Learning with Errors

Chethan Kamath

IST Austria

April 22, 2015

Table of contents

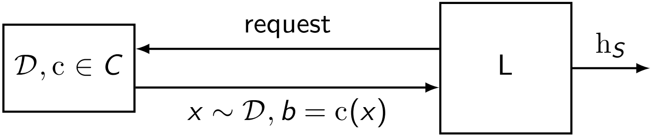
BACKGROUND

Notation

- $\mathcal{X}$: input set; $\mathcal{Y}$: binary label-set $\{0, 1\}$
- $\mathcal{D}$: distribution on the input set
- χ, η: distribution of the noise
- C: concept class, c: target concept
- $\mathrm{R}(\mathrm{h})$: generalisation error for a hypothesis h

$$\mathrm{R}(\mathrm{h}) := \mathbb{P}_{x \sim \mathcal{D}}(\mathrm{h}(x) \neq \mathrm{c}(x))$$

PAC Model



$\mathcal{D}, \mathrm{c} \in C$
request
$x \sim \mathcal{D}, b = \mathrm{c}(x)$
L
h_S

PAC Model

- Definition[1] A concept class C is called PAC-learnable if there exists an algorithm L and a function $q_0 = q_0(\epsilon, \delta)$ s.t. for any
  1. $\epsilon > 0$ (accuracy: approximately correct)
  2. $\delta > 0$ (confidence: probably)
  3. distribution $\mathcal{D}$ on $\mathcal{X}$
  4. target concept $\mathrm{c} \in C$

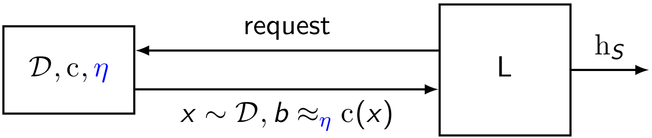
  outputs a hypothesis $\mathrm{h}_S \in C$ s.t. for any sample size $q \geq q_0$:

$$\mathbb{P}_{S \sim \mathcal{D}^q}(\mathrm{R}(\mathrm{h}_S) \leq \epsilon) \geq (1 - \delta)$$

- If L runs in poly$(1/\epsilon, 1/\delta)$-time, C is efficiently PAC-learnable
- Distribution-free

[1] Valiant, 1984

Noisy-PAC Model



Noisy-PAC Model

- Definition[2] A concept class C is efficiently learnable in presence of random classification noise if there exists an algorithm L and a function $q_0 = q_0(\epsilon, \delta)$ s.t. for any
  1. $\epsilon > 0$ (accuracy: approximately correct)
  2. $\delta > 0$ (confidence: probably)
  3. distribution $\mathcal{D}$ on $\mathcal{X}$
  4. target concept $\mathrm{c} \in C$

  and fixed noise-rate $\eta < 1/2$ outputs a hypothesis $\mathrm{h}_S \in C$ s.t. for any sample size $q \geq q_0$:

  $$\mathop{\mathbb{P}}_{S \sim \mathcal{D}^q}(\mathrm{R}(\mathrm{h}_S) \leq \epsilon) \geq (1 - \delta)$$

  and L runs in poly$(1/\epsilon, 1/\delta)$-time

[2] Angluin and Laird, 1998

LEARNING PARITY WITH NOISE

The Parity Function: Definition

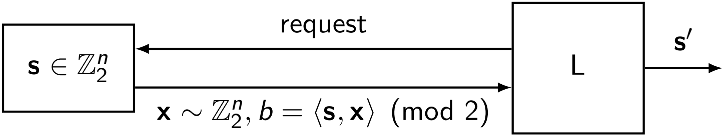
- Denoted by $\mathrm{f}_{\mathbf{s}}$, where $\mathbf{s} \in \mathbb{Z}_2^n$ determines it
- The value of the function is given by the rule

$$\mathrm{f}_{\mathbf{s}}(\mathbf{x}) := \langle \mathbf{s}, \mathbf{x} \rangle \pmod 2$$

- $C := \{\mathrm{f}_{\mathbf{s}} : \mathbf{s} \in \mathbb{Z}_2^n\}$ and $|C| = 2^n$

