richardwu.ca

CS 485/685 Course Notes

Machine Learning: Statistical and Computational Foundations

Shai Ben-David • Winter 2019 • University of Waterloo

Last Revision: January 30, 2019

Table of Contents

Abstract

These notes are intended as a resource for myself; past, present, or future students of this course, and anyone interested in the material. The goal is to provide an end-to-end resource that covers all material discussed in the course displayed in an organized manner. These notes are my interpretation and transcription of the content covered in lectures. The instructor has not verified or confirmed the accuracy of these notes, and any discrepancies, misunderstandings, typos, etc. as these notes relate to course's content is not the responsibility of the instructor. If you spot any errors or would like to contribute, please contact me directly.

1 January 8, 2019

1.1 What is machine learning?

In machine learning, we aim to construct a program that takes as input **experiences** and produces as output **expertise**, or what we have learned from the experience.
We can then apply the **expertise** to produce useful programs such as a spam filter.
An example of learning in nature is **bait shyness**: rats who become sick from eating poisoned bait will become more cautious of food of similar characteristic in the future. Since rats will become more cautious of bait in the future, a delayed poison mechanism (rat is poisoned only 2 days after consuming the bait) is necessary for effective bait by de-associating poison from the bait.
Another example is an experiment called **pigeon superstition** by Skinner (1947): pigeons are starved in a cage with various objects. At random intervals, food is dispersed to satiate the pigeons. Eventually, each pigeon develops a "superstition": they each associate one arbitrary behaviour (e.g. a specific object or a specific movement) that results in food being dispersed.
On the contrary, Garcia (1996) tried a similar experiment to bait shyness with rats where poisoned and un-poisoned bait were identical in characteristic. Whenever a rat approached poisoned bait, a stimulus (e.g. bell ringing, electric shock) was applied to the rat. Surprisingly, the rats did not associate the arbitrary stimulus to the poisoning. This is contrary to the pigeon superstition: this can be explained by evolution (future generations are those that can become aware of poisonous bait) and the fact that rats have **prior knowledge** that poisoning comes from the bait itself, not some arbitrary stimulus.

1.2 Why do we need machine learning?

We desire machines to perform learning because machines can **process lots of data** and are (generally) **fast**.
We desire machines to *learn* because some tasks are simply to complex to hardcode in (e.g. image recognition). Some tasks we do not fully understand how to solve with hardcoded rules. Furthermore, learning allows adaptivity where the machine can constantly learn from new experiences and inputs.

1.3 Types of machine learning

Supervised and unsupervised Machine learning can be generally classified as either **supervised** or **unsupervised**.

Supervised learning takes labelled examples as experience and tries to re-produce these labels on future examples by learning rules. Spam detection may be supervised learning.

Unsupervised learning does not require labelled training data. Examples of unsupervised learning is outlier detection and clustering.

Semi-supervised learning takes as input both labelled and unlabelled data and sits between supervised and unsupervised.

Reinforcement learning also sits between supervised and unsupervised: the machine knows only the rules of the environment and takes actions until a reward (i.e. label) is produced. The machine then learns to label intermediary actions to the final reward produced in the episode (sequence of actions that resulted in the reward).

Passive and active We can also distinguish between **passive** and **active** learning: the former simply takes observed data whereas the latter involves actively performing experiments and interpreting the consequences of the experiments.

Teacher Machine learning can be guided by a "teacher" i.e. how the random sample used as input is generated. Teachers may be **indifferent**, **helpful** or **adversarial**. Helpful teachers produce hints and try to guide the program in the right direction whereas adversarial tries to fool the program.

Batch and online **Batch** learning is learning from a relatively large corpus of data before producing expertise. In contrast **online** learning requires the program to learn as experience is streamed and may result in more mistakes being made.

2 January 10, 2019

2.1 Components of a model

For example sakes, suppose we observe a number of papayas and assign them a score $\in [0, 1]$ for color and hardness. We then label each one as either tasty or not tasty. Using our observations, we would like to predict in the future the tastiness of papayas based on their color and hardness score.

