**1.Decision Tree**

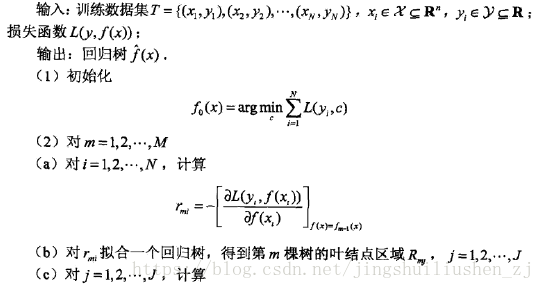
Decision tree is a very commonly used machine learning algorithm for classification and regression tasks. A decision tree is a tree structure in which each internal node represents a judgment on an attribute, and each branch represents a judgment on a output, and finally each leaf node represents a classification result.

**2.** **Gradient Boosting**

Gradient Boosting is a powerful machine-learning technique that achieves state-of-the-art results in a variety of practical tasks. For a number of years, it has remained the primary method for learning problems with heterogeneous features, noisy data, and complex dependencies: web search, recommendation systems, weather forecasting, and many others.

It is backed by strong theoretical results that explain how strong predictors can be built by iterative combining weaker models (base predictors) via a greedy procedure that corresponds to gradient descent in a function space. Most popular implementations of gradient boosting use decision trees as base predictors.

**算法描述**(是否需要)：



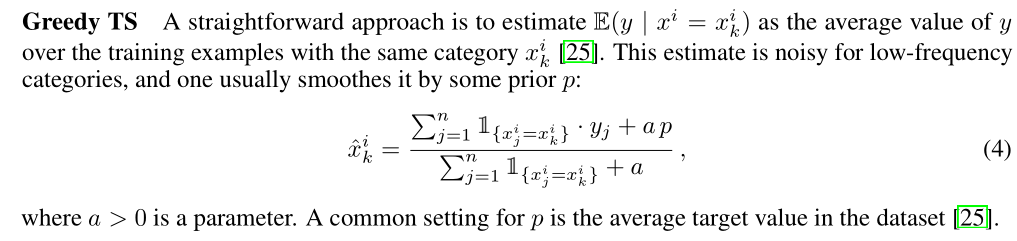
**3.Catboost**

Catboost is a machine-learning algorithm which based on Gradient Boosting Decision Tree. And this algorithm is good at handling categorical features and does well in reducing overfitting. Catboost has some features that are different from other gradient boosting techniques.

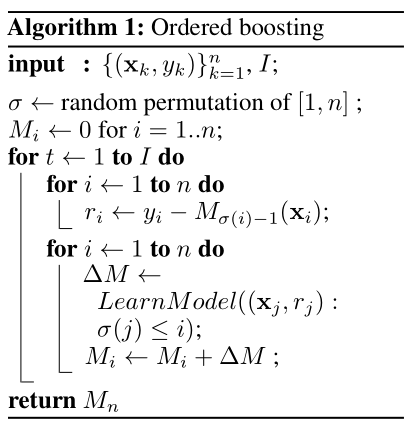
**Firstly**, Catboost uses symmetric tree as base predictor. Catboost claims that symmetric trees help avoid overfitting, increase reliability, and enable greatly accelerated predictions and so on.

**Secondly**, it handles categorical features in a special way. As shown in Equation1, catboost calculate calculate the frequency of occurrence of a category, add a hyper-parameter, and generate a new numerical features. this strategy requires disrupting the data set before training. Meanwhile, catboost algorithm uses different combinations of categorical features. When constructing the first node, only one feature is selected. When it comes to the second one, considering the selected feature and any of the categorical feature of combinations, choose the best of them. This is how the greedy algorithm is used to generate combinations.

Equation 1(是否需要)



**Thirdly**, Catboost presents an solution to avoid the "biased pointwise gradient estimates". Model is built using all training sets (except for the i-th data), and then a correction tree is built using the 1st to i-1st data and accumulatesto the original model .



**For implementation**, we use the features of the item recalled from the offline training set as the training data to train Catboost classifier. Catboost parameters are shown in Table 1. The values of the “iterations” are determined by the function of earlystopping to prevent overfitting. “max\_ depth” is set to None, and the sampling scale of data and features can be adjusted flexibly to get better performance. We can use 'F1', 'AUC' or even 'Recall' metrics for “eval\_metrics” because the positive and negative samples are unbalanced and we tend to put more attention on the recall of the positive samples and the positive samples rank. Finally the class\_weights parameter is used to weight the positive samples to mitigate the positive and negative sample imbalance.

Table 1 Partial parameters for Catboost

|  |  |
| --- | --- |
| iterations | maximum number of base tree preditors |
| max\_depth | maximum depth of the tree |
| learning\_rate | control the speed for parameters update |
| loss\_function | Specific loss function to minimize |
| class\_weights | class weights for samples |
| eval\_metrics | The metric used for overfitting detection and best model selection |
| subsample | Data sampling rate |
| colsample\_bytree | Feature sampleing rate |