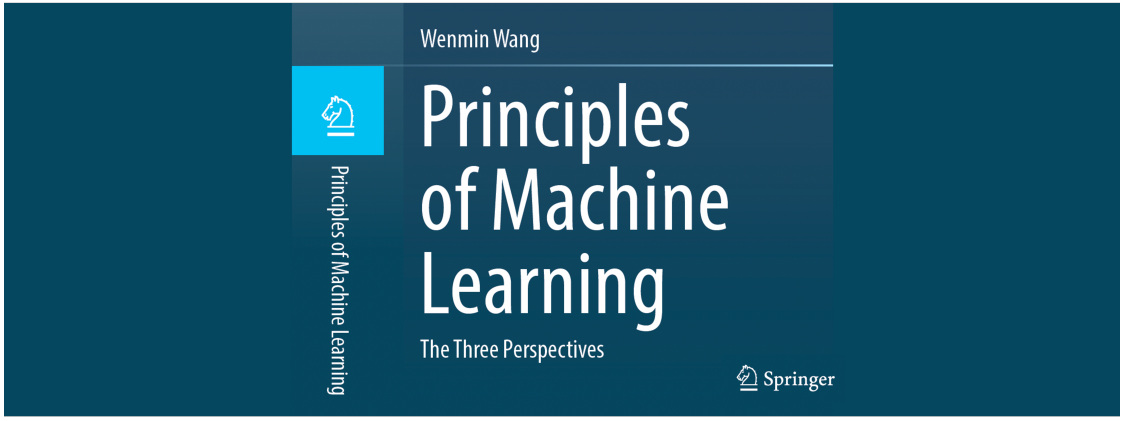

Wenmin Wang
Principles of Machine Learning
The Three Perspectives
Principles of Machine Learning
Springer

Part III Paradigms

8 Supervised Learning Paradigm

Contents

What is Supervised Learning

Supervised learning is one of the three major paradigms of machine learning. It is to learn from a large amount of labeled data, means experiences.

Definition: Supervised Learning

Supervised Learning (SL) is a learning paradigm with labeled data that is used to train the learning algorithm to obtain the best learner called the hypothesis. This hypothesis is then used to predict unknown input data to obtain corresponding output results.

The labeling refers to marking the expected output value for each input data, forming a data pair. E.g., labeling an email as spam or non-spam, and labeling a handwritten Arabic number as the corresponding number.

What is Supervised Learning

The hypothesis is also known as the model of supervised learning.

After training, the hypothesis can be used to predicate for new unseen data.

> It is a way of “teaching” the learning algorithm,
> like that a “teacher” gives a course.

It has the longest-standing, most content-rich in machine learning.

Its application field is extremely wide, for example:

image classification, optical character recognition (OCR), handwriting recognition, information retrieval, recommendation systems, spam mail detection, speech recognition, bioinformatics, and chemical biology.

Formal Description

Supervised learning can be represented as a 5-tuple $\langle \mathcal{X},\ \mathcal{Y}, S, H, \mathcal{L} \rangle$. Where $\mathcal{X}$: input space, $\mathcal{Y}$: output space, S: training sample set, H: hypothesis set, $\mathcal{L}$: loss function.

Using $P(\boldsymbol{x})$ from $\mathcal{X}$ to obtain n independent and identically distributed (i.i.d.) data:

$$D = \{\boldsymbol{x}_i \mid \boldsymbol{x}_i \in \mathcal{X} \text{ and } i = 1, \ldots, n\}.$$

$\forall \boldsymbol{x}_i \in \mathcal{X}\ (i = 1, \ldots, n)$, using $P(y|\boldsymbol{x})$ to label the corresponding target output $y_i \in \mathcal{Y}$.

Based on $P(\boldsymbol{x}, y) = P(y|\boldsymbol{x})\, P(\boldsymbol{x})$ to form the labeled training sample set:

$$S = \{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, n\} \subseteq \mathcal{X} \times \mathcal{Y}.$$

Hypothesis set H is the algorithm designed for a task of supervised learning:

$$H : \mathcal{X} \to \mathcal{Y}\ .$$

Formal Description

Using S to train the H and obtain a $h \in H$ with smallest expected error, so that predicted value $h(\boldsymbol{x}) = \hat{y}$ closest to target output y.

