Machine learning

Overview of suppervised learning

Hamid Beigy

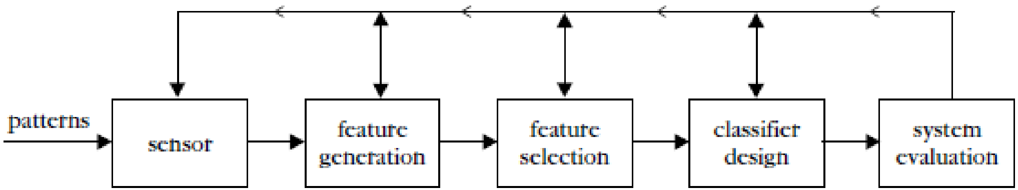
Sharif University of Technology

November 1, 2021

Table of contents

Introduction

In order to classify a pattern, the following stages must be used.



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

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}.$$

  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 $t_i \in \{1, 2, \ldots, C\}$, the problem is known as classification.
- In some situation, multiple classes are associated to each input x, and the problem is called multi-label classification.
- When $t_i \in \mathbb{R}$, the problem is known as regression.

Classification

- In classification, the goal is to find a mapping from inputs X to outputs t, where $t \in \{1, 2, \ldots, C\}$ with C being the number of classes.
- When $C = 2$, the problem is called binary classification. In this case, we often assume that $t \in \{-1, +1\}$ or $t \in \{0, 1\}$.
- When $C > 2$, the problem is called multi-class classification.

Family car

We want to learn the class of a family car. We have a set of examples of cars, and we have a group of people that we survey to whom we show these cars. The people look at the cars and label them; the cars that they believe are family cars are positive examples, and the other cars are negative examples.

- After discussion with experts, each car represented by two features: price (x_1) and engine power (x_2). Thus each car is represented by the following 2-dimensional feature vector.

$$x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

- Each car (feature vector) is labeled as

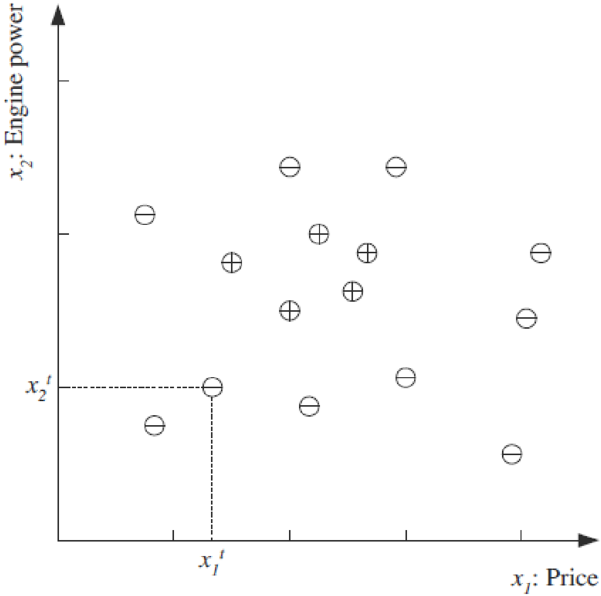
$$h(x) = \begin{cases} 1 & \text{if the car is a family car (positive example)} \\ 0 & \text{if the car is not a family car (negative example)} \end{cases}$$

- Each car in the training set is represented by an ordered pair (x, t) and the training set containing

$$S = \{(x_1, t_1), (x_2, t_2), \ldots, (x_N, t_N)\}.$$

- Each label is generated from a concept $c \in \mathbb{C}$, where $\mathbb{C}$ is called a concept class.

- The training data now can be plotted in the 2-D space (x_1, x_2), where car i is a data point and its label is given by t_i.



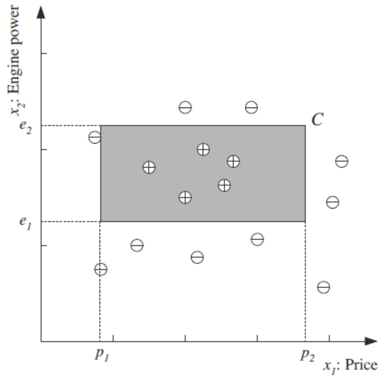
- The learning algorithm should find a particular hypotheses $h \in H$ to approximate $\mathbb{C}$ as closely as possible.
- The expert defines the hypothesis class H, but he can not say the values for e_1, e_2, p_1, p_2.
- We choose H and the aim is to find $h \in H$ that is similar to $\mathbb{C}$. 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.

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

- After further discussion with experts and the analysis of the data, we believe that for a family car, its price and engine power should be in a certain range.

$$(p_1 \leq x_1 \leq p_2)\&(e_1 \leq x_2 \leq e_2)$$

- The above equation assumes H to be a rectangle in 2-D space.
- For suitable values e_1, e_2, p_1, p_2, the above equation fixes $h \in H$ from the set of axis aligned rectangles.



