pertinence
simple solutions for complex processes

www.pertinence.com

Introduction to Learning Theory

Olivier Bousquet
Pertinence
Paris, France
olivier.bousquet@pertinence.com

Machine Learning Summer School, Canberra, February 2006

pertinence

Outline

1. Learning Theory: What and Why?
   (a) Settings and assumptions
   (b) No Free Lunch
   (c) Consistency
2. Ingredients for Learning Bounds
   (a) Fundamental inequalities
   (b) Deviation/concentration inequalities
   (c) Union bound
3. Implications
   (a) How to use bounds
   (b) Connection to the design of algorithms

pertinence

Goal of this course

- Some thoughts about learning and its foundations (main focus)
- Keep things simple: basic techniques, finite sets of hypotheses
- Show some of the important phenomena, give insights into proof techniques
- Discuss about the meaning: how to use and how not to use a bound

pertinence

Not covered

- History of the domain
- General results, detailed proofs
- Advanced topics, e.g. influence of noise and loss functions

Learning Theory: What?

First of all, what is a theory?

> [THEORY] A set of statements or principles devised to **explain** a group of facts or phenomena, especially one that has been **repeatedly** tested or is widely accepted and can be used to **make predictions** about natural phenomena.

- Aim is to **model** a phenomenon (i.e. provide a schematic description of it, that accounts for its properties)
- so that we can **understand** it (i.e. use the model to further study the characteristics of the phenomenon)
- and **predict** it (i.e. derive consequences that can be tested)

Some more definitions

Difference between deduction and induction.

> [DEDUCTION] The process of reasoning in which a conclusion follows necessarily from the stated premises; inference by reasoning from the general to the specific.

This is how mathematicians prove theorems from axioms.

> [INDUCTION] The process of deriving general principles from particular facts or instances.

This is how physicists create theories from observing Nature.
Also: Transduction (from specific to specific in one step, without making the axioms/theory/model explicit)

What is a good theory?

- Intuitively, the goal is to provide a simple and concise account of observations.
- A good theory should be able to explain/predict many phenomena, with a simple model.
- The principles on which rely a theory are called assumptions. For example:

  > [Principle of Relativity] The Laws of Physics are the same in all inertial frames of reference
- Assumptions cannot be proved or disproved (there is no principle from which they are deduced) but their consequences can be tested.
- A good theory should make few assumptions.

What is Learning?

> [LEARNING] To gain knowledge, comprehension, or mastery of through experience or study.

- We will focus on experience, not study (too easy).
- We consider that there is a

In Machine Learning, these are synonymous

- learning
- generalization
- induction
- theory making
- modelling

Recursion

- So far, we have investigated what is a theory and how it should be constructed.
- Why is this needed? (Do we study what is a theory when we study the theory of relativity?)
- Building a theory is the process of induction (i.e. learning).

→ Are we not going in circles?

Recursion

Rephrasing:

- Induction is the process of building theories
- Learning Theory's main focus is the phenomenon of induction
- How can we create a theory about theory building?

A lot of philosophical issues are involved, we need to formalize things in order to go further.

Learning Theory: Why?

[Kurt Lewin] There is nothing so practical as a good theory

As for all theories, learning theory should

- Try to understand systems that learn from data
- Provide a framework for studying their properties
- Allow to derive consequences in the form of "predictions" of which systems may work best

Hopefully, it should guide us toward designing better learning algorithms!

Inductive principles

Unfortunately, Induction is not like Deduction.

- Deduction can be justified (if principles are correct, consequences are also).
- Justifying Induction means justifying principles. This raises many philosophical issues.

Example 1: Probability of Sunrise Tomorrow

What is the probability p that the sun will rise tomorrow? (given we observed it rising on each of the previous d days)

Example 1: Probability of Sunrise Tomorrow

What is the probability p that the sun will rise tomorrow? (given we observed it rising on each of the previous d days)

- p cannot be defined (tomorrow is not identical to yesterday and there cannot be an experiment to test what happens tomorrow).
- $p = 1$, because the sun always rose in the past.
- $p = \frac{d+1}{d+2}$ obtained from Bayes rule (assuming uniform prior).
- Use physics to estimate the probability of the sun to explode tomorrow
  - ⋆ compute the proportion of stars that explode per day
  - ⋆ compare the sun to other stars with similar properties

Results are a high probability of rising again. Justification involves comparison with past "similar" situations (similar is highly subjective).

Example 2: extend a sequence of integers

You observe the sequence

1,2,4,7,...

What comes next?

