2020

Machine Learning

LAB 5

SAINAN DONG

Table of contents

1. Introduction

In development of Machine learning, algorithms are extremely essential. There are three kinds of algorithms in ML, we only discuss nearest neighbor, linear regression, regression forest for handling multiple regression methods, and gaussian process in this report.

1.1 Background and Importance

Machine learning is a branch of artificial intelligence, which mainly studies how to let machines learn from past experiences, model the uncertainty of data and make predictions in the future. In the field of machine learning, algorithms are usually divided into three categories: supervised learning, semi-supervised learning and unsupervised learning. "Supervision" refers to whether the machine can see the label of the sample during the learning phase. Semi-supervised learning is also commonly referred to as weakly supervised learning, which aims to automatically learn parameters in unlabeled samples with a small number of examples. As for unsupervised learning, people are often exposed to the problem of clustering: by analyzing the similarity of data samples, similar data can be combined into clusters. In the era of big data, another hotspot in the field of machine learning is to combine systems and algorithms to design large-scale and distributed machine learning algorithms and systems, so that machine learning algorithms can work in a multi-processor and multi-machine cluster environment and process more data.

1.2 Topic of Report

This report focuses on four algorithms which are used for implementation and critical analysis of multiple regression methods. The four method are Nearest neighbor, Linear regression, Regression forest, Gaussian process respectively. A data set which called toy problem is randomly generated to validated the four algorithms. In this project, I write a short script that calculate root mean square error (RMSE) from data

vector as an evaluation criterion to proceed performance comparison of different algorithms in terms of RMSE.

2. Methods

In this section, I only discuss the method of two algorithms, regression forest and gaussian process.

2.1 Regression forest

Random forest is a classifier containing multiple decision trees, is to establish a forest in a random way. The forest consists of a lot of decision trees, but these decision trees are not related. Regression forest is an algorithm that integrates multiple trees with the idea of integrated Learning. Its basic unit is the decision tree, and its essence belongs to the Ensemble Learning method, a big branch of machine Learning. Integrated learning is to use a series of learning devices to learn, and integrate the learning methods through a certain rule, in order to achieve better learning effect than a single device. Integration learning solves a single prediction problem by building several models and combining them, it works by generating multiple classifiers, or models, to learn and predict independently.

2.1.1 Principle of regression forest algorithm

Regression forest is composed of multiple decision trees. For each tree, the training set they used was taken back from the total training set. When training the nodes of each tree, the features used are extracted from all the features in a certain proportion and randomly without putting back. Its working principle is mainly to generate multiple classifiers or models to learn and make predictions independently.

2.1.2 The construction method of regression forest

The construction of regression forest consists of random sampling and complete splitting.

1) Random sample

   Regression forest samples the input data in rows and columns, but the two sampling methods are different. In the case of row sampling, the method adopted is to put back the sample, it means there may be duplicate samples in the sample set obtained by sampling. If the input samples size is N, then the size of samples sampled is also N. This makes the input samples of each tree are not all samples during the training, so it is relatively difficult to overfit. For column sampling, the method adopted is to select M samples (M <M) from M characteristics in accordance with a certain proportion of sampling without replacement.

2) complete splitting.

   In the process of building decision tree, each node of decision tree should be completely divided to split, until the node cannot be divided. This approach was adopted to construct the decision tree of a certain leaf node or is unable to continue to split, either in all samples is point to the same classifier

The construction process of each tree is as follows:

1) N is the number of training examples, and M is the number of variables.
2) In terms of m, the number of variables used when making decisions on a node.
3) From N training cases, a repeatable sampling method was used to sample N times to form a set of training sets, and the tree was used to predict the category of the remaining variables and to evaluate the errors.
4) For each node, m variables are randomly selected based on the point. According to these m variables, calculate the best way to divide them.
5) For every tree in the forest, no pruning technique is used. Every tree can grow intact.
6) The correlation between any two trees in the forest and the classification ability of each tree in the forest are two important factors influencing the classification effect of random forest. The higher the correlation between any

two trees, the higher the error rate. The greater the classification ability of each tree, the lower the error rate of the whole forest.

