Finding Relevant Variables in PAC Model with Membership Queries

David Guijarro[1⋆], Jun Tarui[2], and Tatsuie Tsukiji[3]

[1] Department LSI, Universitat Politècnica de Catalunya, Jordi Girona Salgado, 1-3, Barcelona 08034, Spain david@lsi.upc.es

[2] Department of Communications and Systems, University of Electro-communications,Chogugaoka, Chofu-shi, Tokyo 182, Japan jun@sw.cas.uec.ac.jp

[3] School of Informatics and Sciences, Nagoya University, Nagoya 464-8601, Japan tsukiji@info.human.nagoya-u.ac.jp

Abstract. A new research frontier in AI and data mining seeks to develop methods to automatically discover relevant variables among many irrelevant ones. In this paper, we present four algorithms that output such crucial variables in PAC model with membership queries. The first algorithm executes the task under any unknown distribution by measuring the distance between virtual and real targets. The second algorithm exhausts virtual version space under an arbitrary distribution. The third algorithm exhausts universal set under the uniform distribution. The fourth algorithm measures influence of variables under the uniform distribution. Knowing the number r of relevant variables, the first algorithm runs in almost linear time for r. The second and the third ones use less membership queries than the first one, but runs in time exponential for r. The fourth one enumerates highly influential variables in quadratic time for r.

1 Introduction : Terminology and Strategy

We propose several algorithms with their own character for automatically finding relevant variables in the presence of many irrelevant ones. Recent application of such algorithms ranges from data mining in the genome analysis to information filtering in the network computing. In these applications, sample data consist of a huge volume of variables, although the target phenomenon may depend on only a few of them. In order for a machine to find such crucial variables, we study algorithms in PAC model with membership queries, and analyze their query and time complexities.

To learn an unknown Boolean function that depends on only a small number of the potential variables, the learner may (1) find a set of relevant variables and (2) combine them into a hypothesis that approximates the target concept with

⋆ Author supported in part by the EC through the Esprit Program EU BRA program under project 20244 (ALCOM-IT) and the EC Working Group EP27150 (NeuroColt II) and by the spanish DGES PB95-0787 (Koala).

an arbitrarily high accuracy. Most inductive learning algorithms do not separate these conceptually different tasks, but rather let them depend on each other. On the other hand, some AI learning papers separate them and execute (1) as a preprocess for (2) (see [5] for a survey). This paper pursues (1) for the learning goal.

Let us begin with fixing notion of *relevant variables.* We say that two Boolean instances are neighbor of each other if they have the same bit length and differs by exactly one bit.

Definition 1. *A variable x is* relevant *to a function f if there exist neighbor instanes A and A' such that $x(A) \neq x(A')$ and $f(A) \neq f(A')$.*

Let $\mathrm{Rel}(r, n)$ be the class of Boolean functions with n variables that depend on only r of them, or equivalently, have at most r relevant variables. In this paper, the target concept is an arbitrary function in $\mathrm{Rel}(r, n)$. In other words, the learner is given the n Boolean variables, say $x_1, \ldots, x_n$, and told that there are at most r relevant variables to the target concept [1].

In this paper we work on *Probably Approximately Correct* (PAC) Learning introduced by Valiant [11], where the target concept f and the target distribution $\mathcal{D}$ are postulated and hidden from the learner. The learner receives a sequence of examples $(A, f(A))$, $(B, f(B))$, $\ldots$ independently and randomly from $\mathcal{D}$. From these examples, the learner must build a hypothesis that can approximate the target concept by an arbitrary accuracy with respect to $\mathcal{D}$.

This paper sets a weaker goal for learning. For a Boolean function h let $\mathrm{rel}(h)$ denotes the set of relevant variables to h.

Definition 2. *A set V of variables is called an α-dominator, $0 \leq \alpha \leq 1$, if there exists a function h such that $\mathrm{rel}(h) \subseteq V$ and $\mathrm{Prob}_A\{h(A) = f(A)\} \geq \alpha$, where A is randomly chosen according to $\mathcal{D}$.*

A weaker goal of learning is to find an $(1 - \varepsilon)$-dominator. Our algorithms may use inner coin flips and achieve this weaker goal with probability $\geq 1 - \delta$ for arbitrary given constants $0 \leq \varepsilon, \delta \leq 1$.

