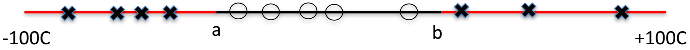
Statistical Learning – Learning From Examples

- We want to estimate the working temperature range of an iPhone.
  - We could study the physics and chemistry that affect the performance of the phone – too hard
  - We could sample temperatures in [-100C,+100C] and check if the iPhone works in each of these temperatures
  - We could sample users' iPhones for failures/temperature
- How many samples do we need?
- How good is the result?

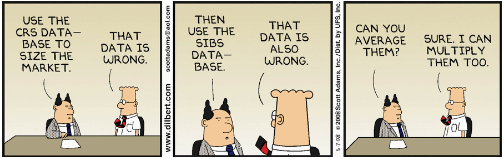


Sample Complexity

Sample Complexity answers the fundamental questions in machine learning / statistical learning / data mining / data analysis:

- Does the data (training set) contains sufficient information to make a valid prediction (or fix a model)?
- Is the sample sufficiently large?
- How accurate is a prediction (model) inferred from a sample of a given size?

Standard statistics/probabilistic techniques do not give adequate solutions



Outline

- Example: Learning binary classification
- Detection vs. estimation
- Uniform convergence
- VC-dimension
- The ε-net and ε-sample theorems
- Applications in learning and data analysis
- Rademacher complexity
- Applications of Rademacher complexity

What's Learning?

Two types of learning:

What's a rectangle?

- "A rectangle is any quadrilateral with four right angles"
- Here are many random examples of rectangles, here are many random examples of shapes that are not rectangles. Make your own rule that best conforms with the examples - Statistical Learning.

Learning From Examples

- We get n random training examples from distribution D. We choose a rule $[a, b]$ conforms with the examples.
- We use this rule to decide on the next example.
- If the next example is drawn from D, what is the probability that we is wrong?
- Let $[c, d]$ be the correct rule.
- Let $\Delta = ([a, b] - [c, d]) \cup ([c, d] - [a, b])$
- We are wrong only on examples in Δ.

What's the probability that we are wrong?

- We are wrong only on examples in Δ.
- The probability that we are wrong is the probability of having a quary from Δ.
- If $\mathit{Prob}(\text{sample from } \Delta) \leq \epsilon$ we don't care.
- If $\mathit{Prob}(\text{sample from } \Delta) \geq \epsilon$ then the probability that n training samples all missed Δ, is bounded by $(1-\epsilon)^n = \delta$, for $n \geq \frac{1}{\epsilon} \log \frac{1}{\delta}$.
- Thus, with $n \geq \frac{1}{\epsilon} \log \frac{1}{\delta}$ training samples, with probability $1-\delta$, we chose a rule (interval) that gives the correct answer for quarries from D with probability $\geq 1-\epsilon$.

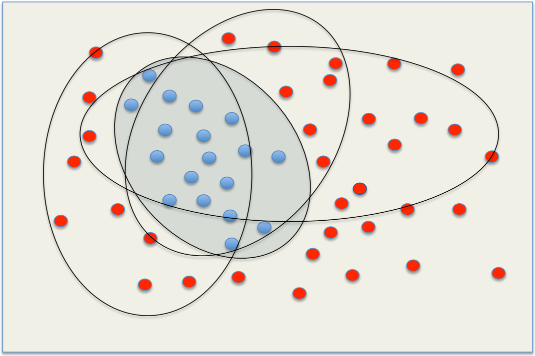
Learning a Binary Classifier

- An unknown probability distribution $\mathcal{D}$ on a domain $\mathcal{U}$
- An unknown correct classification – a partition c of U to *In* and *Out* sets
- Input:
  - Concept class $\mathcal{C}$ – a collection of possible classification rules (partitions of U).
  - A training set $\{(x_i, c(x_i)) \mid i = 1, \ldots, m\}$, where $x_1, \ldots, x_m$ are sampled from $\mathcal{D}$.
- Goal: With probability $1 - \delta$ the algorithm generates a *good* classifier.