Example 2: sequence of integers

528 hits on The On-Line Encyclopedia of Integer Sequences

- Maximum number of pieces formed when slicing a pancake with n cuts $u_{n+1} = u_n + n \Rightarrow$ 1,2,4,7,11,16... (A000124)
- $u_{n+2} = u_{n+1} + u_n + 1 \Rightarrow$ 1,2,4,7,12,20,... (A000071)
- Tribonacci numbers $u_{n+3} = u_{n+2} + u_{n+1} + u_n \Rightarrow$ 1,2,4,7,13,24,... (A000073)
- Binary expansion: 1,10,100,111,1000,1011,... odd number of 1's (Odious numbers) $\Rightarrow$ 1,2,4,7,8,11,13,14... (A000069)
- Or decimal expansions of π and e interleaved $\Rightarrow$ **1**,2,**4**,7,**1**,1,**5**,8,**9**
- and why not : $u_{n+1} = u_n$ pour $n > 3 \Rightarrow$ 1,2,4,7,7,7...

$\longrightarrow$ which one is the simplest ?

Example 3: sequence of integers [Hutter]

Sequence:
$2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, ?$
What comes next?

- 61, since this is the next prime
- 60, since this is the order of the next simple group

Conclusion: We prefer answer 61, since primes are a more familiar concept than simple groups.

Example 4: sequence of digits [Hutter]

Extend
$14159265358979323846264338327950288419716939937?$

- Looks random?!
- Frequency estimate: n = length of sequence. k_i = number of occured $i \Rightarrow$ Probability of next digit being i is $\frac{i}{n}$. Asymptotically $\frac{i}{n} \longrightarrow \frac{1}{10}$ (seems to be) true.
- But we have the strong feeling that (i.e. with high probability) the next digit will be 5 because the previous digits were the expansion of π.
- Conclusion: We prefer answer 5, since we see more structure in the sequence than just random digits.

Inductive Principle

- Is there a principle for providing a "good" answer?
- Occam's razor: Use the simplest explanation consistent with past data. In other terms: build the simplest theory that accounts for the observations and use it for predicting future observations.
- Issue: simple is not an objective notion.

$\longrightarrow$ can we obtain general statements although there is no general principle?

Probability: a nice tool for reasoning

Probability theory allows to formalize reasoning under uncertainty.
Several interpretations exist:

- Frequentism: probabilities are relative frequencies. (e.g. the relative frequency of tossing head.)
- Objectivism: probabilities are real aspects of the world. (e.g. the probability that some atom decays in the next hour)
- Subjectivism: probabilities describe someone's degree of belief. (e.g. it is (im)plausible that extraterrestrians exist)

Examples from [Hutter]

Probabilities as Frequencies

- The frequentist interprets probabilities as relative frequencies.
- Repeat an experiment n times. If an event occurs $k(n)$ times, define the relative frequency of the event as $k(n)/n$.
- The limit $lim_{n\to\infty}k(n)/nn$ is defined as the probability of the event.
- Easy but limited: not possible to perform such experiments in many cases!

Probabilities as Intrinsic Properties

- For the objectivist probabilities are real aspects of the world.
- The outcome of an experiment may be physically random.
- Probabilities give the expected frequency of each outcome if one were to repeat the experiment.
- Probabilities could be measured in a frequentist way, but they pre-exist.
- Objective probabilities satisfy axioms ($P(X) \geq 0$, $P(\Omega) = 1$, $P(A \cup B) = P(A) + P(B) - P(A \cap B)$)

Probabilities as Degrees of Belief

- For the subjectivist probabilities characterize someone's degree of belief in an event occurring.
- It is natural to assume that plausibilities/beliefs $Bel(\cdot|\cdot)$ can be represented by real numbers, that the rules qualitatively correspond to common sense, and that the rules are mathematically consistent.
- Cox's theorem: Beliefs have the same properties as probabilities

Bayes' Rule

- Allows to update probabilities/beliefs based on observations.
- Let D be some observation, $P(h)$ the probability of hypotheses prior to the observation, and $P(D|h)$ the probability to observe D under the hypothesis h.
- The posterior probability $P(h|D)$ of h given the observation is

$$P(h|D) = \frac{P(D|h)P(h)}{\sum_{h'} P(D|h')P(h')}$$

Note: this requires to have prior probabilities (i.e. someone's beliefs, or notion of simplicity).

Probabilities and Proofs

What do we gain using probability calculus?

- Nothing: if the starting point is wrong, the endpoint also.
- Probability-based results are not more valid than others (e.g. using Bayes rule does not guarantee anything)
- Probabilities simply provide a nice framework for reasoning (with uncertainty).

To relate probabilities to a real-world phenomenon, one needs an "interpretative hypothesis" (e.g. events with probability zero never occur).

The need for assumptions

- We mentioned we want to "predict" or "generalize"
- This can work only if
  - ⋆ the future looks like the past (prediction/forecasting)
  - ⋆ or the unseen looks like the seen (generalization)
- Hence we need to assume some kind of common underlying phenomenon

The need for assumptions (2)

- Our goal is to keep assumptions to a minimum
- Can we completely avoid assumptions?
- Yes if we aim at competitive results: we do not aim at predicting well, but predicting almost as well as someone else.