Input The **input** to our learner consists of three parts:

Domain set (X) It is the set of our explanatory variates, in this case $[0, 1] \times [0, 1]$ corresponding to the color score and the hardness score.

Label set (Y) It is the set of our labels: tasty and not tasty.

Training set Our observations i.e. $\{(x_1, y_1), \ldots, (x_m, y_m)\} \subset X \times Y$

Output The **output** of our learner is a **prediction rule** $h : X \to Y$ i.e. the function we learn that maps our papaya scores to a label.

Simple generating model There exists some underlying (unknown) generating process of the population we are interested in (papayas). There is some unknown **probability distribution D over X** and some unknown **labelling rule** $f : X \to Y$.

Together (D, f) describes the generation of papayas.

Success measure We define some metric to measure how well our learner learns the underlying generating model. For example

$$L_{(D,f)}(h) = Pr_{x \sim D}\big[h(x) \neq f(x)\big]$$

We would like to minimize $L_{(D,f)}(h)$ to find the optimal learner h.

2.2 Empirical Risk Minimization (ERM)

As a first strategy, a learner can employ **Empirical Risk Minimization (ERM)** whereby it picks an h that minimize the errors on the *training sample*.

There is however an issue with this strategy: suppose we learn a rule where we label tasty for papayas with scores that exactly match our tasty papayas' scores and not tasty for everything else. That is

$$h_S(x) = \begin{cases} \text{tasty} & \text{if } (x, \text{tasty}) \in S \\ \text{not tasty} & \text{otherwise} \end{cases}$$

We define the **empirical loss (risk)** over a sample $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$

$$L_S(h) = \frac{|\{i \mid h(x_i) \neq y_i\}|}{m}$$

Note the above strategy give us exactly zero empirical error on our sample set S for any generating process but is obviously not a very robust strategy as it **overfits** to our sample.
Suppose the generating process is such that D is the uniform distribution over $[0,1] \times [0,1]$ and let

$$f(x_1, x_2) = \begin{cases} \text{tasty} & \text{if both coordinates in } [0, 1, 0.9] \\ \text{not tasty} & \text{otherwise} \end{cases}$$

Note that the empirical risk is $L_S(h) = 0$, but the risk on our population is $L_{(D,f)}(h) = (0.8)^2 = 0.64$ (since we would be predicting incorrectly for infinitely many points in the region $[0.1, 0.9] \times [0.1, 0.9]$).

2.3 Introducing prior knowledge with inductive bias

For the papaya example above, we could incorporate some prior knowledge such that tasty papayas belong in some rectangular region of color and hardness scores (which we must learn). We could have also assumed the tasty papayas belong in some linear halfspace, or some arbitrary region that we can learn.
More formally, prior knowledge are a set of rules H (set of functions from X to Y) assumed by the learner to contain a good predictor: H is a **hypothesis class**.
Reformulating our previous ERM strategy, ERM_H picks $h \in H$ that minimizes empirical risk over the training set, that is we pick h^* where

$$h^* \in \text{argmin}_{h \in H} L_S(h)$$

Under the following assumptions ERM_H has good success guarantees:

Assumption 1 (Realizability) $\exists h \in H$ such that $L_{(D,f)}(h) = 0$

Assumption 2 S is picked iid by D and labelled by f (i.e. our sample is representative)

3 January 15, 2019

3.1 Finite hypothesis classes

Theorem 3.1. Let H be a **finite set** of predictors (our hypothesis class). Assume our two assumptions from above hold. Then every ERM_H learning rule is guaranteed to converge to a zero-loss predictor as the sample size tends to infinity.
Namely for every ERM_H learner A and every $\epsilon > 0$

$$Pr_{S \sim X \times f}\big[L_{(D,f)}(A(S)) > \epsilon\big] \to 0$$

as $|S| \to \infty$ (this is exactly convergence in probability where $Pr\big[L_{(D,f)}(A(S))\big] \to 0$).

