,” *IEEE Trans. Inform. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[8] D. L. Donoho, “Compressed sensing,” *IEEE Trans. Inform. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[9] S. Dasgupta, “Coarse sample complexity bounds for active learning,” in *Neural Information Processing Systems*, 2005.

[10] R. L. Rivest, A. R. Meyer, and D. J. Kleitman, “Coping with errors in binary search procedure,” *J. Comput. System Sci.*, pp. 396–404, 1980.

[11] J. Spencer, “Ulam’s searching game with a fixed number of lies,” in *Theoretical Computer Science*, 1992, pp. 95:307–321.

[12] R. Castro and R. Nowak, “Upper and lower bounds for active learning,” in *44th Annual Allerton Conference on Communication, Control and Computing*, 2006.

[13] ——, “Minimax bounds for active learning,” *IEEE Trans. Info. Theory*, pp. 2339–2353, 2008.