The $\mathcal{L}$ is to measure the error between $\hat{y}$ and y, and to minimize its expected error:

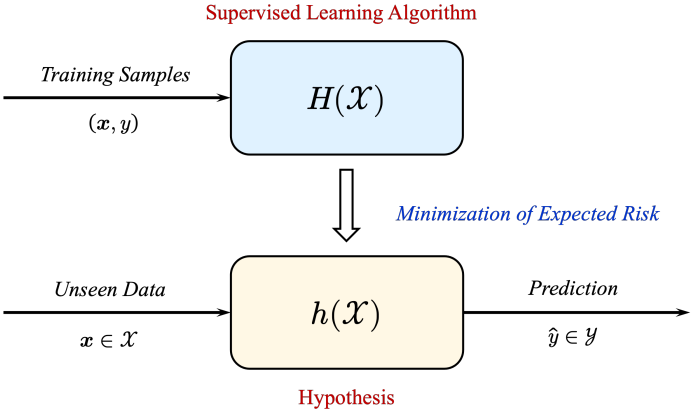
$$\mathcal{L}(h(\boldsymbol{x}), y) = \mathcal{L}(\hat{y}, y) = \arg\min_{h \in H} \mathbb{E}[h(\boldsymbol{x}) - y] .$$

That is to say, $\mathcal{L}(\hat{y}, y)$ is a mapping from output space to real numbers $\mathbb{R}$:

$$\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R} .$$

The $\mathcal{L}$ depends on the task and algorithm in supervised learning. Here, the $\mathcal{L}(\hat{y}, y)$ is just an abstract representation. The choice of $\mathcal{L}$ is a determining factor for selecting $h(\boldsymbol{x}) = \hat{y}$. The $\mathcal{L}$ also affects the convergence speed of supervised learning.

Illustrated Description

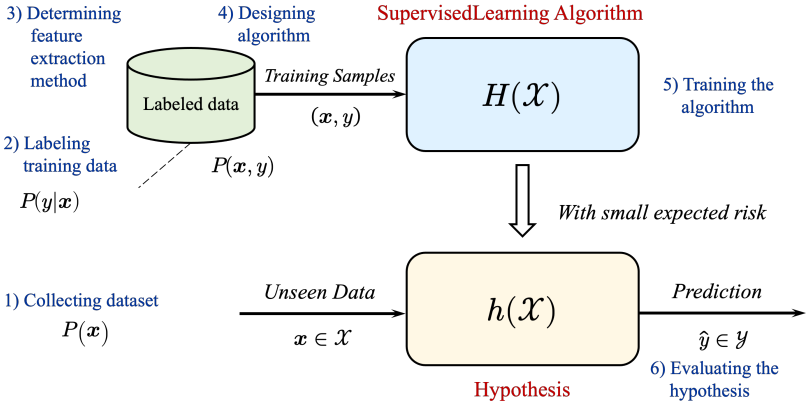


A schematic diagram of supervised learning process: the training process of hypothesis $h(\boldsymbol{x}) \in H(\mathcal{X})$, and the process of using trained hypothesis $h(\mathcal{X})$ to predict unknown data.

Operational Steps

1) **Collecting dataset**: That is, collecting data and constructing corresponding datasets, including training dataset, validation dataset, and test dataset.
2) **Labeling training data**:Training data needs to be labeled with its target output, forming data pairs, thereby constructing a training dataset.
3) **Determining feature extraction method**: Hand-crafted or learned features. The former uses feature descriptor and algorithm to extract features manually, and the latter through deep neural network to learn features automatically.
4) **Designing algorithm**: Design a supervised learning algorithm to solve a specific task.
5) **Training the algorithm**: Use the labeled training sample set to train the algorithm, to obtain the optimal hypothesis.
6) **Evaluating the hypothesis**: With validation dataset to evaluate the hypothesis, verify its accuracy and generalization ability.

