Basis

Ruichen Wang

August 7, 2019

Some Basics

C4.5, ID3, CART, K-means, SVM, Kernel function, SMO, EM, Naive bayes, Variational inference, PageRank, Adaboost, KNN, Data cleaning, 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, Maxpooling, 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, Gibbs sampling, t-SNE, Learning to Rank, mAP, NDCG, Hash.

---

1 ID3, C4.5, CART difference?

What is Decision Tree (DT)?

DTs are non parametric superivised learning method, can be used for classification and regression. DTs need little data preparation. No need NOrmalization. White box model. Easily overfitting, sensitive to variation, because may build complete different trees, can be mitigated by using ensemble. Based on greedy opt to solve NP-complete problem, may reach local optimal. Need balance dataset.

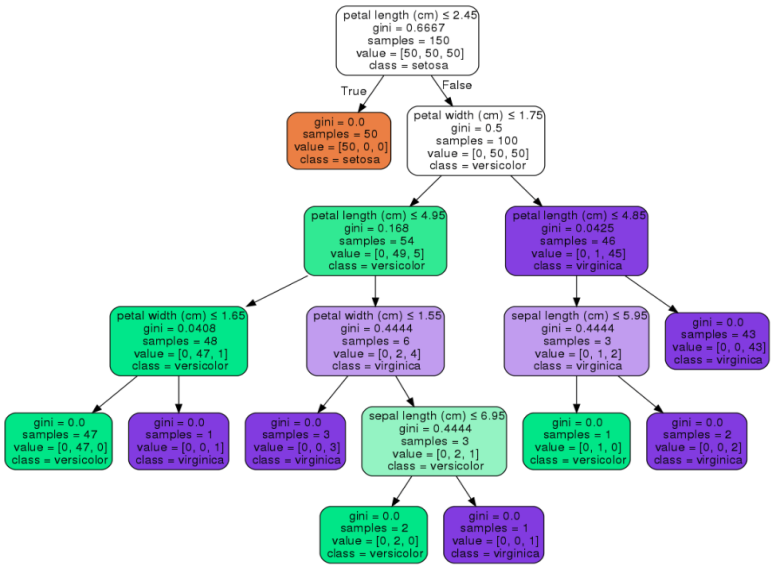

Figure 1: A decision tree using gini.

Pratical Tips:

- DT tend overfit if using a large number of features, Using a ratio to sample the features.
- Consider perform feature dimensionality reduction (PCA, Feature selection).
- Control max depth, control min samples per leaf to prevent overfitting

ID3 (info gain)
Entropy $H(S)$ is a measure of uncertainty of data.

$$H(S) = \sum_{x \in S} -p(x)logp(x)$$

Information Gain $IG(S)$ is a measure of differenece in entropy before and after S splited on attribute A.

$$IG(S, A) = H(S) - \sum_{t \in A} p(t)H(t) = H(S) - H(S|A)$$

C4.5 (gain ratio) C4.5 is the successor to ID3. Can handle continuous attribute.

$$GainRatio(S, A) = \frac{IG(S, A)}{IV(A)}$$

CART(Gini impurity) CART using binary split, Split based on one variable. Pruning mainly has two types : pre and post. CART use CCP (cost complexity pruning), which is defined by $\frac{errorrate}{treecomplexity}$

$$Gini(A,S) = 1 - \sum_{t \in A} p_t^2$$

2 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.

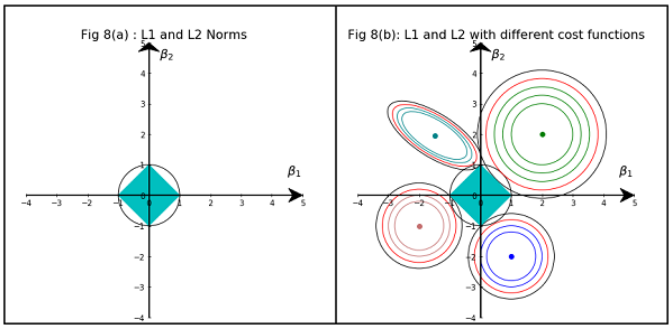

Figure 2: L1 and L2 norm.

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.

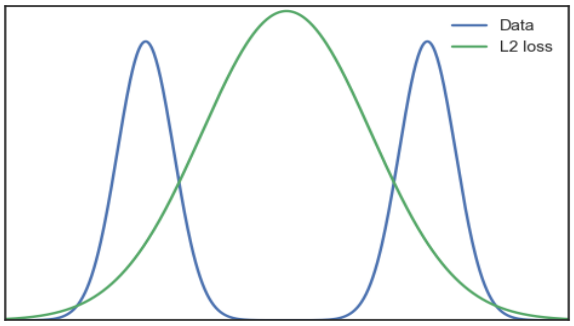

Figure 3: Gaussian and multinomial.

