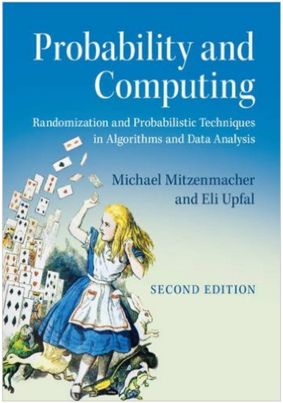
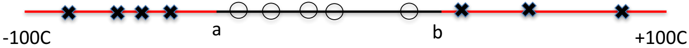
CS155/254: Probabilistic Methods in Computer Science

Chapter 14.1: Sample Complexity - Statistical Learning Theory



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 and Uniform Convergence

Given a function f, with values in a bounded domain (say $f \in [0,1]$), and a sample $x_1, \ldots, x_n$ from a distribution $\mathcal{D}$, we can estimate $\mathbf{E}_{\mathcal{D}}[f]$ using Hoeffding's inequality:

$$Pr(|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - E[f]| \geq \epsilon) \leq 2e^{-2n\epsilon^2} = \delta,$$

or

$$Pr\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \sqrt{\frac{\delta/2}{2n}} \leq E[f] \leq \frac{1}{n}\sum_{i=1}^{n} f(x_i) + \sqrt{\frac{\delta/2}{2n}}\right) \geq 1-\delta$$

We can estimate the probability of an event A using $Pr_{\mathcal{D}}(A) = E_{\mathcal{D}}[1_{x\in A}]$.

We have a well understood relation between ϵ, δ and n,

$$\epsilon \approx \sqrt{\frac{\delta}{n}}.$$

Sample Complexity and Uniform Convergence

For a single function f, $f \in [0,1]$, and a sample $x_1, \ldots, x_n$ from a distribution $\mathcal{D}$,

$$Pr(|\frac{1}{n}\sum_{i=1}^{n} f(x_i) - E[f]| \geq \epsilon) \leq 2e^{-2n\epsilon^2} = \delta,$$

or

$$Pr\left(\frac{1}{n}\sum_{i=1}^{n} f(x_i) - \sqrt{\frac{\delta/2}{2n}} \leq E[f] \leq \frac{1}{n}\sum_{i=1}^{n} f(x_i) + \sqrt{\frac{\delta/2}{2n}}\right) \geq 1-\delta$$

We have a well understood relation between ϵ, δ and n, $\epsilon \approx \sqrt{\frac{\delta}{n}}$.

How does this relation change when we use the sample $x_1, \ldots, x_n$ to estimate the expectations of m different functions?

With a union bound we get $\epsilon \approx \sqrt{\frac{\delta m}{n}}$. Can we do better?

Sample Complexity, Uniform Convergence and Statistical Learning

We have a distribution $\mathcal{D}$ on $\mathcal{X}$, and a collection of functions (hypothesis) $\mathcal{F} : \mathcal{X} \rightarrow \mathcal{Y}$.

We want to identify a function $f \in \mathcal{F}$ that best models the relation between $\mathcal{X}$ and $\mathcal{Y}$ with respect to a loss function $\ell(f(x), y)$ (the penalty for returning $f(x)$ when the "correct" value is y).

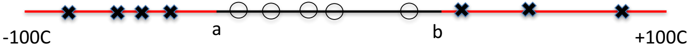
Empirical risk minimization:
Given a sample (training set) $(x_1, y_1), \ldots, (x_n, y_n)$,
approximate $f^* = \arg\min_{f \in \mathcal{F}} E[\ell(x, f(x))]$,
using $\tilde{f}^* = \arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \ell(x, f(x_i))$.

Uniform convergence: The sample needs to simultaneously estimate the expected loss of all the functions in $\mathcal{F}$. We need to bound

$$\Pr(\sup_{f \in \mathcal{F}} |\sum_{i=1}^{n} f(x_i) - E[f]| \geq \epsilon)$$

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?



