Trustworthy Machine Learning

Statistical Query

Sangdon Park

POSTECH

Statistical Queries (SQ)

Efficient Noise-Tolerant Learning From Statistical Queries

Michael Kearns *
AT&T Laboratories – Research
Florham Park, New Jersey

June 15, 1998

1 Introduction

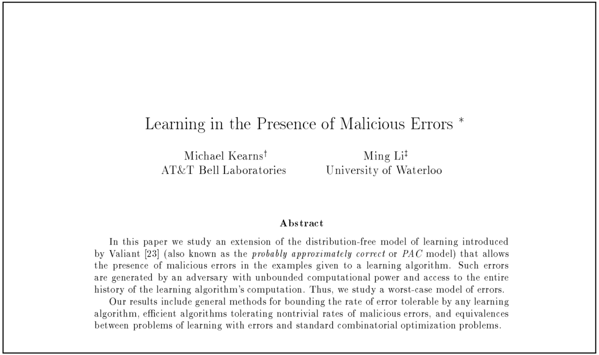
In this paper, we study the extension of Valiant's learning model [32] in which the positive or negative classification label provided with each random example may be corrupted by random noise. This extension was first examined in the learning theory literature by Angluin and Laird [1], who formalized the simplest type of white label noise and then sought algorithms tolerating the highest possible rate of noise. In addition to being the subject of a number of theoretical studies [1, 22, 31, 17], the classification noise model has become a common paradigm for experimental machine learning research.

Angluin and Laird provided an algorithm for learning boolean conjunctions that tolerates a noise rate approaching the information-theoretic barrier of 1/2. Subsequently, there have been some isolated instances of efficient noise-tolerant algorithms [20, 27, 29], but little work on characterizing which classes can be efficiently learned in the presence of noise, and no general transformations of Valiant model algorithms into noise-tolerant algorithms. The primary contribution of the present paper is in making significant progress in both of these areas.

We identify and formalize an apparently rather weak sufficient condition on learning algorithms in Valiant's model that permits the immediate derivation of noise-tolerant learning algorithms. More precisely, we define a natural restriction on Valiant model algorithms that allows them to be reliably and efficiently simulated in the presence of arbitrarily large rates of classification noise. This allows us to obtain efficient noise-tolerant learning algorithms for practically every concept class for which an efficient learning algorithm in the

*Author's address: AT&T Laboratories – Research, Room A235, 180 Park Avenue, Florham Park, New Jersey 07932. Electronic mail address: mkearns@research.att.com.

Learning with Noise



Learning in the Presence of Malicious Errors *

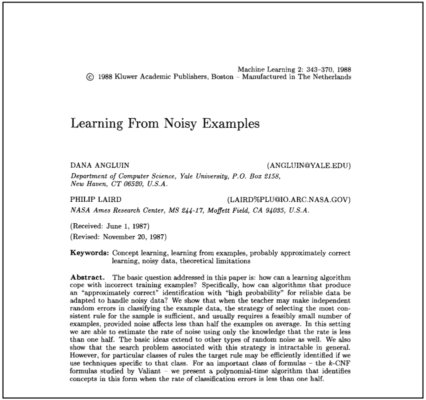
Michael Kearns† AT&T Bell Laboratories

Ming Li‡ University of Waterloo

Abstract

In this paper we study an extension of the distribution-free model of learning introduced by Valiant [23] (also known as the *probably approximately correct* or *PAC* model) that allows the presence of malicious errors in the examples given to a learning algorithm. Such errors are generated by an adversary with unbounded computational power and access to the entire history of the learning algorithm's computation. Thus, we study a worst-case model of errors.

Our results include general methods for bounding the rate of error tolerable by any learning algorithm, efficient algorithms tolerating nontrivial rates of malicious errors, and equivalences between problems of learning with errors and standard combinatorial optimization problems.



Machine Learning 2: 343–370, 1988

Learning From Noisy Examples

DANA ANGLUIN (ANGLUIN@YALE.EDU)
Department of Computer Science, Yale University, P.O. Box 2158, New Haven, CT 06520, U.S.A.

PHILIP LAIRD (LAIRD%PLU@IO.ARC.NASA.GOV)
NASA Ames Research Center, MS 244-17, Moffett Field, CA 94035, U.S.A.