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.

3 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 [**?**], 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.

4 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.

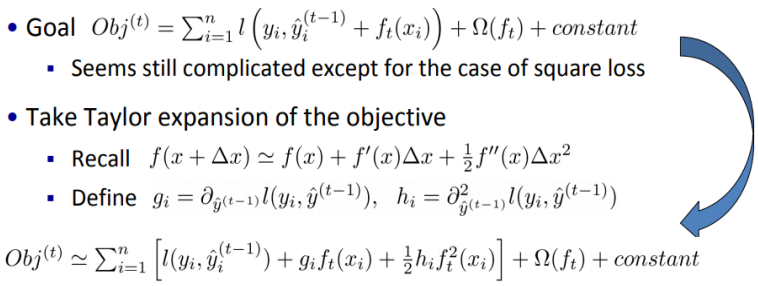

Figure 4:

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.

5 GD, BGD, SGD, momentum, adagrad, adam

GD refers to gradient descent, Using to negative gradient to optimize the ojbective function. Batch gradient descent (BGD) computes the gradient on the whole dataset, which if often slow and intractable. Can not be applied to online model which data is on the fly.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta)$$

BGD guaranteed to converge to the global minimum for convex problem.

SGD perform an update for each training example. Usually slowly decrease the learning rate.

$$\theta = \theta - \eta \cdot \nabla_\theta J(\theta; x^i; y^i)$$

Mini-batch gradient descent perform an update for every mini batch of n train examples.

Momentum adagrad,adam, rmsprop etc are all opt strategy.

6 EM Algorithm and VI

In the topic of clustering, it can be classified as hard and soft clustering. EM is a soft clustering method which is a generative model which trys calculate parameters of probability distribution. EM algorithm is used for maximum likelihood estimates of parameters. In bayesian statistics, it is often used to obtain the mode of the posterior marginal distributions of parameters.

E-step computes the expected value of $l(\theta; X, Z)$ given the observed data X, and current estimate θ_{old}.

$$Q(Z; \theta_{old}) = P(Z|X, \theta_{old})$$

M-step maximizing over θ the expectation computed.

$$\begin{aligned}\theta_{new} :&= \arg\max_{\theta} -KL(Q(Z)||P(Z|X,\theta)) \\ &= \arg\max_{\theta} \sum_i \sum_{z^i} Q_i(z^i) log \frac{p(x^i, z^i; \theta)}{Q_i(z^i)}\end{aligned}$$

In variational inference, we know that

$$logp(x) = KL(q(z)||p(z|x)) + (E_q[logp(z,x)] - E_q[logq(z)])$$

The difference of EM and VB is the kind of results they provide: **EM is just a point, VB is a distribution**. However, they also have similarities. EM and VB can both be interpreted as minimizing some sort of distance between the true value and our estimate, which is the Kullback-Leibler divergence.
For example:

When the Gaussian model involves latent variables and parameters only, Expectation Maximization is enough to solve the model.

If, on top of the latent variables, the parameters becomes random with prior distributions, the Variational Inference (or Variational Bayes) method is used.

7 Precision and Recall and F-measure and mAP

Accuracy Precision

- Meaningless, most case 99% documents are none-relevant

$$Accuracy = \frac{TP + TN}{N}$$

Precision

- Intuition: how much junk did we give to the user ?

$$Precision = \frac{TP}{TP + FP}$$

Recall

- Intuition: how much good stuff did we miss ?

$$Recall = \frac{TP}{TP + FN}$$

F-measure

- Intuition: combine recall and precision

$$F_\beta = \frac{(\beta^2 + 1) \cdot P \cdot R}{\beta^2 P + R}$$

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

However, all of this may be misleading. They could be the same system.

