Machine learning theory

Introduction

Hamid Beigy

Sharif University of Technology

February 6, 2023

Table of contents

Introduction

What is machine learning?

Definition (Mohri et. al., 2018)

Computational methods that use experience to improve performance or to make accurate predictions.

Definition (Mitchell, 1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.

Example (Spam classification)

- Task: determine if emails are spam or non-spam.
- Experience: incoming emails with human classification.
- Performance Measure: percentage of correct decisions.

Why we need machine learning?

We need machine learning because

1. Tasks are too complex to program but they are performed by animals/humans such as driving, speech recognition, image understanding, and etc.
2. Tasks beyond human capabilities such as weather prediction, analysis of genomic data, web search engines, and etc.
3. Some tasks need adaptivity. When a program has been written down, it stays unchanged. In some tasks such as optical character recognition and speech recognition, we need the behavior to be adapted when new data arrives.

Types of machine learning

Machine learning algorithms based on the information provided to the learner can be classified into different groups.

1. Supervised/predictive vs unsupervised/descriptive vs reinforcement learning.
2. Batch vs online learning
3. Passive vs Active learning.
4. Cooperative vs adversarial teachers.

Applications of machine learning i

1. Supervised learning:
    - Classification:
        - Document classification and spam filtering.
        - Image classification and handwritten recognition.
        - Face detection and recognition.
    - Regression:
        - Predict stock market price.
        - Predict temperature of a location.
        - Predict the amount of PSA.
2. Unsupervised/descriptive learning:
    - Discovering clusters.
    - Discovering latent factors.
    - Discovering graph structures (correlation of variables).
    - Matrix completion (filling missing values).
    - Collaborative filtering.
    - Market-basket analysis (frequent item-set mining).
3. Reinforcement learning:
    - Game playing.
    - robot navigation.

The need for probability theory

- A key concept in machine learning is uncertainty.
- Data comes from a process that is not completely known.
- This lack of knowledge is indicated by modeling the process as a random process.
- The process actually may be deterministic, but we don't have access to complete knowledge about it, we model it as random and we use the probability theory to analyze it.

Supervised learning

Supervised learning

- In supervised learning, the goal is to find a mapping from inputs X to outputs t given a labeled set of input-output pairs

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_m, t_m)\}.$$

  S is called training set.

- In the simplest setting, each training input x is a $D-$dimensional vector of numbers.
- Each component of x is called feature, attribute, or variable and x is called feature vector.
- In general, x could be a complex structure of object, such as an image, a sentence, an email message, a time series, a molecular shape, a graph.
- When $t_i \in \{-1, +1\}$ or $t_i \in \{0, 1\}$, the problem is classification.
- When $t_i \in \mathbb{R}$, the problem is known as regression.

Supervised learning

Classification

Classification

- Given a hypothesis space H, the learning algorithm should find a particular hypotheses $h \in H$ to approximate C as closely as possible.
- We choose H and the aim is to find $h \in H$ that is similar to C. This reduces the problem of learning the class to the easier problem of finding the parameters that define h.
- Hypothesis h makes a prediction for an instance x in the following way.

$$h(x) = \begin{cases} 1 & \text{if } h \text{ classifies } x \text{ as a positive example} \\ 0 & \text{if } h \text{ classifies } x \text{ as a negative example} \end{cases}$$

Classification (Cont.)

- In real life, we don't know $c(x)$ and hence cannot evaluate how well $h(x)$ matches $c(x)$.
- We use a small subset of all possible values x as the training set as a representation of that concept.
- Empirical error (risk)/training error is the proportion of training instances such that $h(x) \neq c(x)$.

$$\hat{\mathbf{R}}(h) = \frac{1}{m}\sum_{i=1}^{m} \mathbb{I}\left[h(x_i) \neq c(x_i)\right]$$

- When $\hat{\mathbf{R}}(h) = 0$, h is called a consistent hypothesis with dataset S.
- For many examples, we can find infinitely many h such that $\hat{\mathbf{R}}(h) = 0$. But which of them is better than for prediction of future examples?
- This is the problem of generalization, that is, how well our hypothesis will correctly classify the future examples that are not part of the training set.

Classification (Generalization)

- The generalization capability of a hypothesis usually measured by the true error/risk.

$$\mathbf{R}(h) = \underset{x \sim \mathcal{D}}{\mathbb{P}} [h(x) \neq c(x)]$$

- We assume that *H includes C*, that is there exists $h \in H$ such that $\hat{\mathbf{R}}(h) = 0$.
- Given a hypothesis class H, it may be the cause that we cannot learn C; that is there is no $h \in H$ for which $\hat{\mathbf{R}}(h) = 0$.
- Thus in any application, we need to make sure that H is flexible enough, or has enough capacity to learn C.