  A classifier is good if the probability that it errs on an item generated from $\mathcal{D}$ is $\leq opt(\mathcal{C}) + \epsilon$, where $opt(\mathcal{C})$ is the error probability of the best classifier in $\mathcal{C}$.

Learning a Binary Classifier

- Out and In items, and a concept class C of possible classification rules



When does the sample identify the correct rule? - The realizable case

- The realizable case - the correct classification $c \in \mathcal{C}$.
- For any $h \in \mathcal{C}$ let $\Delta(c, h)$ be the set of items on which the two classifiers differ: $\Delta(c, h) = \{x \in U \mid h(x) \neq c(x)\}$
- Algorithm: choose $h^* \in \mathcal{C}$ that agrees with all the training set (there must be at least one).
- If the sample (training set) intersects every set in

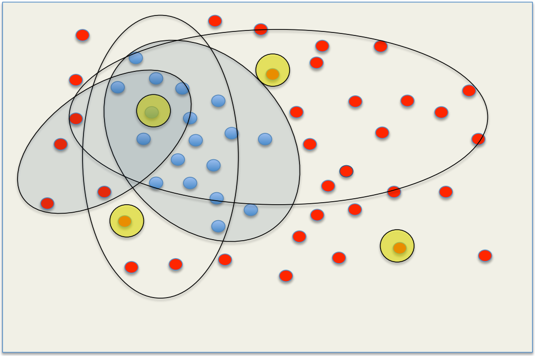
$$\{\Delta(c, h) \mid Pr(\Delta(c, h)) \geq \epsilon\},$$

then

$$Pr(\Delta(c, h^*)) \leq \epsilon.$$

Learning a Binary Classifier

- Red and blue items, possible classification rules, and the sample items



When does the sample identify the correct rule? The unrealizable (agnostic) case

- The unrealizable case - c may not be in $\mathcal{C}$.
- For any $h \in \mathcal{C}$, let $\Delta(c, h)$ be the set of items on which the two classifiers differ: $\Delta(c, h) = \{x \in U \mid h(x) \neq c(x)\}$
- For the training set $\{(x_i, c(x_i)) \mid i = 1, \ldots, m\}$, let

$$\tilde{P}r(\Delta(c, h)) = \frac{1}{m} \sum_{i=1}^{m} \mathbf{1}_{h(x_i) \neq c(x_i)}$$

- Algorithm: choose $h^* = \arg\min_{h \in \mathcal{C}} \tilde{P}r(\Delta(c, h))$.
- If for every set $\Delta(c, h)$,

$$|Pr(\Delta(c, h)) - \tilde{P}r(\Delta(c, h))| \leq \epsilon,$$

then

$$Pr(\Delta(c, h^*)) \leq opt(\mathcal{C}) + 2\epsilon.$$

where $opt(\mathcal{C})$ is the error probability of the best classifier in $\mathcal{C}$.

If for every set $\Delta(c, h)$,

$$|Pr(\Delta(c, h)) - \tilde{P}r(\Delta(c, h))| \leq \epsilon,$$

then

$$Pr(\Delta(c, h^*)) \leq opt(\mathcal{C}) + 2\epsilon.$$

where $opt(\mathcal{C})$ is the error probability of the best classifier in $\mathcal{C}$. Let $\bar{h}$ be the best classifier in $\mathcal{C}$. Since the algorithm chose h^*,

$$\tilde{P}r(\Delta(c, h^*)) \leq \tilde{P}r(\Delta(c, \bar{h})).$$

Thus,

$$\begin{aligned} Pr(\Delta(c, h^*)) - opt(\mathcal{C}) &\leq \tilde{P}r(\Delta(c, h^*)) - opt(\mathcal{C}) + \epsilon \\ &\leq \tilde{P}r(\Delta(c, \bar{h})) - opt(\mathcal{C}) + \epsilon \leq 2\epsilon \end{aligned}$$

