HOML - Chapter 4 - Training Models

Regressão Linear

Regressão linear busca encontrar parâmetros O que serão aplicados aos features para encontrar y.

Fórmula direta

Normal Equation. Podemos aplicar a fórmula direta pra descobrir os parâmetros ideais, mas tem o problema que **datasets com uma quantidade grande de features sofrem.**

Qualidades: é uma fórmula direta, escala bem quando o dataset tem poucos features mas muitas entradas. Não precisa normalizar!

Problemas: não escala bem coom muitos features O(n^2)

SVD

É o usado pelo scikit learn, é basicamente a mesma coisa que a fórmula normal, só que trabalha um pouco melhor, possui as mesmas qualidades e problemas.

Gradient descent e suas variações:

Batch Gradient Descent

Ao invés de utilizar uma fórmula direta, faz correções aos poucos, utilizando todos os dados disponíveis.

Temos os parametros X e O, e a função de custo que varia em relação a O. Quando tiramos a derivada, podemos saber o quão perto estamos do valor ótimo (se a derivada for próxima de zero estaremos próximos). O algoritmo roda em n passos, e para cada passo, On é alterado por d(custo)/d0n * On * learning rate.

Não funciona bem com datasets grandes, mas sim quando há várias features.

better suited for cases where there are a large number of features or too many training instances to fit in memory.

Qualidades: citado acima

Problemas: não escala bem com grandes datasets

Nota: caso os features não estejam escalados de maneira correta, levará muito mais tempo pro gradient descent chegar ao ponto ideal.

Nota 2: um possível problema de gradient descent poderia ser local minima, o algoritmo convergir a um ponto que não é o minimo global, mas com o MSE isso não acontece, pois não há mínimos locais

Stochastic Gradient Descent

Ao invés de utilizar todos os valores do dataset para cada iteração de aprimoramento, ele utiliza apenas uma entrada aleatória.

Qualidades: é extremamente rápido

Problemas: não tem o maior nível de precisão (porém é na maioria das vezes satifatório). Uma solução para esse problema é diminuir o learning rate com o passar do tempo.

Nota: ao usarmos uma cost function irregular, o stochastic pode ser mais preciso que o batch, devido à sua aleatoriedade.

Nota2: é importante dar um shuffle no algoritmo, se não ele pode pegar várias entradas bem similares, diminuindo a precisao do algoritmo.

Mini Batch Gradient Descent

Utiliza um conjunto aleatório do dataset para cada passo de aprendizado

The main advantage of Mini-batch GD over Stochastic GD is that you can get a performance boost from hardware optimization of matrix operations, especially when using GPUs.

Polinomial Regression

É uma regressão mas que consegue trabalhar de uma maneira não linear.

Algo interessante é o uso da tecnica de polynomial features. Vamos supor que nós temos um problema complexo para resolver, mas temos poucas variáveis... Nós podemos utilizar a polinomial features: pf(x,2) →[x,x^2]; se utilizarmos pf([x,y],2) → [x^2,y2,xy,x,y]. Aumenta em muito a quantidade de features e consegue relacionar dois features distintos.

O problema com polinomial regression é que ele pode dar overfitting. Uma solução para esse problema poderia ser a utilização de técnicas que limitam o tamanho dos parâmetros O.

Regularized Linear Models

Ridge Regression

Adiciona α ∑ θ^2 na função de custo (quanto maior o a, mais generalizada é a curva), então à medida que os O vão subindo, o custo vai aumentando, restringindo assim esses valores. Isso resulta em uma função de predição mais generalizada (não dá overfitting).

Pode ser utilizada conjuntamente com Gradient Descent ou Closed Form

Lasso Regression

A mesma coisa que a Ridge Regression, porém o termo adicionado é α ∑i=1|θ i |.

An important characteristic of Lasso Regression is that it tends to eliminate the weights of the least important features (i.e., set them to zero)

Elastic Net

A mistura entre Ridge e Lasso, utiliza os dois termos, cada um com seu peso.

So when should you use plain Linear Regression (i.e., without any regularization), Ridge, Lasso, or Elastic Net? It is almost always preferable to have at least a little bit of regularization, so generally you should avoid plain Linear Regression. Ridge is a good default, but if you suspect that only

a few features are useful, you should prefer Lasso or Elastic Net because they tend to reduce the useless features' weights down to zero, as we have discussed. In general, Elastic Net is preferred over Lasso because Lasso may behave erratically when the number of features is greater than the number of training instances or when several features are strongly correlated.