Unfortunately, random examples are not enough as information resource for identifying relevant variables of a given Boolean concept, because:

- There can be computationally hard problems even though we can information-theoretically identify the function. For instance consider the class of conjunctions of an $(\log n)$-bit majority and an $(\log n)$-bit parity. This class (contained in $\mathrm{Rel}(2 \log n, n)$) is believed to be hard to predict within polynomial even under the uniform distribution [3]. The problem there is that we may have enough information to identify the function (using a universal set) but it is computationally hard to discover the set of relevant variables.
- Moreover, due to reduction from set cover, it is NP-hard in general to discover a set of relevant variables on which one can build an accurate hypothesis [8].

[1] We can tune our algorithms up into adaptive versions for r by guessing and doubling r, that is to execute the algorithms by guessing $r = 2^0, 2^1, 2^2, \ldots$ until they succeed.

We will thus allow algorithms to use more active query in addition with random examples, that are *membership queries* introduced by Angluin [1].

Definition 3. *A membership query for an instance A about the target concept f is a request for the value $\mathrm{MQ}_f(A) = f(A)$ to the membership oracle MQ_f.*

In empirical test, reverse engineer or information search, however, membership queries are much more expensive than random examples. We will thus present econimical algorithms in Section 3 that spend at most $r \log n$ membership queries.

The algorithms in this paper follow a common learning strategy; Find witnesses for relevance and execute binary search on them. The algorithms differ in methods to discover witnesses for relevance.

Definition 4. *For an instance A and a set V of variables A_V and A^V are instances such that*

$$x(A_V) = \begin{cases} x(A) & \text{if } x \in V, \\ 0 & \text{otherwise.} \end{cases} \qquad x(A^V) = \begin{cases} 1 - x(A) & \text{if } x \in V, \\ x(A) & \text{otherwise.} \end{cases}$$

Definition 5. *A witness for relevance outside of V about the target concept f is a pair (A, B) of instances A and B such that $A_V = B_V$ and $f(A) \neq f(B)$.*

Given a witness (A, B) for relevance, binary search finds a relevant variable as follows. It flips half of the different bits between A and B, and ask the membership query for the obtained instance C. If $f(A) \neq f(C)$ then it repeats the argument on a new witness (A, C), otherwise on (B, C). Either case reduces the number of different bits in the witnesses by half. This divide-and-query argument repeats until reaching to a variable x such that $x(A) \neq x(B)$.

Lemma 1. *Given an arbitrary witness for relevance outside of V, the binary search outputs a relevant variable $x \notin V$ by using at most $\log n$ membership queries in $O(n \log n)$ time.*

The proof is folklore (see e.g. [7, Lemma 2.4]). Note that $\log n$ is a query complexity lower bound in finding one relevant variable from a given witness for relevance, hence $r \log n$ is a lower bound in finding r relevant variables.

2 Measuring the Distance between Virtual and Real Targets

This section provides a distribution-free algorithm that runs in time almost linear to r. It measures the distance between virtual and real targets and if the distance is large then the algorithm discovers a new relevant variable.

Definition 6. *The virtual target f_V on a set V of variables about the target concept f is a Boolean function f_V with $\mathrm{rel}(f_V) \subseteq V$ such that $f_V(A_V) = f(A_V)$ holds for any instance A.*

In the mistake bound model with membership queries, Blum, Hellerstein and Littlestone [4] measures $\mathrm{Prob}_A\{f(A) = h_V(A)\}$ for a temporal hypothesis h to find a new relevant variable that is not yet implemented in h. Here we measure $\mathrm{Prob}_A\{f(A) = f_V(A)\}$, the distance between f and f_V. If it is large, then our algorithm finds a witness for relevance (A, A_V) with high probability.

