6.867 Machine Learning Sep 5, 2019

Lecture 1: Introduction

Instructors: Devavrat Shah (Lecturer), David Sontag, Suvrit Sra *Scribes: Huitao Shen*

1 Background

Logistics. See the Information Sheet on the stellar site.

What is Machine Learning. The term "Machine Learning" was coined by MIT alumnus Arthur Samuel[1] in 1959. It evolved from many fields including Statistical Learning, Pattern Recognition and so on. The goal of machine learning is to make computers "learn" from "data"[2]. From an end user's perspective, it is about understanding your data, make predictions and decisions. Intellectually, it is a collection of models, methods and algorithms that have evolved over more than a half-century now.

Machine Learning vs Statistics. Historically both disciplines evolved from different perspectives, but with similar end goals. For example, Machine Learning focused on "prediction" and "decisions". It relied on "patterns" or "model" learnt in the process to achieve it. Computation has played key role in its evolution. In contrast, Statistics, founded by statisticians such as Pearson and Fisher, focused on "model learning". To understand and explain "why" behind a phenomenon. Probability has played key role in development of the field. As a concrete example, recall the ideal gas law $PV = nRT$ for Physics. Historically, machine learning only cared about ability to predict P by knowing V and T, did not matter how; on the other hand, Statistics did care about the precise form of the relationship between P, V and T, in particular it being linear. Having said that, in current day and age, both disciplines are getting closer and closer, day-by-day, and this class is such an amalgamation.

Machine Learning vs Artificial Intelligence. Artificial Intelligence's stated goal is to *mimic human behavior in an intelligent manner*, and to do what humans can do but really well, which includes artificial "creativity" and driving cars, playing games, responding to consumer questions, etc. Traditionally, the main tools to achieve these goals are "rules" and "decision trees". In that sense, Artificial intelligence seeks to create *muscle* and *mind* of humans, and *mind* requires learning from data, i.e. Machine Learning. However, Machine Learning helps learn from data beyond mimicking humans. Having said that, again the boundaries between AI and ML are getting blurry day-by-day.

2 Course Structure

The course contains four parts:

- Part I. Supervised Learning (L2-11, 43%). Learning from data to predict.
- Part II. Unsupervised Learning (L12-18, 30%). Understanding the structure within the data.
- Part III. Probabilistic Modeling (L19-20, 9%). Probabilistic view to model complex scenarios.
- Part IV. Decision Making (L21-24, 18%). Using data to make decisions.

[1] See `https://g.co/kgs/Lj3v3k` to read more about Arthur Samuel.
[2] What is learning? Some food for thought: `https://goo.gl/5R1m4S`.

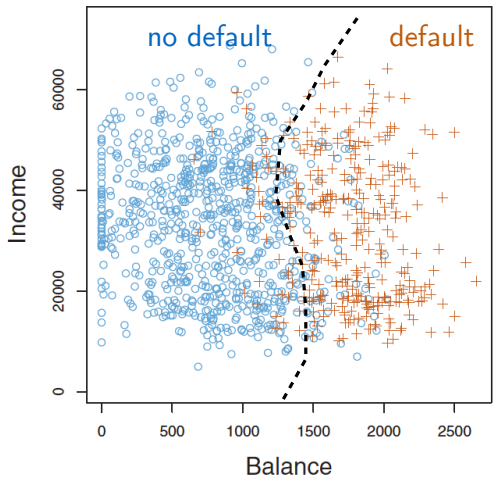
2.1 Supervised Learning

The goal of supervised learning is to predict *target* using *input* / *features*, and a model is learned to do so. This can be sufficiently summarized as

$$target = f(features)$$

For classification tasks, the *target* is categorical or takes discrete values (e.g. hot or cold). For regression tasks, the *target* takes any real value (e.g. temperature). The model type reflects our *belief* about the reality and different model leads to different algorithm. The philosophy of supervised learning is: *future of the past equals future of the future*.

