Foundations for Machine Learning

L. Y. Stefanus

TU Dresden, June-July 2019

machine learning: automated detection of meaningful patterns in data

SOURCE: DRIBBBLE.COM BY DALANEY LAGRANGE

Synopsis

The topic of this course is mathematical foundations for Machine Learning. This course provides a theoretical account of the fundamental ideas underlying machine learning and the mathematical derivations that transform these principles into practical algorithms, such as algorithms appropriate for big data learning. We will discuss Valiant's PAC (Probably Approximately Correct) learning model, the ERM (Empirical Risk Minimization) learning rule, the No-Free-Lunch Theorem, the VC (Vapnik-Chervonenkis) dimension, and CNN (convolutional neural networks).

We will also look into implementations in PyTorch, a popular machine learning software tool using the programming language Python.

References

- Eli Stevens and Luca Antiga. Deep Learning with PyTorch. Manning Publications, 2019/2020.
- Shai Shalev-Shwartz and Shai Ben-David. Understanding Machine Learning: From Theory to Algorithms. Cambridge University Press, 2014.
- Ian Goodfellow and Yoshua Bengio and Aaron Courville. Deep Learning. MIT Press, 2016.

Prerequisites:

- Probability Theory
- Linear Algebra
- Algorithm Design & Analysis
- Python Programming

Schedule

Week 1

- Monday, 17th June from 14:50 - 16:20 pm in APB 2026
- Tuesday, 18th June from 14:50 - 16:20 pm
- Friday, 21st June from 09:20 - 10:50 am

Week 2

- Monday, 24th June from 14:50 - 16:20 pm
- Tuesday, 25th June from 14:50 - 16:20 pm
- Wednesday, 26th June from 14:50 - 16:20 pm
- Thursday, 27th June from 16:40 - 18:10 pm
- Friday, 28th June from 09:20 - 10:50 am

Schedule (cont.)

Week 3

- Monday, 1st July from 14:50 - 16:20 pm
- Tuesday, 2nd July from 14:50 - 16:20 pm
- Wednesday, 3rd July from 14:50 - 16:20 pm
- Thursday, 4th July from 16:40 - 18:10 pm
- Friday, 5th July from 09:20 - 10:50 am

Exam

The written examination will be on Monday, 8th July 2019 from 14:50 - 16:20 pm at HÜLβEBAU/S186/H

What is learning?

- Learning is the process of converting experience into expertise or knowledge.
- The input to a learning algorithm is training data (representing experience), and the output is some expertise (taking the form of a computer program that can perform some task).

Machine Learning Theory

- Inductive inference and generalisation
  - Should be able to predict on unseen examples
- Fundamental Questions
  - How to learn? What is learnable ?
  - How can we know that what we learned is true?

Problems:

- What is the training data our programs will access?
- How can the process of learning be automated?
- How can we evaluate the success of such a process (namely, the quality of the output of a learning program)?

- Let’s look at some examples from naturally occurring animal learning.
- Some of the most fundamental issues in ML arise already in that context.

Bait Shyness

- Rats Learning to Avoid Poisonous Baits: When rats encounter food items with novel look or smell, they will first eat very small amounts, and subsequent feeding will depend on the flavor of the food and its physiological effect. If the food produces an ill effect, the novel food will often be associated with the illness, and subsequently, the rats will not eat it.

- There is a learning mechanism in play here.
- The animal used past experience with some food to acquire expertise in detecting the safety of this food. If past experience with the food was negatively labeled, the animal predicts that it will also have a negative effect when encountered in the future.

A person who eats a novel food and then gets ill shortly after, whether or not the food caused the sickness, may become so averse to the food as to never be able to eat it again. Studies exploring how such taste aversions are formed have reshaped theories of learning.

“TASTE, SICKNESS, AND LEARNING”
BY TERRY L. DAVIDSON, ANTHONY L. RILEY.
AMERICAN SCIENTIST,
MAY-JUNE 2015,
VOLUME 103, NUMBER 3
PAGE 204.