Operational Steps



Classic Tasks of Supervised Learning

The classic tasks of supervised learning mainly include three types.

- **Classification**: the most common task in supervised learning, i.e., for known categories, it involves training the classification algorithm with a set of labeled data, to obtain its hypothesis, and then use it to predict the category of unknown data. The output result is discrete categories.
- **Regression**: also known as regression analysis, involves training to obtain the linear or nonlinear relationship between input and output variables through a set of labeled data, and then uses it to predict the output value of unknown data. The output result is continuous values.
- **Ranking**: training through a set of labeled data to obtain the ordered relationship between input data, and then use it to predict the relative relationship between unknown data.

Classic Tasks of Supervised Learning

The difference between classic tasks is mainly reflected in the output space $\mathcal{Y}$.

Therefore, based on the formalized description of supervised learning, we further formalize its output description for each task:

- **Classification**:
  its output space $\mathcal{Y}$ is known discrete categories, $\mathcal{Y} = \{c_1, c_2, \ldots, c_k\}$.
- **Regression**:
  its output space $\mathcal{Y}$ is continuous real numbers, $\mathcal{Y} \subseteq \mathbb{R}$.
- **Ranking**:
  its output space $\mathcal{Y}$ is a set with relative order after ranking, $\mathcal{Y} = \langle y, \preccurlyeq \rangle$, where: $y \in \mathcal{Y}$, and the symbol $\preccurlyeq$ is used to represent an ordered relationship.

About the Bias-Variance Problem

In supervised learning, the hypothesis obtained after training will have errors when predicting unknown data, namely the bias and variance.

These are a pair of parameters that can easily conflict: reducing bias will lead to an increase in variance; and vice versa.

This is called the “bias-variance problem” or the “bias-variance dilemma”.

The bias-variance problem is the core issue of supervised learning, specifically classification and regression tasks.

But there is no such problem in the paradigms of unsupervised learning and reinforcement learning.

Bias and Variance

The bias is an error caused by the erroneous hypothesis in supervised learning.

Definition: (Bias)

Bias, mathematically, is the difference between the expected value of prediction and the target value of the label, that is:

$$\text{Bias}\left[h\left(\boldsymbol{x}\right)\right] = \mathbb{E}\left[h\left(\boldsymbol{x}\right)\right] - y.$$

Bias is a measure of the accuracy of prediction derived from the hypothesis in supervised learning.

Since the average predicted value of the hypothesis converges to the expected value, it can be considered that the bias is approximately equal to the difference between the average predicted value and the expected value.

Bias and Variance

The variance is an error from sensitivity to small fluctuations in the training set.

Definition: (Variance)

Variance, mathematically, is the expectation of the square of the difference between the predicted value and the expected value of prediction, that is:

$$\mathrm{Var}\left[h\left(\boldsymbol{x}\right)\right] = \mathbb{E}\left[\left(h\left(\boldsymbol{x}\right) - \mathbb{E}\left[h\left(\boldsymbol{x}\right)\right]\right)^2\right].$$

Variance is a measure of the precision of prediction derived from the hypothesis in supervised learning.

Similarly, since the average predicted value of the hypothesis converges to the expected value, the variance can be seen as the degree to which the predicted value of the hypothesis deviates from its average predicted value.

Bias-Variance Tradeoff

The bias-variance tradeoff is a comprehensive consideration of bias and variance.

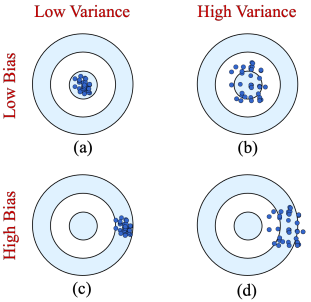
Bias is the distance that the predicted point deviates from bullseye, and variance is the degree of dispersion of predicted point.

(a) When bias and variance are low, the prediction of unknown data is most accurate.

(b) When bias is low and variance is high, the predicted points are scattered around bullseye.

(c) When bias is high and variance is low, the predicted points deviate from bullseye.

(d) When bias and variance are high, the predicted points are scattered away from bullseye.