Classification. Examples of classification include: identify handwritten digits, email spam filtering, detecting malicious network connection based on network log information or predicting whether a client will default on her/his credit based on the client's features. For example, suppose we have access to a client's features or attributes in terms of the (credit card) balance and income. Consider Figure 2.1. It plots available data with X axis representing (credit card) balance and Y axis representing income. The color of the point is blue if *no default* and brown if *default*. Pictorially, the classifier is trying to learn a boundary as shown in Figure 2.1 which separate *no default* from *default*.



Formally, the data are labeled observations of the following form: $(x_1, y_1), \ldots, (x_N, y_N)$. The goal is to learn a model that maps *attribute* (or *feature*) x to *label* (or *target*) y so that given *attribute* x, we can predict corresponding *unknown (discrete) label* y. That is, to learn a function f such that $y = f(x)$ (and sometimes also whats the *confidence*).

Model. Various approaches for learning f can be categorized as

- Linear: Logistic regression, Support Vector Machine (SVM), Linear Discriminant Analysis (LDA), Perceptron
- Non-linear (parametric): Quadratic Discriminant Analysis (QDA), Polynomial, Neural Networks
- Non-parametric: Kernels, Nearest Neighbors

Algorithm. How to find f? Among all possible choices of f, choose the one that *fits* the data the best. That is, solve optimization: *empirical risk minimization (ERM)*:

$$\text{Minimize} \sum_{i=1}^{N} \text{loss}\,(y_i, f(x_i)) \text{ over all possible } f.$$

Stochastic Gradient Descent (SGD) is a method to solve this optimization problem. This is where Optimization meets Machine Learning.

6.036 vs 6.867. 6.036 (or equivalent undergraduate class) discusses the "How" or "mechanics" of such approaches. In this class, we expect that you know the "How" for much of supervised learning and decision making. That is, more than 60% of this class. So, what will we do in 6.867 (since $> 60\%$ is already done!)?
To start with, we will learn "Why" behind the "How". We will utilize *Probability* as our formal language. We will discuss estimators and theoretical guarantees, and generalization: does a good model fit on *historical data* lead to ability to predict *future*? Finally, we will have 40% of the course discusses unsupervised learning / probabilistic modeling to understand the structure within the data.

The Language of Probability. To understand "Why", effectively we need to "logically deduce" what we do starting with appropriate goals and axioms. The axioms that are relevant are that of Probability. In particular, to reason about what we do in Machine Learning, we will utilize the language of probability. And probability is entirely based on the three key axioms. Formally, there is a probability space Ω, events $\mathcal{F}$ in it, and a probability function $\mathbb{P} : \mathcal{F} \to [0, 1]$.

- Axiom 1. $\mathbb{P}(A) \geq 0$, for all $A \in \mathcal{F}$.
- Axiom 2. $\mathbb{P}(\Omega) = 1$.
- Axiom 3. $\mathbb{P}(\cup_{i=1}^{\infty} E_i) = \sum_{i=1}^{\infty} \mathbb{P}(E_i)$, if $E_i \cap E_j = \varnothing$, for all $i \neq j$.

Exercise. What is the probability of an empty set $\mathbb{P}(\varnothing)$?
Solution. Our intuition tells us the probability must be zero. How to prove it from the three axioms? For all $A \in \mathcal{F}$, according to axiom 3 we have $\mathbb{P}(A) + \mathbb{P}(A^c) = 1$, where A^c is the complement of A with respect to the whole space Ω. Then let $A = \Omega$ and according to axiom 2, $\mathbb{P}(\varnothing) = 1 - \mathbb{P}(\Omega) = 0$. □

The above exercise is a simple example of logical deduction starting from the axioms of probability. In a sense, this is what we will do to explain "why".
Before proceeding further, it is important to wonder – "Is it possible to have a different set of probability axioms?" This is a question hotly debated in the first half of last century. At the end of the day, *All roads lead to Rome*: All sorts of reasonable hypothesis about beliefs / decision making lead to axioms of probability[3].

