Machine Learning in Communications
Lecture 1: Machine Learning Basics

Harpreet S. Dhillon

Wireless@VT, Department of Electrical & Computer Engineering
Virginia Tech, Blacksburg, VA

https://www.dhillon.ece.vt.edu
hdhillon@vt.edu

JTG/IEEE Information Theory Society Summer School
IIT Kanpur

Acknowledgments

- Some parts of this short course are drawn from a graduate course that I co-taught with Prof. R. M. Buehrer on this topic at VT in Fall 2019. My graduate student M. A. Abd-Elmagid helped us by serving as the teaching assistant.
- Thanks to Prof. D. Batra for his course on ML at Virginia Tech that introduced me to this topic.
- All the case studies presented in this course are based on the joint work with my graduate students, especially C. Saha and K. Bhogi.
- I am grateful to the National Science Foundation for supporting our work that directly or indirectly contributed to these case studies.

Course Modules

- Module 1: Introduction and Background
  - Machine learning basics
  - Role of machine learning in communications
  - Case Study on *Determinantal Learning in Wireless Networks* demonstrating the role of ML for approximating algorithms
- Module 2: Estimation Theory Perspective of Machine Learning
  - Statistical estimation
  - Popular supervised learning algorithms will be interpreted as ML and MAP estimators of appropriate underlying statistical models
- Module 3: Theory-Guided Machine Learning in Communications
  - Introduction to Theory-Guided ML
  - Introduction to unsupervised learning
  - Case Study on *k-means Clustering on a Grassmann Manifold for MIMO Codebook Design*
- Module 4: Unsupervised Learning
  - Mixture Models and Expectation Maximization
  - Case study on *Gradient Compression for Federated Learning*

Useful References

ISL G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning*. New York: Springer Texts in Statistics, 2013.

ESL T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning*. New York: Springer Series in Statistics, 2001.

DL I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

UML S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms.* Cambridge University Press, 2014.

MLPP K. Murphy, *Machine Learning: A Probabilistic Perspective*. MIT press, 2012.

PRML C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

Conventional Design Flow for Communication Systems

- Step 1 is to acquire domain specific knowledge.
  - Example includes the knowledge of the connection of random movement of charged particles with thermal noise.
- Step 2 is to develop physics-based mathematical models.
  - Example is an Additive Gaussian White Noise (AWGN) channel.
- Step 3 is to develop algorithms (ideally with optimality guarantees).
  - This often requires applying optimization algorithms that also require domain specific knowledge.
- Observation: The design of current systems is essentially driven by the construction of a mathematical model that describes the physics of the underlying setup (within the limitations of that model).

An Alternate Design Flow using Machine Learning

- Step 1 is to acquire a lot of data.
  - Made possible by unprecedented availability of data.
- Step 2 is to train a machine learning model.
  - Made possible by unprecedented availability of computational resources.
- One can then use the trained "black box machine" to carry out the desired task.
- Key observations:
  - Access to data and computational resources is the key.
  - Domain specific knowledge is useful in Step 2.

What is Machine Learning?

- Mitchell (1997) provided this definition: "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." (Chapter 5 of DL)
- Example: Decoding a BPSK signal at the receiver.
  - T: Decode a signal at the receiver.
  - E: Observe the received signal for a known transmitted bit.
  - P: Probability of bit error (or bit error rate).

Types of Machine Learning Algorithms

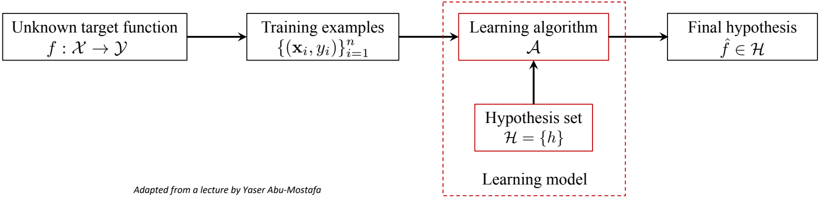
- Supervised learning
  - Involves estimating an output (called label or response) based on one or more inputs (called predictors, features, or attributes).
  - The supervising outputs are included in the training data.
- Unsupervised learning
  - Training data only include predictor or feature values. No supervising output is provided.
  - The task is to discover structures (often clusters) from this data.
- Reinforcement learning
  - The setting of reinforcement learning is slightly different. It involves "agents" that take actions to perform a specific task. For each action, the agents will get a "reward". The goal is to construct a strategy that maximizes some notion of cumulative reward.
  - *Even though reinforcement learning is also useful in communications, we will not be able to cover it in this course because of limited time.*

Essential Components of a Machine Learning Problem

We need three things in order to be able to define a *meaningful* machine learning problem:

- There is an underlying *pattern*.
- It is not possible to describe that pattern mathematically.
- We have data to *learn* that pattern.