Proof. Let B_ϵ denote the set of all hypotheses in H that have error $> \epsilon$ i.e.

$$B_\epsilon = \{h \in H \mid L_{(D,f)}(h) > \epsilon\}$$

Let our set of "misleading" samples be

$$M = \{S \mid |S| = m \text{ and } \exists h \in B_\epsilon \text{ s.t. } L_S(h) = 0\}$$

(samples where we have a zero empirical loss but has $> \epsilon$ loss on the true population: misleading because it tricks us that the $h \in B_\epsilon$ re-constructs f with zero error when in fact it does not).
Note that

$$Pr_S\big[L_{(D,f)}(A(S)) > \epsilon\big] \leq Pr\big[S \in M\big]$$

that is the probability that our sample does not perform better than ϵ on our true population is bounded by the probability of picking a misleading sample (this is not just an equality since we also have samples S where $L_{(D,f)}(A(S)) > \epsilon$ and $L_S(A) > 0$).

Lemma 3.1. We claim

$$Pr_{|S|=m}\big[S \in M\big] \leq |H|(1-\epsilon)^m$$

Proof. Consider any $h \in B$ (where obviously $h \neq f$). There exists a "disagreement" region D where for any x, $h(x) \neq f(x)$ (either $h(x) = +, f(x) = -$ or $h(x) = -, f(x) = +$).
For our sample S of size m, we know that $S \subseteq D^c$ (our sample cannot be in the disagreement region since $L_S(h) = 0$ i.e. our sample is perfect; empirical loss is zero so it must agree with f).
Note that since $L_s(h) > \epsilon$ (i.e. h disagrees with f on a region of proportion at least ϵ, our D), then the region where h and f agree is at most of proportion $1 - \epsilon$ (i.e. D^c).
Choose m sample points iid from D^c is thus

$$Pr_S\big[L_S(h) = 0\big] \leq (1-\epsilon)^m$$

□

Lemma 3.2 (Union bound)**.** Given two set of events A, B we know $P(A \cup B) \leq P(A) + P(B)$.

Note that $Pr[S \in M] = Pr[\text{for some } h \in B, L_S(h) = 0]$ is the union of all misleading hypotheses $h \in B$, thus

$$\begin{aligned} Pr\big[\text{for some } h \in B, L_S(h) = 0\big] &\leq \sum_{h \in B} Pr(L_S(h) = 0) \\ &= |B|(1-\epsilon)^m \\ &< |H|(1-\epsilon)^m \end{aligned}$$

Note that $1 - \epsilon \leq e^{-\epsilon}$ thus $Pr[S \in M] \leq |H|e^{-\epsilon m}$ which goes to 0 as $m \to \infty$ as desired. □

4 January 17, 2019

4.1 Probably Approximately Correct (PAC) learning

In our previous theorem with ERM_H with inductive bias we showed it could do well but only under **strong assumptions**.

Our goal is to prove similar guarantees but with more realistic/relaxed assumptions. Specifically, we would like to relax our assumption that there exists a *detrministic* f that generates the true distribution D (the labels Y) over domain X.
That is, the relax model given domain of instances X and label set Y, the data is generated by a probability distribution D over $X \times Y$. We denote our set of predictors $h : X \to Y$ as H.
Our relaxed model defines the empirical loss as

$$L_S(h) = \frac{|i \mid h(x_i) \neq y_i|}{|S|}$$

and the true loss over our probability distribution D over $X \times Y$

$$L_D(h) = P(\alpha(x, y) \mid h(x) \neq y)$$

Definition 4.1 (PAC learnable (Leslie Valiant 1984))**.** A *hypothesis class* H is **PAC learnable** if there exists a function $m_H(\epsilon, \delta) : (0, 1) \times (0, 1) \to \mathbb{N}$ and a learner A (map from labelled samples to functions $h : X \to Y$) such that for every $\epsilon, \delta \in (0, 1)$ for every distribution D over X and every labelling function $f \in H$, if $m' \geq m(\epsilon, \delta)$ and a labelled sample $S = \{(x_1, f(x_1)), \ldots, (x_m, f(x'_m))\}$ generated iid according to D and labelled by f, then

$$Pr_{S \sim (D^m, f)}\big[L_{(D,f)}(A(S)) > \epsilon\big] < \delta$$

That is: the error is bounded by ϵ (approximately), and the probability of error is bounded by δ (probably) for some large enough sample size.