Detection vs. Estimation

- Input:
  - Concept class $\mathcal{C}$ – a collection of possible classification rules (partitions of U).
  - A training set $\{(x_i, c(x_i)) \mid i = 1, \ldots, m\}$, where $x_1, \ldots, x_m$ are sampled from $\mathcal{D}$.
- For any $h \in \mathcal{C}$, let $\Delta(c, h)$ be the set of items on which the two classifiers differ: $\Delta(c, h) = \{x \in U \mid h(x) \neq c(x)\}$
- For the realizable case we need a training set (sample) that with probability $1 - \delta$ intersects every set in

$$\{\Delta(c, h) \mid Pr(\Delta(c, h)) \geq \epsilon\} \quad (\epsilon\text{-net})$$

- For the unrealizable case we need a training set that with probability $1 - \delta$ estimates, within additive error ϵ, every set in

$$\Delta(c, h) = \{x \in U \mid h(x) \neq c(x)\} \quad (\epsilon\text{-sample}).$$

Uniform Convergence Sets

Given a collection R of sets in a universe X, under what conditions a finite sample N from an arbitrary distribution $\mathcal{D}$ over X, satisfies with probability $1-\delta$,

1. $$\forall r \in R, \ \Pr_{\mathcal{D}}(r) \geq \epsilon \Rightarrow \ r \cap N \neq \emptyset \qquad (\epsilon\text{-net})$$

2. for any $r \in R$,

$$\left| \Pr_{\mathcal{D}}(r) - \frac{|N \cap r|}{|N|} \right| \leq \varepsilon \qquad (\epsilon\text{-sample})$$

Learnability - Uniform Convergence

Theorem

In the realizable case, any concept class $\mathcal{C}$ can be learned with $m = \frac{1}{\epsilon}(\ln|\mathcal{C}| + \ln\frac{1}{\delta})$ samples.

Proof.

We need a sample that intersects every set in the family of sets

$$\{\Delta(c, c') \mid Pr(\Delta(c, c')) \geq \epsilon\}.$$

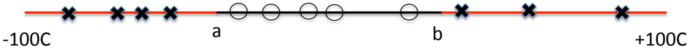
There are at most $|\mathcal{C}|$ such sets, and the probability that a sample is chosen inside a set is $\geq \epsilon$.
The probability that m random samples did not intersect with at least one of the sets is bounded by

$$|\mathcal{C}|(1-\epsilon)^m \leq |\mathcal{C}|e^{-\epsilon m} \leq |\mathcal{C}|e^{-(\ln|\mathcal{C}| + \ln\frac{1}{\delta})} \leq \delta.$$

□

How Good is this Bound?

- Assume that we want to estimate the working temperature range of an iPhone.
- We sample temperatures in [-100C,+100C] and check if the iPhone works in each of these temperatures.



Learning an Interval

- A distribution $\mathcal{D}$ is defined on universe that is an interval $[A, B]$.
- The true classification rule is defined by a sub-interval $[a, b] \subseteq [A, B]$.
- The concept class $\mathcal{C}$ is the collection of all intervals,

$$\mathcal{C} = \{[c, d] \mid [c, d] \subseteq [A, B]\}$$

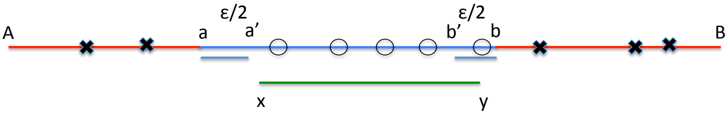
Theorem

There is a learning algorithm that given a sample from $\mathcal{D}$ of size $m = \frac{2}{\epsilon} \ln \frac{2}{\delta}$, with probability $1 - \delta$, returns a classification rule (interval) $[x, y]$ that is correct with probability $1 - \epsilon$.