Pigeon Superstition

- The psychologist B. F. Skinner placed a bunch of hungry pigeons in a cage. An automatic mechanism had been attached to the cage, delivering food to the pigeons at regular intervals with no reference whatsoever to the birds' behavior.
- The hungry pigeons went around the cage, and when food was first delivered, it found each pigeon engaged in some activity (pecking, turning the head, etc.). The arrival of food reinforced each bird's specific action, and consequently, each bird tended to spend some more time doing that very same action.

Pigeon Superstition

- That, in turn, increased the chance that the next random food delivery would find each bird engaged in that activity again.
- What results is a chain of events that reinforces the pigeons' association of the delivery of the food with whatever chance actions they had been performing when it was first delivered. They subsequently continue to perform these same actions diligently.

Pigeon Superstition
and Operant Conditioning (Learning)

https://www.youtube.com/watch?v=I_ctJqjlrHA

Bait Shyness vs Pigeon Superstition

- What made the rats' learning more successful than that of the pigeons?
- What distinguishes learning mechanisms that result in superstition from useful learning?
- This question is crucial to the development of automated learners.
- While human learners can rely on common sense to filter out random meaningless learning conclusions, we must provide well defined principles that will protect the program from reaching senseless or useless conclusions.
- The development of such principles is a central goal of the theory of machine learning.

Bait Shyness and Prior Knowledge

- Rats fail to acquire conditioning between food and electric shock or between sound and nausea.
- The bait shyness mechanism in rats turns out to be more complex than what one may expect. In experiments carried out by Garcia (Garcia & Koelling 1996), it was demonstrated that if the unpleasant stimulus that follows food consumption is replaced by, say, electrical shock (rather than nausea), then no conditioning occurs. Even after repeated trials in which the consumption of some food is followed by the administration of unpleasant electrical shock, the rats do not tend to avoid that food. Similar failure of conditioning occurs when the characteristic of the food that implies nausea (such as taste or smell) is replaced by a vocal signal.
- The rats seem to have some built in prior knowledge telling them that, while temporal correlation between food and nausea can be causal, it is unlikely that there would be a causal relationship between food consumption and electrical shocks or between sounds and nausea.

- One distinguishing feature between the bait shyness learning and the pigeon superstition is the incorporation of prior knowledge that biases the learning mechanism. This is also referred to as inductive bias.
- The pigeons in the experiment are willing to adopt any explanation for the occurrence of food. However, the rats “know" that food cannot cause an electric shock and that the co-occurrence of noise with some food is not likely to affect the nutritional value of that food. The rats' learning process is biased toward detecting some kind of patterns while ignoring other temporal correlations between events.

- It turns out that the incorporation of prior knowledge, biasing the learning process, is inevitable for the success of learning algorithms (this will be formally stated as the No-Free-Lunch theorem later).
- The development of tools for expressing domain expertise, translating it into a learning bias, and quantifying the effect of such a bias on the success of learning is a central theme of the theory of machine learning.
- The stronger the prior knowledge (or prior assumptions) that one starts the learning process with, the easier it is to learn from further examples. However, the stronger these prior assumptions are, the less flexible the learning is. More on these issues later.

When Do We Need Machine Learning?

- Two aspects of a given problem may call for the use of programs that learn and improve on the basis of their “experience”: the problem's complexity and the need for adaptivity.

Tasks That Are Too Complex to Program

- Tasks Performed by Animals/Humans: There are numerous tasks that we human beings perform routinely, yet our introspection concerning how we do them is not sufficiently elaborate to extract a well defined program. Examples of such tasks include driving, speech recognition, and image understanding.

- Tasks beyond Human Capabilities: Another wide family of tasks that benefit from machine learning techniques are related to the analysis of very large and complex data sets: astronomical data, turning medical archives into medical knowledge, weather prediction, analysis of genomic data, Web search engines, and electronic commerce.

- Adaptivity. One limiting feature of programmed tools is their rigidity. Once the program has been written down and installed, it stays unchanged. However, many tasks change over time or from one user to another. Machine learning tools -- programs whose behavior adapts to their input data -- offer a solution to such issues; they are, by nature, adaptive to changes in the environment they interact with. Typical successful applications of machine learning to such problems include programs that decode handwritten text, spam detection programs, and speech recognition programs.