(Received: June 1, 1987)

(Revised: November 20, 1987)

Keywords: Concept learning, learning from examples, probably approximately correct learning, noisy data, theoretical limitations

Abstract. The basic question addressed in this paper is: how can a learning algorithm cope with incorrect training examples? Specifically, how can algorithms that produce an "approximately correct" identification with "high probability" for reliable data be adapted to handle noisy data? We show that when the teacher may make independent random errors in classifying the example data, the strategy of selecting the most consistent rule for the sample is sufficient, and usually requires a feasibly small number of examples, provided noise affects less than half the examples on average. In this setting we are able to estimate the rate of noise using only the knowledge that the rate is less than one half. The basic ideas extend to other types of random noise as well. We also show that the search problem associated with this strategy is intractable in general. However, for particular classes of rules the target rule may be efficiently identified if we use techniques specific to that class. For an important class of formulas – the *k*-CNF formulas studied by Valiant – we present a polynomial-time algorithm that identifies concepts in this form when the rate of classification errors is less than one half.

- SQ generalizes learning with random classification noise.

What is Learning from Statistical Queries (SQ)?

- TL;DR: a generalized version of PAC learning for designing classification noise-tolerant PAC learning algorithms.
  - PAC learning: access to $\text{EX}(h^*, \mathcal{D})$ for a labeled example
  - PAC learning with random classification noise: access to $\text{EX}^{\eta}(h^*, \mathcal{D})$ for a labeled example
  - SQ: access to $\text{STAT}(h^*, \mathcal{D})$ for a statistic of labeled examples

What is Learning from Statistical Queries (SQ)?

- TL;DR: a generalized version of PAC learning for designing classification noise-tolerant PAC learning algorithms.
  - PAC learning: access to EX$(h^*, \mathcal{D})$ for a labeled example
  - PAC learning with random classification noise: access to EX$^\eta(h^*, \mathcal{D})$ for a labeled example
  - SQ: access to STAT$(h^*, \mathcal{D})$ for a statistic of labeled examples
- Main difference: access to the estimate of statistics from multiple samples, instead of one sample

What is Learning from Statistical Queries (SQ)?

- TL;DR: a generalized version of PAC learning for designing classification noise-tolerant PAC learning algorithms.
  - PAC learning: access to $\text{EX}(h^*, \mathcal{D})$ for a labeled example
  - PAC learning with random classification noise: access to $\text{EX}^{\eta}(h^*, \mathcal{D})$ for a labeled example
  - SQ: access to $\text{STAT}(h^*, \mathcal{D})$ for a statistic of labeled examples
- Main difference: access to the estimate of statistics from multiple samples, instead of one sample
- Recall adversarial examples and noisy labels

What is Learning from Statistical Queries (SQ)?

- TL;DR: a generalized version of PAC learning for designing classification noise-tolerant PAC learning algorithms.
  - PAC learning: access to $\text{EX}(h^*, \mathcal{D})$ for a labeled example
  - PAC learning with random classification noise: access to $\text{EX}^\eta(h^*, \mathcal{D})$ for a labeled example
  - SQ: access to $\text{STAT}(h^*, \mathcal{D})$ for a statistic of labeled examples
- Main difference: access to the estimate of statistics from multiple samples, instead of one sample
- Recall adversarial examples and noisy labels
- Why we have to learn? maybe useful for differential privacy and unlearning

PAC Learning with Random Classification Noise

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a PAC-learning algorithm for $\mathcal{H}$ with random classification noise if for any $\varepsilon > 0$, $\delta > 0$, $0 \leq \eta \leq \frac{1}{2}$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ separable by h^*, and for some minimum sample size n' (which depends on ε, δ, η, $\mathcal{D}$), the following holds with any sample size $n \geq n'$:

$$\mathbb{P}\left\{L(\mathcal{A}(\mathcal{S})) \leq \varepsilon\right\} \geq 1 - \delta,$$

where $\mathcal{S} := ((x_1, y_1), \ldots, (x_n, y_n))$ and $(x_i, y_i) \sim \mathrm{EX}^{\eta}(h^*, \mathcal{D})$.

