Lecture-01: Introduction

1 Introduction

Goal: Provide a data driven framework for inference, prediction, decision, and model construction.
Statistical framework: Statistical assumptions about the underlying phenomena, i.e. on the data generation process.

This framework allows a formal theory that can define learning, generalization, overfitting, and characterize the performance of learning algorithms, leading to design of better algorithms.

2 Definitions and terminology

Following are the terms we will use frequently.

2.1 Input and output space

Definition 2.1 (Examples and features). Examples are items or data instances. An **example** is typically represented by a vector $x \in \mathcal{X}$, where the components of an example are its **features**.

Definition 2.2 (Input space). The set of all possible *examples* is called the **input space** and denoted by $\mathcal{X}$.

Remark 1. Feature extraction from examples is domain-specific task done by the experts, and is critical to the successful prediction. If an example has N attributes and all of them can be represented by real numbers, then the feature set $\mathcal{X} = \mathbb{R}^N$ with $N \geqslant 1$.

Definition 2.3 (Labels). Labels or targets are values of categories assigned to examples.

Remark 2. Labels are discrete for classification and real-valued for regression.

Definition 2.4 (Output space). The set of all possible *labels* is called the **output space** and denoted by $\mathcal{Y}$.

Remark 3. For binary classification, output space could be $\mathcal{Y} = \{0,1\}$ or $\{-1,1\}$.

Definition 2.5 (Prediction space). The set of all *predicted labels* is called the **prediction space** and denoted by $\mathcal{Y}'$.

Definition 2.6. Set of predictions $\mathcal{Y}'$ may not necessarily be equal to the set of labels $\mathcal{Y}$.

2.2 Concept and hypothesis set

Definition 2.7 (Concepts). A mapping from input space to output space is called a **concept** and denoted by $c : \mathcal{X} \to \mathcal{Y}$. The set of all concepts is called the **concept class** and denoted by C.

Example 2.8. For binary classification an output space is $\mathcal{Y} = \{0,1\}$, and any concept c can be identified by the set $A_c = \{x \in \mathcal{X} : c(x) = 1\}$ such that $c(x) = \mathbb{1}_{\{x \in A_c\}} = \mathbb{1}_{A_c}(x)$.

Example 2.9. The set of all triangles, rectangles, circles, lines in the plane are all examples of concept classes.

Definition 2.10 (Hypothesis). The set of all possible candidate concepts that map features to predicted labels is called the **hypothesis class** and denoted by $H \subseteq (\mathcal{Y}')^{\mathcal{X}}$. A *consistent* hypothesis set contains the concept to learn, and an *inconsistent* hypothesis set doesn't contain it.

Remark 4. Let $c : \mathcal{X} \to \mathcal{Y}$ be the true concept, and $h : \mathcal{X} \to \mathcal{Y}'$ be a hypothesis, then $y = c(x)$ is the label for an example x and $y' = h(x)$ is the predicted output for a hypothesis h.

2.3 Sample

Assumption 2.11. All examples in $\mathcal{X}$ are identically and independently distributed (*i.i.d.*) with a fixed but unknown underlying distribution D.

Definition 2.12 (Sample). We have a **sample** $x \in \mathcal{X}^m$ of size m generated *i.i.d.* according to the distribution D. For a concept $c : \mathcal{X} \to \mathcal{Y}$, we have a **labeled sample** $z \in (\mathcal{X} \times \mathcal{Y})^m$ such that $z_i = (x_i, c(x_i))$.

There are three kinds of samples.

1. Training sample: examples/samples to train a learning algorithm.
2. Validation sample: samples to tune the free parameters of the learning algorithm.
3. Test sample: samples to evaluate the performance of the learning algorithm.

2.4 Loss function

Definition 2.13 (Loss function). Loss function measures the difference or loss between predicted and the true label, and denoted by $L : \mathcal{Y} \times \mathcal{Y}' \to \mathbb{R}_+$.

Example 2.14 (Hamming loss function). When $\mathcal{Y} = \mathcal{Y}'$ is discrete, the Hamming loss function $L_H(y, y') \triangleq \mathbb{1}_{\{y \neq y'\}}$ is bounded.

Example 2.15 (Euclidean loss function). When $\mathcal{Y} = \mathcal{Y}' \subseteq \mathbb{R}$, the Euclidean loss function $L_E(y, y') = (y - y')^2$ is unbounded for unbounded label sets.

Definition 2.16 (Generalization error). Given a hypothesis $h \in H$, the target concept $c \in C$, and an underlying distribution D from which an example $X \in \mathcal{X}$ is generated *i.i.d.*, the generalization error or risk of hypothesis h is defined as

$$R(h) \triangleq \mathbb{E}L(c(X), h(X)).$$

Example 2.17. For Hamming loss function $L_H(y, y') = \mathbb{1}_{\{y \neq y'\}}$, the generalization risk is given by

$$R(h) = \mathbb{E}\mathbb{1}_{\{c(X) \neq h(X)\}} = P\{c(X) \neq h(X)\}.$$

Definition 2.18. If the sample distribution D and concept c are known, then the optimal hypothesis h^* that minimizes the generalization error is given by

$$h^* = \arg\min_{h \in H} R(h).$$

Remark 5. The generalization error of a hypothesis is not directly accessible to the learner since both the distribution D and concept c are unknown. However, one can measure the *empirical error* of a hypothesis on the labeled sample z.