Make a prediction of the result:

1) Select test characteristics, use the rule of each randomly created decision tree to predict the result, and save the predicted result (target).
2) Count votes for each forest target.
3) The prediction target with the most votes is taken as the final prediction of the random forest algorithm.

2.2 Gaussian process

The gaussian process use the measure of homogeneity between points as the kernel function to predict the value of unknown points from the input training data.

2.2.1 Principle of gaussian process

The purpose of regression is to find a function that describes a given set of data points as closely as possible. This process is called fitting data with functions. For a given set of training data points, there may potentially be an infinite number of functions that can be used for fitting. The gaussian process provides an elegant solution by assigning each of these functions a probability value. The mean of the probability distribution represents the most likely representation of the data. Moreover, the probabilistic approach allows us to incorporate confidence in the prediction into the outcome of the regression.

2.2.2 The method of gaussian process

First, we shift the perspective from a continuous function to a discrete representation of a function: we are more interested in predicting the value of the function at specific

points, called test points X, than in finding an implicit function.Instead, we call the training data Y. So, the key point behind the gaussian process is that all the values of the function are derived from the multivariate gaussian distribution. That means that the joint probability distribution $p(x, y)$ spans the space of possible values of the function that we want to predict. The combined distribution of the test data and the training data has $|x| + |y|$ dimensions. What we're interested in is conditional probability $P(X|Y)$.

Now We've got the basic framework of the gaussian process, we need kernel function to construct this distribution. One advantage of kernels is that they can be combined to form a more specialized kernel which allows experts in a field to add more information and make predictions more accurate.

3. Validation on a toy problem

3.1 Generate dataset for toy problem:

By importing random, I randomly generate three lists which are X1, X2, X3, and these lists are nested list, each of them contains 100 lists, each small list contains 7 data. Meanwhile, I build a list called Y, it contains the result of the computation of X1, X2, X3 with three coefficients which are a, b, c. I have made a equal to 1, b equal to 2, and c equal to 3. I have defined the relationship between X1, X2, X3 and Y, that is Y = a * X1 + b * X2 + c * X3.

Therefore, I have got all data I want. Then I import pandas to merge all the data (X1, X2, X3, Y) into one dataframe. After that I transform the whole dataframe to array, and save it as CSV file, named as “toy problem.csv”.

3.2 Validation procedure

I create a new notebook to validate the four algorithms. I use the method I introduced above to generate and calculate the data for dataset of toy problem, and then I copied the method I load the data and the four algorithms into this notebook to test and check.

4. Experiments and analysis

4.1 Experiment results

4.1.1 Sarcos result

Algorithm	RMSE
Nearest neighbor:	5.02723298106817
Linear regression	5.6052146191036797
Regression forest	5.7163346809985445
Gaussian process	5.438570800663122

4.1.2 Toy problem result

Algorithm	RMSE
Nearest neighbor:	1.3655659532483575
Linear regression	1.4302270068296734
Regression forest	
Gaussian process	2.031629486813969

4.2 Analysis

4.2.1 Nearest neighbor

From the experiment result we can find that KNN method is the best method for accuracy.

KNN classification method is a non-parametric classification technology, which can achieve high classification accuracy for unknown and non-normal distribution

data, and has many advantages such as clear concept and easy implementation. However, there are also some problems, such as too much computation in the classification process, too much dependence on the sample base and inapplicability of the distance function to measure similarity.

When I use KNN method, I find there are many advantages.

1) The algorithm is simple and intuitive, easy to implement
2) Do not need to generate additional data to describe the rules, its rule is the training data itself, there is no requirement for data consistency which means noise can exist.
3) Although this method also relies on the limit theorem in principle, it is only related to a very small number of adjacent samples when making category decisions. Therefore, this method can avoid the problem of unbalanced sample size.

But I also find some shortcoming while testing.

1) The speeds of learning and classifying are slow in large training set.

   The nearest neighbor classifier is a lazy learning method based on instance learning, because it is a classifier constructed according to the given training samples. It is to store all the training samples first, and temporarily carry out calculation processing when classification is required. It is necessary to calculate the similarity between the samples and each sample in the training sample database to obtain the nearest K samples. For high-dimensional samples or large sample sets, the time and space complexity are high, and the time cost is $O(mn)$, where m is the spatial characteristic dimension of the vector space model, and n is the size of the training sample set.
2) It strongly depends on the size of simple.