Lemma 2. *Let m be any integer with $(1-\varepsilon)^m \le \delta$. Draw a sample O of m examples. If V is not an $(1-\varepsilon)$-dominator then $f(A) \ne f(A_V)$ happens for some $(A, f(A)) \in O$ with probability greater than $1-\delta$.*

Proof. Suppose that V is not an $(1-\varepsilon)$-dominator. Then the distance between f and f_V is greater than ε, hence $f(A) = f(A_V)$ happens for each $(A, f(A)) \in O$ with probability $< (1-\varepsilon)^m \le \delta$.

Now we implement this lemma in an algorithm MsrDist and analyze its performance.

Procedure MsrDist
  Input: Integers n, r and m_0.
  Output: A set of relevant variables.
  Set $V := \emptyset$ and $m := 0$.
  Do until either $|V| = r$ or $m = m_0$
    Draw a random example $(A, f(A))$ and increment m by 1.
    Let $\beta := \mathrm{MQ}_f(A_V)$.
    If $f(A) \ne \beta$ then Do
      Set $m := 0$.
      Execute the binary search on (A, A_V) and put an obtained variable into V.
    EndDo
  EndDo
  Output V.
EndProc

Theorem 1. *MsrDist finds an $(1-\varepsilon)$-dominator (a set of relevant variables) with probability greater than $1-\delta$. For some $m_0 = O(\varepsilon^{-1}\log(r/\delta))$, MsrDist uses at most $m_0 r$ random examples, at most $r(m_0 + \log n)$ membership queries and runs in $O(nr(m_0 + \log n))$ time.*

Proof. Fix an arbitrary $m_0 = O(\varepsilon^{-1}\log(r/\delta))$ such that $(1-\varepsilon)^{m_0} \le \delta/r$. Then, in view Lemma 2, in each stage of the Do loop in the procedure MsrDist where V is not an $(1-\varepsilon)$-dominator, each example $(A, f(A))$ may be a witness (A, A_V) for relevance outside of V with probability greater than δ/r. Since MsrInf can draw m_0 examples in the stage, it will succeed in find a witness with probability greater than $1 - (1 - \frac{\varepsilon}{er})^{m_0} \ge 1 - \delta/r$. Since there are at most r stages, due to the union bound, every stage will succeed with probability $\ge 1 - \delta/r \cdot r \ge 1 - \delta$, so with this probability MsrDist outputs an $(1-\varepsilon)$-dominator.

3 Exhausting Virtual Version Spaces

This section presents two algorithms that saves membership queries than one in the previous section. One algorithm is distribution-free and exhausts virtual version spaces, while another works under the uniform distribution and exhausts an r-universal.

Intuitively, a version space is the set of hypotheses under consideration in inductive learning. Haussler [9] proposed to exhaust the version space by throwing the hypotheses away that are inconsistent with drawn examples until only accurate hypotheses may remain. In this section, for a set V of variables,

Definition 7. *the* virtual version space *on V is the set of Boolean functions h with* $\mathrm{rel}(h) \subseteq V$.

Let V be the set of already found relevant variables. Then virtual version spaces may expand at each moment that a new relevant variable is discovered and added to V. Unless V is an $(1-\varepsilon)$-dominator, exhausting the virtual version space on V is shown to provide a witness for relevance outside of V with high probability.

Lemma 3. *Let m be any integer such that $(1-\varepsilon)^m 2^{2^r} \leq \delta$. Draw a sample O of m random examples. If V is not an $(1-\varepsilon)$-dominator then there exists a witness (A, B) for relevance outside of V with $(A, f(A))$,$(B, f(B)) \in O$ with probability at least $1-\delta$.*

Proof. Suppose that V is not an $(1-\varepsilon)$-dominator. Then for every function h in the virtual version space on V, $h(A) \neq f(A)$ holds with probability greater than ε, so O is consistent with h with probability $< (1-\varepsilon)^m$. The union bound thus implies that O does not provide any witness for relevance outside of V with probability $< (1-\varepsilon)^m 2^{2^r} \leq \delta$.

We now implement this lemma in an algorithm ExhVVS that finds an $(1-\varepsilon)$-dominator. For each new example $(A, f(A))$, ExhVVS checks over the old examples $(B, f(B))$ in a stock that whether (A, B) is a witness for relevance outside of V. If ExhVVS finds a new witness it discards all the old examples and make the stock empty.