Bullseye of bias-variance tradeoff.

Bias-Variance Decomposition

Bias-variance decomposition is a very useful and widely used tool for analyzing the expected generalization error of supervised learning models.

Taking mean squared error (MSE) to introduce the bias-variance decomposition.

First, assume that there is no noise in training samples, $y = f(\boldsymbol{x})$:

$$\text{MSE}(\boldsymbol{x}) = \mathbb{E}\left[(y - h(\boldsymbol{x}))^2\right] = \mathbb{E}\left[(f(\boldsymbol{x}) - h(\boldsymbol{x}))^2\right].$$

On the right side of the equation, subtract and add $\mathbb{E}[h(\boldsymbol{x})]$:

$$\begin{aligned}\mathbb{E}\left[(f(\boldsymbol{x}) - h(\boldsymbol{x}))^2\right] &= \mathbb{E}\left[(f(\boldsymbol{x}) - \mathbb{E}[h(\boldsymbol{x})] + \mathbb{E}[h(\boldsymbol{x})] - h(\boldsymbol{x}))^2\right] \\ &= \mathbb{E}\left[((f(\boldsymbol{x}) - \mathbb{E}[h(\boldsymbol{x})]) + (\mathbb{E}[h(\boldsymbol{x})] - h(\boldsymbol{x})))^2\right].\end{aligned}$$

Bias-Variance Decomposition

By applying the equation of the square of the sum and further expanding it, we have:

$$\begin{aligned}\mathbb{E}\left[(f(\boldsymbol{x})-h(\boldsymbol{x}))^2\right] &= \mathbb{E}\left[(f(\boldsymbol{x})-\mathbb{E}[h(\boldsymbol{x})])^2\right]+\mathbb{E}\left[(\mathbb{E}[h(\boldsymbol{x})]-h(\boldsymbol{x}))^2\right]+\\ &\quad 2\mathbb{E}\left[(f(\boldsymbol{x})-\mathbb{E}[h(\boldsymbol{x})])(\mathbb{E}[h(\boldsymbol{x})]-h(\boldsymbol{x}))\right]\\ &= \mathbb{E}\left[(f(\boldsymbol{x})-\mathbb{E}[h(\boldsymbol{x})])^2\right]+\mathbb{E}\left[(\mathbb{E}[h(\boldsymbol{x})]-h(\boldsymbol{x}))^2\right]+\\ &\quad 2\left(f(\boldsymbol{x})\mathbb{E}[h(\boldsymbol{x})]-\mathbb{E}[h(\boldsymbol{x})]^2-f(\boldsymbol{x})\mathbb{E}[h(\boldsymbol{x})]+\mathbb{E}[h(\boldsymbol{x})]^2\right).\end{aligned}$$

The third term on the right side of above equation is zero. Therefore, we get:

$$\begin{aligned}\mathbb{E}\left[(\boldsymbol{y}-h(\boldsymbol{x}))^2\right] &= E\left[(f(x)-E[h(x)])^2\right]+E\left[(E[h(x)]-h(x))^2\right]\\ &= (\text{Bias}[h(x)])^2+\text{Var}[h(\boldsymbol{x})].\end{aligned}$$

Bias-Variance Decomposition

Secondly, let S contain noise ϵ, i.e. $y = f(\boldsymbol{x}) + \epsilon$. Omitting the intermediate steps:

$$\begin{aligned}
\mathbb{E}\left[(y - h(\boldsymbol{x}))^2\right] &= \mathbb{E}\left[(f(\boldsymbol{x}) + \epsilon - h(\boldsymbol{x}))^2\right] \\
&= \mathbb{E}\left[(f(\boldsymbol{x}) - \mathbb{E}[h(\boldsymbol{x})])^2\right] + \mathbb{E}\left[(\mathbb{E}[h(\boldsymbol{x})] - h(\boldsymbol{x}))^2\right] + \mathbb{E}\left[\epsilon^2\right] \\
&= (\text{Bias}[h(\boldsymbol{x})])^2 + \text{Var}[h(\boldsymbol{x})] + \text{Var}(\epsilon).
\end{aligned}$$

