

机器学习是一门人工智能的科学，该领域的主要研究对象是人工智能，特别是在经验学习中改善具体算法的性能

机器学习核心功能：让分析算法无需人类干预的显示程式即可对最新数据进行学习

## 机器学习的发展

- 神经科学，认知科学
- 数学
- 计算

## 机器学习

学习的目的在于使期望风险最小化，但可利用信息只有样本，期望风险往往无法计算（经验风险最小化归纳原则）

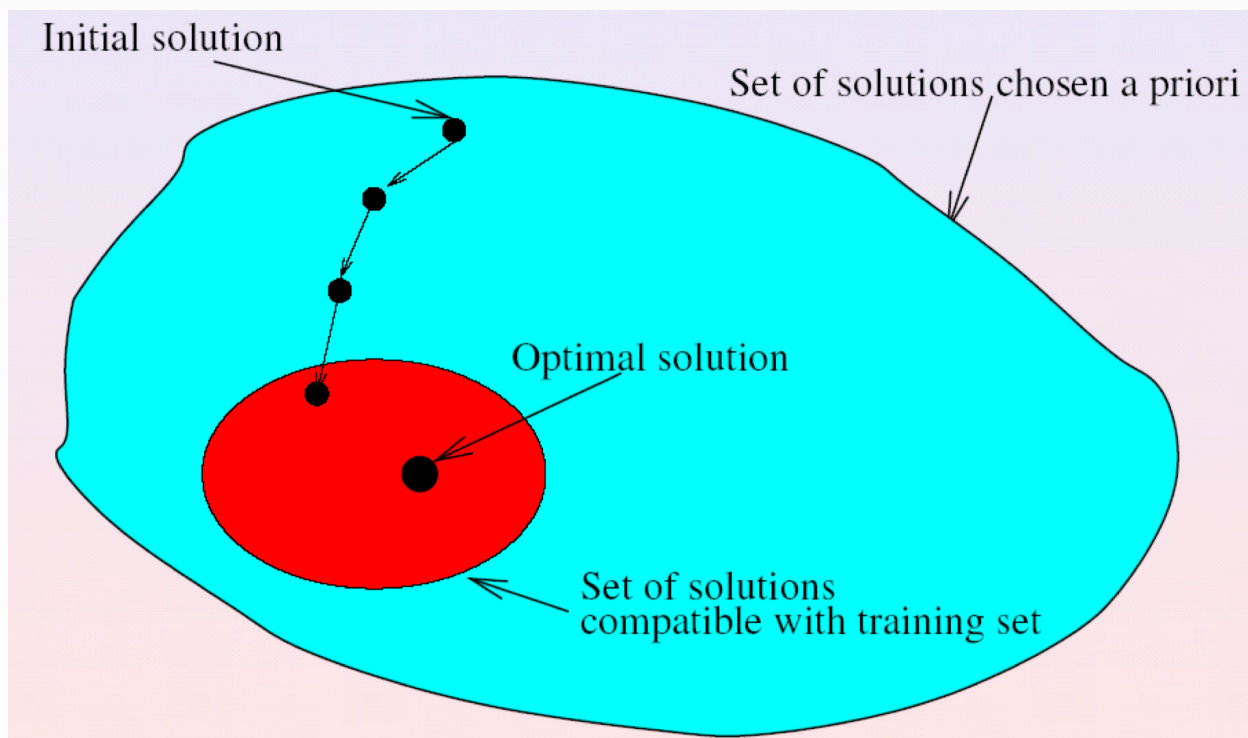
## 机器学习的核心功能

让分析算法无需人类干预和显式程序即可对最新数据进行学习

这就允许数据科学家根据典型数据集创建一个模型，然后利用算法自动概括和学习这些范例和新的数据源

## 机器学习的过程

机器学习的过程可以看作是最优方案搜索的过程



## 机器学习三类基本问题

### 分类

$$L(f(X, w), y) = \begin{cases} 1, & f(X, w) \neq y \\ 0, & f(X, w) = y \end{cases}$$

### 回归

$$L(f(X, w), y) = (f(X, w) - y)^2$$

### 概率密度估计

$$L(p(X, w)) = -\log(p(X, w))$$

## 学习分类

### 有监督学习

标定的训练数据

## 典型方法

- 全局：BN, NN, SVM, Decision Tree
- 局部：KNN、CBR(Case-base reasoning)

Case-based reasoning (CBR), broadly construed, is the process of solving new problems based on the solutions of similar past problems. An auto mechanic who fixes an engine by recalling another car that exhibited similar symptoms is using case-based reasoning.

