

Machine learning

Introduction

Hamid Beigy

Sharif University of Technology

February 13, 2023





1. What is machine learning?
2. Types of machine learning
3. Outline of course
4. Supervised learning
5. Summary
6. References

What is machine learning?



The field of *machine learning* is concerned with the question of how to construct computer programs that automatically improve with the experience.

Definition (Mohri et. al., 2012)

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

Definition (Mitchel 1997)

A computer program is said to *learn*

- from *training 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 (Checkers learning problem)

Class of task T : playing checkers.

Performance measure P : percent of games won against opponents.

Training experience E : playing practice game against itself.

Example (Handwriting recognition learning problem)

Class of task T : recognizing and classifying handwritten words within images.

Performance measure P : percent of words correctly classified.

Training experience E : a database of handwritten words with given classifications.

Example (Robot driving learning problem)

Class of task T : driving a robot on the public highways using vision sensors.

Performance measure P : average distance travelled before an error.

Training experience E : a sequence of images and steering command recorded.



We need machine learning because

1. Tasks are too complex to program
 - Tasks performed by animals/humans such as driving, speech recognition, image understanding, and etc.
 - Tasks beyond human capabilities such as weather prediction, analysis of genomic data, web search engines, and etc.
2. 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 three main groups.

1. **Supervised/predictive learning:** The goal is to learn a mapping from **inputs** x to **outputs** y given the **labeled set** $S = \{(x_1, t_1), (x_2, t_2), \dots, (x_N, t_N)\}$. x_k is called feature vector.
 - When $t_i \in \{1, 2, \dots, C\}$, the learning problem is called **classification**.
 - When $t_i \in \mathbb{R}$, the problem is called **regression**.
2. **Unsupervised/descriptive learning:** The goal is to find interesting pattern in data $S = \{x_1, x_2, \dots, x_N\}$. Unsupervised learning is arguably more typical of human and animal learning.
3. **Reinforcement learning:** Reinforcement learning is learning by interacting with an environment. A reinforcement learning agent learns from the consequences of its actions.



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.



- **Probability theory** can be applied to any problem involving **uncertainty**.
- A key concept in machine learning is **uncertainty**. In machine learning, uncertainty comes in many forms:
 - What is the best prediction about the future given some past data?
 - What is the best model to explain some data?
 - What measurement should I perform next?
- 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.
- The probabilistic approach to machine learning is closely related to the field of statistics, but differs slightly in terms of its emphasis and terminology¹.

¹<http://www-stat.stanford.edu/~tibs/stat315a/glossary.pdf>

Outline of course



1. Introduction to machine learning & probability theory
2. Supervised learning:
 - Linear models for regression
 - Classifiers based on Bayes decision theory
 - Linear & Nonlinear models for classification
 - Combining classifiers
 - Evaluating classifiers
 - Computational learning theory
3. Unsupervised/descriptive learning:
 - Feature selection & Feature extraction/dimensionality reduction
 - Clustering & clustering evaluation
4. Reinforcement learning:
 - Reinforcement model & model-based learning
 - Monte-carlo & Temporal difference methods
5. Advanced topics:
 - Statistical learning theory
 - Graphical models
 - Deep & semi-supervised & Active & online learning
 - Large scale machine learning



1. Evaluation:

Mid-term exam 25% 1402-01-30

Final exam 25%

Homework 30%

Quiz 10%

Paper & Project 10% **Hard deadline for choosing paper: 1401-01-30**

2. **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/14012-40717.html>

5. TAs :

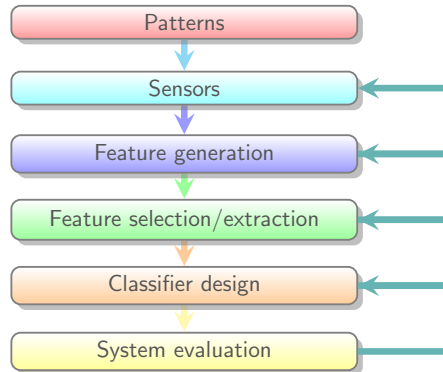
Mohaddeseh Mirbeygi

Supervised learning



The basic stages in design of a classification system.

Basic stages in design of a classification system





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

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

S is called **training set**.

2. In the simplest setting, each training input x is a D -dimensional vector of numbers.
3. Each component of x is called **feature**, **attribute**, or **variable** and x is called **feature vector**.
4. 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.
5. When $t_i \in \{1, 2, \dots, C\}$, the problem is known as **classification**.
6. In some situation, multiple classes are associated to each input x , and the problem is called **multi-label classification**.
7. When $t_i \in \mathbb{R}$, the problem is known as **regression**.

