Teaching and compressing for low VC-dimension

Farnam Mansouri,
Advisor: Dr. Adish Singla

Sharif University of Technology

fmansouri@ce.sharif.edu

November 22, 2019

Overview

1. Introduction
    - VC-Dimension and PAC-Learning
    - Sample Compression Schemes
    - Basic Definitions
2. An Upper Bound on RTD
3. Upper Bound for Compression Scheme
    - Constructing Compression Map for Compression Scheme
    - Constructing Reconstruction Map for Compression Scheme
    - Correctness of Construction for Compression Scheme
    - Compression Size of Compression Scheme

VC-Dimension and PAC-Learning

Theorem

For every Concept class C with VC-Dimension d, and distribution μ over X and $\epsilon, \delta > 0$, and integer m satisfying $2(2m+1)^d(1-\epsilon/4) < \delta$. Let $c \in C$ and $Y = \{x_1, ..., x_m\}$, be a set of m independent samples from μ. Then the probability there exists $c' \in C$ so that $c_{|Y} = c'_{|Y}$ and $\mu(\{x : c(x) \neq c'(x)\})$ is at most δ.

Definition

Every probability distribution μ on X induces the (pseudo) metric

$$dist_\mu(c, c') = \mu(\{x : c(x) \neq c'(x)\})$$

VC-Dimension and PAC-Learning (cont.)

Definition ($\epsilon -$ *seprating* set w.r.t. μ)

$S \subseteq C$ is called $\epsilon -$ *seprated* w.r.t. μ if $\forall c, c' \in S : dist_{\mu}(c, c') > \epsilon$

Definition ($\epsilon -$ *approximating* set)

A concept class is called $\epsilon -$ *approximating* set if it is maximal $\epsilon -$ *seperating* set and is denoted by $A_{\mu}(C, \epsilon)$.
Set A is maximal if $\forall c \notin A : \exists r \in A : dist_{\mu}(c, r) \leq \epsilon$, and r is called rounding of c in A.

Theorem (Haussler Theorem)

If $d = VCD(C)$ and S is $\epsilon -$ seprated w.r.t. μ then

$$|S| \leq e(d+1)(\frac{2e^2}{\epsilon})^d \leq (\frac{4e^2}{\epsilon})^d$$

Sample Compression Schemes

Definition (Compression Schemes with Information Q)

Let $|X| = n$ and

$$L_C(k_1, k_2) = \{(Y, y) : Y \subseteq X, k_1 \leq |Y| \leq k_2, y \in C_{|Y}\}$$

be the set of labeled samples from C, of sizes between k_1 and k_2. A k-sample compression scheme for C with information Q, for a finite set of Q, consists of tow maps κ, ρ for which the following hold:

- κ (the compression map)

  $$\kappa : L_C(1, n) \to L_C(0, k) \times Q$$

  takes (Y, y) to $((Z, z), q)$ with $Z \subseteq Y$ and $y_{|Z} = z$.

Sample Compression Schemes (cont.)

Definition (Compression Schemes with Information Q (cont.))

- ρ (the reconstruction map)

$$\rho : L_C(o, k) \times Q \to \{0, 1\}^X$$

is so that for all (Y, y) in $L_C(1, n)$,

$$\rho(\kappa(Y, y))_{|Y} = \{(Y, y)\}$$

Sample Compression Schemes (cont.)

Theorem (Littlestone-Warmuth)

Let $c \in C$, and μ distribution on X, and $Y = (x_1, ..., x_m)$ set of independent samples from μ. Let κ, ρ be a k-sample compression scheme for C with additional information Q, and $h = \rho(\kappa(Y, y))$. Then,

$$Pr_{\mu^m}(dist_\mu(h, c) > \epsilon) < |Q| \sum_{j=0}^{k} \left(\begin{array}{c} m \\ j \end{array} \right) (1-\epsilon)^{m-j}$$

Examples of Compression Schemes

Note that: Here we are looking for examples of **k-sample compression scheme with no additional**

rectangles: Consider Class of axis parallel rectangles in $\mathcal{R}^2$; the point within a rectangles are labeled '1', and others '0'. Now compression function only saves the leftmost, rightmost, top and bottom point, so he always saves only 4 points with the labels. Consider the smallest rectangle consistent with all 4, now label every sample according to this rectangle, it is guaranteed to be consistent with original samples. Note that VC-Dimension of this class is also 4.