Note that the sample size is independent of the size of the concept class $|\mathcal{C}|$, which is infinite.

Learning an Interval

- If the classification error is ≥ ε then the sample missed at least one of the the intervals [a,a’] or [b’,b] each of probability ≥ ε/2



Each sample excludes many possible intervals.
The union bound sums over overlapping hypothesis.
Need better characterization of concept's complexity!

Proof.

Algorithm: Choose the smallest interval $[x, y]$ that includes all the "In" sample points.

- Clearly $a \leq x < y \leq b$, and the algorithm can only err in classifying "In" points as "Out" points.
- Fix $a < a'$ and $b' < b$ such that $Pr([a, a']) = \epsilon/2$ and $Pr([b, b']) = \epsilon/2$.
- If the probability of error when using the classification $[x, y]$ is $\geq \epsilon$ then either $a' \leq x$ or $y \leq b'$ or both.
- The probability that the sample of size $m = \frac{2}{\epsilon} \ln \frac{2}{\delta}$ did not intersect with one of these intervals is bounded by

$$2(1 - \frac{\epsilon}{2})^m \leq e^{-\frac{\epsilon m}{2} + \ln 2} \leq \delta$$

□

- The union bound is far too loose for our applications. It sums over overlapping hypothesis.
- Each sample excludes many possible intervals.
- Need better characterization of concept's complexity!

Probably Approximately Correct Learning (PAC Learning)

- The goal is to learn a concept (hypothesis) from a pre-defined concept class. (An interval, a rectangle, a k-CNF boolean formula, etc.)
- There is an unknown distribution D on input instances.
- Correctness of the algorithm is measured with respect to the distribution D.
- The goal: a polynomial time (and number of samples) algorithm that with probability $1 - \delta$ computes an hypothesis of the target concept that is correct (on each instance) with probability $1 - \epsilon$.

Formal Definition

- We have a unit cost function $Oracle(c, D)$ that produces a pair $(x, c(x))$, where x is distributed according to D, and $c(x)$ is the value of the concept c at x. Successive calls are independent.
- A concept class $\mathcal{C}$ over input set X is PAC learnable if there is an algorithm L with the following properties: For every concept $c \in \mathcal{C}$, every distribution D on X, and every $0 \leq \epsilon, \delta \leq 1/2$,
  - Given a function $Oracle(c, D)$, ϵ and δ, with probability $1 - \delta$ the algorithm output an hypothesis $h \in \mathcal{C}$ such that $Pr_D(h(x) \neq c(x)) \leq \epsilon$.
  - The concept class $\mathcal{C}$ is efficiently PAC learnable if the algorithm runs in time polynomial in the size of the problem,$1/\epsilon$ and $1/\delta$.

So far we showed that the concept class "intervals on the line" is efficiently PAC learnable.

Learning Axis-Aligned Rectangle

- Concept class: all axis aligned rectangles.
- Given m samples $\{x_i, y_i, class\}$, $i = 1, \ldots, m$.
- Let R' be the smallest rectangle that contains all the positive examples. $A(R')$ the corresponding algorithm.
- Let R be the correct concept. W.l.o.g. $Pr(R) > \epsilon$
- Define 4 sides each with probability $\epsilon/4$ of R: r_1, r_2, r_3, r_4.
- If $Pr(A(R')) \geq \epsilon)$ then there is an $i \in \{1, 2, 3, 4\}$ such that

$$Pr(R' \cap r_i) \geq \epsilon/4,$$

and there were no training examples in $R' \cap r_i$

$$Pr(A(R')) \geq \epsilon) \leq 4(1 - \epsilon/4)^m$$

Learning Axis-Aligned Rectangle - More than One Solution

- Concept class: all axis aligned rectangles.
- Given m samples $\{x_i, y_i, class\}$, $i = 1, \ldots, m$.
- Let R' be the smallest rectangle that contains all the positive examples.
- Let R'' be the largest rectangle that contain no negative examples.
- Let R be the correct concept.