Hence there are two points of view:

- Given assumptions, what can we do at best?
- Given no assumptions, can we match others' performance?

Settings

Examples of commonly considered settings

1. Off-line supervised learning: training pairs are given, the goal is to produce a model that predicts well
2. Semi-supervised: training pairs and unlabeled data are given, the goal is to produce a model that predicts well
3. Transductive: training pairs and unlabeled data are given, the goal is to predict well on the unlabeled examples
4. On-line: repeated transductive learning (one new example at a time, prediction to be performed at each step)
5. Variants of on-line: actions instead of predictions (e.g. reinforcement learning)

Settings vs Assumptions

It is important to distinguish between the assumptions, the goals and the algorithms!

- Data generation mechanism: only relevant for proving theorems.
- Protocol: important for designing algorithms
- Success measure: sometimes impossible to practically measure. May serve as a guide for designing algorithms.
- Type of analysis: what we want to prove.
- Further restrictive assumptions: used for designing algorithms or proving restricted results.

Settings and Algorithms

→ Algorithms may do something different than optimizing directly the success (often impossible, or intractable)

→ Algorithms may perform well under various settings

Data Generation Mechanisms

- Bayes: the function is sampled and the data also
- IID: the data is sampled iid from unknown P
- Transduction: the data is fixed, the split is random
- On-line stochastic: random source but not necessarily independent (Markov...)
- On-line deterministic: no assumption, the data and its order is fixed
- On-line adversarial: the data is generated by an opponent

Protocols

- Off-line: examples given altogether
- On-line: examples given one at a time (order may or may not matter)

Information available to the learner may differ

- Error function
- Error value at past predictions
- Error of other predictors

Success Measures

- Off-line: Expected error (average error on future instances generated in the same way)
- On-line: Cumulated error (sum of the errors made at each step)
- On-line: Expected future cumulated error (average cumulated error on future instances generated in the same way)

Type of analysis

- Worst-case: performance under the worst possible data generation mechanism
- Average-case: average over possible data generation mechanisms (requires some weighting)
- This case: performance on the given problem

What do we want to prove for a given learning algorithm?

- Expectation vs Probability
- Relative error bounds
- Oracle inequalities

Example: Bayesian Inference

- Data generation mechanism: function sampled from prior, data sampled iid from prior and labeled by function.
- Protocol: off-line, all examples given at once.
- Success measure: expected error
- Type of analysis: average under the prior
- Further restrictive assumptions: noise of a specific form/intensity

These can be considered independently: one can use the same setting but perform a different analysis, or consider different generation mechanisms (e.g. IID)
Also, Bayes rule can be used in other settings. It is only optimal under all the above assumptions and for the above type of analysis.

Our Goal

We will

- consider a worst-case analysis
- try to avoid assumptions as much as possible
- look for best algorithms

The quantity of interest looks like

$$\min_{\text{predictor}} \max_{\text{problem}} \text{Success(predictor, problem)}$$

Is this reasonable?

- We will show that this has no reason to be small. This will be the no-free-lunch theorems. They come in various flavours depending on how deep you dig.

- There are several ways in which you can make this quantity high: for fixed n, for varying n and fixed problem...

Minimax estimation

The classical minimax approach is to consider the following quantity:

$$\min_{\text{predictor}} \max_{\text{problem in a class}} \text{Success(predictor, problem)}$$

Sometimes (if minimum success is not zero) one considers

$$\min_{\text{predictor}} \max_{\text{problem}} \text{Success(predictor, problem)} - \text{Best(problem)}$$

This is fine if we can guarantee that the data generation mechanism is indeed of the posited form. If this is not the case, this type of quantity is not very useful.

Learnability

- Another similar point of view is to tell whether a class can be learned.

- Several settings (identification in the limit, inductive inference, PAC learning)

- The question is whether, for a given class of functions, one can design an algorithm such that any function in the class would be recovered by the algorithm with enough training data.

Model identification

Sometimes one even measures the success by how similar is the predictive model to the "true" model.

- This is not interesting in practice (one can never verify the identity of two models since most often one can only access data)

- We prefer error minimization

- Similarly, you can hope but not ensure that the best function is in the class you consider

Updated goal

We prefer a competitive approach.
Indeed, we do not want to impose restrictions on the way the data is generated, but instead compare our performance to some references.

$$\min_{\text{predictor}} \max_{\text{problem,reference}} \text{Success(predictor, problem)} - \text{Success(reference, problem}$$

This is an important point!