Probabilistic View of Classification. In the language of probability, both attributes X and labels Y are random variables. Especially, Y is discrete-valued random variable. The conditional distribution $\mathbb{P}(Y|X)$ is of interest. Suppose labels take value 1 (e.g. default) or -1 (e.g. no default), given attribute $X = x$. An ideal classier, also known as *Bayes classifier*, which in the context of binary classification, predicts

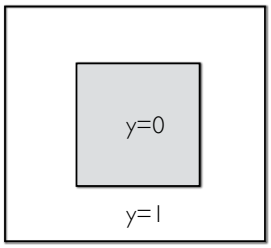
$$\hat{Y}(x) = \begin{cases} 1 & \text{if } \mathbb{P}(Y = 1|X = x) \geq 1/2 \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

The performance metric of interest is mis-classification probability, i.e. $\mathbb{P}(\hat{Y}(X) \neq Y)$.

Exercise. Prove that Bayes classifier minimize the mis-classification error amongst all possible classifiers.

Generalization. Probabilistic view will help us understand how to choose the loss function and how well our model *generalizes*. In terms of generalization and overfitting, you should trust your data, but only so much. Consider the following example: We have observations (x_i, y_i), $i = 1, \ldots, n$. Here attributes x_i are points distributed uniformly in the unit square. The label is generated according to the following rule: As sketched in the figure below, $y_i = 0$ when the corresponding $\mathbf{x}_i$ lies in the shaded square and $y_i = 1$ otherwise. The area of the shaded square is $1/2$.

[3] A good set of readings include [Cox46], [Sav12] and [dF17]



Pretend we do not know the true label rule and would like to to find a model to approximate it based on the observations. The function fit,

$$f(x) = \begin{cases} y_i, & \text{if } x = x_i, \\ 0, & \text{otherwise,} \end{cases}$$

which assigns every observed points to the correct label y_i and assign all unseen points to 0, is a perfect fit for the observation. However, since the possibility we encounter the same points in the set $\{(x_i, y_i), i = 1, \ldots, n\}$ in the future is zero, we will most certainly assign all future points to 0 and this function is simply as bad as "random" function! This is overfitting.

In order to prevent overfitting, empirically, we use *cross-validation* – split data into three parts: *train*, (*validate*) and *test*, or/and K*-fold* cross-validation. To explain why this the right thing to do, we shall discuss the notion of *generalization* that utilizes the view that data is generated per an unknown underlying probability distribution. Methodically, we shall use *regularization* and again probabilistic formalism will help explain why (or why not) it works well. Probabilistic view, again will come to our rescue to explain the *implicit* regularization that is implemented by modern methods (e.g. 'dropout') of neural networks.

Regression. Some examples of regression include predict wage given age, year, and education level. Formally, the data are labeled observations of the following form: $(x_1, y_1), \ldots, (x_N, y_N)$. The goal is to learn a model that maps *attribute* (or *feature*) x to *label* (or *target*) y so that given *attribute* x, we can predict corresponding *unknown (continuous) label* y. That is, to learn a function f such that $y = f(x)$ (and sometimes also what is the *confidence interval*).

In the language of probability, both attributes X and labels Y are random variables. Now, Y is continuous-valued random variable. The conditional distribution $\mathbb{P}(Y|X)$ is of interest. Given attribute $X = x$, we estimate $\hat{Y}(x)$ to minimize estimation error. One the most common estimation error is $\mathbb{E}\left[(Y - \hat{Y}(x))^2 | X = x\right]$, which is minimized by $\hat{Y}(x) = \mathbb{E}[Y|X = x]$. Finally, we should determine *predictive* distribution. $\mathbb{E}[Y|X = x]$ is unknown. The model fit for regression means to find the best fit for $f(x) \approx \mathbb{E}[Y|X = x]$ using observed data.

Exercise. Prove $\hat{Y}(x) = \mathbb{E}[Y|X = x]$ minimizes the estimation error $\mathbb{E}\left[(Y - \hat{Y}(x))^2 | X = x\right]$.