Remark 4.1. The number of required samples is determined regardless of D and f.

Some weaknesses of this definition:

- Realizability assumption ($h \in H$ such that $L_{(D,f)}(h) = 0$): the learner has strong prior knowledge.
- The labelling rule is *deterministic*: the label of any x is fully determined by X.
- The training distribution and test distribution are the sample: this may be unobtainable in some cases.

4.2 Finite hypothesis H is PAC learnable

Theorem 4.1. Any finite H is PAC learnable.

Proof. Recall if H is finite then

$$Pr_{S \sim (D^m, f)}\big[L_{(D,f)}(A(S)) > \epsilon\big] < |H|e^{-m\epsilon}$$

Our claim holds if $|H|e^{-m\epsilon} \leq \epsilon$. Solving for m

$$m \geq \frac{\ln(|H|) + \ln\left(\frac{1}{\delta}\right)}{\epsilon}$$

□

4.3 Real intervals on real domain is PAC learnable

Theorem 4.2. Let $X = \mathbb{R}$ and H is the class of all real intervals where $H_{int} = \{h_{(a,b)} \mid a \leq b\}$ where

$$h_{(a,b)}(x) = \begin{cases} 1 & x \in [a,b] \\ 0 & \text{otherwise} \end{cases}$$

then H_{int} is PAC learnable.

Proof. Suppose we have a sample generated by $f \in H_{int}$ where all our positive examples and only positive examples lie within an interval of $X = \mathbb{R}$ (since $f \in H_{int}$ so it labels positive examples only in a real interval), e.g.

$$\ldots \quad 0 \quad 0 \quad 1 \quad 1 \qquad 1 \quad \ldots$$

where $S = \{(x_1, y_1), \ldots, (x_m, y_m)\}$. Our learner A could be such that $A(S) \in H_{int}$ where

$$a(S) = \min\{x_i \mid (x_i, 1) \in S\}$$
$$b(S) = \max\{x_i \mid (x_i, 1) \in S\}$$

We show that this rule A is a successful PAC learner for H_{int}.
Given ϵ, m let us upper bound the probability that an m-size sample will lead A to out h with $> \epsilon$ error.
Denote $B_\epsilon = \{h \in H \mid L_{(D,f)}(h) > \epsilon\}$ (bad hypotheses) and $M = \{S \mid A(S) \in B\}$ (set of misleading samples).
Note that a sample $S \in M$ is misleading if our minimum and maximum positive samples cover a "small" region of the actual interval specified $f \in H_{int}$ our arbitrary labelling function.
That is

$$Pr(S \in M) = Pr\big(S \text{ does not hit intervals } [\min(f), min(h)] \text{ or } [\max(h), \max(f)]\big)$$

S is a misleading sample only if S does not hit either the interval from $\min(f)$ to $\min(f) + weight_D(\epsilon/2)$ or interval form $\max(f) - weight_D(\epsilon/2)$ to $\max(f)$ (where $weight_D(\epsilon/2)$ is defined as the region R immediately to the right/left where $Pr_D(x \in R) = \epsilon/2$).
That is

$$Pr(S \in M) \leq \big(1 - \frac{\epsilon}{2}\big)^m + \big(1 - \frac{\epsilon}{2}\big)^m$$

where each sample misses both intervals of weight/probability $\frac{\epsilon}{2}$.
Therefore

$$Pr_{S \sim (D^m, f)}\big[L_{(D,f)}(A(S)) > \epsilon\big] \leq 2\big(1 - \frac{\epsilon}{2}\big)^m$$

for some $m(\epsilon, \delta)$ such that $2\big(1 - \frac{\epsilon}{2}\big)^m < \delta$, thus H_{int} is PAC learnable. □