- Suppose binary classification
- η: a noise rate
- $\mathrm{EX}^{\eta}(h^*, \mathcal{D})$: the noisy example oracle that randomly flips the label with η probability
- If $\eta = \frac{1}{2}$, no hope to learn.

Setup for SQ

- $\mathcal{D}$: a distribution over $\mathcal{X}$
- Binary classification, *i.e.*, $\mathcal{Y} := \{0, 1\}$

Setup for SQ

- $\mathcal{D}$: a distribution over $\mathcal{X}$
- Binary classification, *i.e.*, $\mathcal{Y} := \{0, 1\}$
- $\chi : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$: a map (*e.g.*, 0-1 loss)
- α: a tolerance parameter
- (χ, α): a statistical query

Setup for SQ

- $\mathcal{D}$: a distribution over $\mathcal{X}$
- Binary classification, *i.e.*, $\mathcal{Y} := \{0, 1\}$
- $\chi : \mathcal{X} \times \mathcal{Y} \to [-1, 1]$: a map (*e.g.*, 0-1 loss)
- α: a tolerance parameter
- (χ, α): a statistical query
- STAT$(h, \mathcal{D}) : (\chi, \alpha) \mapsto [-1, 1]$: a statistical query oracle (*i.e.*, a data source)
- $v \in [0, 1]$: The response of STAT$(h^*, \mathcal{D})$, where

$$\left| \mathop{\mathbb{E}}_{x \sim \mathcal{D}} \{\chi(x, h^*(x))\} - v \right| \leq \alpha.$$

  - Suppose that the statistical query oracle satisfies this with probability one.

Learning from SQ

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a statistical query algorithm for $\mathcal{H}$ if for any $\varepsilon > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ over $\mathcal{X}$, and for some minimum number of queries n' (which depends on $\varepsilon, \delta, \mathcal{D}$), the following holds with any number of queries $n \geq n'$:

$$L(\mathcal{A}(\mathcal{S})) \leq \varepsilon,$$

where $\mathcal{S} := (v_1, \ldots, v_n)$ and $v_i \sim \mathsf{STAT}(h^*, \mathcal{D})$.

Learning from SQ

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a statistical query algorithm for $\mathcal{H}$ if for any $\varepsilon > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ over $\mathcal{X}$, and for some minimum number of queries n' (which depends on $\varepsilon, \delta, \mathcal{D}$), the following holds with any number of queries $n \geq n'$:

$$L(\mathcal{A}(\mathcal{S})) \leq \varepsilon,$$

where $\mathcal{S} \coloneqq (v_1, \ldots, v_n)$ and $v_i \sim \mathrm{STAT}(h^*, \mathcal{D})$.

- Suppose that the statistic is efficiently computed.

Learning from SQ

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a statistical query algorithm for $\mathcal{H}$ if for any $\varepsilon > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ over $\mathcal{X}$, and for some minimum number of queries n' (which depends on $\varepsilon, \delta, \mathcal{D}$), the following holds with any number of queries $n \geq n'$:

$$L(\mathcal{A}(\mathcal{S})) \leq \varepsilon,$$

where $\mathcal{S} := (v_1, \ldots, v_n)$ and $v_i \sim \mathsf{STAT}(h^*, \mathcal{D})$.

- Suppose that the statistic is efficiently computed.
- No confidence parameter $1 - \delta$; it is required in the statistical query oracle.

Learning from SQ

Definition (simplified definition)

An algorithm $\mathcal{A}$ is a statistical query algorithm for $\mathcal{H}$ if for any $\varepsilon > 0$, $h^* \in \mathcal{H}$, and $\mathcal{D}$ over $\mathcal{X}$, and for some minimum number of queries n' (which depends on $\varepsilon, \delta, \mathcal{D}$), the following holds with any number of queries $n \geq n'$:

$$L(\mathcal{A}(\mathcal{S})) \leq \varepsilon,$$

where $\mathcal{S} := (v_1, \ldots, v_n)$ and $v_i \sim \mathsf{STAT}(h^*, \mathcal{D})$.

- Suppose that the statistic is efficiently computed.
- No confidence parameter $1 - \delta$; it is required in the statistical query oracle.
- If an algorithm is a SQ algorithm, then it is a tolerant algorithm for random classification noise.