Types of Machine Learning

- Supervised vs Unsupervised
  - Ex. Spam detection vs outlier detection
  - Intermediate scenario: reinforcement learning
- Batch vs Online
- Teacher: Cooperative vs Indifferent vs Adversarial
- Learner: Active vs Passive

Many applications

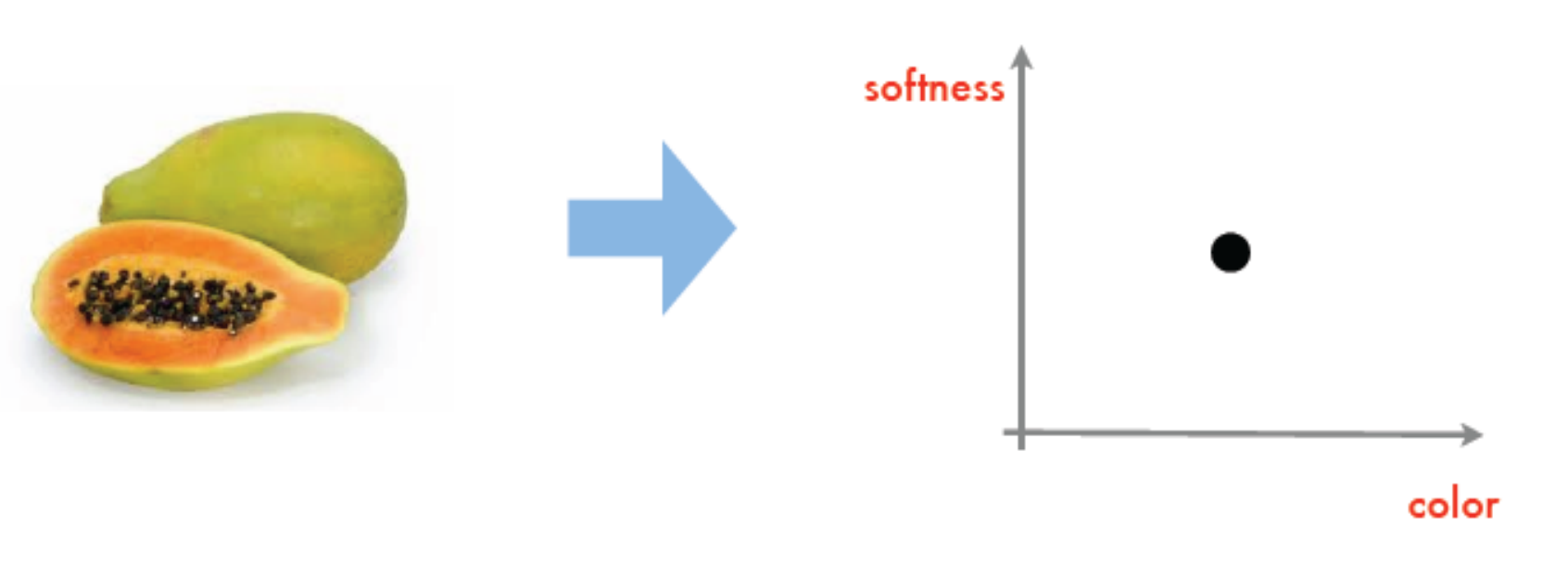
- AI: Object recognition, face detection, autonomous driving, text categorization, speech-to-text, voice recognition, ...
- Science: Gene expression, drug design, medical imaging, climate, astronomy, …
- Web applications: Search engines, spam detection, machine translation ...
- Economy: E-commerce, trades, ...

A Show Case: Papaya Tasting

- We've just arrived in some small pacific island
- We soon find that papayas are a significant ingredient in the local diet
- How can we know if a papaya is tasty?

A Show Case: Papaya Tasting

- Based on previous experience with other fruits, we decide to use two features:



A Show Case: Papaya Tasting

- Our goal is to find a prediction rule:

Formal Model

- Domain set, X: This is the set of objects that we may wish to label.
- Label set, Y: The set of possible labels.
- A prediction rule, $h : X \rightarrow Y$: used to label future examples. This function is called a predictor, a hypothesis, or a classifier.

Example

- $X = R^2$ representing color and softness of papayas.
- $Y = \{-1, +1\}$ or $\{0, 1\}$ representing tasty or non-tasty
- $h(x) = 1$ if x is within the inner rectangle