5 January 22, 2019

5.1 Agnostic PAC learning (more general learning model)

Our previous definition of PAC learnable is still too unrealistic. Namely we are going to:

1. Remove the *realizability* assumption: we do not require there exist $h \in H$ such that $L_D(h) = 0$ (although we do not necessarily remove the requirement that the labelling rule is in H)

2. Remove the deterministic labelling requirement (allow smae x to show up with different labels)

We then assume there exists some probability distribution D over "abstract" set Z whereby data is generated iid.
Our new notion of loss is we are given some function $l(\text{hypothesis}, z \in Z)$ which is real-valued.
For example we may have for email spam detection:

Example 5.1. Let $Z = X \times \{0,1\}$ where X is the set of emails and 0/1 denotes not spam or spam.
Let $h : X \to \{0,1\}$ (our hypothesis function).
Let

$$l(h,(x,y)) = \begin{cases} 1 & \text{if } h(x) \neq y \\ 0 & \text{if } h(x) = y \end{cases}$$

Given ta training sample $S = (z_1, \ldots, z_m)$ and a predictor (hypothesis function) h the **empirical loss** of h on S is now

$$L_S(h) = \frac{1}{m} \sum_{i=1}^{m} l(h, z_i)$$

Also we define the **true loss** of h as (for true distribution D over Z)

$$L_D(h) = \mathbb{E}_{Z \sim D}(l(h,z))$$

Remark 5.1. In our example with $Z = X \times \{0,1\}$ and $l_{0,1}$ our new definition of empirical and true loss $L_S(h)$ and $L_D(h)$ are equivalent to our definitions in PAC learnable.

However our individual loss function could be arbitrarily defined:

Example 5.2. Suppose we want to predict tomorrow's temperature from today's measurements.
Let $Z = \big(\text{today's measurements} \times \text{tomorrow's temp}\big)$.
Let $h : \text{today's measurements} \to [-50, +50]$.
We define loss as $l(h,(x,y)) = |h(x) - y|$ (L1 norm).

Example 5.3 (K-means clustering)**.** Suppose we would like to pick k locations for a chain of stores in KW.
Each h will represent a set of k potential locations $h = (\mu_1, \ldots, \mu_k)$.
Therefore we let $Z =$ location of customers seeking a store and we define our loss to be

$$l((\mu_1, \ldots, \mu_k), z) = \min_{1 \leq i \leq k} |z - \mu_i|$$

i.e. the loss is the L1 distance between a customer z and the closest location μ_i.
Our training data would then be a sample $S = (z_1, \ldots, z_m)$ which is a record of past customers.

We will now explore how learning is achieved under this more general model.
The prior knowledge of the learner is again modeled by a set H of possible predictors (class of hypotheses). Our **input** is a training set $S = (z_1, \ldots, z_m)$ generated iid by some unknown D over Z. The **output** of the learner is a predictor h.
Our goal is the minimize the *true loss* $L_D(h)$.

Remark 5.2. If we know the distribution D over Z then we could solve our problem without learning, but of course all we know is S.

Definition 5.1 (Agnostic PAC learnable)**.** A class H is **agnostic PAC learnable** if $\exists m : (0,1) \times (0,1) \to \mathbb{N}$ and a learner A (mapping S's to h's) such that $\forall \epsilon \forall \delta$, for all distribution D over Z, and for all $m \geq m(\epsilon, \delta)$ we have

$$Pr_{z \sim D^m}\left[L_D(A(S)) \geq \min_{h \in H} L_D(h) + \epsilon\right] < \delta$$

Remark 5.3. Agnostic PAC learnable is almost identical to PAC learnable except we do not assume a lower bound of 0 on $L_D(A(S))$: instead we lower bound it with $\min_{h\in H} L_D(h)$ plus some small ϵ.
Furthermore we no longer assume a determinisitic f and instead describe a D over Z.

Remark 5.4. More correctly, we require only $\inf_{h\in H} L_D(h)$: we need not require an attainable minimum.

How can we learn in this new model? In many cases ERM_H is still a good strategy.

