JMLR: Workshop and Conference Proceedings vol 35:1–7, 2014

Open Problem: The Statistical Query Complexity of Learning Sparse Halfspaces

Vitaly Feldman VITALY@POST.HARVARD.EDU
IBM Research - Almaden and Simons Institute, UC Berkeley

Abstract

We consider the long-open problem of attribute-efficient learning of halfspaces. In this problem the learner is given random examples labeled by an unknown halfspace function f on $\mathbb{R}^n$. Further f is r-sparse, that is it depends on at most r out of n variables. An attribute-efficient learning algorithm is an algorithm that can output a hypothesis close to f using a polynomial in r and $\log n$ number of examples (Blum, 1992). Despite a number of attempts and some partial progress, there are no efficient algorithms or hardness results for the problem. We propose a potentially easier question: what is the query complexity of this learning problem in the statistical query (SQ) model of Kearns (1998). We show that, as in the case of general PAC learning, the query complexity of attribute-efficient SQ learning of any concept class can be characterized by a combinatorial parameter of the concept class. The proposed question is then equivalent to estimating the value of this parameter for the concept class of halfspaces. A potentially simpler problem is to estimate this parameter for the concept class of decision lists, a subclass of halfspaces.

Background

Learning of halfspaces in the presence of a large number of irrelevant attributes is one of the fundamental problems in machine learning theory and practice. In this problem we are given examples over $\mathbb{R}^n$ labeled by a halfspace $\mathsf{sign}(w \cdot x \geq \theta)$. Further the vector w is r-sparse, that is, has at most r non-zero coordinates. We are interested in the common case when r is much smaller than n, for example $r = \log n$. It is well known that the VC dimension of all r-sparse halfspaces is $O(r \log n)$.Therefore r-sparse halfspaces can be PAC learned using $O(r \log n/\epsilon)$ examples to accuracy $1 - \epsilon$, although the best known algorithm requires $n^{\Theta(r)}$ time steps. Given $O(n/\epsilon)$ examples this learning task becomes easy since general halfspaces are learnable efficiently with that many examples in polynomial time. The question of whether efficient learning if possible with fewer examples is one of the long standing open problems in learning theory (Blum, 1992; Blum and Langley, 1997). This problem is also interesting in the general context of problems exhibiting an intriguing gap between the number of samples sufficient to solve the problem information-theoretically and the number of samples sufficient to solve the problem efficiently. Such problems have been studied in the past and have attracted renewed attention more recently Decatur et al. (1999); Servedio (2000); Feldman (2007); Shalev-Shwartz et al. (2012); Berthet and Rigollet (2013); Daniely et al. (2013).

Learning in the presence of a large number of irrelevant attributes was explicitly formalized by Avrim Blum as *attribute-efficient* learning (1992). A PAC learning algorithm[1] for a class of concepts $\mathcal{C}$ is said to be attribute-efficient if the number of examples it uses is polynomial in the VC dimension of $\mathcal{C}$. For most classes studied this is equivalent to being polynomial in $r \log n$, where

[1] This definition was originally stated in the context of online mistake-bound model (Littlestone, 1987).

r is the number of variables that can influence the value of the target concept $f \in \mathcal{C}$. In further discussion, for simplicity we restrict our attention to learning over $\{0,1\}^n$ domain for which most of prior work is stated.

Littlestone's (1987) seminal Winnow algorithm leads directly to an algorithm that learns r-sparse halfspaces using $O(\log n/(\epsilon\gamma_W^2))$ examples, where γ_W is the $\ell_{\infty,1}$ margin of the target halfspace on the input distribution ($1/\gamma_W$ is also equal to the smallest total weight of an integer weight representation of f). In particular, this leads to attribute-efficient learning whenever the margin is at least $1/\text{poly}(r)$. This is the case, for example, for disjunctions and majorities of r variables (disjunctions can also be learned attribute-efficiently via an algorithm of Haussler (1988)). Unfortunately, in general, γ_W of an r-sparse halfspaces over $\{0,1\}^n$ can be as low as $r^{-\Omega(r)}$ (Håstad, 1994) and thus for $r \geq \log n$ this approach does not improve on the trivial $O(n/\epsilon)$ sample complexity bound.