Early Stopping

Uma tecnica bastante interessante é a utilização de Early Stopping, que é parar o modelo quando ele chega em seu mínimo (esse mínimo deve ser alcançado no validation dataset).

Logistic Regression

É quando regressões (lineares ou polinomiais) são utilizadas para disponibilizar probabilidades (40% de ser um urso e 60% de ser um cachorro). Ela utiliza uma sigmoid function.

Softmax Regression

É usada para prever mais de uma classe, sem utilizar diversos classificadores binomiais.

Ainda preciso estudar os detalhes matemáticos.

Questões:

1. Which Linear Regression training algorithm can you use if you have a training set with millions of features?

Batch gradient descent, mini-batch and stochastig gradient descent.

2. Suppose the features in your training set have very different scales. Which algorithms might suffer from this, and how? What can you do about it?

Scaling é necessário para todos os Gradient Descents, então só não seria problemático para a fórmula normal e SVC. Para resolvermos esse problema é necessário apenas que nós normalizemos os nossos dados. O problema causado estaria no tempo utilizado para encontrar um boa solução, sem a normalização seria necessário vários passos a mais nos algorítmos de Gradient Descent.

3. Can Gradient Descent get stuck in a local minimum when training a Logistic Regression model?

Eu imagino que sim. Vamos checar no livro. E na verdade não. A função de custo é convexa assim como o RMSE e MSE, não tendo mínimos locais.

4. Do all Gradient Descent algorithms lead to the same model, provided you let them run long enough?

Não! Muitas vezes eles chegam em resultados bastante similares, mas a depender dos hyperparâmetros utilizados eles podem ter resultados completamente diferentes, por exemplo se utilizarmos um stochastic gradient descent com learning rate muito grande, ele provavelmente nunca vai nem encontrar um resultado bom, enquanto que um batch gradient descent com um pequeno learning rate, utilizando uma função de custo convexa, sempre alcançará um bom resultado, dado que esperemos o suficiente.

5. Suppose you use Batch Gradient Descent and you plot the validation error at every epoch. If you notice that the validation error consistently goes up, what is likely going on? How can you fix this?

Estamos provavelmente sofrendo de overfitting, uma técnica plausível para esse cenário seria o Early Stopping, ou seja utilizar os parâmetros de quando o erro do validation estava em seu mínimo.

6. Is it a good idea to stop Mini-batch Gradient Descent immediately when the validation error goes up?

Depende dos parâmetros, se você está utilizando batches grandes, faz sentido, pois a probabilidade de ter encontrado o mínimo já é bastante razoável, mas caso o batch seja pequeno, é melhor esperar algumas iterações a mais.

7. Which Gradient Descent algorithm (among those we discussed) will reach the vicinity of the optimal solution the fastest? Which will actually converge? How can you make the others converge as well?

Stochastic gradient descent vai ser o mais rápido. De maneira pura, o único que irá convergir é o batch. Para que os outros converjam podemos diminuir o learning rate à medida que o algoritmo avança.

8. Suppose you are using Polynomial Regression. You plot the learning curves and you notice that there is a large gap between the training error and the validation error. What is happening? What are three ways to solve this?

O seu moedlo está sofrendo de training bias, ele está com overfitting, devemos então generalizar o nosso modelo. Lasso Regression, Ridge Regression e Elastic Net Regression são 3 possíveis soluções. Por último você pode tentar aumentar o datamnho do seu dataset

9. Suppose you are using Ridge Regression and you notice that the training error and the validation error are almost equal and fairly high. Would you say that the model suffers from high bias or high variance? Should you increase the regularization hyperparameter α or reduce it?

Underfitting = High bias. Decrease a.

10. Why would you want to use: a. Ridge Regression instead of plain Linear Regression (i.e.,

without any regularization)?

No caso onde precisemos lidar com dados não lineares e particularmente onde o nosso modelo esteja com o bias elevado.

b. Lasso instead of Ridge Regression?

Quando você achar que alguns features são irrelevantes

c. Elastic Net instead of Lasso?

Lasso não funciona muito bem quando o númeto de features é maior que o número de instancias de treinamento ou quando há uma forte correlação entre os dados

11. Suppose you want to classify pictures as outdoor/indoor and daytime/nighttime. Should you implement two Logistic Regression classifiers or one Softmax Regression classifier?

Dois Logistic Regression, pois a diferenciação entre os binários é mais fácil de ser encontrada.

12. Implement Batch Gradient Descent with early stopping for Softmax Regression (without using Scikit-Learn).