- Restricted parity function: $\mathrm{f}_{\mathbf{s}}$ depends on only the first k bits if all non-zero components of $\mathbf{s}$ lies in the first k bits

Learning the Parity Function



Find $\mathbf{s}$, given

$$\langle \mathbf{s}, \mathbf{x}_1 \rangle = b_1 \pmod 2$$
$$\vdots$$
$$\langle \mathbf{s}, \mathbf{x}_q \rangle = b_q \pmod 2$$

where $\mathbf{s} \in \mathbb{Z}_2^n$, $\mathbf{x}_i \sim \mathbb{Z}_2^n$ ($\mathcal{D}$=uniform), $b_i \in \mathbb{Z}_2$ and $q \in \mathrm{poly}(n)$

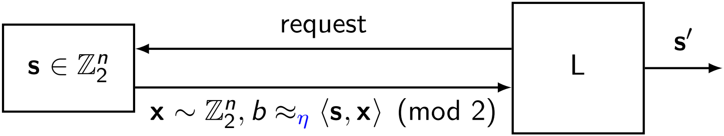
It is possible to learn $\mathbf{s}$ using $\mathrm{O}(n)$ samples and $\mathrm{poly}(n)$ time:

Gaussian elimination

Learning for arbitrary $\mathcal{D}$ possible[3]

[3] Helmbold *et al.*, 1992

Learning Parity with Noise



Find $\mathbf{s}$, given

$$\begin{aligned} \langle \mathbf{s}, \mathbf{x}_1 \rangle &\approx_\eta b_1 \pmod 2 \\ \langle \mathbf{s}, \mathbf{x}_2 \rangle &\approx_\eta b_2 \pmod 2 \\ &\vdots \\ \langle \mathbf{s}, \mathbf{x}_q \rangle &\approx_\eta b_q \pmod 2 \end{aligned}$$

where $\mathbf{s} \in \mathbb{Z}_2^n$, $\mathbf{x}_i \sim \mathbb{Z}_2^n$, $b_i \in \mathbb{Z}_2$, $q \in \text{poly}(n)$ and $\eta < 1/2$

Let $\mathcal{A}_{\mathbf{s},\chi}$ denote this distribution

Hardness of LPN: Intuition

- Consider applying Gaussian elimination to the noisy samples to find the first bit
  - Find $S \subset [q]$ s.t. $\sum_{i \in S} \mathbf{x}_i = (1, 0, \ldots, 0)$
  - But the noise is amplified: solution correct only with probability $1/2 + 2^{-\Theta(n)}$
  - Therefore, the procedure needs to be repeated $2^{\Theta(n)}$ times

- Alternative: maximum likelihood estimation of $\mathbf{s}$ using $\mathrm{O}(n)$ samples and $2^{\mathrm{O}(n)}$ time

Hardness of LPN

- Statistical Query[4] Model: the learning algorithm has access to statistical queries, that is instead of the label, it get the probability of a property holding for the particular example
- C is learnable in SQ-model imples it is learnable in the Noisy-PAC model
- LPN: Hard to learn efficiently in the SQ-model

[4] Kearns, 1998

BKW ALGORITHM

Overview

- Best known algorithm for LPN
- Solves LPN in time $\mathrm{O}(2^{n/\log n})$

- "Block-wise" Gaussian elimination
- Works by iterative "zeroising"

- Focus: LPN on uniform distribution; algorithm works for arbitrary distributions

Setting

- Two parameters: a and b s.t. $n \geq ab$
- Each sample is partitioned into a blocks of size b. That is, a sample, $\mathbf{x} = x_1, \ldots, x_n \in \mathbb{Z}_2^n$ is split as

$$\underbrace{x_1, \ldots, x_b}_{\text{block } 1} \cdots \underbrace{x_{b(i-1)+1}, \ldots, x_{b(i-1)+b}}_{\text{block } i} \cdots \underbrace{x_{k-b}, \ldots, x_n}_{\text{block } a}$$

- Definition: V_i, i-sample

  V_i: the subspace of $\mathbb{Z}_2^{ab}$ consisting of those vectors whose last i blocks have all bits equal to zero
  i-sample of size s: a set of s vectors independently and uniformly distributed over V_i.