Supervised learning

Regression

Regression

- In regression, $c(x)$ is a continuous function. Hence the training set is in the form of

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_m, t_m)\}, t_k \in \mathbb{R}.$$

- In regression, there is noise added to the output of the unknown function.

$$t_k = f(x_k) + \epsilon \qquad \forall k = 1, 2, \ldots, m$$

  $f(x_k) \in \mathbb{R}$ is the unknown function and ϵ is the random noise.

- The explanation for the noise is that there are extra hidden variables that we cannot observe.

$$t_k = f^*(x_k, z_k) + \epsilon \qquad \forall k = 1, 2, \ldots, N$$

  z_k denotes hidden variables

- Our goal is to approximate the output by function $g(x)$.
- The empirical error on the training set S is

$$\hat{\mathbf{R}}(h) = \frac{1}{m}\sum_{k=1}^{m}\left[t_k - g(x_k)\right]^2$$

- The aim is to find $g(.)$ that minimizes the empirical error.
- We assume that a hypothesis class for $g(.)$ has a small set of parameters.

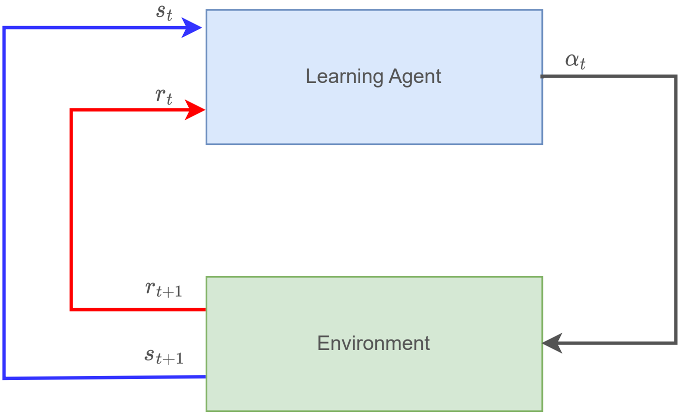
Reinforcement learning

Introduction

- Reinforcement learning is what to do (how to map situations to actions) so as to maximize a scalar reward/reinforcement signal
- The learner is not told which actions to take as in supervised learning, but discover which actions yield the most reward by trying them.
- The trial-and-error and delayed reward are the two most important feature of reinforcement learning.
- Reinforcement learning is defined not by characterizing learning algorithms, but by characterizing a learning problem.
- Any algorithm that is well suited for solving the given problem, we consider to be a reinforcement learning.
- One of the challenges that arises in reinforcement learning and other kinds of learning is tradeoff between exploration and exploitation.

Introduction

- A key feature of reinforcement learning is that it explicitly considers the whole problem of a goal-directed agent interacting with an uncertain environment.



Unsupervised learning

Introduction

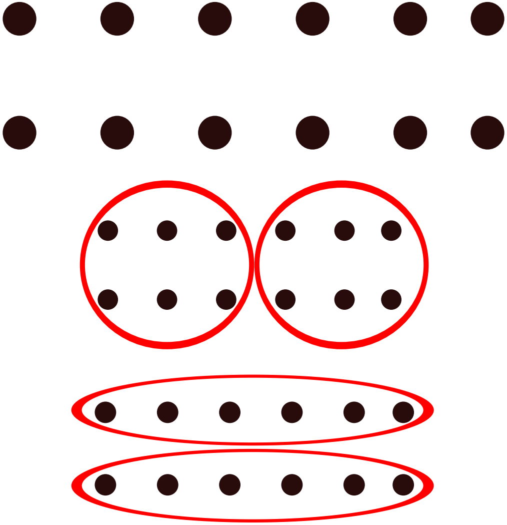
- Unsupervised learning is fundamentally problematic and subjective.
- Examples :
  1. Clustering : Find natural grouping in data.
  2. Dimensionality reduction : Find projections that carry important information.
  3. Compression : Represent data using fewer bits.
- Unsupervised learning is like supervised learning with missing outputs (or with missing inputs).

Clustering

- Given data

$$X = \{x_1, x_2, \dots, x_m\}$$

learn to understand the data, by re-representing it in some intelligent way.



Machine learning theory

Introduction

What is machine learning theory?

