Week 6 Report

Yunhao Liu

Yh-liu18@mails.tsinghua.edu.cn

1. Definition and applications of Machine Learning

Definition: In particular, we define machine learning as a set of methods that can automatically detect patterns in data, and then use the uncovered patterns to predict future data, or to perform other kinds of decision making under uncertainty.Machine Learning relies on the lens of probabilistic modeling and inference.

Classification: supervised learning , unsupervised learning and Reinforcement learning.

① **Supervised learning approach:**The answer y is already known. The goal is to learn a mapping from input x to output y. Given a labeled set of intput -output pairs D = {(xi, yi)},D is called data set(samples or instances). In the simplest setting, each training input xi is a D-dimensional vector of numbers. These D-dimensional vectors are defined as attributes or feature. y can also be anything,but in most methods we assume that y is a categorical or nominal variable from some finite set.When y is discrete, the problem is known as classification or pattern recognition. When y is continuous,the problem is known as regression.The set of x is called attribute space,while the set of y is called label space.

② **Unsupervised learning approach:** Only input x is given.The goal is to find interesting patterns in the data.we use clustering which means dividing the training set into several groups, and each subset may help us to find some potential concepts and the intrinsic attributes.

③ **Reinforcement learning**:we set different reward signals to learn how to act or behaving in different occasions.

Applications:

① document classification, the goal is to classify a document, such as a web page or email message, into one of C classes

② Image classification and handwriting recognition

③ Face detection and recognition

2. Some basic knowledge:

Hypothesis space defines the class of functions mapping the input space to the output space.(a space of functions)

Induction： from instances to general rules->generalization.

Deduction： from general to specific->specification.

Training set and Testing set: We divide the given data set into two parts,one part is made up of training samples to produce the mapping(training set) and the other is made up of testing samples to revise or correct the mapping(testing set).

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}(f(\boldsymbol{x}_i)-y_i)^2 \ .$$

Expectation: discrete:

Continuous:

$$E(f;\mathcal{D}) = \int_{\boldsymbol{x}\sim\mathcal{D}} (f(\boldsymbol{x}) - y)^2 p(\boldsymbol{x}) \mathrm{d}\boldsymbol{x} .$$

$$\bar{f}(\boldsymbol{x}) = \mathbb{E}_D[f(\boldsymbol{x}; D)]$$

Bias: abs(the output y in learning function on the training set-the real output y), means the accuracy of the learning function itself. It is decided by the complexity of the function.For example, the linear function generally have a higher bias than x^n.

$$bias^2(\boldsymbol{x}) = (\bar{f}(\boldsymbol{x}) - y)^2$$

Variance: abs(the output y in learning function on the testing set – the real output y) means the stability of the function.For example,we selected 500 groups of samples ,every group has ten samples.We produce a function for each group.The linear function usually have a lower variance than the x^n function(the difference between different functions is smaller).

$$var(\boldsymbol{x}) = \mathbb{E}_D\left[(f(\boldsymbol{x}; D) - \bar{f}(\boldsymbol{x}))^2\right]$$

$$\varepsilon^2 = \mathbb{E}_D\left[(y_D - y)^2\right]$$

$$\begin{aligned}
E(f;D) &= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - y_D)^2\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}) + \bar{f}(\boldsymbol{x}) - y_D)^2\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2\right] + \mathbb{E}_D\left[(\bar{f}(\boldsymbol{x}) - y_D)^2\right] \\
&\quad + \mathbb{E}_D\left[2(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))(\bar{f}(\boldsymbol{x}) - y_D)\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2\right] + \mathbb{E}_D\left[(\bar{f}(\boldsymbol{x}) - y_D)^2\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2\right] + \mathbb{E}_D\left[(\bar{f}(\boldsymbol{x}) - y + y - y_D)^2\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2\right] + \mathbb{E}_D\left[(\bar{f}(\boldsymbol{x}) - y)^2\right] + \mathbb{E}_D\left[(y - y_D)^2\right] \\
&\quad + 2\mathbb{E}_D\left[(\bar{f}(\boldsymbol{x}) - y)(y - y_D)\right] \\
&= \mathbb{E}_D\left[(f(\boldsymbol{x};D) - \bar{f}(\boldsymbol{x}))^2\right] + (\bar{f}(\boldsymbol{x}) - y)^2 + \mathbb{E}_D\left[(y_D - y)^2\right] ,
\end{aligned}$$

$$E(f;D) = bias^2(\boldsymbol{x}) + var(\boldsymbol{x}) + \varepsilon^2$$

Occam's razor : We generally prefer those models that are simpler when other performance is similar.