$$R' \subseteq R \subseteq R''$$

- Define 4 sides (in for R', out for R'') each with probability $1/4$ of R: r_1, r_2, r_3, r_4.

$$Pr(A(R')) \geq \epsilon) \leq 4(1 - \epsilon/4)^m$$

Learning Boolean Conjunctions

- A Boolean literal is either x or $\bar{x}$.
- A conjunction is $x_i \wedge x_j \wedge \bar{x}_k \ldots$.
- $\mathcal{C} =$ is the set of conjunctions of up to $2n$ literals.
- The input space is $\{0, 1\}^n$

Theorem

The class of conjunctions of Boolean literals is efficiently PAC learnable.

Proof

- Start with the hypothesis $h = x_1 \wedge \bar{x}_1 \wedge \ldots x_n \wedge \bar{x}_n$.
- Ignore negative examples generated by $Oracle(c, D)$.
- For a positive example $(a_1, \ldots, a_n)$, if $a_i = 1$ remove $\bar{x}_i$, otherwise remove x_i from h.

Lemma

At any step of the algorithm the current hypothesis never errs on negative example. It may err on positive examples by not removing enough literals from h.

Proof.

Initially the hypothesis has no satisfying assignment. It has a satisfying assignment only when no literal and its complement are left in the hypothesis. A literal is removed when it contradicts a positive example and thus cannot be in c. Literals of c are never removed. A negative example must contradict a literal in c, thus is not satisfied by h. □

Analysis

- The learned hypothesis h can only err by rejecting a positive examples. (it rejects a input unless it had a similar positive example in the training set.)
- If h errs on a positive example then in has a literal that is not in c.
- Let z be a literal in h and not c. Let

$$p(z) = Pr_{a \sim D}(c(a) = 1 \text{ and } z = 0 \text{ in } a).$$

- A literal z is"bad" If $p(z) > \frac{\epsilon}{2n}$.
- Let $m \geq \frac{2n}{\epsilon} \ln(2n) + \ln \frac{1}{\delta}$. The probability that after m samples there is any bad literal in the hypothesis is bounded by

$$2n(1 - \frac{\epsilon}{2n})^m \leq \delta.$$

Two fundamental questions:

- What concept classes are PAC-learnable with a given number of training (random) examples?
- What concept class are efficiently learnable (in polynomial time)?

A complete (and beautiful) characterization for the first question, not very satisfying answer for the second one.

Some Examples:

- Efficiently PAC learnable: Interval in R, rectangular in R^2, disjunction of up to n variables, 3-CNF formula,...
- PAC learnable, but not in polynomial time (unless $P = NP$): DNF formula, finite automata, ...
- Not PAC learnable: Convex body in R^2, $\{\sin(hx) \mid 0 \leq h \leq \pi\}$,...

Uniform Convergence [Vapnik – Chervonenkis 1971]

Definition

A set of functions $\mathcal{F}$ has the *uniform convergence* property with respect to a domain Z if there is a function $m_{\mathcal{F}}(\epsilon, \delta)$ such that

- for any $\epsilon, \delta > 0$, $m(\epsilon, \delta) < \infty$
- for any distribution D on Z, and a sample $z_1, \dots, z_m$ of size $m = m_{\mathcal{F}}(\epsilon, \delta)$,

$$Pr(\sup_{f \in \mathcal{F}} |\frac{1}{m} \sum_{i=1}^{m} f(z_i) - E_{\mathcal{D}}[f]| \leq \epsilon) \geq 1 - \delta.$$

Let $f_E(z) = \mathbf{1}_{z \in E}$ then $\mathbf{E}[f_E(z)] = Pr(E)$.