Example: 1-sample

$$\underbrace{x_1, \ldots, x_b}_{\text{block } 1} \cdots \underbrace{x_{b(i-1)+1}, \ldots, x_{b(i-1)+b}}_{\text{block } i} \cdots \underbrace{0, 0, \ldots, 0}_{\text{block } a}$$

Main Theorem

Theorem[5] LPN can be solved with a sample-size and total computation time $\mathrm{poly}((\frac{1}{1-2\eta})^{2^a}, 2^b)$.

Corollary LPN for constant noise-rate $\eta < 1/2$ can be solved with sample-size and total computation time $2^{\mathrm{O}(n/\log n)}$.

Proof: Plug in $a = (\log n)/2$ and $b = 2n/\log n$

[5] Blum *et al.*, 2003

Zeroising

Input: i-samples $\mathbf{x}_1, \ldots, \mathbf{x}_s$

Output: $(i+1)$-samples $\mathbf{u}_1, \ldots, \mathbf{u}_{s'}$

Zeroise$_i(\mathbf{x}_1, \ldots, \mathbf{x}_s)$.

1. Partition $\mathbf{x}_1, \ldots, \mathbf{x}_s$ based on the values in block $a - i$
2. For each partition p pick a vector $\mathbf{x}_{j_p}$ at random
3. Zeroise by $\mathbf{x}_{j_p}$ to each of the other vectors in the partition
4. Return the resulting vectors $\mathbf{u}_1, \ldots, \mathbf{u}_{s'}$

Lemma

1. $\mathbf{u}_1, \ldots, \mathbf{u}_{s'}$ are $(i+1)$-samples with $s' \geq s - 2^b$
2. Each vector in $\mathbf{u}_1, \ldots, \mathbf{u}_{s'}$ is written as the sum of two vectors in $\mathbf{x}_1, \ldots, \mathbf{x}_s$
3. The run-time $\mathrm{O}(s)$

Main Algorithm

Input: s labelled examples $(\mathbf{x}_1, b_1), \ldots, (\mathbf{x}_s, b_s)$

Output: set $S \subset [s]$ s.t. $\sum_{i \in S} \mathbf{x}_i = (1, 0, \ldots, 0)$

Solve($\mathbf{x}_1, \ldots, \mathbf{x}_s$):

1. For $i = 1, \ldots, a-1$, iteratively call Zeroise$_i(\cdot)$
2. Let $\mathbf{u}_1, \ldots, \mathbf{u}_{s'}$ be the resulting $(a-1)$-samples
3. If $(1, 0, \ldots, 0) \in \{\mathbf{u}_1, \ldots, \mathbf{u}_{s'}\}$ output the index of the 2^{a-1} vectors subset of $\mathbf{x}_1, \ldots, \mathbf{x}_s$ that resulted in $(1, 0, \ldots, 0)$

The first bit of $\mathbf{s}$ is: $\sum_{i \in S} b_i \pmod 2$

Analysis

- If $s = a2^b$, then $s' \geq 2^b$
- Probability of output is $(1 - 1/e)$
- Probability that output is correct is $\geq 1/2 + 1/2(1 - 2\eta)^{2^{a-1}}$
- Repeat poly$((\frac{1}{1-2\eta})^{2^a}, b)$ times to reduce the error probability

Main Algorithm

- The rest of the bits of $\mathbf{s}$ can be found using Solve$(\cdot)$ on cycling shifting all the examples.
- Thus the effective computation time is poly$\left(\left(\frac{1}{1-2\eta}\right)^{2^a}, 2^b\right)$

- Recall: Restricted parity function depends only on k bits of $\mathbf{s}$
- If $k = \mathrm{O}(\log n)$ then we can learn the parity in $\mathrm{O}(n)$
- Leads to separation between SQ-Model (where restricted-LPN is hard) and the noisy-PAC model

CRYPTOGRAPHY FROM LPN

"In some sense, cryptography is the opposite of learning."
– Shalev-Schwartz and Ben-David

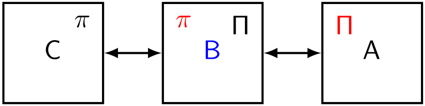
Cryptography 101

How to build protocols?

