Regressão Logística Multinomial aplicada ao Reconhecimento de Dígitos Manuscritos

Pedro Arthur Santos Gama

[1]Instituto de Computação – Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro – Brasil

pedroasg@dcc.ufrj.br

Resumo. *O propósito principal do projeto é implementar um algoritmo de classificação de digitos escritos a mão utilizando bibliotecas de ciência de dados, explorando a teoria de Regressão Logística Multinomial*

1. Introdução

1.1. Motivação

Ao longo do curso exploramos e implementamos modelos baseados em Regressão Linear e Regressão Logística Binomial. Sobre esse último, a fim de explorar um pouco mais sobre o entendimento dos métodos de Classificação, o projeto visa implementar um algoritmo de reconhecimento baseado na Regressão Logística Multinomial, que possui como base a implementação de 3 ou mais classes para a classficação das entradas.

1.2. Especificação do Projeto

O projeto foi pensado visando ser dividido em duas partes: Uma implementação simples e teórica da Regressão Logística Binomial e a implementação da Regressão Logística Multinomial propriamente. Em ambas as partes temos, ao final de cada implementação e treino, modelos que recebem de entrada vetores, e retornam uma probabilidade de que o valor previsto seja aquele encontrado pelo modelo.

1.2.1. Regressão Logística Binomial

A Regressão Logística Binomial é uma implementação que visa classificar a base de dados a ser treinada em classes definidas, se limitando a apenas 2 classes,

No início do projeto temos um caso simples e teórico para nos basearmos a fim de termos um comparativo ao final do projeto entre as implementações, facilitando a compreensão das técnicas de classificação. Em relação ao modelo, temos um conjunto de dados, chamados dados de treino, que são as nossas **Entradas** para esse momento de treino do modelo já estando classificadas a partir da ocorrência ou não de uma doença qualquer. Como possuímos apenas a variável idade, o nosso modelo pode ser escrito como um sigmóide da seguinte forma.

Modelo utilizado:

$$f(x) = \frac{1}{1 + e^{(-1 \cdot (c_0 + c_1 \cdot x))}}$$

Da mesma forma , visando , diminuir o erro associado à escolha dos coeficientes da função, devemos implementar, dado que o modelo da Regressão Logísitca é não linear, o método do Gradiente Descente, visando achar o mínimo local.

Cálculo do erro associado (função custo):

$$f(x,y) = -\frac{1}{8} \cdot (y \cdot \log(\frac{1}{1+e^{-1\cdot(c+ax)}}) + (1-y) \cdot \log(1 - \frac{1}{1+e^{-1\cdot(c+ax)}})$$

Nesse caso fica evidente a facilidade do caso base, dado o que estamos avaliando e o tipo de modelo que estamos usando, Regressão Linear Binomial, não necessitando de uma implementação de biblioteca para esse caso, especificamente.

1.2.2. Regressão Logística Multinomial

Entendido o caso base apresentado, partimos para o propósito principal do projeto: A implementação do algoritmo de classficação dos dígitos manuscritos. Para essa implementação, um tanto mais complexa, não podemos fazer uma aborgem quase direta como feito anteriomente, a partir daqui utilizaremos algumas ferramentas que facilitam a implementação dos modelos de Regressão Logística e que foram utilizados ao longo do projeto para cálculos e visualização.

1.2.3. Ferramentas

-Scikit Learn : Sendo a pricipal ferramenta utilizada no projeto, tanto para o import do dataset usado para os testes quanto para o uso dos modelos de Regressão Logística

- Outras Bibliotecas: Outras bibliotecas também foram implementadas e que contribuiram para a facilitação do projeto como: Sympy, Numpy e Math, implementadas para a manipulação matemática, utilizadas principalmente no caso base para : realizar a Diferenciação do modelo, transformação dos dados em vetores, uso do logaritmo, exponenciação, entre outros. E para visualização gráfica o uso do MatplotLib, para o plot da função modelada, a sigmóide.

2. Dataset

Para o Dataset, foi utilizado o **Mnist**, um dataset público do próprio Scikit Learn voltado para o uso em algoritmos de classificação. No dataset temos uma base já organizada, onde temos labels(rótulos) e os dados(as imagens) associados. Para cada imagem o label correspondente se refere a um dígito que a imagem desenhada representa

3. Metodologia

O Dataset vem organizado e implementado de uma forma que facilita a utilização do mesmo. Voltemos ao exemplo do caso base inical. No exemplo possuíamos apenas uma condição analisada para a classificação probabilística de se o indivíduo possuía a doença ou não, no caso agora temos uma série de classes a serem verificadas, o modelo não é mais de 2 classes, mas possui mais possibilidades para os dados de entrada além de possuir mais valores a serem analisados. Na implementação de um algoritmo de classificação

voltado para imagens, as entradas que forneceremos ao nosso algortimo deverá ser dada como um vetor, um array onde cada elemento representa um pixel da imagem. Daqui já tiramos a primeira grande contribuição do dataset do Scikit, nele encontramos imagens no formato 8x8, de apenas 64 pixels, isso se dá, pois, o modelo contará com 64 coeficientes, e para fazermos um bom fit, ou seja, para podermos treiná-lo bem, as imagens já vem reduzidas para que a quantidade de imagens do dataset, algo próximo de 1700 , seja o suficiente. Associado a isso, temos que o valor de cada píxel a ser treinado e testado, está na escala gray-scale. Tal característica facilita a implementação do modelo, visto que não precisamos nos preocupar ainda mais com a dimensionalidade da matriz; as 64 variáveis possuem apenas um escalar. Vale ressaltar que em uma imagem em RGB o tratamento da imagem e o método poderiam ser distintos devido a complexidade, necessitando possivelmente, por exemplo, da implementação de redes neurais, o que fugiria do escopo. Asssim, utilizamos o modelo de Regressão Linear do Scikit para criarmos o nosso modelo e o Mnist para fazermos o treino com a base de dados em formato de array equivalente ao tamanho em pixels da imagem.

4. Resultados

Para o modelo da Regressão Linear Multinomial o mesmo apresentou uma taxa de acerto de mais de 95 por cento, mostrando que o modelo teve um bom fit dos dados. Da mesma forma o projeto conta com a implementação de um caso simples, porém prático e de fácil entendimento a cerca da regressão logística Binomial, servindo como base para o entedimento do modelo proposto de classificação de imagem.

5. Conclusão

Os modelos de classificação se mostram bem mais complicados quando saímos de um caso em que temos 2 classes para mais classes, e para a sua implementação se tornar mais viável se torna necessário o uso de bibliotecas como o SciKit Learn. Da mesma forma o problema ao analisar imagens se torna tão mais complexo quanto for o tamanho da imagem e em qual escala está sendo implementada, RGB ou Gray-Scale. No caso do projeto em questão, foi trabalhado o uso de um dataset com uma quantidade de pixels baixas por padrão, contudo, para datasets com tamanho em pixels maior do que o aqui implementado, faz-se necessário o uso de alguma técnica de redução de tamanho de imagem e adequação do padrão de cores para o facilitar a implementação. O tamanho da base de treino também é de suma importância para a construção de bons coeficientes do modelo.

6. Referências

Documentação Scikit-learn

Canal Eduardo— Ciência dos Dados

Canal Hashtag Treinamentos

Portal Python Guides