Upper and lower bounds

- To prove an upper bound: exhibit an algorithm for which you can prove a bound on the error which is independent of the data generation mechanism
- To prove a lower bound: for each possible algorithm, exhibit a data generation mechanism which misleads the algorithm

Notation

Let us introduce (finally) some notation

- $\mathcal{X}$ input space (often $\mathcal{X} \subset \mathbb{R}^d$)
- $\mathcal{Y}$ output space (often $\mathcal{Y} = \{0, 1\}$)
- $n \in \mathbb{N}$ sample size, i.e. number of observed pairs so far $(X_i, Y_i) \in \mathcal{X} \times \mathcal{Y}$
- $\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$ loss function (often $1_{[y \neq y']}$)

Notation (2)

- Cumulated loss for $f : \mathcal{X} \to \mathcal{Y}$:

$$L(f) = \sum_{i=1}^{n} \ell(f(x_i), y_i)$$

(in the on-line setting, $f(x_i)$ may depend on the past observations)

- Expected loss

$$L(f) = \mathbb{E}\ell(f(X), Y)$$

- Predictor constructed from n observations: f_n

Minimax lower bounds: weak and strong

Consider the quantity

$$\min_{f_n} \max_{P} L(f_n)$$

- To prove a lower bound, we need to find a "bad" P.
- This P may depend on f_n and on n.
- Whether it depends on n or not makes a difference.
- Indeed, one is often interested as the decrease as n increases.

No free lunch

Simplest form due to [Wolpert96]

> On average over all possible data generation mechanisms (assuming there is a finite number of them), all algorithms have the same error when measured on future instances

Interpretation

- If A_1 is better than A_2 on a problem, it will be worse on another one
- One has to restrict the considered problems
- "my algorithm is better" means "the prior implemented by my algorithm is better suited to these databases"

So what do we do?

- Given values of f: $f(x_1), \ldots, f(x_n)$, what could be the value at some $x \neq x_1, \ldots, x_n$?
- If we allow all possible functions, it is clear that no generalization will be possible: all values are equally possible
- We need to make choices and express preferences or assumptions about the possible forms of functions we expect: we need to choose a prior
- Data only does not lead to generalization (if all possibilities look the same, nothing will be inferred).

The IID assumption

- Examples are pairs (Xi,Yi) drawn
  - ⋆ independently
  - ⋆ from an unknown
  - ⋆ but fixed distribution P
- Quantity of interest: expected loss

$$\mathbb{E}L(f_n)$$

Expectation over the possible training samples

Criticism of NFL1 under IID

- In the finite setting of Wolpert, with enough examples one achieves minimal error (fixed problem, n increasing)
- Wolpert considers off-sample error. But there is a slight difference between expected and off-sample errors ($\sqrt{1 + r \log n / n}$ where r is the number of repetitions in the training set).
- In the continuous case ($\mathcal{X}$ continuous) they are the same.

Good and Bad news

- Good news: there exists universally consistent algorithms
  - ⋆ When the sample size goes to infinity, the error of the algorithm converges to the best possible error (under P).
- Bad news: this does not help (Devroye et al. 96)
  - ⋆ NFL2: for a fixed sample size, the error of the algorithm can be arbitrarily close to the worst possible error (for some P)
  - ⋆ NFL3: the rate of convergence of a universally consistent algorithm can be arbitrarily slow (for some P)

Consistency under IID

Usually consistency means

$$\mathbb{E}L(f_n) - L^* \to 0$$

when n increases.
In a countable space, with enough data, all $x \in \mathcal{X}$ will eventually be observed (several times) so that consistency can be achieved without generalization!
In a continuous space, measurability comes to the rescue.

No-Free-Lunch 2

Finite training sequence
Binary classification setting.
Theorem 1. [DGL96, Thm 7.1] *For any $\epsilon > 0$, any n and any algorithm f_n, there exists P (with $L^* = 0$) such that*

$$\mathbb{E}L(f_n) \geq 1/2 - \epsilon.$$

Even though there are consistent algorithms, on finite samples their performance may be arbitrarily bad.
But this is a 'weak' statement (P depends on n).

No-Free-Lunch 3

Slow rates

Theorem 2. [DGL96, Thm 7.2] *For any non-increasing sequence a_n converging to zero, and any algorithm f_n, there exists P (with $L^* = 0$) such that*

$$\mathbb{E}L(f_n) \geq a_n.$$

This is a 'strong' statement.

Proof ideas

- NFL2: construct a problem with features such that the solution is one of the features. If this has not been observed yet the performance will be bad. Choose the probabilities such that this occurs often.
- NFL3: same thing on a countable space (the probability mass of the unseen example has to be larger than a_n)

Comments

- Consistency is easy in the countable case
- Consistency relies on measurability in the continuous case
- No-Free-Lunch results only rely on not having seen enough (off-sample error)