1. What are the intrinsic properties of a given learning problem that make it hard or easy to solve?
2. How much do you need to know ahead of time about what is being learned in order to be able to learn it effectively?
3. Why are simpler hypotheses better?
4. How do we formalize machine learning problems (for eg. online , statistical)?
5. How do we pick the right model to use and what are the tradeoffs between various models?
6. How many instances do we need to see to learn to given accuracy?
7. How do we design learning algorithms with provable guarantees on performance?

Example

Problem

1. *Suppose that you have a coin that has an unknown probability θ of coming up heads.*
2. *We must determine this probability as accurately as possible using experimentation.*

1. Let the outcome of a single toss at trial i denoted by

$$x_i = \begin{cases} 1 & \text{head} \\ 0 & \text{tail} \end{cases}$$

2. Tossing the coin m times and recording the outcomes as $x_1, \ldots, x_m$.
3. Since outcomes $x_1, \ldots, x_m$ are independet, then, $\mathbb{P}[x_i = 1] = \theta$. What would be a reasonable estimate of θ?
4. By Law of Large Numbers, in a long sequence of independent coin tosses, the relative frequency of heads will eventually approach the true value of θ with high probability.

$$\hat{\theta} = \frac{1}{m} \sum_i x_i$$

5. Using Chernoff bound, we have $\mathbb{P}\left[|\hat{\theta} - \theta| > \epsilon\right] \leq 2e^{-2\epsilon^2 m}$ and $m \geq \frac{1}{2\epsilon^2} \log\left(\frac{2}{\delta}\right)$, where $1 - \delta$ specifies the confidence of estimation.

Machine learning theory

1. There are two basic questions
   1.1 How large of a sample do we need to achieve a given accuracy with a given confidence?
   1.2 How efficient can our learning algorithm be?
2. The first question is within statistical learning theory.
3. The second question is within computational learning theory.
4. However, there are some overlaps between these two fields.

Outline of course

Outline of course i

Outline of course

1. Introduction
2. Part 1 (Theoretical foundation)
    - 2.1 Consistency and PAC models
    - 2.2 Learning by uniform convergence
    - 2.3 Empirical and structural risk minimization
    - 2.4 Growth functions, VC-dimension, covering number, ...
    - 2.5 Learning by non-uniform convergence, SRM, and MDL
    - 2.6 Generalization bounds
    - 2.7 Regularization and stability of algorithms
    - 2.8 Analysis of kernel learning
    - 2.9 Computational complexity and running time of learning algorithms
    - 2.10 PAC-MDP model for reinforcement learning
    - 2.11 Theoretical foundation of clustering
    - 2.12 Convex learning problems
3. Part 2 (Analysis of algorithms)
    - 3.1 Linear classification
    - 3.2 Boosting
    - 3.3 SVM and Kernel based learning
    - 3.4 Regression

Outline of course ii

- 3.5 Learning automata
- 3.6 Reinforcement learning
- 3.7 Ranking
- 3.8 Online learning
- 3.9 Active learning
- 3.10 Semi-supervised learning
- 3.11 Deep learning

4. Part 3 (Advanced topics)
    - 4.1 Radamacher Complexity
    - 4.2 PAC-Bayes theory
    - 4.3 Universal learning
    - 4.4 Other advanced topics
5. Part 4 (Applications)
    - 5.1 Learnability of some problems in Social networks
    - 5.2 Analysis of some graph-based problems
    - 5.3 Generalization bound for Domain adaptation
    - 5.4 Generalization bound for sampling-based algorithms

Course evaluation

1. Evaluation:

Mid-term exam	25%	1402-02-06
Final exam	25%	
Homeworks	30%	
Quiz	10%	
Paper & Project	10%	Explore a theoretical or empirical question and present it.

2. Deadline for choosing paper: 1401-02-06

   You must send the selected paper to Mrs. Mirbeygi with me as a cc.

3. Class Link: `https://vc.sharif.edu/beigy`
4. Course Website:

   `http://sharif.edu/~beigy/14012-40718.html`
5. TAs :