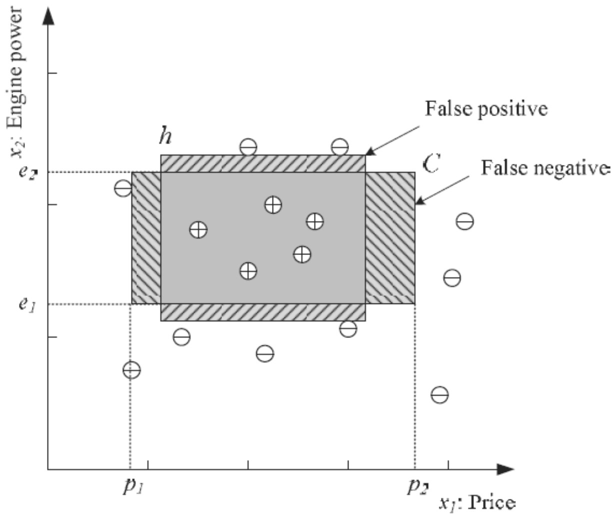
- In real life, we don't know $c(x)$ and hence cannot evaluate how well $h(x)$ matches $c(x)$.
- We use a small subset of all possible values x as the training set as a representation of that concept.
- Empirical error (risk)/training error is the proportion of training instances such that $h(x) \neq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

$$E_E(h|S) = \frac{1}{N} \sum_{i=1}^{N} I[h(x_i) \neq c(x_i)]$$

- When $E_E(h|S) = 0$, h is called a consistent hypothesis with dataset S.
- For family car, we can find infinitely many h such that $E_E(h|S) = 0$. 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.

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

$$E_T(h|S) = \underset{x \sim D}{\textbf{Prob}}[h(x) \neq c(x)] \tag{1}$$



Most specific hypothesis (h_s)

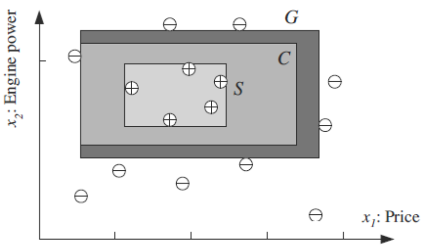
The tightest/smallest rectangle that includes all positive examples and none of the negative examples.

Most general hypothesis (h_g)

The largest rectangle that includes all positive examples and none of the negative examples.

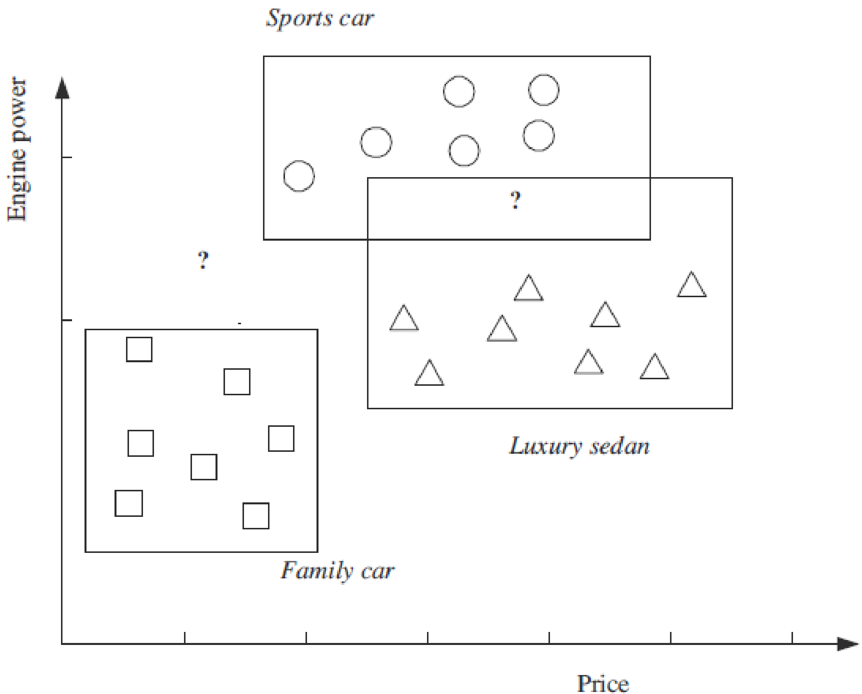
Version space

Version space is the set of all $h \in H$ between h_s and h_g.



- We assume that H includes $\mathbb{C}$, that is there exists $h \in H$ such that $E_E(h|S) = 0$.
- 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 $E_E(h|S) = 0$.
- Thus in any application, we need to make sure that H is flexible enough , or has enough capacity to learn $\mathbb{C}$.

How extend two-class classification to multiple class classification?