1. Assume a "hard" problem π (*e.g.*, factorisation, discrete-log)
2. Build a protocol Π on π
3. Aim: η is hard $\implies$ Π is not breakable $\equiv$
   Π is breakable $\implies$ π is not hard

Reductions: $\pi \leq \Pi$

1. Assume an adversary A against Π and use it to break π



2. Since η is assumed to be hard, this leads to a contradiction.

Recall: LPN

Find $\mathbf{s}$, given

$$\begin{aligned}\langle \mathbf{s}, \mathbf{x}_1 \rangle &\approx_\eta b_1 \pmod 2\\ \langle \mathbf{s}, \mathbf{x}_2 \rangle &\approx_\eta b_2 \pmod 2\\ &\vdots\\ \langle \mathbf{s}, \mathbf{x}_q \rangle &\approx_\eta b_q \pmod 2\end{aligned}$$

where $\mathbf{s} \in \mathbb{Z}_2^n$, $\mathbf{x}_i \sim \mathbb{Z}_2^n$, $b_i \in \mathbb{Z}_2$, $q \in \mathrm{poly}(n)$ and $\eta < 1/2$

Learning with Errors: LPN for higher moduli

Find $\mathbf{s}$, given

$$\begin{aligned} \langle \mathbf{s}, \mathbf{x}_1 \rangle &\approx_\chi b_1 \pmod{p} \\ \langle \mathbf{s}, \mathbf{x}_2 \rangle &\approx_\chi b_2 \pmod{p} \\ &\vdots \\ \langle \mathbf{s}, \mathbf{x}_q \rangle &\approx_\chi b_q \pmod{p} \end{aligned}$$

where $\mathbf{s} \in \mathbb{Z}_p^n, \mathbf{x}_i \sim \mathbb{Z}_p^n$, $b_i \in \mathbb{Z}_p$, $q \in$ poly and
χ is a probability distribution on $\mathbb{Z}_p$

LPN=LWE if $p = 2$ and $\chi(0) = 1 - \eta, \chi(1) = \eta$

Hardness of LWE

- Conjectured to be hard to break
- Lattice problems reduce[6] to LWE for appropriate choice of p and χ
  - Example: $p = \mathrm{O}(n^2)$, $\alpha = \mathrm{O}(\sqrt{n} \log n)$ and $\chi = \bar{\Psi}_\alpha$, discrete Gaussian on $\mathbb{Z}_p$ with s.d. αp
  - For the above parameters $\mathsf{SVP}, \mathsf{SIVP} \leq \mathsf{LWE}$
    - SVP: shortest-vector problem
    - SIVP: shortest independent vectors problem
- The above parameters used for the encryption scheme

[6] Regev, 2005

REGEV'S ENCRYPTION SCHEME

Encryption Scheme: Definitions

Consists of three algorithms $\Pi = \{\mathsf{K}, \mathsf{E}, \mathsf{D}\}$

Key Generation. $\mathsf{K} : \mathbb{N} \to \mathcal{K}$

$(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \mathsf{K}(1^n)$

Encryption. $\mathsf{E} : \mathcal{M} \to \mathcal{C}$

$c \xleftarrow{\$} \mathsf{E}(m, \texttt{pk})$

Decryption. $\mathsf{D} : \mathcal{C} \to \mathcal{M} \cup \{\perp\}$

$m' \leftarrow \mathsf{D}(c, \texttt{sk})$

Requirements:

1. Correctness: for all $(\texttt{pk}, \texttt{sk}) \xleftarrow{\$} \mathsf{K}(1^n)$, $m \xleftarrow{\$} \mathcal{M}$

$$\mathsf{D}(\mathsf{E}(\texttt{pk}, m), \texttt{sk}) = m$$

2. Security: ciphertext c should not leak any information about the plaintext m

Bit-Encryption from LWE

- Bit-Encryption: $\mathcal{M} = \{0, 1\}$
- Parameters:
  1. $n \in \mathbb{N}$: the security parameter
  2. p: prime modulus of the underlying group ($p = \mathrm{O}(n^2)$)
  3. ℓ: length of the public key ($\ell = 5n$)
  4. $\chi = \bar{\Psi}_\alpha$