Definition 2.19 (Empirical error). For a hypothesis $h \in H$, a target concept $c \in C$, and an labeled sample $z \in (\mathcal{X} \times \mathcal{Y})^m$ The *empirical error* is defined as

$$\hat{R}_z(h) = \frac{1}{m} \sum_{i=1}^{m} L(c(x_i), h(x_i)).$$

Definition 2.20 (Supervised learning). The **supervised learning** is selection of a hypothesis $h_z \in H$ to minimize the empirical error with respect to loss function L. That is,

$$h_z = \arg\min_{h \in H} \hat{R}_z(h).$$

Remark 6. The empirical error of a hypothesis is the average error over the sample x, while the generalization error is the expected error based on the distribution D. We see that $\mathbb{E}\hat{R}_z(h) = R(h)$ by the linearity of expectations. We will see later that $\hat{R}_z(h) \approx R(h)$ with high probability for large m.

Remark 7. For Hamming loss function $L_H(y, y') = \mathbb{1}_{\{y \neq y'\}}$, the empirical risk $\hat{R}_z(h) = \frac{1}{m} \sum_{i=1}^{m} \mathbb{1}_{\{h(x_i) \neq c(x_i)\}}$ is an empirical average of m *i.i.d.* Bernoulli random variables.

2.5 Learning stages

Learning stages for a given sample (collection of labeled examples).

1. Randomly partition into training, validation, and test sample.
2. Associate features to examples.
3. Fix free learning parameters and pick a hypothesis.
4. Pick the hypothesis with best performance on validation sample.
5. Predict labels of the test examples.
6. Evaluated the algorithm using the test labels.

A learning algorithm is called *consistent* if there are no errors on the training data. A consistent algorithm may perform very poorly on test data, if the learning class is highly complex. This is the difference between memorization and generalization.

3 What is machine learning?

Machine learning is computational methods to improve performance or make predictions using *experience*. Experience is the past information available to the learner. Information maybe readily available as digitized human-labeled training sets, or can be obtained via interaction with environment.

Two main practical objectives of machine learning are:

- accurate predictions of unseen items, and
- design of efficient, robust, and scalable prediction algorithms.

The quality of machine learning algorithms is measured by

- time complexity: running time of the algorithm,
- space complexity: memory requirements of the algorithm, and
- sample complexity: sample size required for the algorithm to learn a family of concepts

The success of prediction depends on size and quality of data instances. The theoretical learning guarantees depend on

- complexity of concept class, and
- size of training sample.

Fundamental algorithmic and theoretical questions that arise are

- Which concept families can be learned, and under what conditions?
- How well can these concepts be learned computationally?

Learning techniques are data-driven methods with relations to computer science, statistics, probability, and optimization.

4 Learning Problems

Learning problems can be broadly classified into following major classes.

1. Classification: assign a category to each item. Applications include document classification, text classification, image classification where number of categories are small. Other applications where there are large or unbounded categories are optical character recognition and speech recognition.

2. Regression: assign a real value to each item. Applications include prediction of stock values or other economic variables, or prediction of physical processes such as temperature, humidity etc.

3. Ranking: assign order to items. Applications include recommendation systems, web search, and natural language processing.

4. Clustering: partition items into homogeneous regions. Clustering is typically used for large unlabeled data sets. Applications include community detection in large data sets.

5. Dimensionality reduction: Transform an initial representation of items to a low dimensional representation preserving some properties of the initial representation. Applications include machine aided compression, preprocessing of digital images.

5 Learning scenarios

1. Supervised learning: The learner receives a sample for training and validation, and makes prediction for all unseen points. This is common scenarios for classification, regression, and ranking.

2. Unsupervised learning: The learner receives unlabeled examples for training and makes predictions for all unseen points. Difficult to quantitatively evaluate the performance of a learner. Clustering and dimensionality reduction are examples of unsupervised learning.

3. Semi-supervised learning: The learner receives a training sample consisting of both labeled and unlabeled data, and make predictions for all unseen points.

4. Transductive inference: The learner receives a labeled training sample along with a set of unlabeled test points, and make predictions for only these test points.

5. Online learning: At each round, the learner receives an unlabeled training example, makes a prediction, receives the true label, and incurs a loss. The objective is to minimize the cumulative loss over all rounds.

6. Reinforcement learning: The learner actively interacts with the environment and receives an immediate reward for each action. The objective is to maximize reward over a course of actions and iterations with the environment.

7. Active learning: The learner adaptively/interactively collects training samples by querying an oracle for new samples. The goal is to achieve comparable performance to the supervised learning with fewer samples.