Supervised learning

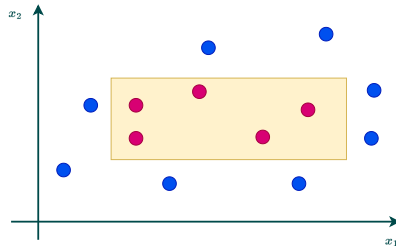
Classification



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

Example (Two-dimensional game)

In this game, the teacher selects some points from a two dimensional plane via sampling. Then labels them based on its situation with a pre-specified rectangle. When the sampled point is in the rectangle, then the label is **1**, otherwise the label is **0**.





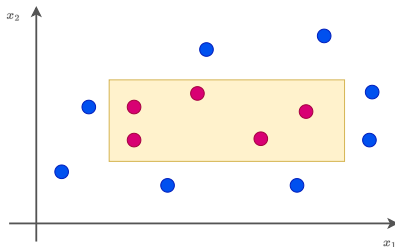
1. Each feature vector is labeled as

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

2. Each point in the training set is represented by an ordered pair (x, t) and the training set containing

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

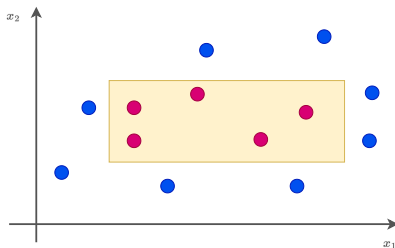
3. Each label is generated from a concept $c \in \mathbb{C}$, where \mathbb{C} is called a **concept class**.
4. The training data now can be plotted in the 2-D space (x_1, x_2) , where point i is a data point and its label is given by t_i .





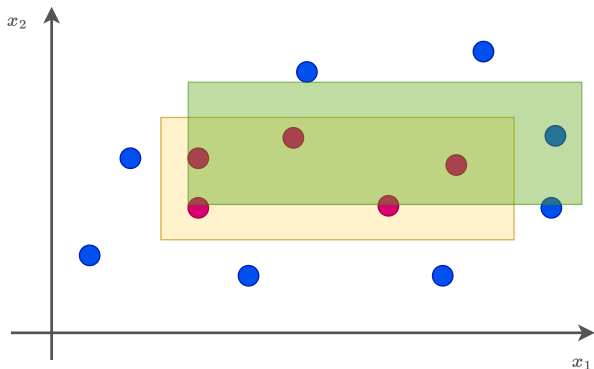
1. The learning algorithm should find a particular hypotheses $h \in H$ to **approximate** \mathbb{C} as **closely as possible**.
2. The expert defines the **hypothesis class** H , but he can not say the values for a, b, c, d .
3. We choose H and the aim is to find $h \in H$ that is similar to \mathbb{C} . This reduces the problem of learning the **class** to the easier problem of finding the **parameters** that define h .
4. Hypothesis h makes a prediction for an instance x in the following way.

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





1. After further discussion with experts and the analysis of the data, we believe that the rectangle is in the form of $(a \leq x_1 \leq b)$ & $(c \leq x_2 \leq d)$.
2. The above equation assumes H to be a rectangle in 2-D space.
3. For suitable values a, b, c, d , the above equation fixes $h \in H$ from the set of **axis aligned rectangles**.





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

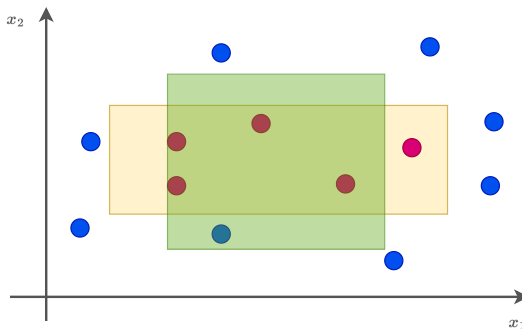
$$E_E(h|S) = \frac{1}{N} \sum_{i=1}^N \mathbb{I}[h(x_i) \neq c(x_i)]$$

4. When $E_E(h|S) = 0$, h is called a **consistent hypothesis** with dataset S .
5. 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?
6. 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.