2.2 Unsupervised Learning

In unsupervised learning, there is no *target*. Only *input / features* are given. The goal is to learn the data distribution. In this course, we are going to cover topics such as dimensionality reduction, matrix estimation, clustering and mixture distribution, and feature extraction (topic model and deep generative model) from unstructured data such as text, audio or image, or for complexity reduction. Examples of unsupervised learning: Finding the principal component of DNA data (dimensionality reduction) [NJB+08], movie recommendation (matrix estimation), analyzing topics in documents (feature extraction: topic model), generating fake faces of celebrities (feature extraction: deep generative model).

2.3 Probabilistic Modeling

Two important topics in probabilistic modeling is incorporating prior knowledge from Bayesian perspective and sampling from distribution when probabilistic model is complex.

Incorporating Knowledge: Bayesian View. Most of the key tasks in machine learning are inference tasks. For example, in prediction we need to infer $\mathbb{P}(Y|X)$. In model learning, we need to infer $\mathbb{P}(\text{parameters}|\text{data})$. The Bayes' rule states that

$$\underset{\text{posterior}}{\mathbb{P}(\text{parameters}|\text{data})} \propto \underset{\text{likelihood}}{\mathbb{P}(\text{data}|\text{parameters})} \times \underset{\text{prior}}{\mathbb{P}(\text{parameters})}$$

The key question is how to select *prior*? This is the *prior* knowledge of the world. One of the classical priors is Gaussian distribution, which for example, leads to ridge regularization in regression.

Sampling from Distribution. A probability distribution can be complex. It may have succinct representation but no closed form formula, and hence difficult to evaluate. For example, we may know

$$\mathbb{P}(X = x) \propto \exp(f(x)) = \frac{1}{Z}\exp(f(x)),$$

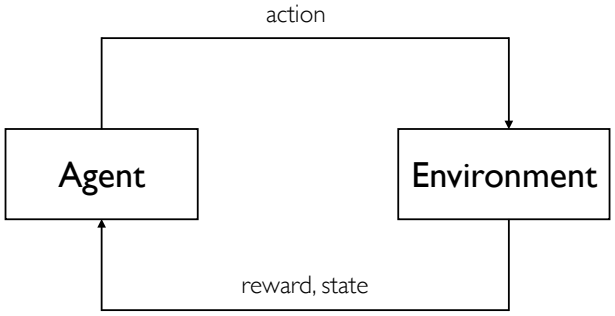
where

$$Z = \int \exp(f(x))dx.$$

This integration can be very hard to evaluate for a general $f(x)$. The key algorithm to evaluate on such complex distributions is Markov Chain Monte Carlo (MCMC)[4] It has specific forms such as Gibbs sampling and Metropolis-Hastings. MCMC works for generic form of distribution.

2.4 Decision Making

In data driven decision making (in presence of uncertainty), we need to learn the model of uncertainty, given observations. The goal is to make "optimal" decision with respect to a long-term objective. The decision vs information *timescale* are critically important. The following diagram summarizes the framework of decision making,



The two key *timescales* are state or environment dynamics, and information dynamics. Depending on the two timescales, there are methods / approaches including optimizing given model of uncertainty, Markov decision process, and reinforcement learning.

	State Dynamics	Information Dynamics
Optimizing Given Model of Uncertainty	No change (or extremely slow)	Lots of historical information
Markov Decision Process	High	Lots of historical information
Reinforcement Learning	High	Minimal information, learn as you go

The fundamental challenge in reinforcement learning is *explore vs exploit*. An example of poor decision is it is difficult to find blue sweater for young girls. To maximize profit (*exploit*), clothes makers choose not to make or make very few blue sweaters such that blue sweaters are hard to find and expensive. An important application of reinforcement learning is automated game player. We'll do a case study on AlphaGoZero.

[4]MCMC is one of the top 10 algorithms of all time [SD00]. Other algorithms include quicksort and fast Fourier transform.

2.5 And then, What Is Not Cover, But Of Interest

We may not be able to cover the following interesting topics in machine learning:

- Active Learning, actively obtain data as each data point is expensive.
- Transfer Learning, transfer data collected for one task to other learning task.
- Semi-supervised Learning, supervised setting with (additional) unsupervised data.
- Causal inference, Hypothesis testing, ...