Where $\text{Var}(\epsilon) = \sigma^2$. The MSE can be expressed in the following form:

$$\text{MSE}(\boldsymbol{x}) = (\text{Bias}[h(\boldsymbol{x})])^2 + \text{Var}[h(\boldsymbol{x})] + \sigma^2.$$

The MSE can also be expressed as:

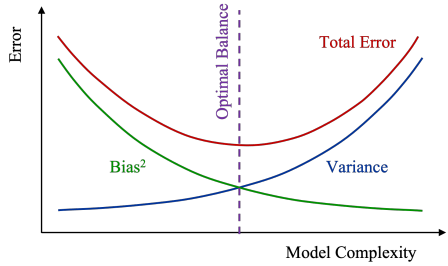
$$\text{Total Error} = \text{Bias}^2 + \text{Variance} + \text{Irreducible Error}.$$

Bias-Variance Decomposition

The meaning is: the total error equals the sum of the three parameters, i.e., square of the bias, variance, and irreducible error.

From this, it can be seen that to build a better model of supervised learning, a suitable balance between bias and variance must be found.

In conclusion, the issue of bias and variance is one of the important factors in supervised learning paradigm.



Bias-variance decomposition.

About the Risk Minimization Principles

The risk minimization principles are also the important factors to consider in the supervised learning paradigm, include:

- Expected risk minimization,
- Empirical risk minimization,
- Structural risk minimization.

Which provide the theoretical performance boundaries for a supervised learning algorithm.

Expected Risk Minimization

The expected risk of a hypothesis $h \in H$ is represented by $R(h)$, which means the expected risk of loss function, i.e.:

$$R(h) = \mathbb{E}_{\mathcal{X},\mathcal{Y}}\left[\mathcal{L}\left(h\left(\boldsymbol{x}\right), y\right)\right] = \int_{\mathcal{X},\mathcal{Y}} \mathcal{L}\left(h\left(\boldsymbol{x}\right), y\right) p\left(\boldsymbol{x}, y\right) d\boldsymbol{x}dy.$$

It is also known as *expected error* or *generalized error*.

Expected risk minimization refers to finding an optimal hypothesis $h^* \in H$ through training, so that its expected risk $R(h)$ is minimized, that is:

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

However, the $p\left(\boldsymbol{x}, y\right)$ is unknown, and thus $R(h)$ is also impossible to calculate. The minimized expected risk h^* is just a theoretical value.

Empirical Risk Minimization

But, we can calculate an approximation, called empirical risk, on training samples.

For a given training samples, $S = \{(\boldsymbol{x}_i, y_i) \mid i = 1, \ldots, n\}$, its empirical risk $R_{emp}(h)$ can be calculated by:

$$R_{emp}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(h\left(\boldsymbol{x}_i\right), y_i\right).$$

Empirical risk is also known as *empirical error*.

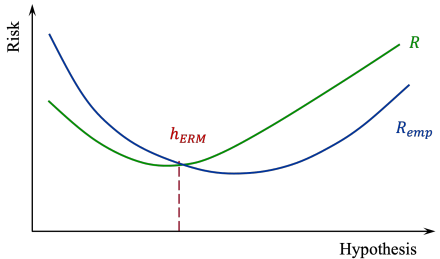
Empirical risk minimization (ERM) is also known as the principle of empirical risk minimization, which means that the learning algorithm should choose a hypothesis that minimizes empirical risk h_{ERM}, that is:

$$h_{ERM} = \arg\min_{h \in H} R_{emp}(h).$$

Empirical Risk Minimization

The performance of a supervised learning algorithm that satisfies ERM principle can be guaranteed to meet the uniform deviation bounds:

$$P\left(\sup_{h\in H}|R_{emp}(h)-R(h)|\geq\epsilon\right)\leq\delta.$$



For all $n\geq 1$ and $\epsilon>0$, the relationship between the δ and growth function $G_H(n)$ is:

$$P\left(\sup_{h\in H}|R_{emp}(h)-R(h)|\geq\epsilon\right)\leq\delta=8G_H(n)\,e^{\frac{-n\epsilon^2}{32}}.$$

This means the generalization bounds.