Example: Stochastic Convex Optimization

Setup:

- $\mathcal{H}$: a set of convex functions
- $\ell(h, z)$: convex and sub-differentiable in h
- separable assumption (*i.e.*, $\mathbb{E}\{\ell(h^*, z)\} = 0$ for some $h^* \in \mathcal{H}$)

Example: Stochastic Convex Optimization

Setup:

- $\mathcal{H}$: a set of convex functions
- $\ell(h, z)$: convex and sub-differentiable in h
- separable assumption (*i.e.*, $\mathbb{E}\{\ell(h^*, z)\} = 0$ for some $h^* \in \mathcal{H}$)

Stochastic convex optimization (in learning):

$$\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim \mathcal{D}} \{\ell(h, z)\}$$

- We use the mirror descent algorithm to solve this.

Mirror Descent Algorithm

- A generalized version of the gradient descent algorithm.
- The mirror descent considers the "geometry" of optimization.

Mirror Descent Algorithm

- A generalized version of the gradient descent algorithm.
- The mirror descent considers the "geometry" of optimization.
- Example: a proximal gradient descent algorithm
  - Minimization:

$$\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim \mathcal{D}} \{\ell(h, z)\}$$

Mirror Descent Algorithm

- A generalized version of the gradient descent algorithm.
- The mirror descent considers the "geometry" of optimization.
- Example: a proximal gradient descent algorithm
  - Minimization:

$$\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim \mathcal{D}} \{\ell(h, z)\}$$

  - Minimization with linear approximation:

$$h_{t+1} \leftarrow \arg\min_{h} \left\{ \eta \underbrace{\left(\ell(h_t, z_t) + \langle \nabla \ell(h_t, z_t), h - h_t \rangle\right)}_{\text{linear approximation}} + \frac{1}{2} \underbrace{\|h - h_t\|_2^2}_{\text{regularizer}} \right\}$$

$$h_{t+1} \leftarrow \arg\min_{h} \left\{ \eta \langle \nabla \ell(h_t, z_t), h \rangle + \frac{1}{2} \|h - h_t\|_2 \right\}$$

Mirror Descent Algorithm

- A generalized version of the gradient descent algorithm.
- The mirror descent considers the "geometry" of optimization.
- Example: a proximal gradient descent algorithm
  - Minimization:

$$\min_{h \in \mathcal{H}} \mathbb{E}_{z \sim \mathcal{D}} \{\ell(h, z)\}$$

  - Minimization with linear approximation:

$$h_{t+1} \leftarrow \arg\min_{h} \left\{ \eta \underbrace{\left(\ell(h_t, z_t) + \langle \nabla \ell(h_t, z_t), h - h_t \rangle\right)}_{\text{linear approximation}} + \frac{1}{2} \underbrace{\|h - h_t\|_2^2}_{\text{regularizer}} \right\}$$

$$h_{t+1} \leftarrow \arg\min_{h} \left\{ \eta \langle \nabla \ell(h_t, z_t), h \rangle + \frac{1}{2} \|h - h_t\|_2 \right\}$$

  - This is equivalent to the conventional gradient descent algorithm, *i.e.,*

$$\eta \nabla \ell(h_t, z_t) + (h_{t+1} - h_t) = 0 \quad \Longrightarrow \quad h_{t+1} = h_t - \eta \nabla \ell(h_t, z_t)$$

Gradient Descent Algorithm with SQ

Algorithm (for $\|\cdot\|_2$):

$$\bar{h} := \frac{1}{T}\sum_{t=1}^{T} h_t \quad \text{where} \quad h_{t+1} \leftarrow \arg\min_{h} \left\{ \eta\langle \bar{g}_t, h\rangle + \frac{1}{2}\|h - h_t\|_2^2 \right\}$$

- $\bar{g}_t := \frac{1}{K}\sum_{k=1}^{K} \nabla\ell(f_t, z_t^{(k)})$
- $\mathsf{STAT}(h^*, \mathcal{D}) = \bar{g}_t$: an "$\alpha$-good" gradient value
- The above algorithm is a SQ algorithm, *i.e.*, $\mathbb{E}\{\ell(\bar{h}, z)\} \leq \varepsilon$.