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

$$E_T(h|S) = \mathbf{Prob}_{x \sim D}[h(x) \neq c(x)] \quad (1)$$





Definition (Most specific hypothesis (h_s))

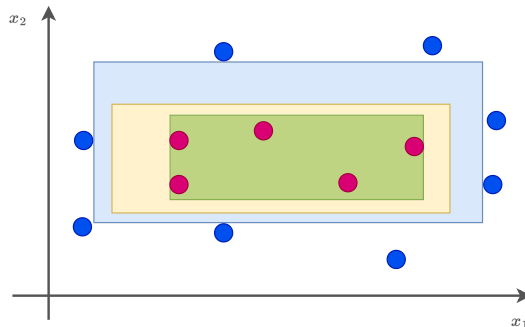
The tightest/smallest rectangle that includes all positive examples and none of the negative examples.

Definition (Most general hypothesis (h_g))

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

Definition (Version space)

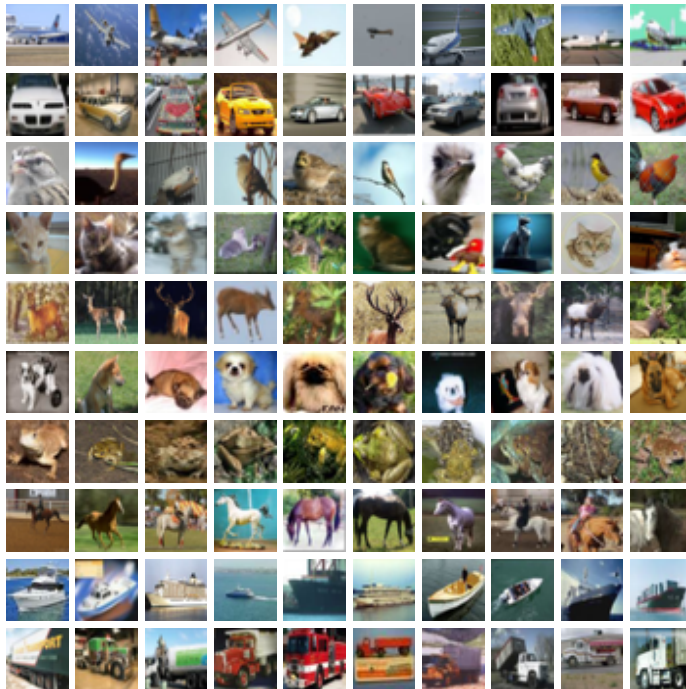
Version space is the set of all $h \in H$ between h_s and h_g .





1. We assume that H includes \mathbb{C} , that is there exists $h \in H$ such that $E_E(h|S) = 0$.
2. Given a hypothesis class H , it may be the cause that we cannot learn \mathbb{C} ; that is there is no $h \in H$ for which $E_E(h|S) = 0$.
3. 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?



Supervised learning

Regression



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

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

2. 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

$$t_k = f(x_k) \quad \forall k = 1, 2, \dots, N$$

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

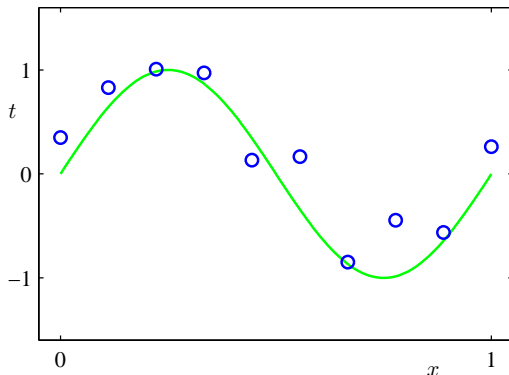
$$t_k = f(x_k) + \epsilon \quad \forall k = 1, 2, \dots, N$$

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



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

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



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

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

z_k denotes hidden variables



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

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

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

Supervised learning

Model selection



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

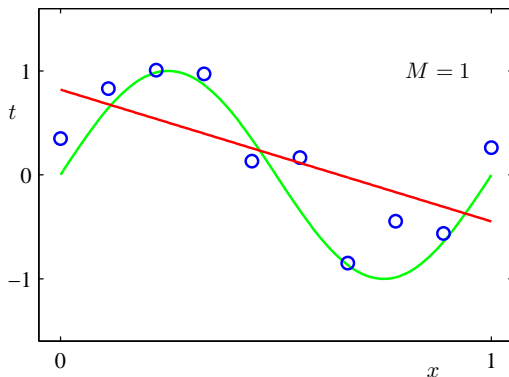
Definition (Inductive bias)

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.