Examples of Compression Schemes (cont.)

intersection closed concept classes: We're trying to find a compression scheme for any intersection closed concept class. Reconsider the definition of spanning set of a concept; Any set of samples and their labels must be consistent with a concept, now we compress those examples to the spanning set of a concept, and we can rebuild the concept by having the intersection of all concepts containing the spanning set. Natarajan proved size of every spanning set is at-most VC-Dimension of the concept class. this indicates that we have a k-sampled compression scheme for the concept class which $k \leq VCD(C)$.

Basic Definitions

Definition (dual class)

We'll define dual of concept class C, $C^* = \{c_x : x \in X\}$, where $c_x : C \to \{0,1\}$ where $c_x(c) = 1$ iff $c(x) = 1$, i.e., C^* is distinct rows of transpose of C.

Lemma (Assouad)

If $VCD(C) = d$, then $VCD(C^*) < 2^{d+1}$.

Proof.

Assume for the sake of contradiction $VCD(C^*) \geq 2^{d+1}$. Define M $(d+1) \times 2^{d+1}$ matrix which is $d+1$ shattered instances now since $VCD(C^*) \geq 2^{d+1}$ there are 2^{d+1} rows in C which are shattered, this indicates that M is subclass of those rows, which means $VCD(C) \geq d+1$, and is contradiction. □

Basic Definitions (cont.)

Definition ($\epsilon-$approximating of dual class)

We'll define $A^*(C, \epsilon) = A_U(C^*, \epsilon)$, where U is uniform distribution.

An Upper Bound on RTD

Theorem

If $VCD(C) = d$*, then* $RTD(C) < d2^{d+3}(\log(4e^2) + \log(\log(|C|))$.

Proof.

We'll denote $\epsilon := |C|^{-\frac{1}{d2^{d+2}}}$. Assume that $\forall c_x, c_{x'} \in C^* : dist_\mu(c_x, c'_x) > \epsilon$. And also without loss of generality imagine non of two rows of C are the same. Using Haussler theorem we'll drive

$$|X| = |C^*| \leq (\frac{4e^2}{\epsilon})^{VCD(C^*)} \leq (\frac{4e^2}{\epsilon})^{2^{d+1}} < (\frac{1}{\epsilon})^{2^{d+2}}$$

which the second inequality comes from Assouad Lemma. We'll drive

$$|C| \leq |X|^d < (\frac{1}{\epsilon})^{d.2^{d+2}} = |C|$$

which is a contradiction. □

An Upper Bound on RTD (cont.)

Proof.

This indicates $\exists c_x, c_{x'}$ which $dist_\mu(c_x, c'_x) = \frac{|\{c \in C : c(x) \neq c(x')\}|}{|C|} \leq \epsilon$. We'll drive

$$|\{c \in C : c(x) = 0 \& c(x') = 1\}| \leq |\{c \in C : c(x) \neq c(x')\}| \leq |C|^{1 - \frac{1}{d2^{d+2}}}.$$

If we take x, x' iteratively after at-most $d2^{d+2} \log(\log(|C|))$ iterations, $|C| < (4e)^2 d2^{d+2}$. After this A. Wigderson and A. Yehudayoff introduced a halving algorithm which finds c^* in less than $\log |C|$ steps.
We'll drive

$$TD_{best}(C) \leq d2^{d+2}(\log(\log(|C|)) + \log(4e^2)).$$

which it indicates $RTD(C) \leq d2^{d+2}(\log(\log(|C|)) + \log(4e^2))$.

□

Upper Bound for Compression Scheme

Theorem

If $VCD(C) = d$, we have k-sample compression scheme with additional information Q, such that $k = O(d2^d \log\log|C|)$ and $\log(Q) \le (k \log k)$.

construction of compression map κ:

Proof.