## 无监督学习

不存在标定的训练数据

### 典型方法

- K-means、SOM....

## 半监督学习

结合（少量的）标定训练数据和（大量的）未标定数据来进行学习

### 典型方法

- Co-training、EM、Latent variables....

## 增强学习(Reinforcement Learning)

外部环境对输出只给出评价信息而非正确答案，学习机通过强化受奖励的动作来改善自身的性能。

## 多任务学习

Learns a problem together with other related problems at the same time, using a shared representation.

## 学习模型

### 单学习模型

- Linear models
- Kernel methods
- Neural networks
- Probabilistic models
- Decision trees

## 组合模型

### Boosting

- 结合低性能学习模型来产生一个强大的分类器组

初始化时对每一个训练例赋相等的权重 $1/n$ ，然后用该学算法对训练集训练 $t$ 轮，每次训练后，对训练失败的训练例赋以较大的权重，也就是让学习算法在后续的学习中集中对比较难的训练例进行学习，从而得到一个预测函数序列 $G_1, \dots, G_m$ ，其中 $G_i$ 也有一定的权重，预测效果好的预测函数权重较大，反之较小。最终的预测函数 $H$ 对分类问题采用有权重的投票方式，对回归问题采用加权平均的方法对新示例进行判别。

- Boosting 的流程

---

**Algorithm 10.1** *AdaBoost.M1.*

---

1. Initialize the observation weights  $w_i = 1/N$ ,  $i = 1, 2, \dots, N$ .
  2. For  $m = 1$  to  $M$ :
    - (a) Fit a classifier  $G_m(x)$  to the training data using weights  $w_i$ .
    - (b) Compute
$$\text{err}_m = \frac{\sum_{i=1}^N w_i I(y_i \neq G_m(x_i))}{\sum_{i=1}^N w_i}.$$
    - (c) Compute  $\alpha_m = \log((1 - \text{err}_m)/\text{err}_m)$ .
    - (d) Set  $w_i \leftarrow w_i \cdot \exp[\alpha_m \cdot I(y_i \neq G_m(x_i))]$ ,  $i = 1, 2, \dots, N$ .
  3. Output  $G(x) = \text{sign} \left[ \sum_{m=1}^M \alpha_m G_m(x) \right]$ .
- 

### Bagging

- 结合多个不稳定学习模型来产生稳定预测

让该学习算法训练多轮，每轮的训练集由从初始的训练集中随机取出的 $n$ 个训练样本组成，某个初始训练样本在某轮训练集中可以出现多次或根本不出现，训练之后可得到一个预测函数序列 $h_1, \dots, h_n$ ，最终的预测函数 $H$ 对分类问题采用投票方式，对回归问题采用简单平均方法对新示例进行判别。

- 不稳定模型：Neural Nets, trees
  - 稳定模型：SVM, KNN.
  - 主动学习 (Active Learning)
-

## What's the similarities and differences between this 3 methods: bagging, boosting, stacking?

All three are so-called "meta-algorithms": approaches to combine several machine learning techniques into one predictive model in order to decrease the variable (bagging), bias (boosting) or improving the predictive force (stacking alias ensemble).

Every algorithm consists of two steps:

Producing a distribution of simple ML models on subsets of the original data.

Combining the distribution into one "aggregated" model.

Here is a short description of all three methods:

**Bagging** (stands for Bootstrap Aggregation) is the way decrease the variance of your prediction by generating additional data for training from your original dataset using combinations with repetitions to produce multisets of the same cardinality/size as your original data. By increasing the size of your training set you can't improve the model predictive force, but just decrease the variance, narrowly tuning the prediction to expected outcome. (Each predictor in ensemble is created by taking a bootstrap sample of the data)

**Boosting** is a two-step approach, where one first uses subsets of the original data to produce a series of averagely performing models and then "boosts" their performance by combining them together using a particular cost function (=majority vote). Unlike bagging, in the classical boosting the subset creation is not random and depends upon the performance of the previous models: every new subsets contains the elements that were (likely to be) mis-classified by previous models.

**Stacking** is a similar to boosting: you also apply several models to you original data. The difference here is, however, that you don't have just an empirical formula for your weight function, rather you introduce a meta-level and use another model/approach to estimate the input together with outputs of every model to estimate the weights or, in other words, to determine what models perform well and what badly given these input data.

	Bagging	Boosting	Stacking
Partitioning of the data into subsets	Random	Giving mis-classified samples higher preference	Various
Goal to achieve	Minimize variance	Increase predictive force	Both
Methods where this is used	Random subspace	Gradient descent	Blending
Function to combine single models	(Weighted) average	Weighted majority vote	Logistic regression

