TP555 - AI/ML
Gustavo Kreuzer Marengo - 835
Lista de Exercícios #2

Regressão Linear

1. Qual técnica de regressão linear você usaria se tivesse um conjunto de treinamento com milhares de features? Explique por quais razões você utilizaria esta técnica.

Utilizaria a técnica de gradiente descendente mini – batch pois como temos um número muito grande de features, logo o conjunto de treinamento total seria muito alto o que levaria a uma convergencia demorada. Com esse método, escolheriamos um conjunto aleatório de exemplos ou atributos a cada interação e calculamos os gradientes para cada um, levando a uma resposta de convergência mais rápida. Porém, é preciso tomar conhecimento que esse algoritmo muitas vezes pode oscilar ao redor do mínimo sem nunca convergir para valores ótimos. Isso pode ser solucionado alterando o passo de aprendizagem ao longo do algoritmo.

2. Suponha que as features (i.e., atributos) do seu conjunto de treinamento tenham escalas muito diferentes. Qual técnica de regressão linear pode sofrer com isso e como? O que pode ser feito para mitigar este problema?

A técnica de batelada, ou batch, pode sofrer com isso pois o mesmo modelo é apresentado a todos os atributos levando a um tempo de convergência muito lento. Utilizando a versão online, ou estocástica, apenas uma instancia aleatória no conjunto de treinamento e calcula o gradiente considerando apenas essa única instancia, levando o algoritmo a uma velocidade mais elevada pois possui poucos dados a serem manipulados a cada iteração.

3. Suponha que você use o gradiente descendente em batelada e plote o erro de cada época. Se você perceber que o erro aumenta constantemente, o que provavelmente está acontecendo? Como você pode consertar isso?

Uma das possíveis causas de o erro estar aumentanto é que o algoritmo utilizado está errado sendo esse o gradiente ascendente no qual visa maximar o erro ao invés de minimiza-lo. Apenas corrigindo para o modelo gradiente descendente do tipo estocástico conseguiremos atingir o ponto ótimo de mínimo da função que minimizará o erro ao máximo possível.

4. Entre os algoritmos baseados no gradiente descendente (GD) que discutimos (batch,
estocástico e mini-batch), qual deles chega mais rapidamente à vizinhança da solução ótima? Qual deles realmente converge? O que você pode fazer para que os outros também convirjam?

O algoritmo GD estocástico chega mais rapidamente a vizinhança da solução ótima. Já o algoritmo GD batch é o que possui a melhor convergencia porem dependendo do passo de aprendizado esse método pode ser demorado. Se utilizarmos a técnica do mini-batch teremos os gradientes calculados em pequenos conjuntos aleatorios de instancias.

5. Em sala de aula, nós discutimos 3 tipos de algoritmos baseados no gradiente descendente (batch, estocástico e mini-batch), porém, o código do mini-batch foi o único que não foi apresentado. Portanto, neste exercício eu peço que vocês
a. Implementem o algoritmo do mini-batch
b. Testem sua implementação com y = 2*x1 + 2*x2 + w, onde x1, x2 e w são M = 1000 valores retirados de uma distribuição aleatória Gaussiana normal padrão (i.e, com média 0 e variância igual a 1) e utilizando a função hipótese h = a1*x1 + a2*x2,
c. Plotem a superfície de erro, a superfície de contorno com os parâmetros a1 e a2 para cada iteração do mini-batch, e o gráfico de iteração versus erro,
d. Encontrem o valor ótimo do passo de aprendizagem (**Dica**: utilizem os gráficos da superfície de contorno com os parâmetros a1 e a2 para cada iteração do mini-batch e o gráfico de iteração versus erro para saber se aquele passo é o ótimo),
e. Comparem os resultados do mini-batch com os resultados obtidos com o GD em batelada (batch) e GD estocástico (**Dica** : para a comparação, usem os códigos que estão nos slides da aula e plotem os gráficos da superfície de contorno com os parâmetros a1 e a2 para cada iteração e o gráfico de iteração versus o erro para GD em batelada e estocástico).
f. Baseando-se nos gráficos do item anterior, a que conclusões vocês podem chegar quanto ao treinamento dos 3 tipos de gradiente descendente?
6. Dada a seguinte função hipótese e assumindo o erro quadrático médio como função de
erro
h = a0 + a1*x + a2*x^2.
Encontre as equações de atualização dos pesos/parâmetros para esta função. Em seguida, utilizando os vetores x e y definidos abaixo, encontre os parâmetros a0, a1 e a2 através do método da regressão de forma fechada e com gradiente descendente em
batelada.
y = 3 + 1.5*x + 2.3*x^2 + w,
onde x é um vetor coluna com M = 1000 valores retirados de uma distribuição aleatória uniformemente distribuída no intervalo de -5 a 5 e w é outro vetor coluna com M valores
retirados de uma distribuição aleatória Gaussiana com média 0 e variância igual a 10.
a. Plote o gráfico do número de iterações versus o erro.
b. Baseado no gráfico acima, encontre o melhor valor para o passo de aprendizagem.
7. Neste exercício você vai utilizar o arquivo ***t raining.csv*** onde a primeira coluna são os
valores de x (feature) e a segunda de y (label). Baixe o arquivo do endereço: training.csv. Após, leia o conteúdo do arquivo, ou seja, os vetores x e y, com os seguintes comandos:

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('training.csv', header=None)
x = df[0].to_numpy()
y = df[1].to_numpy()
fig = plt.figure(figsize=(10,10))
plt.plot(x, y, 'b.')
```

Em seguida, utilize o algoritmo do ***gradiente descendente em batelada*** para encontrar
os parâmetros de cada uma das seguintes funções hipóteses.
a. h = a0 + a1*x

b. h = a0 + a1*x + a2*x^2
c. h = a0 + a1*x + a2*x^2 + a3*x^3
d. h = a0 + a1*x + a2*x^2 + a3*x^3 + a4*x^4
Para cada uma das funções hipótese acima faça o seguinte:
a. Encontre os valores ótimos dos parâmetros através do método de forma fechada, i.e., equação normal, ou também conhecida como método dos mínimos quadrados.
b. Encontre as equações de atualização dos parâmetros assumindo o erro quadrático médio como função de erro.
c. Encontre o valor ótimo do passo de aprendizagem.
d. Plote um gráfico que mostre x vs. y e x vs. h, ou seja, um gráfico comparando os dados originais com a estimativa (i.e., hipótese) da função que gerou y.
e. Plote um gráfico com do número de iterações versus o erro.
Em seguida responda às seguintes perguntas
A. Qual das funções hipótese acima aproxima melhor a função alvo (target), ou seja, qual produz o menor erro ao final do treinamento?
A função hipotese de grau 4 aproxima-se mais da função alvo, produzindo uma maior aproximação nos pontos disponíveis.

B. Dado que você encontrou os parâmetros que otimizam cada uma das funções hipótese acima (ou seja, você agora tem um modelo treinado que pode predizer o resultado para novos exemplos), use os dados contidos no arquivo predicting.csv e calcule o erro quadrático médio para cada um dos modelos (i.e., função hipótese). Qual função hipótese resulta no menor erro quadrático médio? O que você consegue concluir a respeito deste resultado?
A função hipótese de grau 2 resulta no menor erro quadrático médio. É possível se concluir que aumentando o grau da função hipótese a mesma se aproximará mais da função original porém o seu erro não será minimizado uma vez que não possível prever possíveis próximos atributos. A hipótese de grau 2 tem melhor previsao de possíveis atributos melhor minimizando o erro.

8. Neste exercício você irá aplicar escalonamento de feature aos dados de treinamento.
Dada a seguinte função objetivo
y = x1 + x2,
onde x1 é um vetor coluna com M = 1000 amostras retiradas de uma distribuição Gaussiana com média 0 e desvio padrão unitário e x2 é um vetor coluna com M amostras retiradas de uma distribuição Gaussiana com média 10 e desvio padrão igual a 10. Utilize o gradiente descendente em batelada com a seguinte função hipótese
h = a1*x1 + a2*x2,
com a1 e a2 iniciais iguais a -20 e -20, respectivamente. Para todos os casos abaixo, treine os modelos com o mesmo número máximo de iterações, por exemplo, 2000 iterações. Pede-se
a. Sem aplicar nenhum escalonamento de features aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros a1 e a2 encontrados durante as iterações (ou seja, o histórico de valores que o algoritmo encontra durante o treinamento do modelo) e o gráfico de erro versus iteração. Não se esqueça de encontrar o valor ótimo para o passo de aprendizagem.
b. Aplique a normalização min-máx aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros a1 e a2 encontrados durante as iterações e o gráfico de erro versus iteração. Não se esqueça de encontrar o valor ótimo para o passo de aprendizagem.
c. Aplique a padronização aos exemplos de treinamento, plote a superfície de erro, a superfície de contorno com os parâmetros a1 e a2 encontrados durante as iterações e o gráfico de erro versus iteração. Não se esqueça de encontrar o

valor ótimo para o passo de aprendizagem.
d. Baseado nos resultados anteriores o que você pode concluir à respeito do escalonamento de features? (**D ica** : Comente a respeito da forma da superfície de erro, do número de iterações necessárias para se alcançar o ponto ótimo, etc. Quanto mais detalhada sua análise dos resultados, melhor será sua avaliação neste exercício.)

Pode-se concluir que com a utilização do escalonamento, os contornos das superfícies de erro terão um formato mais circular, com isso ajudando a acelerar a convergência do algoritmo de gradiente descendente.