Machine Learning

100 QUESTIONS

THOMPSON WONG

Introduction

MACHINE LEARNING

THOMPSON WONG

This book is written during the learning of machine learning, it summarized the frequent questions asked by junior or the major points that make people confuse. The original plan is 100 questions, it may be adjusted depends on the practice. Hope it will help you in machine learning field. Enjoy !!!

ABOUT AUTHOR

Thompson Wong
Data Scientists | Machine Learning | Startup
https://kinshing0930.wixsite.com/portfolio

Thompson, experienced with data analysis such as equipment optimization, fault detection and diagnosis, performance evaluation; software development includes analytics framework development, machine learning pipelines and application, RESTful API, optimization algorithm design; research and development for next generation technology; dashboard solution using Grafana.

Benefiting from the various working experience in different scale company, a solid experience and good soft skill is developed such as engineering mindset for problem solving, design thinking for the product development, efficiency and effectiveness in software features delivery, leadership to coordinate the works from different stakeholders, communication between client request and team capability.

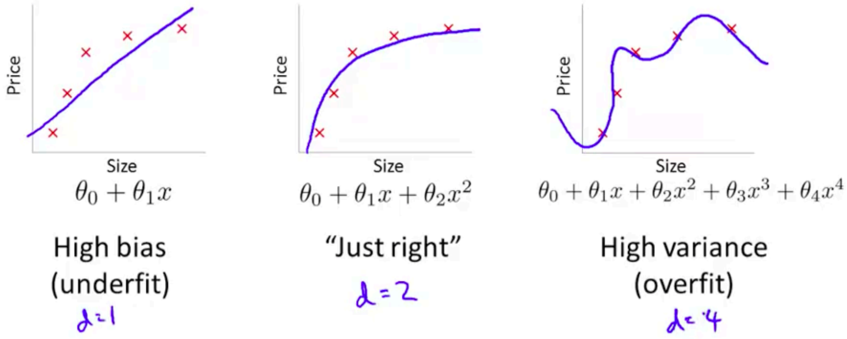
Communication to digest the needs, design thinking to make a innovation and breakthrough, engineering to solve problem step by step, leadership to coordinate the colleagues and make everyone satisfied in their job assignment.

Cost function, loss function, objective function

QUESTION 1

BIAS AND VARIANCE

Bias and variance is a basic concept to illustrate the machine learning model performance, like overfitting and underfitting.

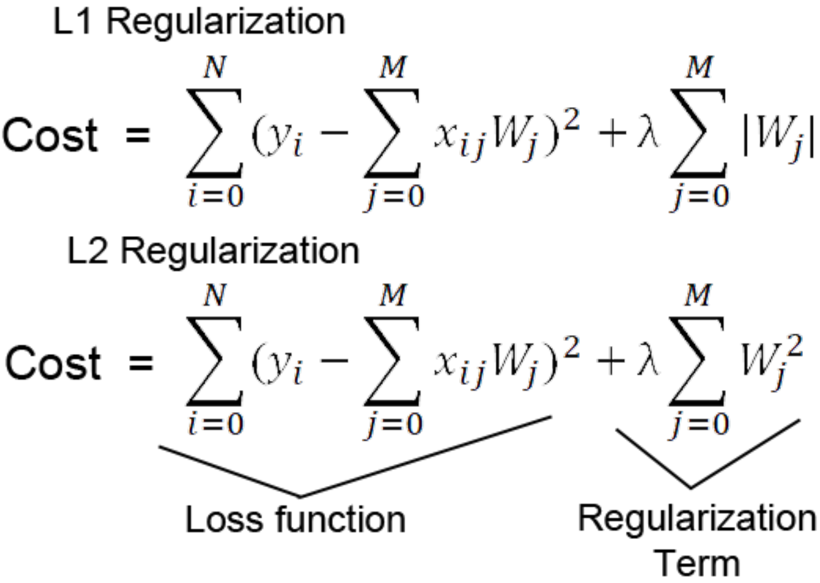


The bias usually represent the underfitting, as insufficient features makes prediction bias in training set. The variance usually represent the overfitting, too much features make machine learning model only perform well in training set, and get fail in newly unseen dataset or test set or validation set. You can comprehend as not generic model.

QUESTION 2

REGULARIZATION

Regularization is an effective things that control the model fitting performance. It is divided by L1 and L2 regularization. The formula is generally written in:



L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N}(y_i - \sum_{j=0}^{M} x_{ij}W_j)^2}_{\text{Loss function}} + \lambda \underbrace{\sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

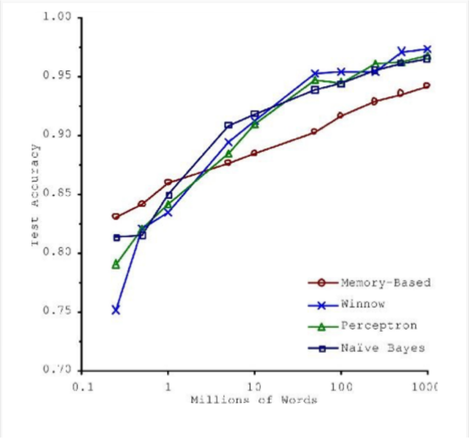
The L1 regularization is usually used in feature selection, L2 regularization is usually used in avoiding overfitting in machine learning. L1 regularization reduce model complexity via feature selection (reducing coefficients parameters/ features amount), L2 regularization reduce complexity by controlling coefficients parameters values.

QUESTION 3

DATA-CENTRIC OR MODEL-CENTRIC?