Structural Risk Minimization

The principle of structural risk minimization (SRM) is to solve the problem of structural risk by balancing the complexity of the model and the fit of the training samples.

Structural risk introduces a sequence of hypothesis sets with a nested structure:

$$H_1 \subset H_2 \subset \cdots \subset H_m \subset \cdots .$$

Given a training samples $S = \{(\boldsymbol{x}_i, y_i) \;\; i = 1, \ldots, n\}$, for each $h \in H$, and $m \in \mathbb{N}$, there exists the structural risk $R_{str}(h)$ as follows:

$$R_{str}(h) = \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}\left(h\left(\boldsymbol{x}_i\right), y_i\right) + \text{pen}\,(m, n) = R_{emp}(h) + \text{pen}\,(m, n) .$$

Where, pen $(\cdot, \cdot)$ denotes the penalty function of complexity.

Structural Risk Minimization

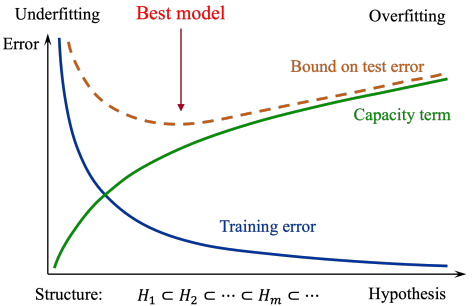
The usual practice is to choose a large model H, and define the regularization matrix of H, which is usually denoted as a norm, hence:

$$R_{str}(h) = R_{emp}(h) + \lambda \|h\|^2.$$

Where the λ is a regularization parameter, used to make an appropriate tradeoff between the model's generalization and complexity.

Structural risk minimization is for each H_m to find the smallest solution $h_{ERM}(\boldsymbol{x})$ in structural risk $R_{str}(h)$, that is:

$$h_{ERM}(\boldsymbol{x}) = \arg\min_{h \in H} [R_{str}(h)].$$



Limitations of Supervised Learning

The supervised learning paradigm is the most widely used paradigm in machine learning field.

But, it requires a large amount of labeled training samples, and accuracy and precision of a model depend on the quantity and quality of training samples.

The labeling of training samples has the following problems:

- The labeling is basically done manually. The time-consuming, labor-intensive, and large workload of manual labeling are self-evident.
- The labeling is often done by domain experts, e.g., the training samples for medical image segmentation require medical experts to label.

Over the years, some variants have gradually evolved, including weakly supervised learning, and semi-supervised learning.

Weakly Supervised Learning

Weakly supervised learning uses low-quality, small-quantity, “weak” labeled training samples. It can be divided into the following three types:

(1) Incomplete supervision:
Only a part of the training data is labeled, which is not enough to carry out complete supervised learning training for the machine learning model.

(2) Inexact supervision:
The labeling of training data is rough and lacks accuracy, which is called coarse-grained labels.

(3) Inaccurate supervision:
The technicians for labeling have limited experience, causing some labels to be wrong, making it difficult to become the ground truth.

Weakly supervised learning does not strictly require labeled training samples, which can objectively greatly reduce the cost of manual labeling.

Semi-Supervised Learning

The semi-supervised learning refers to training using a large amount of unlabeled data in addition to a small amount of labeled data.

In fact, semi-supervised learning was originally a human learning method, and its has theoretical and practical value.

Due to the related costs of labeling data, it is very difficult to obtain a sufficient number of accurately labeled training samples, while obtaining unlabeled data does not involve the problem of labeling costs.

Many experiments have shown that semi-supervised learning can significantly reduce the cost of labeled data and significantly improve the accuracy of the model.

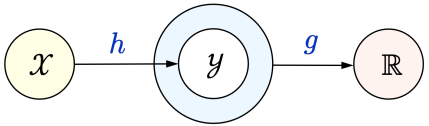
Semi-supervised learning can be seen as a special case of weakly supervised learning. Conversely, weakly supervised learning can be seen as a generalization of semi-supervised learning.