Partial progress has been made on an important special case of decision lists which have margin as low as $2^{-\Omega(r)}$. By representing decision lists as polynomial threshold functions and using the Winnow algorithm, Klivans and Servedio (2006) gave an algorithm that uses $2^{\tilde{O}(r^{1/3})} \log n/\epsilon$ examples and runs in time $n^{\tilde{O}(r^{1/3})}/\epsilon$. Their approach also gives other points in the trade-off between the running time and the number of examples and was strengthened in a recent work of Servedio et al. (2012). Nearly tight lower bounds are known for this approach and its generalizations (see App. A). For the case when the distribution over the inputs is uniform (or sufficiently close to it) Long and Servedio (2006) show that decision lists are learnable attribute-efficiently in polynomial time and halfspaces are learnable attribute-efficiently albeit with worse $2^{O(1/\epsilon^2)}$ dependence on the accuracy parameter.

Attribute-efficient Statistical Query Learning

We consider learning of sparse halfspaces in the statistical query (SQ) model of Kearns (1998). In this model the learning algorithm has access to statistical queries instead of random examples. An SQ oracle for input distribution $\mathcal{D}$ and target function f provides answers to statistical queries. A query is given by a bounded function of an example $\phi : \{0,1\}^n \times \{-1,1\} \to [-1,1]$ and the oracle responds with some value v that satisfies $|v - \mathbf{E}_{x\sim\mathcal{D}}[\phi(x, f(x))]| \leq \tau$. Here τ is referred the *tolerance* of the query. A valid answer to a query of tolerance τ can be found with probability $1-\delta$ using $O(\log(1/\delta)/\tau^2)$ random examples and, naturally, $\Omega(1/\tau^2)$ examples are in general necessary to obtain such an estimate with probability $\geq 1/2$. Therefore tolerance of the query corresponds to the number of examples available to the learning algorithm (in (Feldman et al., 2013) a variant of the SQ oracle is described that makes this correspondence explicit and tight). Given the correspondence above, a natural way to define attribute-efficient SQ learning of $\mathcal{C}$ is as learning in which tolerance of SQs used by the algorithm is lower bounded by the inverse of a polynomial in the VC dimension of $\mathcal{C}$.

Essentially all known upper bounds for PAC learning also hold for the SQ model up to, possibly, polynomial factors (with learning of parity functions via Gaussian elimination being the only known exception). The same holds for the known attribute-efficient upper bounds that we are aware of (without the exception since there is no attribute-efficient version of Gaussian elimination). In the most important case of the Winnow algorithm (and its use in expanded feature spaces) this can be easily derived either by analyzing the Winnow algorithm directly or by using the boosting-based approach for learning r-sparse halfspaces of Jackson and Craven (1996) instead (it has the same dependence on γ_W).

SQ complexity

An important property of the SQ model is that the query complexity of SQ learning $\mathcal{C}$ over a distribution $\mathcal{D}$ can be characterized via a geometric property of $\mathcal{C}$ and $\mathcal{D}$. For the case of weak PAC learning this was first shown by Blum et al. (1994) and has since been extended to many other settings (*e.g.* (Feldman, 2012)). A lower bound on the query complexity of the SQ algorithm gives a lower bound on its running time. Remarkably, for all known "natural" concept classes their query complexity is the same (up to polynomial factors) as the running time of the SQ learning algorithm (see (Feldman and Kanade, 2012) for a more detailed discussion). Therefore both upper and lower bounds on the SQ complexity can shed light on computational complexity of the learning problem.

To characterize the complexity of attribute-efficient SQ learning we define the following generalization of SQ-DIM in (Blum et al., 1994).

Definition 1 *For $\gamma > 0$, a class of $\{-1,1\}$-valued functions $\mathcal{C}$ and a distribution $\mathcal{D}$ over some domain X we say that SQ-DIM$(\mathcal{C},\mathcal{D},\gamma) = d$ if d is the largest such that there exist d functions $f_1,\ldots,f_d \in \mathcal{C}$ such that for every $1 \leq i \neq j \leq d$, $|\mathbf{E}_{x\sim\mathcal{D}}[f_i(x)f_j(x)]| \leq \gamma$. We define SQ-DIM$(\mathcal{C},\gamma) = \max_{\mathcal{D}}\{$SQ-DIM$(\mathcal{C},\mathcal{D},\gamma)\}$.*

We prove (in App. B) the following generalization of results in (Blum et al., 1994).