We'll construct κ with induction of $|C|$.

Initial Condition: if $|C| \le (4e^2)^{d2^{d+1}}$ then S. Floyd and M. K. Warmuth have introduce a k-sampled compression scheme with $k \le log|C| = O(d.2^d)$ and no additional information.

Induction Step: $\exists\epsilon : \epsilon|C| = (\frac{1}{\epsilon})^{d2^d}$, now using Haussler theorem we know that:

$$|A^*(C,\epsilon)| \le (\frac{2e^2}{\epsilon})^{VCD(C^*)}(\frac{2e^2}{\epsilon})^{2^{d-1}} < (\frac{1}{\epsilon})^{2^d} < |X|$$

□

Constructing Compression Map for Compression Scheme

Proof.

which the second inequality comes from Assouad Lemma, and the last one comes from Saur's lemma. Let $(Y, y) \in L_C(1, n)$, then we'll denote $r(x)$ rounding of x in $A * (C, \epsilon)$. From now on we'll assume $Q = (F, T)$, which f is a function and T is an integer, which is number of inputs of f.

- **Case 1:** If $\exists x \in Y, c \in C : c(r(x)) \neq c(x)$, we'll denote $y = C_{|Y}$. we'll define:
  - $C' = \{C_{|X} - \{x, r(x)\} : c' \in C, c'(x) = c(x), c'(r(x)) = c(r(x))\}$
  - $Y' = Y - \{x, r(x)\}, y' = y_{|Y}$

  now using induction we know there is a $\kappa : (Y', y') \to ((Z', z'), (f', T'))$, define Z, z, f, T as:
  - $Z_{|Z'} = Z', z(x) = y(x)$
  - $f(0, ..., T-1) = f'(0, ..., T-1), f(T) = x, T = T' + 1$

□

Constructing Compression Map for Compression Scheme (cont.)

Proof.

- **Case 2:** If $\forall x \in Y, c \in C : c(r(x)) = c(x)$, we'll denote $y = C_{|Y}$. We'll also define $r(Y) = \{r(y) : y \in Y\}$ which $r(Y) \subseteq A^*(C, \epsilon)$ since $\forall xinX : r(x) \in A * (C, \epsilon)$. We'll define:
  - $C' = C_{|A^*(C,\epsilon)}$
  - $Y' = r(Y)$, $\forall x' \in Y' : y'(x') = y(s(x'))$, which $\forall x' \in r(Y) : s(x') \in Y, r(s(x')) = x'$, i.e., $s = r^{-1}$, but s isn't guaranteed to have a reverse that's why we define it this way.

  Now since $|A^*(C, \epsilon)| < |X|$ by using induction we know there is a $\kappa : (Y', y') \to ((Z', z'), (f', T'))$, define Z, z, f, T as:
  - $Z = \{s(x') : x' \in Z'\}, z(x) = z'_{|r(x)}$
  - $f(0, ..., T - 1) = f'(0, ..., T - 1), f(T) =$ not defined , $T = T' + 1$

□

Constructing Reconstruction Map for Compression Scheme

construction of reconstruction map ρ:

Proof.

This time we'll prove by induction on $|X|$ since every time T reduces $|X|$ reduces too, we can prove by induction on T.

Initial Condition: $T = 0$, this situation as mentioned before is already solved.

Induction Step:

- **Case 1:** We'll denote $x = f(T)$. We'll define:
  - $X' = X - \{x, r(x)\}$
  - $Y' = r(Y)$
  - $\forall x' \in Y' : y'(x') = y(s(x'))$, which $\forall x' \in r(Y) : s(x') \in Y, r(s(x')) = x'$, i.e., $s = r^{-1}$, but s isn't guaranteed to have a reverse that's why we define it this way.
  - $Z = \{s(x') : x' \in Z'\}, z(x) = z'_{|r(x)}$
  - $f(0, ..., T-1) = f'(0, ..., T-1), f(T) =$ not defined , $T = T' + 1$

□

Constructing Reconstruction Map for Compression Scheme (cont.)

Proof.

