Q & A

Ruichen Wang

February 20, 2019

Abstract

Some basic questions worth thinking.

Things you should know...

C4.5, ID3, K-means, SVM, Kernel function, SMO, EM, Naive bayes, Variational inference, PageRank, Adaboost, KNN, CART, Linear regression, Logistic regression, LDA, PCA, Random Forest, Bagging & stacking, Softmax, GBDT, Xgboost, LightGBM, PCA, Max-entorpy, Gradient diminishing, FM, LFM, MF, SVD, SVD++, Simhash, Max likelihood, ALS, L1 norm, L2 norm, Discriminate & Generative model, Entorpy, Cross-entropy, KL divergence, SGD, BGD, MBGD, Adam, Newton's method, Unbiased estimator, F1 score, Recall, Prercision, AUC, ROC, Cross validation, Bias-Variance tradeoff, Loss function, Overfitting & underfitting, DBSCAN, TF-IDF, Text rank, Word2vec, Glove, Fast-text, Collaborative Filtering, User-cf, Item-cf, Dropout, Batch norm, Max-pooling, Avg pooling, Global avg pooling, Respective field, LR normalization, Online learning, FTRL, FOBOS, RDA, Tree row & column sampling, ANN, Tree normalization, RNN, LSTM, GRU, CRF, RNN & CNN, KDtree, Resnet, Seq2seq, Deep wise CNN, 1X1 kernel, SVM multi-classification, Attention, Conv complexity, Pearson correlation

Why L1 regulation generates sparsity? L2 regulation cause blur?

Firstly, why do we want the result matrix to be sparse?

Consider 1 million dimension, calculate the inner product between w and x need a lot of computation. If the w can be sparse, the inner product will only be performed on the non-zero columns.

Or consider another situation, in some scenario, there are free data and many features, which is often called as **'small n, large p problem'**. If $n \ll p$, then our model will be very complex, our w will be a singular matrix ($|w| = 0$). In other words, **overfitting**.

One way to control overfitting is adding a regularization term to the loss function. Rigde ($l_2 norm$) and LASSO ($l_1 norm$) regression are two very common regression ways.

$$J(w) = Loss(x) + \lambda ||w||_2^2$$

$$J(w) = Loss(x) + \lambda ||w||_1$$

Assume we use loss using MSE, the target function can also be denoted as :

$$\min_w \frac{1}{n}||y - Xw||^2 \quad s.t. \lambda||w||_2^2 \leq C$$

$$\min_w \frac{1}{n}||y - Xw||^2 \quad s.t. \lambda||w||_1 \leq C$$

Back to the problem, intuitivly, the target loss will alway intersect at the coordinate axis when using l1 norm. Imaging high dimension situation, the angles will certainly more likely to be intersected, while the ball will not.

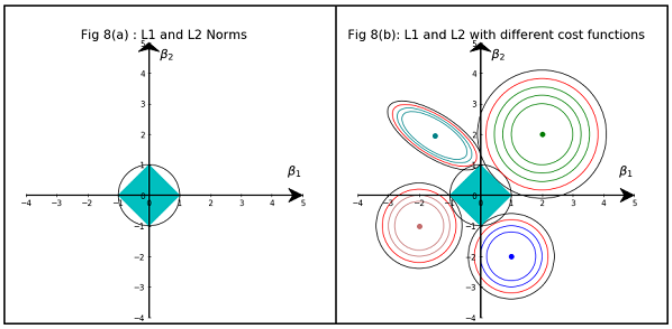

Figure 1: L1 and L2 norm.

For more math proof, see *http://freemind.pluskid.org/machine-learning/sparsity-and-some-basics-of-l1-regularization*

Why L2 regulation cause blur?

In generative models, eg.VAE, L2 norm / L2 loss / MSE tend to yield blurry images. We try to explain this in probabilistic settings. In Gaussian distribution, it defines as :

$$p(x|\mu, \sigma^2) = \frac{1}{Z} exp\left(-\frac{||\mu - x||^2}{2\sigma^2}\right)$$

$$logp(x|\mu, \sigma^2) \propto exp\left(-\frac{1}{2}||x_\mu - x||^2\right)$$

As we can see, minimizing MSE, is same as maximizing the log likelihood of gaussian, we make the assumption that our x comes from a gaussian.

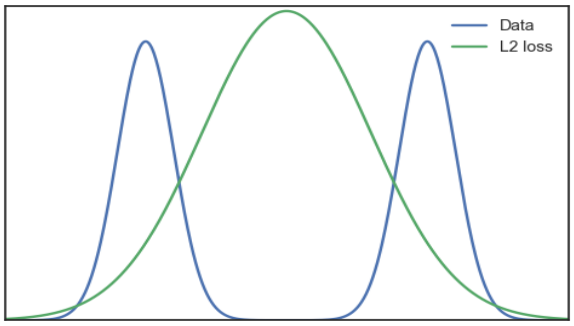

Figure 2: Gaussian and multinomial.

In reality, there often many hidden variables (multinomial) controls the x. A simple example, we have white and black dogs as dataset x. maximizing the likelihood will blur the two and generate gray dogs.

One-hot encoding for gbdt?

It is often the case that we have continuous and categorical features. One-hot encoding may produce very sparse variables. Tree based algorithm tries to increase information gain on the data it's splitting at any given level. If the data is very sparse, the one-hot encoded feature may be ignored as they are far too sparse.

Then how to using categorical features in trees?

In xgboost [1], it treat every input as numerical, It may be helpful is the categories as small. Otherwise, it may downgrade the performance.

In lightGBM, it said that a tree built on one-hot features tends to be unbalanced and needs to grow very deep to achieve good accuracy. It's automaticly provide a optimal solution $O(k * logk)$ to find the best optimal partition. This often performs better than one-hot encoding.

For more info, see lightgbm api.

xgboost, lightGBM, gbdt, GBM?

Gradient tree boosting is also known as gradient boosting machine (GBM) or gradient boosted regression tree (GBRT).

xgboost using taylor expansion to approximate the objective function. Regularization is defined as number of leafs plus l2 norm of leaf values.

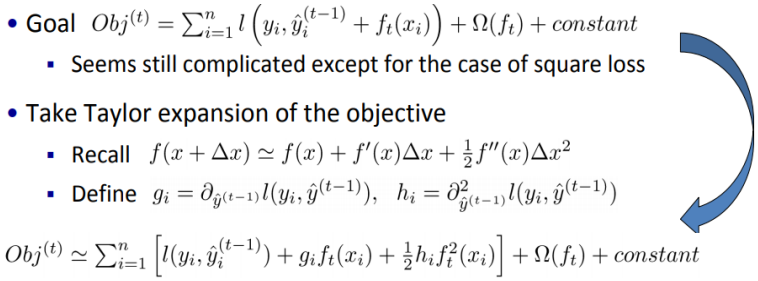

Figure 3:

for detail see pdf in ml papers basis.

lightgbm choose the leaf with max delta loss to grow(leaf wise, best first), while most tree grow by level wise. Using max depth to control overfitting.

bagging vs boosting vs stacking

bagging Bootstrap Aggregating

xgb rf lr difference

user-cf item-cf difference and application scenarios

svm vs lr

lstm vs gru

References

[1] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. In Balaji Krishnapuram, Mohak Shah, Alexander J. Smola, Charu C. Aggarwal, Dou Shen, and Rajeev Rastogi, editors, *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016*, pages 785–794. ACM, 2016.