Theorem 2 *If SQ-DIM$(\mathcal{C},\mathcal{D},\gamma) = d$ then any SQ algorithm that learns $\mathcal{C}$ over $\mathcal{D}$ using queries of tolerance $\tau \geq \sqrt{\gamma/2}$ and outputs a hypothesis with accuracy $\geq 1/2+\tau/2$ needs at least $d\tau^2/2-1$ statistical queries. Further, $\mathcal{C}$ can be SQ learned over $\mathcal{D}$ to accuracy $1/2+\gamma/4$ using at most d queries of tolerance $\gamma/8$.*

If SQ-DIM$(\mathcal{C},\gamma) = d$ and the learning algorithm has access to unlabeled samples in addition to SQs then the weak learner above can be converted to a *distribution-independent* learner with accuracy $1-\epsilon$ via standard results in hypothesis boosting. When used with a *smooth* boosting algorithm such as (Servedio, 2003; Feldman, 2010) this approach uses $O(d\log(1/\epsilon)/\gamma^2)$ queries of tolerance $\Omega(\gamma\epsilon)$ and poly$(d, 1/\epsilon, 1/\gamma)$ unlabeled examples. The unlabeled examples are used to obtain an empirical approximation $\hat{\mathcal{D}}_i$ to each distribution $\mathcal{D}_i$ produced by the boosting algorithm. This is needed to find a maximal set or nearly uncorrelated f_i's for $\mathcal{D}_i$ (formal details are easy to fill in and are omitted in this note).

Definition 1 and Theorem 2 can be extended to learning with accuracy $1-\epsilon$ using the results in (Feldman, 2012). However, for the problem of learning r-sparse halfspaces even weak learning appears to be hard and therefore understanding this simpler dimension is a natural starting point. Given the definitions and the characterization we can pose our questions formally:

Problem 3 *Let $\mathcal{H}_r$ denote the class of r-sparse halfspaces over $\{0,1\}^n$. Does there exist $\gamma = (r\log n)^{-O(1)}$ such that SQ-DIM$(\mathcal{H}_r,\gamma) = n^{O(1)}$? We also ask the same question for the class of length-r decision lists.*

Weaker non-trivial results about the trade-off between γ and SQ-DIM$(\mathcal{H}_r,\gamma)$ are also of interest.

Acknowledgments

The author thanks Sasha Sherstov and Greg Valiant for sharing many thoughtful ideas and discussions about this problem.

References

Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *COLT*, pages 1046–1066, 2013.

A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

A. Blum, M. Furst, J. Jackson, M. Kearns, Y. Mansour, and S. Rudich. Weakly learning DNF and characterizing statistical query learning using Fourier analysis. In *Proceedings of STOC*, pages 253–262, 1994.

Avrim Blum. Learning boolean functions in an infinite attribute space. *Machine Learning*, 9:373–386, 1992.

H. Buhrman, N. Vereshchagin, and R. de Wolf. On computation and communication with small bias. In *Proceedings of IEEE Conference on Computational Complexity*, pages 24–32, 2007.

Amit Daniely, Nati Linial, and Shai Shalev-Shwartz. More data speeds up training time in learning halfspaces over sparse vectors. In *NIPS*, pages 145–153, 2013.

S. Decatur, O. Goldreich, and D. Ron. Computational sample complexity. *SIAM Journal on Computing*, 29(3):854–879, 1999.

V. Feldman. Attribute efficient and non-adaptive learning of parities and DNF expressions. *Journal of Machine Learning Research*, (8):1431–1460, 2007.

V. Feldman. Evolvability from learning algorithms. In *Proceedings of STOC*, pages 619–628, 2008.

V. Feldman. Distribution-specific agnostic boosting. In *Proceedings of Innovations in Computer Science*, pages 241–250, 2010.

V. Feldman. A complete characterization of statistical query learning with applications to evolvability. *Journal of Computer System Sciences*, 78(5):1444–1459, 2012.

