L1—gzy

ZhengyanGuo

April 2019

1. Introduction to machine learning

- Basic terminology:

Attribute = Attribute value such as color shape weight

data set(training set & testing set)

feature vector such as a melon with green color，big shape

- Hypothesis space:

defines the class of functions mapping the input space to the output space.

- Types of machine learning:

1.supervised learning:(classification & regression):instances with label information

2.semi-supervised learning

3.unsupervised learning(clustering):instances with unlabel information(through clustering to classify)

- Basic concepts:

1. Induction(instance -> general ; generalization) and deduction (general -> specific; specification)
2. Parametric and non-parametric model : whether the model has a fixed number of parameters.

   Parametric: easier, faster to use but the model is limited.

Non-paramatic: flexible, but is computationally intractable for large datasets.

3. About KNN(non-parametric): don't work well with high dimensionality input due to **curse of dimensionality**， lack of generalization for new data.
4. Linear regression and Logistic regression (classification) (both are Parametric);
5. Overfitting and underfitting：

we should avoid trying to model every minor variation in the input, since this is more likely to be noise than true signal。

- Model selection：

$$E(f;D)=\frac{1}{m}\sum_{i=1}^{m}\mathbb{I}\left(f\left(\boldsymbol{x}_i\right)\neq y_i\right) .$$

We care about generalization error, which is the expected value of the misclassification rate when averaged over future data. So we partition the training set into two parts: one to train the model, and the other to select the model complexity(vaildation set).so we can know vaildation set is different from test set.

I think we use the validation set to test our model's generalization ability. (generalization error: error in the new dataset;

Training error (or empirical error): error in the training set)

So we need some methods to selcet the two sets:

- Hold-out

Devide the dataset into training set and testing set directly, and we use the method of " stratified sampling ". We always do the above steps many times, and use the average result as the final result

Dilemma: the number of testing set's data: too few: can't guarantee the accuracy. too many: the training set will be small. the training takes up 2/3-4/5.

- Cross validation

1. Divide the dataset into k subsets.
2. for i=1->k: select k-1 subsets as training set and the last one as the testing set
3. use the average as the result
4. return

- Bootstrapping

1. For i=1->m: randomly copy one data and put it into D' set, mark the selected data
2. Use the unmarked data as the testing data and D' as the training data;

- No Free Lunch Theorem:

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

- Bias-variance decomposition:

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

1. bias: difference between the expected and the real label.

   describe the fitting ability of the learning algorithm.

2. variance: difference between the expected and real result.Describe the change of data how to impact result.

3. noise: describe the difficulty of the problem itself.

- Decision tree

  We use recursive to create a decision tree.

  We have some key points: the cases may result return(or the cases that the nodes will be marked as leaf nodes), exploration situation(when the sample of the attribute's one value is not empty, then generate the node);

- Pruning: Prepruning, Postpruning

  1. Both are used to improve the generalizing ability of the decision. Pruning is to slove the problem of overfitting.
  2. We judge each node's contribution to the generalization ability, if it is negative, then we should mark it as the leaf node to improve the generalizing ability.

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

- Neuron model:

1. threshold: the standard to judge whether the neuron will "actived".
2. M-P neuron model: the neuron accepts n signal, each signal has their weight, and use this input to produce the output. (we use Sigmoid function instead step function).

- Perceptron:

  Made up of two layers of neurons.

  We can use perceptron to represent and/or/not. (through use different threshold and input weight)