6 January 24, 2019

6.1 Finite hypothesis classes are agnostic PAC learnable

Claim. Given a **finite** hypothesis class H, H is agnostic PAC learnable.

We first require some definitions.

Definition 6.1 (ϵ-representative)**.** A sample $S = (z_1, \ldots, z_m)$ is **ϵ-representative of** H with respect to a distribution D if for any $h \in H$ we have $|L_S(h) - L_D(h)| < \epsilon$.

Claim. If S is $\frac{\epsilon}{2}$-representative of H wrt D then for any ERM_H learner A

$$L_D(A(S)) \leq \min_{h\in H} L_D(h) + \epsilon$$

Proof. Note that for any $h \in H$

$$\begin{aligned} L_D(A(S)) &\leq L_S(A(S)) + \frac{\epsilon}{2} && \frac{\epsilon}{2} - \text{representative} \\ &\leq L_S(h) + \frac{\epsilon}{2} && A(S) \text{ is } ERM_H \\ &\leq L_D(h) + \epsilon && \frac{\epsilon}{2} - \text{representative} \end{aligned}$$

sine this holds for any h then $L_D(A(S)) \leq \min_{h\in H} L_D(h) + \epsilon$. □

Question 6.1. For a given distribution D and finite hypothesis class H, how do we determine m large enough such that

$$L_D(A(S)) \leq \min_{h\in H} L_D(h) + \epsilon$$

i.e. such that H is agnostic PAC learnable?

Proof. Note that

$$\begin{aligned} & P_{S\sim D^m}\left[\forall h \in H \text{ s.t. } |L_S(h) - L_D(h)| \leq \frac{\epsilon}{2}\right] > 1 - \delta \\ =& D^m\left[\{S \mid \exists h \in H \text{ s.t. } |L_S(h) - L_D(h)| > \frac{\epsilon}{2}\}\right] < \delta \\ \iff & \bigcup_{h\in H} D^m\left[\{S \mid h \in H \text{ s.t. } |L_S(h) - L_D(h)| > \frac{\epsilon}{2}\}\right] < \delta \end{aligned}$$

where $D^m\big[\{S\}\big]$ is the total probability mass of $\{S\}$ in D^m.
Recall $L_S(h) = \frac{1}{m}\sum_{i=1}^m l(h, z_i)$ and $L_D(h) = \mathbb{E}_{Z\sim D}\big[l(h, z)\big]$. Let $\theta_i = l(h, z_i)$ and let $L_D(h) = \mu$, thus $\mathbb{E}(\theta_i) = \mu$.

Note that on the LHS we have

$$\begin{aligned}
&\bigcup_{h\in H} D^m\left[\{S \mid h \in H \text{ s.t. } |L_S(h) - L_D(h)| > \frac{\epsilon}{2}\}\right] \\
&\leq \sum_{h\in H} D^m\left[\{S \mid h \in H \text{ s.t. } |L_S(h) - L_D(h)| > \frac{\epsilon}{2}\}\right] \\
&\leq \sum_{h\in H} P\left[\left|\frac{1}{m}\sum \theta_i - \mu\right| > \frac{\epsilon}{2}\right] \\
&\Rightarrow 2|H| \exp\left(-2m\left(\frac{\epsilon}{2}\right)^2\right) < \delta
\end{aligned}$$

where the second last inequality follows from **Hoeffding's inequality** (assuming $l \in [0,1]$):

Theorem 6.1 (Hoeffding's inequality)**.** Let $\theta_1, \ldots, \theta_n$ be random variables where $\mathbb{E}(\theta_i) = \mu$ and $a \leq \theta_i \leq b$. Then

$$P\left[\left|\frac{1}{m}\sum \theta_i - \mu\right| > \epsilon\right] \leq 2\exp\left(\frac{-2m\epsilon^2}{(b-a)^2}\right)$$

Solving for m we get

$$m \geq \frac{2\log\left(\frac{2|H|}{\delta}\right)}{\epsilon^2}$$

□

Since there exists such a function $m(\epsilon, \delta)$ a finite hypothesis class H is agnostic PAC learnable.