Bit-Encryption from LWE

Key Generation, $\mathsf{K}(1^n)$:

1. Secret key: $\mathtt{sk} := \mathbf{s} \xleftarrow{\$} \mathbb{Z}_p^n$
2. Public key: $\mathtt{pk} := \{\mathbf{x}_i, b_i\}_{i=1}^{\ell}$, where

$$\mathbf{x}_1, \ldots, \mathbf{x}_\ell \xleftarrow{\$} \mathbb{Z}_p^n,\ e_1, \ldots, e_\ell \xleftarrow{\$} \chi \text{ and } b_i := \langle \mathbf{x}_i, \mathbf{s} \rangle + e_i$$

Encryption, $\mathsf{E}(m, \mathtt{pk})$:

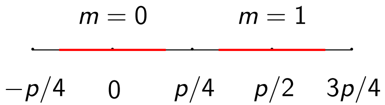
1. Choose random $S \subset [\ell]$
2. $c := \begin{cases} (\sum_{i \in S} \mathbf{x}_i, \sum_{i \in S} b_i) & \textbf{if } m = 0 \\ (\sum_{i \in S} \mathbf{x}_i, \lfloor p/2 \rfloor + \sum_{i \in S} b_i) & \textbf{if } m = 1 \end{cases}$

Decryption, $\mathsf{D}(c, \mathtt{sk})$: Note that $c = (\mathbf{x}, b)$

1. $m' := \begin{cases} 0 & \textbf{if } b - \langle \mathbf{x}, \mathbf{s} \rangle \text{ is closer to 0 than } \lfloor p/2 \rfloor \text{ (modulo } p\text{)} \\ 1 & \text{otherwise} \end{cases}$

Correctness

- Intuition: since the noise is sampled from appropriate discrete Gaussian, it does not drown the message
- Argument
  - Decryption: $e := \sum_{i \in S} e_i = \begin{cases} b - \langle \mathbf{x}, \mathbf{s} \rangle & \textbf{if } m = 0 \\ b - \langle \mathbf{x}, \mathbf{s} \rangle - \lfloor p/2 \rfloor & \textbf{if } m = 1 \end{cases}$



  - Error in decryption only if $e < p/4$
  - Let's χ^* denote the distribution of e
  - Claim: for $\chi = \bar{\Psi}_\alpha$

$$\mathop{\mathbb{P}}_{\mathbf{e} \sim \chi^*}(e < p/4) > 1 - \delta \text{ for some } \delta > 0$$

Security

- Distributions involved:
  1. $\mathcal{A}_{\mathbf{s},\eta}$: LWE sampling
  2. $\mathcal{C}_m$: ciphertext corresponding to encryption of bit m
  3. $\mathcal{U}$: uniform distribution on $\mathbb{Z}_p^n \times \mathbb{Z}_p$
- $\mathcal{X} \overset{\mathsf{D}}{\not\equiv} \mathcal{Y}$: denotes that D distinguishes $\mathcal{X}$ from $\mathcal{Y}$

- Argument
  1. Assume that the ciphertexts are distinguishable
  2. $\exists \mathsf{A}$ s.t. $\mathcal{C}_0 \overset{\mathsf{A}}{\not\equiv} \mathcal{C}_1 \implies$
  3. $\exists \mathsf{A}'$ s.t. $\mathcal{C}_0 \overset{\mathsf{A}'}{\not\equiv} \mathcal{U}$ [shifting + averaging] $\implies$
  4. $\exists \mathsf{A}''$ s.t. $\mathcal{A}_{\mathbf{s},\eta} \overset{\mathsf{A}''}{\not\equiv} \mathcal{U}$ [Leftover Hash Lemma]

More LWE

- Post-Quantum Cryptosystems
- Fully-Homomorphic Encryption[7]

[7] Brakerski and Vaikuntanathan, 2011

Sources

Mohri *et al.*– *Foundations of Machine Learning*

Shalev-Schwartz and Ben-David – *Understanding Machine Learning*

Regev – *On Lattices, Learning with Errors, Random Linear Codes, and Cryptography*

Blum *et al.*– *Noise-Tolerant Learning, the Parity Problem and the SQ Model*

THANK YOU!