   Mohaddeseh Mirbeygi

Relevant journals

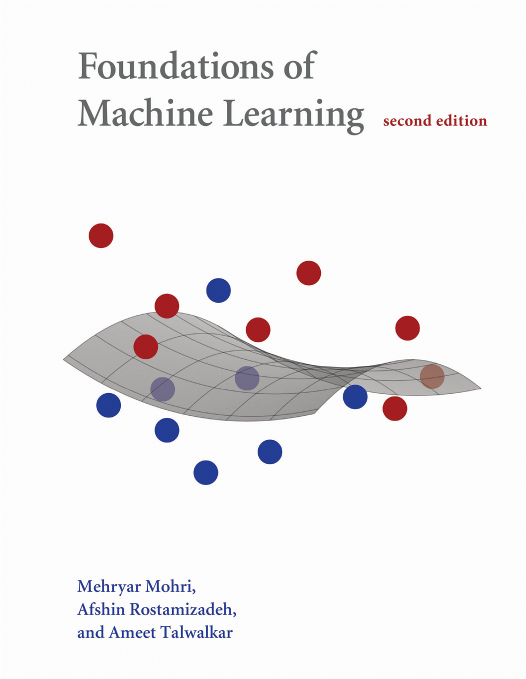
1. IEEE Trans on Pattern Analysis and Machine Intelligence
2. Journal of Machine Learning Research
3. Pattern Recognition
4. Machine Learning
5. Neural Networks
6. Neural Computation
7. Neurocomputing
8. IEEE Trans. on Neural Networks and Learning Systems
9. Annuals of Statistics
10. Journal of the American Statistical Association
11. Pattern Recognition Letters
12. Artificial Intelligence
13. Data Mining and Knowledge Discovery
14. IEEE Transaction on Cybernetics (SMC-B)
15. IEEE Transaction on Knowledge and Data Engineering
16. Knowledge and Information Systems

Relevant conferences

1. Neural Information Processing Systems (NIPS)
2. International Conference on Machine Learning (ICML)
3. European Conference on Machine Learning (ECML)
4. Asian Conference on Machine Learning (ACML)
5. Conference on Learning Theory (COLT)
6. Algorithmic Learning Theory (ALT)
7. Conference on Uncertainty in Artificial Intelligence (UAI)
8. Practice of Knowledge Discovery in Databases (PKDD)
9. International Joint Conference on Artificial Intelligence (IJCAI)
10. IEEE International Conference on Data Mining series (ICDM)

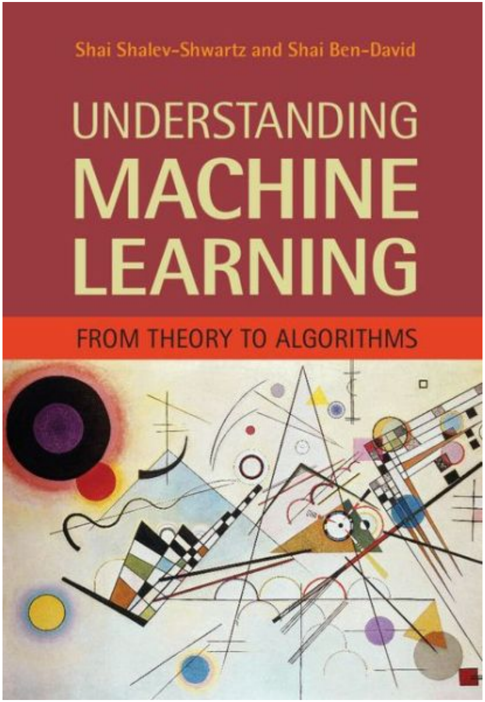
References

Main references



Foundations of Machine Learning second edition

Mehryar Mohri,
Afshin Rostamizadeh,
and Ameet Talwalkar



Shai Shalev-Shwartz and Shai Ben-David

UNDERSTANDING MACHINE LEARNING

FROM THEORY TO ALGORITHMS

Readings

1. Chapter 1 of Foundations of machine learning (Mohri, Rostamizadeh, and Talwalkar 2018).
2. Chapter 1 of Understanding machine learning: From theory to algorithms (Shalev-Shwartz and Ben-David 2014).

References

- Anthony, Martin and Peter L. Bartlett (1999). *Learning in Neural Networks : Theoretical Foundations*. Cambridge University Press.
- Anthony, Martin and Norman Biggs (1992). *Computational Learning Theory : An introduction*. Cambridge University Press.
- Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Berlin, Heidelberg: Springer-Verlag.
- Cesa-Bianchi, Nicolò and Gábor Lugosi (2006). *Prediction, learning, and games*. Cambridge University Press.
- Devroye, Luc, Laszlo Gyorfi, and Gabor Lugosi (1996). *A probabilistic theory of pattern recognition*. Springer.
- Kearns, Michael J. and Umesh Vazirani (1994). *An Introduction to Computational Learning Theory*. MIT Press.
- Mohri, Mehryar, Afshin Rostamizadeh, and Ameet Talwalkar (2018). *Foundations of Machine Learning*. Second Edition. MIT Press.
- Shalev-Shwartz, Shai and Shai Ben-David (2014). *Understanding machine learning: From theory to algorithms*. Cambridge University Press.

Questions?