Uniform Convergence and Learning

Definition

A set of functions $\mathcal{F}$ has the *uniform convergence* property with respect to a domain Z if there is a function $m_{\mathcal{F}}(\epsilon, \delta)$ such that

- for any $\epsilon, \delta > 0$, $m(\epsilon, \delta) < \infty$
- for any distribution D on Z, and a sample $z_1, \dots, z_m$ of size $m = m_{\mathcal{F}}(\epsilon, \delta)$,

$$Pr(\sup_{f \in \mathcal{F}} |\frac{1}{m} \sum_{i=1}^{m} f(z_i) - E_{\mathcal{D}}[f]| \leq \epsilon) \geq 1 - \delta.$$

- Let $\mathcal{F}_{\mathcal{H}} = \{f_h \mid h \in H\}$, where f_h is the loss function for hypothesis h.
- $\mathcal{F}_H$ has the uniform convergence property $\Rightarrow$ an ERM (Empirical Risk Minimization) algorithm "learns" $\mathcal{H}$.
- The *sample complexity* of learning $\mathcal{H}$ is bounded by $m_{\mathcal{F}_{\mathcal{H}}}(\epsilon, \delta)$

Uniform Convergence - 1971, PAC Learning - 1984

Definition

A set of functions $\mathcal{F}$ has the *uniform convergence* property with respect to a domain Z if there is a function $m_{\mathcal{F}}(\epsilon, \delta)$ such that

- for any $\epsilon, \delta > 0$, $m(\epsilon, \delta) < \infty$
- for any distribution D on Z, and a sample $z_1, \ldots, z_m$ of size $m = m_{\mathcal{F}}(\epsilon, \delta)$,

$$Pr(\sup_{f \in \mathcal{F}} |\frac{1}{m} \sum_{i=1}^{m} f(z_i) - E_{\mathcal{D}}[f]| \leq \epsilon) \geq 1 - \delta.$$

- Let $\mathcal{F}_{\mathcal{H}} = \{f_h \mid h \in H\}$, where f_h is the loss function for hypothesis h.
- $\mathcal{F}_H$ has the uniform convergence property $\Rightarrow$ an ERM (Empirical Risk Minimization) algorithm "learns" $\mathcal{H}$. PAC efficiently learnable if there a polynomial time ϵ, δ-approximation for minimum ERM.

Uniform Convergence

Definition

A set of functions $\mathcal{F}$ has the *uniform convergence* property with respect to a domain Z if there is a function $m_{\mathcal{F}}(\epsilon, \delta)$ such that

- for any $\epsilon, \delta > 0$, $m(\epsilon, \delta) < \infty$
- for any distribution D on Z, and a sample $z_1, \ldots, z_m$ of size $m = m_{\mathcal{F}}(\epsilon, \delta)$,

$$Pr(\sup_{f \in \mathcal{F}} |\frac{1}{m}\sum_{i=1}^{m} f(z_i) - E_{\mathcal{D}}[f]| \leq \epsilon) \geq 1 - \delta.$$

VC-dimension and Rademacher complexity are the two major techniques to

- prove that a set of functions $\mathcal{F}$ has the uniform convergence property
- charaterize the function $m_{\mathcal{F}}(\epsilon, \delta)$

Some Background

- Let $f_x(z) = \mathbf{1}_{z \leq x}$ (indicator function of the event $\{-\infty, x\}$)
- $F_m(x) = \frac{1}{m}\sum_{i=1}^{m} f_x(z_i)$ (empirical distributed function)
- Strong Law of Large Numbers: for a given x,

$$F_m(x) \rightarrow_{a.s} F(x) = Pr(z \leq x).$$

- Glivenko-Cantelli Theorem:

$$\sup_{x \in \mathbf{R}} |F_m(x) - F(x)| \rightarrow_{a.s} 0.$$

- Dvoretzky-Keifer-Wolfowitz Inequality

$$Pr(\sup_{x \in \mathbf{R}} |F_m(x) - F(x)| \geq \epsilon) \leq 2e^{-2n\epsilon^2}.$$

- VC-dimension characterizes the uniform convergence property for arbitrary sets of events.