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

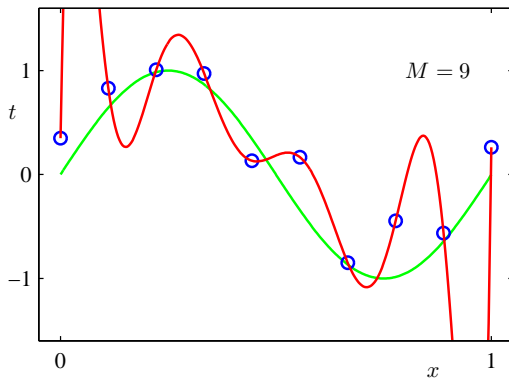


1. For best generalization, we should choose the right model that match the complexity of the hypothesis with the complexity of the function underlying data.
2. If the hypothesis is less complex than the function, we have **underfitting**





1. If the hypothesis is more complex than the function, we have **overfitting**

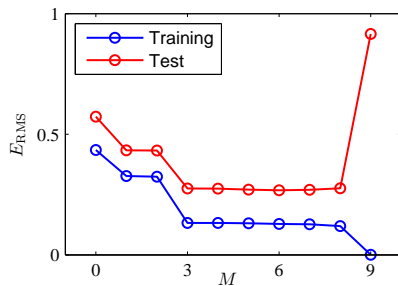


2. There are trade-off between three factors

- Complexity of hypotheses class
- Amount of training data
- Generalization error



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



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

Summary



1. 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 $p(x, t)$.
2. In order to have successful learning, three decisions must take
 - Select appropriate model ($g(x|\theta)$)
 - Select appropriate loss function

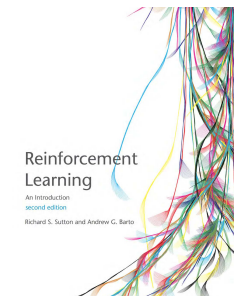
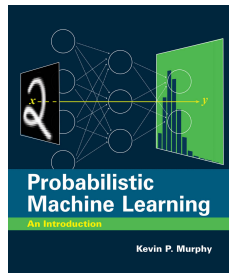
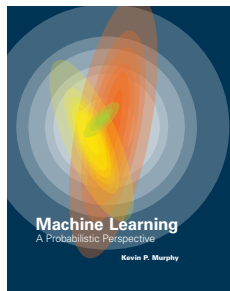
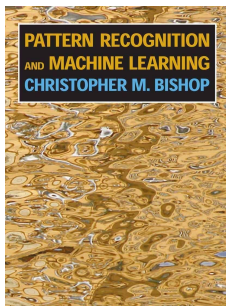
$$E_E(\theta|S) = \sum_k L(t_k, g(x; \theta))$$

- Select appropriate optimization procedure

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

References

1. Main books



2. Other useful books are

- Tom M. Mitchell, [Machine Learning](#), (Mitchell 1997)
- E. Alpaydin, [Introduction to Machine Learning](#), (Alpaydin 2014)
- T. Hastie, R. Tibshirani, and J. Friedman, [The Elements of Statistical Learning: Data Mining, Inference and Prediction](#), (Hastie, Tibshirani, and Friedman 2009)
- C. Szepesvari, [Algorithms for Reinforcement Learning](#)(Szepesvari 2010)



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



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 (ACML2013)
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)



1. Packages:

- R <http://www.r-project.org/>
- Weka <http://www.cs.waikato.ac.nz/ml/weka/>
- RapidMiner <http://rapidminer.com/>
- MOA <http://moa.cs.waikato.ac.nz/>








2. Datasets:

- UCI Machine Learning Repository <http://archive.ics.uci.edu/ml/>
- StatLib <http://lib.stat.cmu.edu/datasets/>
- Delve <http://www.cs.toronto.edu/~delve/data/datasets.html>



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).



-  Alpaydin, Ethem (2014). *Introduction to Machine Learning*. 3rd ed. MIT Press.
-  Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag.
-  Hastie, Trevor, Robert Tibshirani, and Jerome Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. 2nd ed. Springer.
-  Mitchell, Tom M. (1997). *Machine Learning*. McGraw-Hill.
-  Murphy, Kevin P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.
-  — (2022). *Probabilistic Machine Learning: An introduction*. The MIT Press.
-  Szepesvari, Csaba (2010). *Algorithms for Reinforcement Learning*. Morgan and Claypool Publishers.

Questions?