   Limitations on the practical application of KNN algorithm are obvious. There are many categories that cannot provide enough training samples, which makes the relatively uniform feature space conditions required by KNN algorithm cannot be met, and the recognition error is large.
3) The function of features is the same.

Compared with the decision tree induction method, the traditional nearest neighbor classifier considers that each attribute has the same function (given the same weight). The distance of the sample is calculated according to all the characteristics (attributes) of the sample. Among these features, some are strongly related to classification, some are weakly related to classification, and some (perhaps most) are not related to classification. In this way, if the similarity of samples is calculated according to the same function of all features, the classification process will be misled.

4.2.2 Linear regression

In the machine learning algorithm, when minimizing the loss function, the gradient descent method can be used to solve it iteratively step by step to obtain the minimized loss function and model parameter values.

Linear regression is the most commonly used algorithm for regression tasks. In its simplest form, it is fitting a dataset with a continuous hyperplane. If there is a linear relationship between variables in the dataset, the degree of fit is quite high.

Linear regression is intuitive to understand and explain, and can be regularized to avoid overfitting. In addition, the linear model can easily update the data model through stochastic gradient descent. However, Linear regression is bad at dealing with nonlinearity, it is not flexible enough to identify complex patterns, and adding the right interaction terms or polynomials is tricky and time consuming

4.2.3 Regression forest

Random forest is a flexible, easy-to-use machine learning algorithm that yields good results in most cases, even without hyperparameter tuning. In the process of constructing regression forest, I find some characters of this method.

1) When the classification data set has many kinds of data, the classifier with high

accuracy can be produced.

2) When the classification dataset is unbalanced, the random forest can balance the error.
3) Random forest can calculate the closeness in each case, which plays a very important role in data mining, detecting deviators and visualizing data.
4) Performs well on large data sets.
5) Ability to assess the importance of each feature in a classification problem.
6) It also has ability to resist overfitting, through the average decision tree, reduce the risk of overfitting.
7) When more than half of the base classifiers fail, the prediction would be wrong. Random forest is very stable, even if a new data point appears in the data set, the whole algorithm will not be affected too much, it will only affect one decision tree, it is difficult to affect all the decision trees

However, it has been observed that data sets in random forests are overfitted if there is noise in the training data for some classification/regression problems. Because of its complexity, they require more time to train than other similar algorithms.

4.2.4 Gaussian process

Gaussian process is a non-parametric model. There are no training model parameters. Once the kernel function and training data are given, the model is uniquely determined. But the kernel function itself has parameters, such as the parameters σ and L of the gaussian kernel, which are called the hyperparameters of the model. The kernel function essentially determines the measure of the similarity of the sample points and affects the shape of the probability distribution of the whole function. The L larger, the function smoother and the variance between training data points more less, whereas the L smaller, the function tend to be more "twists and turns", predicted variance between training data points larger. σ directly control the size of variance, the bigger σ, variance more lager, and vice versa.

How to select the optimal kernel parameters is very important. Maximizes the probability of occurrence of y under these two hyperparameters, and finds the optimal parameter by maximizing Marginal log-likelihood.

Marginal logarithm likelihood:

$$log\, P(y/\sigma, l) = \log N\left(0, K_{yy}(\sigma, L)\right) = -\frac{1}{2}y^T k_{yy}^{-1} y - \frac{1}{2} log\left|k_{yy}\right| - \frac{N}{2} log(2\pi)$$

By maximizing edge likelihood, gaussian process regression can give a good regularization effect without cross validation.

In the process of realizing the gaussian process, I find some shortcomings of this method.

1) The gaussian process is a non-parametric model, and all data points must be inverted in each inference. For gaussian process regression without any optimization, n sample points time complexity is probably $O(n^3)$, complexity is $O(n^2)$, gaussian process become intractable when handle large amounts of data.
2) In the gaussian process regression, the prior is a gaussian process, and likelihood is also gaussian, so the obtained posterior is still a gaussian process. In the problem of likelihood not subject to gaussian distribution (such as classification), it is necessary to approximate the obtained posterior to make it still a gaussian process.