Learning an Interval From Examples

- The domain is $[A, B] \subset (-\infty, +\infty)$. There is an unknown distribution D on $[A, B]$
- There is an unknown classification of the domain to an interval of points in class In, the rest are in class Out.
- The algorithm gets n random labeled examples, $(point, class)$, from the distribution D (the "training set").
- The algorithm chooses a rule $r = [x, y]$ based on the examples.
- We use this rule to decide on unlabeled points drawn from D (the "test set").
- Let $r^* = [a, b]$ be the correct rule.
- Let $\Delta(r, r^*) = ([a, b] - [x, y]) \cup ([x, y] - [a, b])$
- We are wrong only on examples in $\Delta(r, r^*)$.

What's the probability that we are wrong?

- The correct classification is $r^* = [a, b]$.
- The algorithm chose $r = [x, y]$.
- The algorithm is wrong only on examples in $\Delta(r, r^*)$.
- The probability that the algorithm is wrong is $Pr_D(\Delta(r, r^*))$.
- For fixed ϵ and δ we want:

$$Prob(\text{select } r \text{ such that } Pr(\Delta(r, r^*)) \geq \epsilon) \leq \delta$$

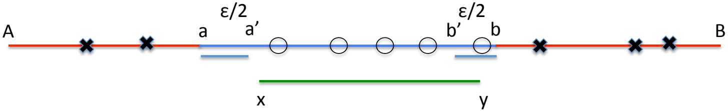
Two probabilities:

1. ϵ - the probability that the rule gives a wrong answer.
2. δ - the probability that the algorithm fails to generate a rule with error $\leq \epsilon$.

Sample Complexity: the minimum number of labeled samples to satisfy both probabilities.

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!

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 $1 - \epsilon$.

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

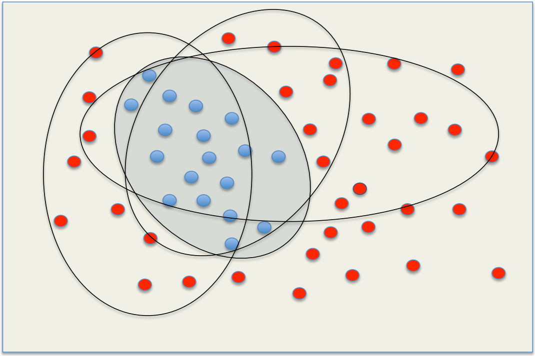
$$2(1 - \frac{\epsilon}{2})^m \leq e^{-\frac{\epsilon m}{2} + \ln 2} = e^{-\frac{\epsilon}{2}\frac{2}{\epsilon}\ln\frac{2}{\delta} + \ln 2} = \delta$$

Learning a Binary Classifier

- An unknown probability distribution $\mathcal{D}$ on a domain $\mathcal{U}$
- An unknown correct classification – a partition c of $\mathcal{U}$ 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 $\mathcal{C}$.
- Realizable case: $c \in \mathcal{C}$, $Opt(C) = 0$.
- Unrealizable case: $c \notin \mathcal{C}$, $Opt(C) > 0$.

Learning a Binary Classifier

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



When does the sample specify a *good* 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

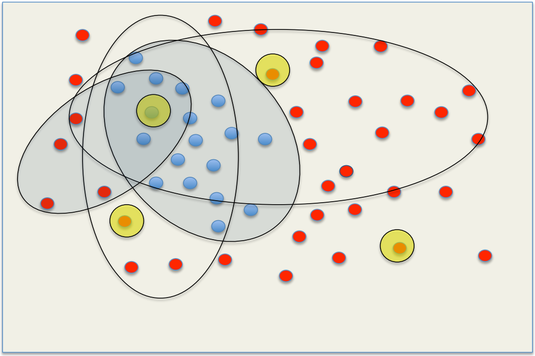
$$\{\Delta(c, h) \mid Pr(\Delta(c, h)) \geq \epsilon\},$$

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 a *good* 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

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

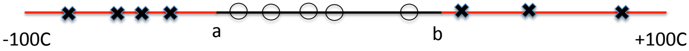
There are at most $|\mathcal{C}|$ such sets, and the probability that a sample is chosen inside a set is $\geq \epsilon$.
The probability that m random samples did not intersect with at least one of the sets is bounded by

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

□

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.

- 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 $1 - \epsilon$.

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, $\{\sin(hx) \mid 0 \leq h \leq \pi\}$,...