But hopefully, things you'll learn this in course will provide systematic foundations to approach these topics.

3 Model Selection: An Example

We have data $x_1, \ldots, x_N$ sampled from a distribution. The goal is to learn the distribution. The assumption is that the data is generated from a Gaussian distribution $\mathcal{N}(\mu, \sigma^2)$. Then the refined goal is to learn the mean and variance. How to learn (parameters, mean and variance)?
A common method is maximum likelihood (ML), that is, choose the parameters that maximize $\mathbb{P}(\text{data}|\text{parameters})$. In this problem, to choose mean, variance from samples, the likelihood is

$$\begin{aligned}\mathbb{P}\left(x_1, \ldots, x_N | \mu, \sigma^2\right) &= \prod_{i=1}^{N} \mathbb{P}\left(x_i | \mu, \sigma^2\right) \\ &= \prod_{i=1}^{N} \frac{1}{(2\pi\sigma^2)^{1/2}} \exp\left(-\frac{(x_i-\mu)^2}{2\sigma^2}\right).\end{aligned}$$

Maximizing likelihood is same as maximizing logarithm of likelihood. This leads to

$$\max_{\mu,\sigma^2} g(\mu, \sigma^2),$$

where

$$g(\mu, \sigma^2) = -\frac{1}{2\sigma^2}\sum_{i=1}^{N}(x_i-\mu)^2 - N\ln\sigma - N\ln\sqrt{2\pi}.$$

This is an optimization problem and its solution is what we desire. For such reasons, optimization is an integral part of Machine Learning.

Exercise. Prove the solution to the above optimization problem is

$$\begin{aligned}\mu_{\text{ML}} &= \frac{1}{N}\sum_{i=1}^{N} x_i, \\ \sigma^2_{\text{ML}} &= \frac{1}{N}\sum_{i=1}^{N}(x_i-\mu_{\text{ML}})^2.\end{aligned}$$

The ML estimation for variance (and standard deviation) is biased. This leads to the Bessel correction for variance:

$$\tilde{\sigma}^2_{\text{ML}} = \frac{1}{N-1}\sum_{i=1}^{N}(x_i-\mu_{\text{ML}})^2.$$

Exercise Prove ML estimation for variance (and standard deviation) is biased and the Bessel correction for variance is unbiased. An estimator $\hat{X}$ of variable X is unbiased if $\mathbb{E}[\hat{X}] = X$.

Exercise. Modify the assumption such that the data is generated from a multivariate Gaussian distribution $\mathcal{N}(\mu, \Sigma)$, $\mu \in \mathbb{R}^d$, $\Sigma \in \mathbb{R}^{d \times d}$.
(a) Compute the ML estimator μ_{ML} and Σ_{ML}.
(b) Is Bessel's correction needed for covariance estimation? If so, identify it.

References

[Cox46] R. T. Cox. Probability, Frequency and Reasonable Expectation. *American Journal of Physics*, 14(1):1–13, 1946. doi:10.1119/1.1990764.

[dF17] Bruno de Finetti. *Theory of Probability: A Critical Introductory Treatment*. Wiley Series in Probability and Statistics. Wiley, 2017.

[NJB+08] John Novembre, Toby Johnson, Katarzyna Bryc, Zoltán Kutalik, Adam R Boyko, Adam Auton, Amit Indap, Karen S King, Sven Bergmann, Matthew R Nelson, Matthew Stephens, and Carlos D Bustamante. Genes mirror geography within Europe. *Nature*, 456:98, Aug 2008. doi:10.1038/nature07331.

[Sav12] Leonard J. Savage. *The Foundations of Statistics*. Dover Books on Mathematics. Dover Publications, 2012.

[SD00] Francis Sullivan and Jack Dongarra. Guest Editors' Introduction: The Top 10 Algorithms. *Computing in Science & Engineering*, 2(1):22–23, Jan 2000. doi:10.1109/MCISE.2000.814652.