Proposta de Estudo Orientado

Regressão Logística - Um singelo resumo

1 O que seria uma Regressão Logística

Regressão logística é uma técnica de IA que visa produzir saídas do tipo “sim”/ “não”, “é”/ “não é”. Nesse trabalho, consideraremos esse tipo de saída como parecer binário. Para emprego dessa técnica pressupõe-se a existência dos seguintes elementos:
-Um conjunto de treino (T), que seja uma amostra dos elementos sobre os quais deverão ser emitidos pareceres binários. Cada elemento desse conjunto deve ser representado por um vetor x_i de dimensão *n*. A dimensão dos vetores que representam os elementos de teste é conhecida por “natureza”. No caso, a natureza dos elementos sobre os quais se deseja emitir um parecer binário. Desse conjunto, deve-se obter uma matriz X, cujas colunas são formadas pelos vetores x_i, “empilhados”verticalmente”
-Um vetor Y, cujas componentes correspondam aos resultados esperados para cada um dos elementos de X. Se, por exemplo, tivermos um conjunto X, de forma que #X seja 6, Y será um vetor de dimensão 6 em que cada componente guarda o resultado do julgamento esperado sobre o elemento de X correspondente.
-Um vetor *w* e outro *b*, de dimensões *n*, que permitirão mapear cada elemento de X em um vetor Z, da mesma dimensão de Y, dado por $Z = w^T X + b$.
Resumidamente, pode-se afirmar que todo trabalho em torno de uma Regressão Logística se resume em ajustar os vetores *w* e *b* de forma que o vetor $\hat{Y} = \sigma(Z)$, esteja cada vez mais próximo de Y. Nas seções a seguir, veremos que, na regressão linear, essa questão é abordada por meio da maximização de uma função probabilidade.

2 A Probabilidade na Regressão Linear

Tornar $\hat{Y}$ mais próximo de Y pode ser alcançado quando cada componente de *w* produz um vetor Z, em que cada componente maximiza a seguinte função de probabilidades

$$P(\hat{y}_i = y_i | x = x_i) = \hat{y}_i^{y_i}.(1 - \hat{y}_i)^{(1-y_i)}$$

Façamos os testes:
-Se $y_i = 1$, $P = \hat{y}_i$, que é máxima para $\hat{y}_i = 1$.
-Fica como exercício o caso de $y_i = 0$
Entretanto, matematicamente é mais interessante maximizar $log(P)$, que é dado por:

$$y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i)$$

Em suma, para que *w* produza $\hat{Y}$ cada vez mais próximos de Y, precisamos ajustar seus valores de forma a maximizar a função $log(P)$, ou, de forma equivalente minimizar, $L = -log(P) \iff L = -(y_i log(\hat{y}_i) + (1 - y_i) log(1 - \hat{y}_i))$. L é conhecida como função custo.

3 Gradient Descent

Para minimizar a função custo (L), a Regressão Logística se utiliza de um algoritmo conhecido por Gradient Descend. Basicamente, omitindo-se inúmeros aspectos formais, dada uma função f(x), esse algoritmo busca o x_n de mínimo, partindo de um valor inicial (x_0). A cada rodado do algoritmo x_{i+1} é obtido somando-se $-\alpha.df/dx$ a x_i, ou seja

$$x_{i+1} = -\alpha.df/dx \tag{1}$$

O parâmetro α é conhecido como passo de aprendizagem. A base desse algoritmo está no fato de a função decrescer quando x toma um sentido contrário ao da derivada. Fica como exercício para o leitor elencar algumas funções em que esse fato pode ser constatado.
No caso específico das Regressões Logísticas, faz-se necessário minimizar a Função Custo (L). As variáveis do domínio são as componentes de *w* e *b*. É possível esquematizar a contribuição das componentes de X nas diferenciais das componentes de w, por meio da definição de Z:

$$Z = \begin{bmatrix} w_1 & w_2 & . & w_n \end{bmatrix} \begin{bmatrix} x_{11} & x_{12} & . & x_{1m} \\ x_{21} & x_{21} & . & x_{2m} \\ x_{31} & x_{32} & . & x_{3m} \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ x_{n1} & x_{n2} & . & x_{nm} \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & . & b_n \end{bmatrix}$$

Existe uma correspondência linear entre as componentes de X, e as de w, sugerida pelo seguinte:

Para w_1
-w_1 está linearmente relacionado a z_1, por x_{11} como coeficiente;
-w_1 está linearmente relacionado a z_2, tendo x_{12} como coeficiente;
-...
-w_1 está linearmente relacionado a z_m, tendo x_{1m} como coeficiente;
Para w_2
-w_2 está linearmente relacionado a z_1, tendo x_{21} como coeficiente;
-w_2 está linearmente relacionado a z_2, tendo x_{22} como coeficiente;
-...
-w_2 está linearmente relacionado a z_3, tendo x_{23} como coeficiente;
-...
-...
Para w_n
-w_n está linearmente relacionado a z_1, tendo x_{n1} como coeficiente;
-w_n está linearmente relacionado a z_2, tendo x_{n2} como coeficiente;
-... -w_n está linearmente relacionado a z_m, tendo x_{nm} como coeficiente;
Como a relação é linear, a matriz abaixo nos ajuda a visualizar as “componentes”dw

$$\begin{bmatrix} dw_1 & dw_1 & dw_1 & . & . & . & dw_1 \\ dw_2 & dw_2 & dw_2 & . & . & . & dw_2 \\ dw_3 & dw_3 & dw_3 & . & . & . & dw_3 \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ . & . & . & . & . & . & . \\ dw_n & dw_n & dw_n & . & . & . & dw_n \end{bmatrix}$$

A partir dessa matriz, pode-se deduzir que:
$dw_1 = x_{11}dz_1 + x_{12}dz_2... + x_{1m}dz_m$
Entre b e z, a relação é de coeficiente 1, daí:
$db = dz_1 + dz_2... + dz_m$
Da forma como foi definida L, $dz_i = \hat{y}_i - y_i$ Agora, basta aplicarmos o algoritmo, ajuntando os w e b por meio da equação (1), utilizando um passo de aprendizagem $\frac{\alpha}{m}$