Vitaly Feldman and Varun Kanade. Computational bounds on statistical query learning. In *COLT*, pages 16.1–16.22, 2012.

Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for planted clique. In *STOC*, pages 655–664. ACM, 2013.

M. Goldmann, J. Håstad, and A. Razborov. Majority gates vs. general weighted threshold gates. *Computational Complexity*, 2:277–300, 1992.

J. Håstad. On the size of weights for threshold gates. *SIAM Journal on Discrete Mathematics*, 7(3): 484–492, 1994.

D. Haussler. Quantifying inductive bias: AI learning algorithms and Valiant's learning framework. *Artificial Intelligence*, 46(2):177–221, 1988.

J. Jackson and M. Craven. Learning sparse perceptrons. In *Advances in Neural Information Processing Systems 8*, pages 654–660, 1996.

M. Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM*, 45(6): 983–1006, 1998.

Adam R. Klivans and Rocco A. Servedio. Toward attribute efficient learning of decision lists and parities. *Journal of Machine Learning Research*, 7:587–602, 2006.

N. Littlestone. Learning quickly when irrelevant attributes abound: a new linear-threshold algorithm. *Machine Learning*, 2:285–318, 1987.

Philip M. Long and Rocco A. Servedio. Attribute-efficient learning of decision lists and linear threshold functions under unconcentrated distributions. In *NIPS*, pages 921–928, 2006.

R. Servedio. Computational sample complexity and attribute-efficient learning. *Journal of Computer and System Sciences*, 60(1):161–178, 2000.

R. Servedio. Smooth boosting and learning with malicious noise. *Journal of Machine Learning Research*, 4:633–648, 2003. ISSN 1533-7928.

Rocco A. Servedio, Li-Yang Tan, and Justin Thaler. Attribute-efficient learning and weight-degree tradeoffs for polynomial threshold functions. In *COLT*, pages 14.1–14.19, 2012.

Shai Shalev-Shwartz, Ohad Shamir, and Eran Tromer. Using more data to speed-up training time. In *AISTATS*, pages 1019–1027, 2012.

Alexander A. Sherstov. Halfspace matrices. *Computational Complexity*, 17(2):149–178, 2008.

B. Szörényi. Characterizing statistical query learning:simplified notions and proofs. In *Proceedings of ALT*, pages 186–200, 2009.

Andrew Yao. Probabilistic computations: Toward a unified measure of complexity. In *FOCS*, pages 222–227, 1977.

Appendix A. Lower Bounds for Generalized Winnow

Essentially the only lower bounds known for learning sparse halfspaces are for the technique of Klivans and Servedio (2006) and its generalizations. The essence of this technique is using the Winnow algorithm over a more general feature space and it relies on representation of functions in $\mathcal{C}$ as low total integer weight halfspaces over the new feature space (recall that the total integer weight is the inverse of γ_W). In (Klivans and Servedio, 2006; Servedio et al., 2012) this feature space consists of k-disjunctions for some $k \leq r$. They give lower bounds for the approach when it is used with this feature space that essentially match their upper bounds. It was observed by Sherstov (2008) that communication complexity lower bounds of Goldmann et al. (1992) imply that representation of all r-dimensional halfspaces over $\{0,1\}^r$ requires either $2^{\Omega(r)}$ features or $2^{\Omega(r)}$ integer weight. Similarly, results of Buhrman et al. (2007) imply analogous lower bounds (with $r^{1/3}$ in place of r) for decision lists (see (Feldman, 2008; Servedio et al., 2012) for more details). This means that this approach to learning r-sparse halfspaces requires either $n^{\Omega(r)}$ features (and hence time) or $2^{\Omega(r)}$ examples.

Appendix B. Proof of Theorem 2

Proof Let $\mathcal{A}$ be a statistical algorithm that uses q queries of tolerance $\tau \geq \sqrt{\gamma/2}$ to learn $\mathcal{C}$ over $\mathcal{D}$. Using the decomposition of a query function into a correlation and target-independent parts (*e.g.* Feldman, 2008) we can assume that all queries of $\mathcal{A}$ are of the form $\mathbf{E}_{x\sim\mathcal{D}}[f(x)g(x)]$ where $g : X \to [-1,1]$ is a bounded function. We simulate $\mathcal{A}$ by answering any query $g : X \to [-1,1]$ of $\mathcal{A}$ with value 0. Let $g_1, g_2, \ldots, g_q$ be the queries asked by $\mathcal{A}$ in this simulation and let g_{q+1} be the output hypothesis of $\mathcal{A}$.