Procedure ExhVVS

Input: Integers n and r.

Output: A set of relevant variables.

Set $O := \emptyset$, $V := \emptyset$, $s := 1$ and $m := 0$.

Initialize $m_0 :=$ the minimum integer such that $(1-\varepsilon)^{m_0} 2^2 \geq \delta/r$.

Do until either $|V| = r$ or $m = m_0$

Draw a random example $(A, f(A))$.

For each $B \in O$ Do

If $A_V = B_V$ and $f(A) \neq f(B)$ then Do

Update $O := \emptyset$, $m := 0$ and $s := s+1$.

Update $m_0 :=$ the minimum integer such that $(1-\varepsilon)^{m_0} 2^{2^s} \geq \delta/r$.

Execute the binary search on (A, B) and put an obtained variable into V.

EndDo

EndDo

Put $(A, f(A))$ into O and increment m by 1.

EndDo

Output V.

EndProc

Theorem 2. *The procedure ExhVVS outputs an $(1-\varepsilon)$-dominator (a set of relevant variables) with probability greater than $1-\delta$. ExhVVS uses at most $m_0 = O(r2^r \log(1/\varepsilon)\log(r/\delta))$ random examples, $r \log n$ membership queries and runs in $O(n(m_0 + r\log n))$ time.*

Proof. If $|V| = s$ then ExhVVS draws m_0 random examples so that $(1-\varepsilon)^{m_0} 2^{2^s} \geq \delta/r$. Therefore, due to Lemma 3, if V is not an $(1-\varepsilon)$-dominator then it derives a witness for relevance outside of V with probability at least δ/r. The remaining argument is the same with the proof of Theorem 1.

Under the uniform distribution, a modification of ExhVVS can find all the relevant variables by exhausting an r-universal set.

Definition 8. *A set U of instances is called r-universal if every r-bits on every set of r variables occurs in some instance A in U.*

Damaschke [7] worked in the exact learning model with only membership queries and studied the numbers of (adaptive and non-adaptive) membership queries for exhausting r-universal set. Here we show that, under the uniform distribution, an enough number of random examples provides an r-universal set and that any of those sets is a sufficient source for witnesses for all relevant variables.

Lemma 4. *Let m be any integer such that $(1-2^{-r})^m 2^r \binom{n}{r} \leq \delta$. Draw m instances independently and randomly under the uniform distribution over the n-bit instance space. Then it forms an r-universal set with probability at least $1-\delta$.*

Proof. We say that the sample hits an r-bits on a set of r variables if an assignment A of some example $(A, f(A))$ in the sample sets those variables to those bits. Then, due to probabilistic independence of examples, the sample does not hit a given r-bits with probability $(1-2^{-r})^m$. Since there are $2^r\binom{n}{r}$ possibility of such r-bits's, the union bound implies that the sample does not hit some r-bits with probability at most $(1-2^{-r})^m 2^r\binom{n}{r} \leq \delta$. Or equivalently, the sample hits every r-bits with probability at least $1-\delta$.

Let ExhUniv be a modification of ExhVVS that does not discard old examples at all; ExhUniv omits updating O and m_0 in the lines 9 and 10 of the procedure ExhVVS.

Theorem 3. *Under the uniform distribution, ExhUniv finds all the relevant variables with probability at least* $1-\delta$ *by at most* $m_0 = O(r2^r \log n \log(1/\delta))$ *random examples and* $r \log n$ *membership queries in* $O(n(m_0 + r\log n))$ *time.*

Proof. Let R be the set of relevant variables and let $s = |R| \leq r$. Let m_0 satisfy $(1-2^{-s})^{m_0} 2^s\binom{n}{s} \leq \delta$. Then, due to Lemma 4, the sample of size m_0 presents an s-universal set with probability $\geq 1-\delta$. Such a sample induces every s-bits on R, so in particular ExhUniv discovers a witness (A, B) for relevance of every variable $x \in R$ such that A_R and B_R are neighbors at x, hence x itself by binary searching on (A, B).