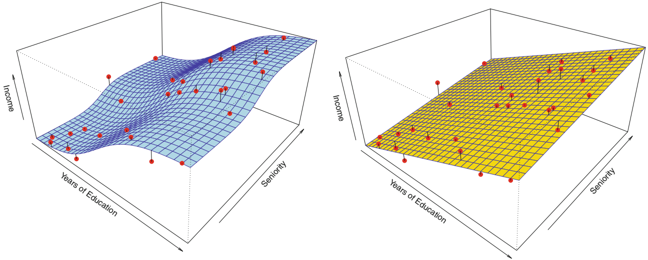
Overview of the Supervised Learning Process



Adapted from a lecture by Yaser Abu-Mostafa

- We "observe" unknown target function through training examples.
- Hypothesis set $\mathcal{H}$ contains all candidate functions that are considered.
- The learning algorithm and hypothesis set together constitute our learning model.
- Learning algorithm will choose the "best" candidate function, which will be denoted by $\hat{f}$.

Example



[ISL, Figures 2.3 and 2.4] First figure shows the true function. Second shows a linear fit for the training data using the function: $\text{Income} = \beta_0 + \beta_1 \times \text{education} + \beta_2 \times \text{seniority}$. Here, β_0, β_1, and β_2 are the model parameters that are being learnt using training data.
Aside: Note that Y does not have a deterministic relationship with X.

Supervised Learning: Summary and Notation

- Purpose: Estimating an output (called label or response) based on one or more inputs (called predictors, features or attributes)
- Features/predictors/attributes: We will denote the predictors by X. When it is a vector, its j^{th} element will be denoted by X_j. The total number of predictors will be denoted by d, which means $1 \leq j \leq d$.
- Lets assume that we have n observations in the training dataset. The value of j^{th} predictor in the i^{th} observation is denoted by x_{ij}.
- The values of predictors in a training dataset can be represented by an $n \times d$ matrix

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1d} \\ x_{21} & x_{22} & \dots & x_{2d} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \dots & x_{nd} \end{bmatrix}. \tag{1}$$

- Output/response: We will denote the response or output by Y. Its value for the i^{th} observation is denoted by y_i.

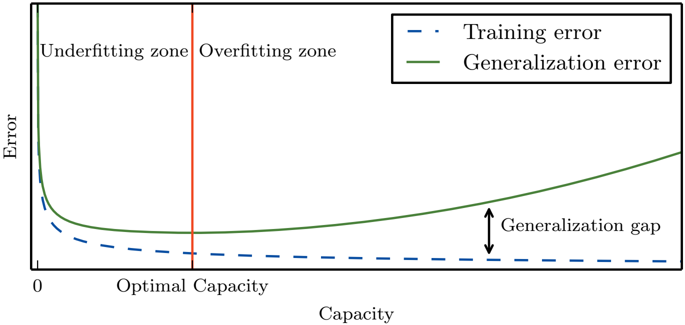
Regression vs. Classification

- A supervised learning problem is categorized as a regression problem or a classification problem based on whether the response variable is continuous or discrete.
  - Regression: If we try to predict the income of a person with a specific seniority and years of education, it is a regression problem.
  - Classification: When the response variable is qualitative (or discrete), the problem becomes a classification problem because the goal is to put the observed response in one of the "classes".

Loss/Cost/Error Function

- $L(y, \hat{y})$: Penalizes errors in our prediction. In other words, this is the penalty of predicting $\hat{y}$ when the correct output is y.
- The choice of a loss function depends on whether we are doing regression or classification. Two examples are:
  - Regression: $L(y, \hat{y}) = (y - \hat{y})^2$.
  - Classification: $0/1$ loss function, where loss is 1 when $\hat{y} \neq y$ and 0 otherwise.
- It is common to assume that error *decomposes* over the dataset, which allows one to write the total loss over the dataset as: $\frac{1}{n}\sum_{i=1}^{n} L(y, \hat{y})$.
- As we will see shortly, we need to be careful with the choice of the loss function as well as how the performance is characterized.

Model Selection



[DL, Figure 5.3] Model complexity (capacity) vs. error.

- The goal is to make sure our learning algorithm works on the new and unseen data. This is termed as *generalization*.
- We care about the generalization error as opposed to the training error. This is why we cannot arbitrarily increase our model complexity with the hope of getting "better" performance.

Why is Learning Hard?

Consider the following simple problem:

- Number of features: d
- Each feature takes a binary value: $x_{ij} \in \{0,1\} \forall i, j$.
- Each response variable is also binary: $y_i \in \{0,1\} \forall i$.

How many mappings are possible for this setting? In other words, what is the size of the hypothesis class: $\mathcal{H} = \{h : \{0,1\}^d \to \{0,1\}\}$?