Data quality is always the major factors affecting the machine learning model performance. Garbage in, garbage out. Spending time for controlling data quality is much better than model algorithm. The existing algorithm is good enough, we all standing on the giant's shoulders.

When you have good enough data quality and quantity, the modelling result will be close to each others in different machine learning algorithms. So don't worry on selecting algorithm or developing new algorithm. Focus on improving data quality, Data-centric AI is greater than model-centric AI. The below figure illustrate the relationship between data quantity, quality and its performance



QUESTION 4

MODEL-BASED VS INSTANCE-BASED LEARNING

Model-based usually knew as supervised learning such as SVM (support vector machine), instance-based learning usually knew as unsupervised learning such as kNN.

In model-based learning, once model is built, the training data can be discarded (e.g. learned weights and bias value in SVM: $y=mx+c$) while instance-based learning use the entire dataset as the model, and looks at the close neighbourhood of the input example in the space of feature vectors and outputs the label that it saw the most often in this close neighbourhood.

Fundamentally, the model-based and instance-based learning has no direct relationship with the terms of data-centric and model-centric. Don’t get confused even the terms are similar.

QUESTION 5

DIFFERENCES BETWEEN RF, XGBOOST, LIGHTGBM, GBDT

XGBoost, lightGBM, GBDT, RF are attributed as **ensemble learning***, and boosting as major techniques to improve model performance while random forest adopts bagging method to improve performance.

Generally, the relationship is similar to the iteration of technologies. Therefore, RF -> GBDT -> XGBoost -> LightGBM. I shall create new question to detailedly explain each algorithms.

QUESTION 6

ENSEMBLE LEARNING

The purpose of ensemble learning is that combining multiple base-estimator's prediction result to improve single estimator's robustness and generalization.

The current ensemble learning is divided by two classes. One is strong relationship, serialization is required for result generation. Another one is no strong relationship between estimators, use parallel method to create multiple estimators. The representative of the former is Boosting, and the representative of the latter is Bagging.

Some terms can also be comprehended as ensemble learning as well such as blending, stacking. People may claim this is different. But no worries, it is the same. Just bear in mind is ok, no need to argue for a long time.

QUESTION 7

RANDOM FOREST

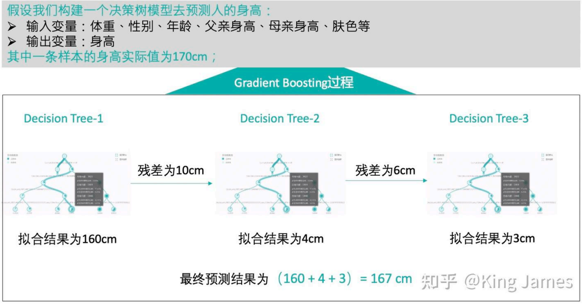
QUESTION 8

GRADIENT BOOSTING DECISION TREE

Gradient boosting decision tree also named as multi additive regression tree. (It is useful, additive telling you it is model use summation.)

The fundamental of gradient boosting decision tree is residuals based learning (fitting the negative gradient). Assuming you have dataset (x_1, y_1), (x_2, y_2), …(x_n, y_n). Then you use a model (linear or whatever like F(x)) to fit these data and minimise the loss function. After model fitted, you found its fitting effect is good but potential to advance. (e.g. residuals examples: 0.8 vs 0.9 | 1.4 vs 1.3). The easier way is creating a new model to fit the residuals of F(x), which equals to y_i - F(x_i).

The training dataset will be (x_1, y_1 - F(x_1)), (x_2, y_2 - F(x_2)), …(x_n, y_n - F(x_n)). The target of next estimator is changed to fit the residuals of former estimator. While next estimator can predict the residuals, and you sum it up, the model is accurate. Allow me to reference a picture for easy illustration.



假设我们构建一个决策树模型去预测人的身高：

- 输入变量：体重、性别、年龄、父亲身高、母亲身高、肤色等
- 输出变量：身高

其中一条样本的身高实际值为170cm；

最终预测结果为（160 + 4 + 3）= 167 cm

After several iterations, the model will converge finally. Thats why boosting is serialization method, and not support parallel like random forest.

Some may confuse, why GBDT use negative gradient. Loss function shall be minimum when ytrue=ypredicted . If you remember, we have regularization in the cost function. Therefore, when model adopts regularization, it is not able to use residuals for fitting, we shall calculate the gradient of loss function. In most of situation, if just fitting the residuals, the loss go smaller but higher regularization. The cost function may not be reduced. The objective function can be expressed by the gradient, uniform unit scale.

QUESTION 9

XGBOOST

QUESTION 10

LIGHTGBM

QUESTION 11

COST FUNCTION, LOSS FUNCTION, OBJECTIVE FUNCTION

Loss function is usually a function defined by different metrics. It is about the difference between y_{true}, $y_{\text{predicted}}$.

$$residuals = loss = y_{true} - y_{predicted}$$

Cost function is more general. It is the summation of loss function over the training set plus regularization (or penalty)

$$cost\ function = \frac{1}{2m} \sum_{i=1}^{n} (y_{true} - y_{predicted})^2 + \frac{1}{2m} \lambda |x|$$

Objective function is the most general term for any function that optimize during training. It is customized. Like a score of specific training.

Loss function is a part of cost function which is a type of an objective function. So you can know it as any function you want to minimize or maximize called objective function.

Don’t get confused in the terms definition, it is interchanged depends on the context, background… all the purpose is for gradient descent. Right?