4 Measuring Influence of Variables

An algorithm in this section measures influence of variables to the target concept under the uniform distribution and outputs highly influential ones. Such an approach for learning has been taken in many AI and data mining research papers and achieves good empirical success (see [5, Section 2.4]). Based on this approach, we will design an algorithm that finds an $(1-\varepsilon)$-dominator under the uniform distribution in time almost quadratic for r, the number of relevant variables.

Definition 9. *The influence* $\mathrm{Inf}(x)$ *of a variable* x *is the number of instances* A *as a fraction of the set of all instances such that* $f(A) \neq f(A')$ *for the neighbor instance* A' *of* A *at* x.

Therefore, $\mathrm{Inf}(x) = 0$ if and only if a variable x is irrelevant, and $\mathrm{Inf}(x) = 1$ if and only if $f = x \oplus g$ for some function g irrelevant with respect to x. To show existence of highly influential variables, Ben-Or and Linial [2] applied the edge-isoperimetric inequality (see [6, Section 16] for edge-isoperimetric inequalities).

Lemma 5 (Edge Isoperimetric Inequality). *For a given Boolean function* $f(x_1, \ldots, x_n)$ *choose* $b \in \{0,1\}$ *and* $1 \leq k$ *such that*

$$2^{-k-1} \leq \mathrm{Prob}_A\{f(A) = b\} \leq 2^{-k}$$

under the uniform distribution on A*. We then have* $\sum_{i=1}^{n} \mathrm{Inf}_f(x_i) \geq k2^{1-k}$.

In order to implement this inequality in a learning algorithm, we need to relatize it on a given set V of variables.

Lemma 6. *If V is not an $(1-\varepsilon)$-dominator then $\sum_{x\notin V} \mathrm{Inf}(x) > 2\varepsilon$.*

Proof. Let h be a Bayes optimal predictor of f on V. That is, (1) $\mathrm{rel}(h) \subseteq V$ and (2) $h(A) = 1$ if and only if $\mathrm{Prob}_B\{f(B) = 1 | B_V = A_V\} \geq 1/2$. We suppose $\sum_{x\notin V} \mathrm{Inf}(x) \leq 2\varepsilon$ and prove that h approximates f with accuracy $\geq 1-\varepsilon$.

For any instance A we let

$$p(A) = 1 - \max\{\mathrm{Prob}_B\{f(B) = 1|B_V = A_V\}, \mathrm{Prob}_B\{f(B) = 0|B_V = A_V\}\}.$$

Then expectation of $p(A)$ on A is $\mathrm{E}_A[p(A)] = \mathrm{Prob}_A\{h(A) \neq f(A)\}$, so it is enough to claim that $\mathrm{E}_A[p(A)] \leq \varepsilon$.

For any instance A let $f_{A,V}$ be the function that fixes the input-bits of f on V by A. Hence $\mathrm{rel}(f_{A,V}) \subseteq \mathrm{rel}(f) - V$. Lemma 5 then promises $2p(A) \leq \sum_{x\notin V} \mathrm{Inf}_{f_{A,V}}(x)$, so taking average on A for both sides derives

$$2E_A[p(A)] \leq \sum_{x\notin V} \mathrm{Inf}_f(x) \leq 2\varepsilon,$$

so we obtain $E_A[p(A)] \leq \varepsilon$.

Uehara et. al. [10] apply the following lemma for finding relevant variables to fundamental Boolean functions (conjunctions, parities, etc).

Lemma 7 (Uehara et. al. [10]). *Let R be any set of r elements in the n element set X. Choose a subset W of X with $|W| = n/r$ uniformly at random. Then $|V \cap W| = 1$ happens with probability $> 1/e$.*

Now, we present an algorithm that applies Lemma 6 and Lemma 7 and finds highly influential variables.