Regression

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

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

- If there is no noise, the task is interpolation and our goal is to find a function $f(x)$ that passes through these points such that we have

$$t_k = f(x_k) \qquad \forall k = 1, 2, \ldots, N$$

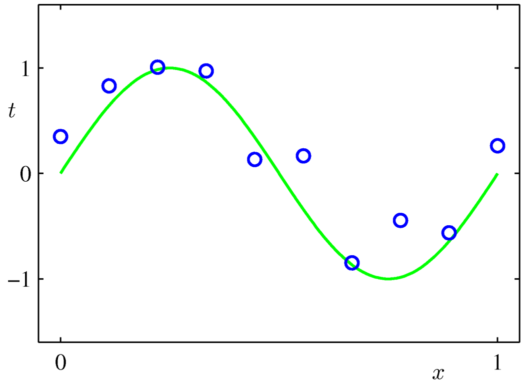
- In polynomial interpolation, given N points, we find $(N-1)$st degree polynomial to predict the output for any x.
- If x is outside of the range of the training set, the task is called extrapolation.
- In regression, there is noise added to the output of the unknown function.

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

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

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

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



- 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

$$E_E(g|S) = \frac{1}{N}\sum_{k=1}^{N}[t_k - g(x_k)]^2$$

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

Model selection

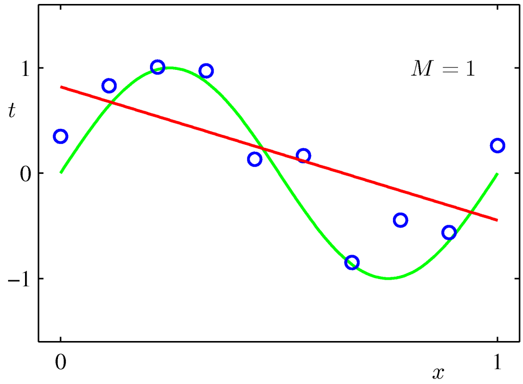
- The training data is not sufficient to find the solution, we should make some extra assumption for learning.

Inductive bias

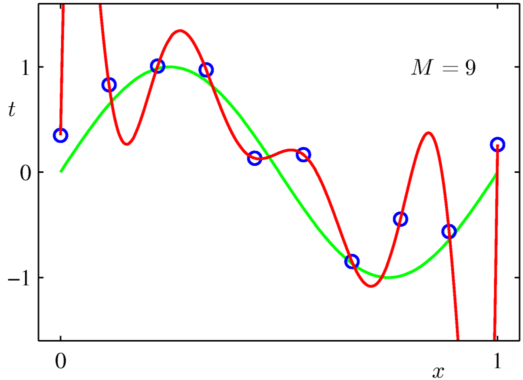
The inductive bias of a learning algorithm is the set of assumptions that the learner uses to predict outputs given inputs that it has not encountered.

- One way to introduce the inductive bias is when we assume a hypothesis class.
- Each hypotheses class has certain capacity and can learn only certain functions.
- How to choose the right inductive bias (for example hypotheses class)? This is called model selection.
- How well a model trained on the training set predicts the right output for new instances is called generalization.
- For best generalization, we should choose the right model that match the complexity of the hypothesis with the complexity of the function underlying data.

- For best generalization, we should choose the right model that match the complexity of the hypothesis with the complexity of the function underlying data.
- If the hypothesis is less complex than the function, we have underfitting

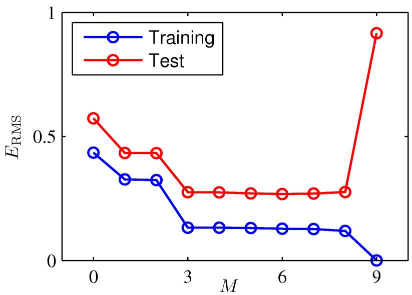


- If the hypothesis is more complex than the function, we have overfitting



- There are trade-off between three factors
  - Complexity of hypotheses class
  - Amount of training data
  - Generalization error

- As the amount of training data increases, the generalization error decreases.
- As the capacity of the models increases, the generalization error decreases first and then increases.



- We measure generalization ability of a model using a validation set.
- The available data for training is divided to
  - Training set
  - Validation data
  - Test data

Summary

Summary

- The training set S
  - A set of N i.i.d distributed data.
  - The ordering of data is not important
  - The instances are drawn from the same distribution $p(x, t)$.
- In order to have successful learning, three decisions must take
  - Select appropriate model $(g(x|\theta))$
  - Select appropriate loss function

$$E_E(\theta|S) = \sum_k L(t_k, g(x;\theta))$$

  - Select appropriate optimization procedure

$$\theta^* = \underset{\theta}{argmin}\ E_E(\theta|S)$$

Readings

1. Chapter 1 of Pattern Recognition and Machine Learning Book (Bishop 2006).
2. Chapter 1 of Machine Learning: A probabilistic perspective (Murphy 2012).
3. Chapter 1 of Probabilistic Machine Learning: An introduction (Murphy 2022).

References i

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.

Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

– (2022). *Probabilistic Machine Learning: An introduction*. MIT Press.

Questions?