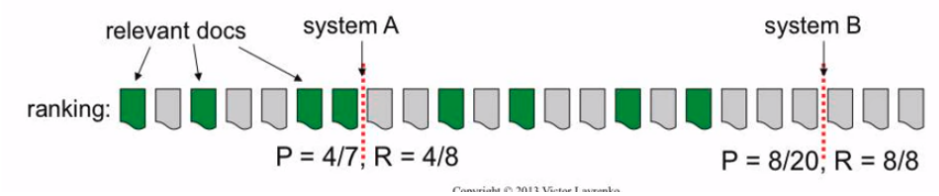

Figure 5: Same system, different score

Relationship between Recall/Precision/F1 in ranking:

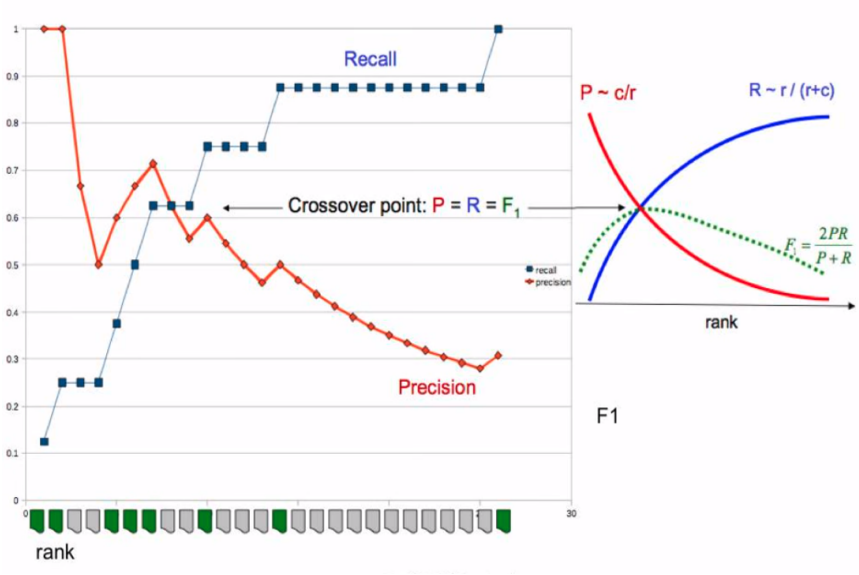

Figure 6: Same system, different score

Mean Average Precision

- Assume user wants to find many relevant docs
- biased towards top of the ranking (related with NDCG)

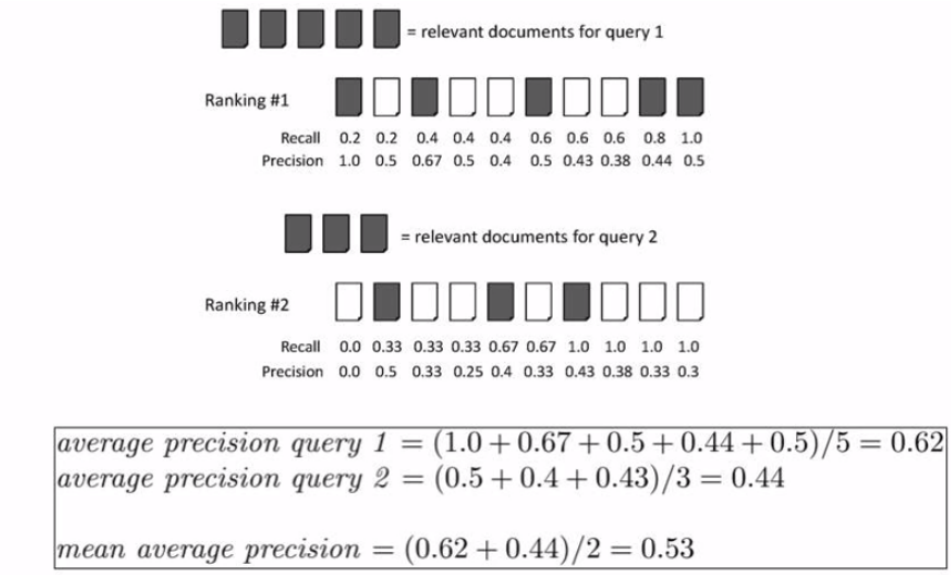

Figure 7: mean average precision

Normalized Discounted Cumulative Gain (NDCG)
When we want none-binary metrics.

$$DCG_k = \sum_{r=1}^{k} \frac{rel_r}{log(r+1)}$$

8 Learning to Rank

Pointwise
Pairwise

- RankNet, RankSVM, RankBoost, GBRank

Listwise

- AdaRank, SoftRank, SVM-MAP, lambdaRank, lambdaMART

9 SVM

Ref: PRML / pluskid blog

For linear classification, we want to find a hyper plane:

$$w^T x + b = 0$$

In SVM, we define functional margin as :

$$\widehat{\gamma} = y \cdot (w^T x + b) = yf(x)$$

Notice we multiply y to ensure $\widehat{\gamma} > 0$ And we define geometrical margin as :

$$\widetilde{\gamma} = y \cdot \frac{w^T x + b}{||w||} = \frac{\widehat{\gamma}}{||w||}$$

The only difference between these two margin is the scale $||w||$. As the functional margin $\widehat{\gamma}$ can scaled by the w.