No Free Lunch Theorem (NLF for short):If an algorithm performs well on a certain class of problems, then it necessarily pays for that with degraded performance on the set of all

remaining problems.(If an algorithm focus more on the personal features instead of general features ,it will not fits more than one category well)

E(error rate):a/m(a means the number of wrong answer,m means the total number)

Accuracy : 1 – E

Error:①training error (empirical error)（the error on the training set ）

②generalization error (the error on the new samples)

Overfitting: the learning mapping recognizes some personal features of the training sets as the features of all objects. It is difficult to solve which is a vital challenge of machine learning.

Underfitting: The learning ability is not good enough,so some features are not found by the learning function. Solution: add branches to the decision tree or increase the rounds in NN learning(not hard to solve)

3. Divide the data set into training set and testing set

Principle we need to obey:

① Testing set is mutually exclusive with the training set

② All members in the testing set are independent co-distributed sampling obtained from the real sample distribution.

Methods of dividing:

① **hold-out**(divide according to percent):choose about 2/3 ~ 4/5 of the data set randomly to make up the training set. In classification problem, each class is subdivided and finally merged.The capacity of the testing set is beyond 30.

② **K-fold cross validation**(fit for large data set):Divide the data set into k subsets(hierarchical sampling method to keep the consistency of data).Each time we take the union set of k-1 subsets as training set, the remaining one as testing set.The average of k test results is the final result.

③ **Bootstrapping**(fit for small data set): Dataset D (m samples) -> generate D'. For i in range (n),Randomly select a sample s, and place its copy into D',

$$\lim_{m\mapsto\infty}\left(1-\frac{1}{m}\right)^m \mapsto \frac{1}{e}\approx 0.368\ ,$$

about 36% data will never be selected.

4. Decision Tree

输入: 训练集 $D=\{(\boldsymbol{x}_1,y_1),(\boldsymbol{x}_2,y_2),\ldots,(\boldsymbol{x}_m,y_m)\}$;
属性集 $A=\{a_1,a_2,\ldots,a_d\}$.
过程: 函数 TreeGenerate(D, A)
1: 生成结点 node;
2: **if** D 中样本全属于同一类别 C **then**
3: 将 node 标记为 C 类叶结点; **return**
4: **end if**
5: **if** $A=\varnothing$ **OR** D 中样本在 A 上取值相同 **then**
6: 将 node 标记为叶结点, 其类别标记为 D 中样本数最多的类; **return**
7: **end if**
8: 从 A 中选择最优划分属性 a_*;
9: **for** a_* 的每一个值 a_*^v **do**
10: 为 node 生成一个分支; 令 D_v 表示 D 中在 a_* 上取值为 a_*^v 的样本子集;
11: **if** D_v 为空 **then**
12: 将分支结点标记为叶结点, 其类别标记为 D 中样本最多的类; **return**
13: **else**
14: 以 TreeGenerate(D_v, $A\setminus\{a_*\}$)为分支结点
15: **end if**
16: **end for**
输出: 以 node 为根结点的一棵决策树

Leaf node:(1)same category ->no need to divide (all the output y of the data set is same) (2)attribute set is empty / all sample are the same in every attributes(There is no other

attribute for further dividing already n dimension || All attributes of the rest samples are same,but y are different.For example,5 right and 2 wrong,we suppose the answer is right)
Root node:(3)the sample set is empty (recursive for the next node)

How to select the best arribute to divide:
Information entropy: An index used for measuring the purity of the sample.

$$\mathrm{Ent}(D) = -\sum_{k=1}^{|\mathcal{Y}|} p_k \log_2 p_k \ .$$

Proportion of Type k := pk
The lower the Ent(D), the higher the purity.
There are V possible values of discrete attribute a:{a1 a2……aV}

$$\mathrm{Gain}(D,a) = \mathrm{Ent}(D) - \sum_{v=1}^{V} \frac{|D^v|}{|D|}\mathrm{Ent}(D^v) \ .$$

We choose the arribute of the largest Gain as the root node.
Pruning: to improve the generalizing ability of decision trees

① Prepruning: (estimate before dividing) : Using the testing set for testing.If dividing cannot improve the ability, mark the current node as a leaf node)
② Postpruning: (buttom-up analyze those non-leaf nodes) : If replacing the node with a leaf node can improve the generalizing ability, then do it)