Procedure MsrInf
  Input: Integers n, r and m_0.
  Output: A set of relevant variables.
  Set $X := \{x_1, \ldots, x_n\}$, $V := \emptyset$, $m := 0$, $n' := n$ and $r' := r$.
  Do until either $r' = 0$ or $m = m_0$
    Draw a random example $(A, f(A))$ and increment m by 1.
    Choose a set of variables $W \subseteq X - V$ with $|W| = n'/r'$ uniformly at random.
    Let $\beta := \mathrm{MQ}_f(A^W)$.ppp
    If $f(A) \neq \beta$ then Do
      Update $m := 0$, $n' := n' - 1$ and $r' := r' - 1$.
      Execute the binary search on (A, A^W), put the obtained variable in V and remove the variable from X.
    EndDo
  EndDo
  Output V.
EndProc

Theorem 4. *Under the uniform distribution, MsrInf finds an* $(1-\varepsilon)$*-dominator (a set of relevant variables) with probability greater than* $1-\delta$ *by at most* $m_0 = O((r/\varepsilon)\log(r/\delta))$ *random examples and* $r(m_0+\log n)$ *membership queries in* $O(nr(m_0+\log n))$ *time.*

Proof. Fix an arbitrary $m_0 = O(\varepsilon^{-1}\log(r/\delta))$ such that $(1-\frac{\varepsilon}{er})^{m_0} \leq \delta/r$. Let R be any set of r variables containing all the relevant variables.

In view of Lemma 7, choosing W as in the procedure, we have $|(R-V)\cap W| = 1$ with probability greater than $1/e$. Moreover, if it is so, $(R-V)\cap W = \{x\}$ happens equally likely for each $x \in R-V$. Therefore, drawing A according to $\mathcal{D}$ and choosing W as in the procedure, $f(A) \neq f(A^W)$ happens with probability greater than $\frac{1}{2er}\cdot\sum_{x\notin V}\mathrm{Inf}_f(x)$. Thus if V is not an $(1-\varepsilon)$-dominator then Lemma 6 promises $\sum_{x\notin V}\mathrm{Inf}(x) > 2\varepsilon$, so (A, A^W) happens to be a witness for relevance outside of V with probability greater than $\frac{\varepsilon}{er}$.

Therefore, in each stage of the Do loop, if V is not an $(1-\varepsilon)$-dominator then a witness for relevance is successfully found with probability greater than $1-(1-\frac{\varepsilon}{er})^{m_0} \geq 1-\delta/r$. The remaining argument is the same with the proof of Theorem 1.

The procedure MsrInf enumerates variables in order of their influence to the target. For example, suppose that there are 100 relevant variables where only three of them have influence 0.1 and the other 97 have only 0.001. MseInf gets the three in precedence with the other 97 with probability $> (0.9)^3 = 0.729$.

Acknowledgments

Osamu Watanabe contributed to this research from the beginning. We would like to thank him for motivation and many technical improvements. We would also like to thank Ryuhei Uehara for his helpful comments.

References

1. D. Angluin. Queries and concept learning. *Machine Learning*, 2:319, 1987.
2. M. Ben-Or and N. Linial. Collective coin flipping. *Advances in Computing Research*, 5, 1989.
3. A. Blum, M. Furst, M.l. Kearns, and R. J. Lipton. Cryptographic primitives based on hard learning problems. In *Proc. CRYPTO 93*, pages 278–291, 1994. LNCS 773.
4. A. Blum, L. Hellerstein, and N. Littlestone. Learning in the presence of finitely or infinitely many irrelevant attributes. *Journal of Computer and System Sciences*, 50(1):32–40, 1995.
5. A.L. Blum and P. Langrey. Selection of relevant features and examples. *Machine Learning*. to be appeared.
6. B. Bollobas. *Combinatorics*. Cambridge Univ. Press, Cambridge, 1986.
7. P. Damaschke. Adaptive versus nonadaptive attribute-efficient learning. In *Proceedings of the 30th Annual ACM Symposium on Theory of Computing (STOC-98)*, pages 590–596, 1998.

8. A. Dhagat and L. Hellerstein. PAC learning with irrelevant attributes. In *Proceedings of the 35th Annual Symposium on Foundations of Computer Science*, pages 64–74, 1994.
9. D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's model. *Artificial Intelligence*, 36(2):177–221, 1988.
10. R. Uehara, K. Tsuchida, and I. Wegener. Optimal attribute-efficient learning of disjunction, parity and threshold functions. In *Proceedings of the 3rd European Conference on Computational Learning Theory*, volume 1208 of *LNAI*, pages 171–184, 1997.
11. L. G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1985.