Label-Free Supervised Learning

The novelty of this approach lies in supervising the training of neural networks with specific constraint data, instead of using labeled data.

The constraint data is derived from prior knowledge, such as known physical laws.

It proposes a solution to the structured prediction problem in computer vision.



Consider a supervised method, by finding h to capture the structure required by g, it is possible to obtain h without providing the label y:

$$\hat{h}^* = \arg\min_{h \in H} \sum_{i=1}^{n} g\left(\boldsymbol{x}_i, h\left(\boldsymbol{x}_i\right)\right) + R\left(h\right).$$

Where $g : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is constraint function, and $R : H \to \mathbb{R}$ is regularization term.

About the No Free Lunch Theorems

There are such theorems in machine learning, that are no free lunch (NFL).

The theorems attempt to reveal the fundamental relationships between the functionalities and the performances of machine learning algorithms.

They are as implied by the popular adage:

> “No such thing as a free lunch.”

Meaning of the Theorems

In the metaphor of “no free lunch”, each “restaurant” (problem-solving process) has a menu listing each “dish” (problem) and its “price” (performance). Suppose each restaurant’s menu is exactly the same, only prices are different.

For a customer who likes to order every dish, the average cost of dining does not depend on the choice of restaurant. However, if a vegetarian and a meat-eater dine on a Dutch treat, the former’s cost will increase.

To reduce the average cost, it is necessary to know in advance: i) what dishes will be ordered, ii) in which restaurant these dishes will be ordered.

That is to say, the performance of problem-solving depends on using prior information to match the process with the problem.

Formally, when the probability of problem instances makes all problem-solving results the same, there is no free lunch.

Two Sub-Theorems

Let R: generalization error, S: training samples, m: number of samples, f: target function, and h: hypothesis.

Theorem 1: (Expectation of Algorithm)

The $\mathbb{E}[R|S]$ can be expressed as a inner product between $P(h|S)$ and $P(f|S)$:

$$\mathbb{E}[R|S] = \sum_{h,f} \text{Error}(h,f)\, P(h|S) P(f|S)\,.$$

Where $\text{Error}(h,f) = (h-f) \cup (f-h)$ is the error between h and f.

The result of above equation also applies to $\mathbb{E}[R|m]$, $\mathbb{E}[R|f,S]$, and $\mathbb{E}[R|f,m]$.

Two Sub-Theorems

Theorem 2: (Expectation of Two Algorithms Are Equivalent)

Consider the error function off-training-set. Let $\mathbb{E}_i[\cdot]$ denote the expectation obtained by ith algorithm. For any two algorithms $P_1(h|S)$ and $P_2(h|S)$, independent of the sampling distribution:

(a) $\mathbb{E}_1[R|f,m] - \mathbb{E}_2[R|f,m] = 0;$

(b) $\mathbb{E}_1[R|f,S] - \mathbb{E}_2[R|f,m] = 0;$

(c) $\mathbb{E}_1[R|m] - E_2[R|m] = 0;$

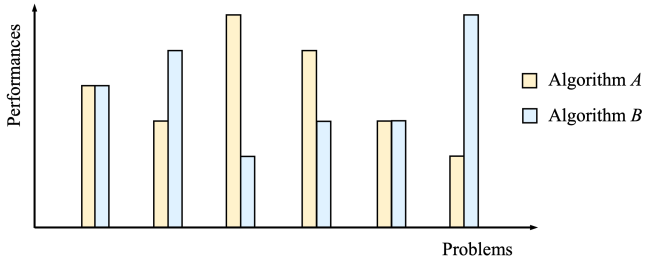
(d) $E_1[R|S] - E_2[R|S] = 0.$

The result of above equation also applies to $\mathbb{E}[R|m]$, $\mathbb{E}[R|f,S]$, and $\mathbb{E}[R|f,m]$.

Application for machine learning

The NFL theorems theorems for machine learning, specifically for supervised learning, refers to the case where the loss function is the misclassification rate.

If the focus is on off-training-set error, there is no a prior difference between learning algorithms.



Thank You