Answer: 2^{2^d}. This is a huge number.

- Implication: Even if you have n training samples, we still have $2^{(2^d - n)}$ unobserved mappings. Hopelessly large search space!
- There can be no learning if you do not assume something about the function!

Statistical Interpretation

- Setting: Let $X \in \mathbb{R}^d$ denote a random input/feature vector and $Y \in \mathbb{R}$ a random output variable. We consider that (X, Y) is sampled from the joint distribution $p(X, Y)$.
- A useful way to think about the connection of this interpretation with function approaximation is in terms of the following statistical model for the joint distribution of X and Y:

$$Y = f(X) + \epsilon,$$

  where ϵ is a zero mean error term, which can be assumed to be independent of X.
- This *additive model* is a useful approximation of the fact that we will seldom have deterministic relationship between X and Y in our datasets.
- Therefore, our objective is to estimate $\hat{Y} = \hat{f}(X)$.

The Utility of Statistical Interpretation

- Setting: $(X, Y) \sim p(X, Y)$, where $X \in \mathbb{R}^d$ is the feature vector and $Y \in \mathbb{R}$ a random output variable.
- Question: Given X, how do we predict Y? In other words, we seek a function $h(X)$ for predicting Y given X.
- Lets consider squared loss function $L(Y, h(X)) = (Y - h(X))^2$.
- Lets determine $h(\cdot)$ that minimizes expected prediction error: $E[(Y - h(X))^2] = E_X E_{Y|X}[(Y - h(X))^2 | X]$.
- It suffices to minimize this function pointwise:

$$h(x) = \arg\min_c E_{Y|X}[(Y - c)^2 | X = x].$$

- The solution of this is $h(x) = E[Y|X = x]$.
  - This is also called the regression function.
  - k-NN directly implements this.

Summary

- A very brief introduction to the basics of machine learning.
- Defined machine learning and introduced types of ML algorithms.
- Introduced supervised learning through statistical and functional approximation viewpoints.
- Discussed model selection briefly.
- Next lecture: Role of machine learning in communications.

Note

- The following four slides were supposed to be covered in Lecture 1. However, they were moved to Lecture 2 to limit the first video recording to 1 hour. They fit within the scope of Lecture 2 as well.

Binary Classification on an Unbalanced Dataset

- Lets assume that each point in our training set has a binary label.
- Assume further that one of the labels occurs very infrequently.
  - Think of a signal detection problem assuming that the message is transmitted very infrequently.
- In many such problems, it is more detrimental if we miss a signal than if we detect a signal that was not there (*false negatives* are more critical than *false positives*).
- Consider the classical example of a medical dataset.
  - Assume that the binary label signifies whether a given patient has a disease or not.
  - It is really critical to detect correctly when a patient has that disease. Otherwise, the treatment may get delayed.
  - On the contrary, if we misclassify a healthy person as having that disease, it is "relatively" easy to handle it (e.g., run more tests).

Binary Classification - Choice of Loss Function

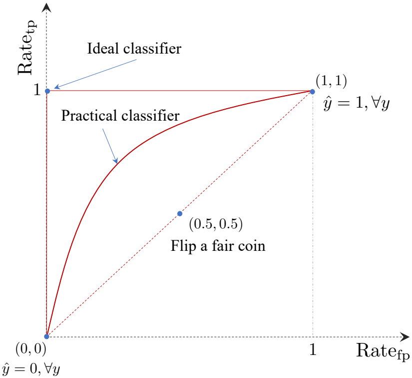
y \ $\hat{y}$	0	1
0	tn 0	fp ?
1	fn ?	tp 0

For the reasons that we already discussed, we may want to put a larger loss for fn. Therefore, simply 0-1 loss function will not work in this case.

Binary Classification - Measuring Accuracy

- Consider a dataset in which only 0.1% of patients have a disease and the rest are healthy. Note that you can easily map this to the signal detection problem as well.
- You propose an algorithm that gives a 99.5% accuracy. Accuracy here is defined as the percentage of points that were correctly classified.
- Is this a good algorithm?
- What about a trivial algorithm that predicts that no one has a disease? In other words, $\hat{y}_i = 0, \forall i$. What is the accuracy of this algorithm?
- Why is this performing better than your algorithm?
- Takeaway: We need to be more careful with how we *measure* accuracy.

Binary Classification - ROC



- Remember the dependence of $\mathrm{Rate_{tp}}$ and $\mathrm{Rate_{fp}}$ in a signal detection problem on the signal detection threshold.
  - The *practical classifier* curve is obtained by changing this threshold.
- This is called Receiver Operating Characteristics (ROC) curve and is one of the standard tools used in machine learning to characterize the performance of classifiers.