In SVM, we want to find the best w to maximize the margin. Here we can define $\widehat{\gamma} = 1$, then we have our objective function:

$$max \frac{1}{||w||},\ s.t., y_i(w^T x_i + b) \geq 1$$

Which is equal to:

$$min \frac{1}{2}||w||^2,\ s.t., y_i(w^T x_i + b) \geq 1$$

This is a convex problem, we can apply **Lagrange multiplier**, convert the problem as:

$$L(w, b, \alpha) = \frac{1}{2}||w||^2 - \sum_{i=1}^{n} \alpha_i (y_i(w^T x_i + b) - 1)$$

$$\min_{w,b} \theta(w) = \min_{w,b} \max_{\alpha_i \geq 0} L(w, b, \alpha) = p^*$$

It means that, when the condition meets, we want to find the best w that minimize $\frac{1}{2}||w||^2$. Here we convert our objective to anther dual problem:

$$\max_{\alpha_i \geq 0} \min_{w,b} L(w, b, \alpha) = d^*$$

as :

$$d^* \leq q^*$$

with $\partial L/\partial w = 0$, $\partial L/\partial b = 0$, we have:

$$L(w, b, \alpha) = \sum \alpha_i - \frac{1}{2} \sum \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$s.t., \alpha_i \geq 0, \sum \alpha_i y_i = 0$$

Slack variable means that svm allow some of noise vector inside the hyperplane. specifically, we have:

$$y_i(w^T x_i + b) \geq 1 - \epsilon_i$$

And our new target is:

$$min \frac{1}{2}||w||^2 + C\sum \epsilon_i$$

Same mathematics as above, we finally get:

$$L(w, b, \alpha) = \sum \alpha_i - \frac{1}{2}\sum \alpha_i \alpha_j y_i y_j x_i^T x_j$$

$$s.t., 0 \leq \alpha_i \leq C, \sum \alpha_i y_i = 0$$

Note the only difference now is that α has a upper limit C.

10 推荐与排序

虽然都是重排序，但搜索和推荐的场景是有很大区别的。

从用户的角度来看，搜索场景中用户的查询需求是显式的、唯一的，对结果有几乎确定的期望（所以搜索引擎都在追求转型问答引擎）。而在推荐场景中，用户的需求没有明确给定，需要系统去推断用户的兴趣，这里用户的兴趣需求是隐式的、多元的。这就决定了两种场景下优化目标函数的不同，前者关注的是搜索结果与显式查询需求的相关性，后者关注的则是推荐内容与用户隐式且多元的兴趣分布的契合度。

从业务的角度来看，搜索场景希望用户停留时间越短越好，点击越少越好；推荐场景则希望用户停留时间越长越好，点击越多越好。这就决定了优化思路的不同，前者的目的是找到最优的若干条结果，最好是第一条就是用户想要的答案（一般不超过10，很少有用户在使用搜索引擎的时候会点击下一页）；而后者的目的是找到用户最喜欢的若干条新闻或物品，最好推荐的每一条新闻、每一件商品用户都喜欢。要想让用户都点击，那就不仅要符合用户兴趣，还必须具有一定的差异性和多样性，满足用户多元化的兴趣需求，而不能让某一类内容扎堆出现。所以前者是一个序列优化问题，而后者则是一个组合优化问题。

举个栗子。

女票说，我想吃苹果。那你就去买苹果，红的、大的、甜的、好吃的，一个就好。多了不要，香蕉不要，橘子不要，榴莲也不要。just follow the order，不要问东问西，不要画蛇添足。这是搜索。

女票啥也不说，但你知道女票喜欢吃苹果、香蕉、柚子、栗子（咦？栗子好像不是水果？管它呢，女票喜欢吃就好）、猕猴桃..... 那你就给她上一个大果盘吧，苹果香蕉切了片，柚子栗子去了皮，猕猴桃最好配点酸奶。你会想早上女票刚刚吃了两个大苹果，那就把苹果换成榴莲吧。你还会想大家都说这个季节的葡萄也很好吃，要不也放三两颗给女票尝尝？让用户不用动脑子，所见全是喜欢的，一块不剩吃个精光。这是推荐。