For real-valued functions g, h over X we use the following inner product $\langle g, h\rangle_D = \mathbf{E}_{x\sim\mathcal{D}}[g(x)h(x)]$ and let $\|g\|_D^2 = \langle g, g\rangle$ be the associated norm. By the definition of SQ-DIM, there exists a set of d functions $\{f_1, \ldots, f_d\} \subseteq \mathcal{C}$ such that for every $i \neq j \leq d$, $|\langle f_i, f_j\rangle_D| \leq \gamma$. In the rest of the proof for conciseness we drop the subscript D from inner products and norms.

To lower bound q, we use a generalization of an elegant argument of Szörényi (2009). For every $k \in [q+1]$ let $A_k \subseteq [d]$ be the set of indices i such that $|\langle f_i, g_k\rangle| > \tau$. To prove the desired bound we prove that following two claims:

1. $\sum_{k\in[q+1]} |A_k| \geq d$;
2. for every $k \in [q+1]$, $|A_k| \leq 2/\tau^2$.

Combining these two immediately implies the desired bound $q \geq d\tau^2/2 - 1$.

To prove the first claim we assume, for the sake of contradiction, that there exists $i \in [d] \setminus (\cup_{k\in[q+1]} A_k)$. Then for every $k \in [q+1]$, $|\langle f_i, g_k\rangle| \leq \tau$. This implies that the replies of our simulation are within τ of $\langle f_i, g_k\rangle$. By the definition of $\mathcal{A}$, this implies that $\mathbf{Pr}_{\mathcal{D}}[g_{q+1}(x) = f_i(x)] \geq 1/2 + \tau/2$ or $\langle f_i, g_{q+1}\rangle \geq \tau$. This contradicts the condition that $i \notin A_{q+1}$.

To prove the second claim we consider upper and lower bounds on the following quantity:

$$\left\langle g_k, \sum_{i\in A_k} f_i \cdot \mathsf{sign}\langle g_k, f_i\rangle \right\rangle .$$

By Cauchy-Schwartz we have that

$$\begin{aligned}\left\langle g_k, \sum_{i\in A_k} f_i \cdot \mathsf{sign}\langle g_k, f_i\rangle \right\rangle^2 &\leq \|g_k\|^2 \cdot \left\| \sum_{i\in A_k} f_i \cdot \mathsf{sign}\langle g_k, f_i\rangle \right\|^2 \\ &\leq \left(\sum_{i,j\in A_k} |\langle f_i, f_j\rangle| \right) \\ &\leq |A_k| + \gamma \cdot (|A_k|^2 - |A_k|).\end{aligned}$$

We also have that

$$\left\langle g_k, \sum_{i\in A_k} f_i \cdot \mathsf{sign}\langle g_k, f_i\rangle \right\rangle = \sum_{i\in A_k} |\langle g_k, f_i\rangle| \geq |A_k|\tau.$$

By combining these two bounds we obtain that

$$|A_k| + \gamma \cdot (|A_k|^2 - |A_k|) \geq |A_k|^2\tau^2.$$

Using the condition $\tau \geq \sqrt{\gamma/2}$ we obtain that $|A_k| \geq |A_k|^2\tau^2/2$ and therefore $|A_k| \leq 2/\tau^2$.

We remark that for simplicity we assumed that $\mathcal{A}$ is deterministic. The argument can be easily generalized to randomized algorithms by considering the success probability on a randomly and uniformly chosen f_i and applying Yao's minimax principle (1977). More details can be found in the analogous argument in (Feldman et al., 2013).

For the other direction of this theorem the algorithm simply finds which of f_i's has the largest correlation with the target function using SQs of tolerance $\gamma/8$. Denote it by h. It then outputs h (if the correlation is positive) or $-h$ (if the correlation is negative) as a hypothesis. It is easy to see that this gives the desired weak learning algorithm. ■