- **Case 1 (Cont.):** By Induction on $((Z', z'), (f', T'))$, we'll get ρ, h'. Output h will be $h_{|X'} = h'$, $h(x) = z(x)$, $h(r(x)) = 1 - z(x)$.
- **Case 2:** Choose s in a way which $\forall z \in Z : r(s(z)) = z$. Now define
  - $X' = A^*(C, \epsilon)$, $C' = C_{|X}$
  - $Z' = r(Z)$
  - $\forall x' \in Z' : (x) = z(s(x'))$
  - $f'(0, ..., T-1) = f(0, ..., T-1)$, $T' = T - 1$

  By Induction on $((Z', z'), (f', T'))$, we'll get ρ, h'. Output h will be $\forall x \in X : h(x) = h'(r(x))$.

□

Correctness of Construction for Compression Scheme

Theorem

- *The Construction we introduced for compression scheme is correct, i.e., if $h = \rho(\kappa(Y, y))$, $h_{|Y} = y_{|Y}$.*
- *Also $Z \subseteq Y$, and $z_{|Z} = y_{|Z}$.*

Proof.

We'll prove it by induction on T, base of the induction is already worked out.

- **Case 1:** If f is defined on T. We know in this case $\exists x \in Y : C_{|Y} = y, c(r(x)) \neq c(X)$. Also by induction $Z' \subseteq Y'$ which indicates $Z = Z' \cup \{x\} \subseteq Y = Y' \cup \{x\} = Y$.
  Now $h_{|\{x,r(x)\}} = C_{|\{x,r(x)\}} = y_{|\{x,r(x)\}}, h_{|Y'} = h'_{|Y'} = y'$ which indicates $h_{|Y} = y$ this means we are correctly decoding y.

□

Correctness of Construction for Compression Scheme (cont.)

Proof.

- **Case 2:** We know $\forall x \in Z, \exists x' \in Z' : x = s(x')$. since range of s, is Y, it follows that $x \in Y$. This shows that $Z \subseteq Y$. Now

  $$h(x) = h'(r(x)) = y'(r(x)) = y(s(r(x)) = y(x)$$

  which first equation comes from definition, second one comes from induction, and third one comes from induction on hypothesis space, and last one comes from definition of case 2. Since $\forall t : r(s(t)) = t$, we'll drive $r(s(r(x))) = r(x)$. Also $\forall x : c(r(x)) = r(X)$. We'll denote $c_1 = s(r(x))$. If $y(c_1) \neq y(x)$, we know $c_1 \in Y$, this indicates $y(c_1) = c(c_1)$, also $c(c_1) = c(x)$. Now if $y(c_1) \neq y(x)$. this means $c(c_1) \neq c(x)$, but $c(r(c_1)) = c(r(x))$, also $c(r(c_1)) = c(r(x))$, which means either $C(r(x)) \neq c(x)$, or $c(r(c_1)) \neq c(c_1)$, which is contradiction.

□

Compression Size of Compression Scheme

- **Case 1:** By the definition of r, $dist_U(c_x, c_{r(X)}) \leq \epsilon$, it follows $\frac{|c' \in C : c'(x) \neq c'(r(x))|}{|C|} \leq \epsilon$. which indicates $|C'| \leq |c' \in C : c'(x) \neq c'(r(x))| \leq \epsilon |C|$.
- **Case 2:** $X(C') = |A^*(C, \epsilon)| \leq (\frac{1}{\epsilon})^{2^d}$, this indicates: $|C'| \leq |X(C')|^d \leq (\frac{1}{\epsilon})^{d.2^d} = |C|^{1 - \frac{1}{d.2^{d-1}}}$.

In either case after $O((d.2^d + 1) \log \log |C|)$ step we'll reach to base in each step $|Z|$ increases once, and at first $|Z| = O(d2^d)$, this indicates that $|Z| = O((d.2^d + 1) \log \log |C|)$, now $Q = (f, T)$, and $T = |Z|$, so for representing each output of f we'll need $\log |Z|$ bit this indicates that $\log |Q| = O(|Z| \log |Z|)$.

The End