Universidade Estadual de Campinas
MS211 - Cálculo Numérico - Turma M- 2º Semestre de 2019

Bruna Dias - 213821
Luana Barros - 201705

Projeto 4 – Diagnóstico de Câncer de Mama

1. Introdução

Este projeto tem como objetivo principal construir um modelo capaz de inferir o diagnóstico de câncer de mama, utilizando como auxílio a aplicação do método dos quadrados mínimos. Para isso utilizaremos uma técnica de aprendizado de máquina chamada *extreme learning machine (EML*), a qual é detalhada na seção 2. Os detalhes da ELM podem ser consultados [1, 2].

2. Metodologia

Para construir um modelo capaz de inferir o diagnóstico de um paciente, iremos utilizar dados coletados de uma biópsia para diagnóstico de câncer de mama realizada anteriormente. Para isso, temos uma matriz $Xtr \, \varepsilon \, R^{dxm}$, em que d = 30 e m = 393, onde a coluna *Xtr(:,k)* contém esses dados para diagnóstico da k-ésima paciente. Temos também um vetor $Ytr \in \{-1, 1\}^m$, o qual contém o valor 1 se a k-ésima paciente foi diagnosticada com câncer maligno e -1 se foi diagnosticada com câncer benigno.
Podemos relacionar a matriz Xtr e o vetor Ytr por meio de uma função:

$$F(Xtr(:,k)) = Ytr(k), \ \forall k = 1, ..., m$$

Ou seja, com base em dados coletados para cada paciente podemos inferir qual tipo de câncer esta possui.

Note que é muito difícil encontrar a função F exata, por isso vamos determinar uma função φ que mais se aproxima de F por meio da técnica *extreme learning machine,* com a rede neural *perceptron de múltiplas camadas* e encontrar seus parâmetros α's por meio do método dos quadrados mínimos.

2.a Definição da função φ

A função φ é definida como:

$$\varphi(x) = \alpha_1 g_1(x) + \alpha_2 g_2(x) + \ldots + \alpha_n g_n(x) = \sum_{i=1}^{n} \alpha_i g_i(x) \quad (1)$$

onde α1, α2, . . . , αn são parâmetros e as funções *g1, g2, . . . , gn* são dadas por

$$g_i(x) = \tanh\left(\sum_{j=1}^{d} w_{ij} x_j + b_i\right) \quad (2)$$

,

em que $\mathbf{w_i} = \left[w_{i1}, \ldots, w_{id}\right] \in \mathbb{R}^d$ e $b_i \in \mathbb{R}$ para todo o *i = 1, . . . , n.* Logo, podemos escrever a função (1) em termos matriciais:

$$\varphi(x) = \boldsymbol{\alpha}^T \tanh(\mathbf{W}x + \mathbf{b}) \quad (3)$$

em que $\boldsymbol{\alpha} = [\alpha_1, ..., \alpha_n]^T$, onde ***W*** é uma matriz de dimensão n x d, cujas linhas correspondem aos vetores **w**i e $\boldsymbol{b} = [b_1, ..., b_n]^T$ é um vetor coluna. Devemos construir a função φ. Para isso, queremos determinar a matriz dos coeficientes ***α,*** e a matriz **G,** de dimensão n x m, tal que **G** = *tanh(**Wx** + **b**)*, cujo elemento G(i, k) corresponde à avaliação da i-ésima função g_i calculada nos dados do vetor de características da k-ésima paciente.

Numa ELM, os vetores $\mathbf{w_i} = \left[w_{i1}, \ldots, w_{id}\right] \in \mathbb{R}^d$ e o escalar bi ∈ R que definem a função gi são gerados aleatoriamente utilizando uma distribuição normal padrão.

2.b Método dos Mínimos Quadrados

Tendo definido $\varphi(Xtr(:,k)) \approx Ytr(k), \ \forall k = 1, ..., m$, e as funções g(i) $\forall i = 1, ..., n$, podemos encontrar cada ***α*** associado a g(i) pelo método dos quadrados mínimos. Encontrar $\varphi(Xtr(:,k))$ que mais se aproxima de *Ytr(k)* significa que a soma dos quadrados dos desvios $(\varphi(Xtr(:,k)) - Ytr(k))$ é mínima, ou seja,

$$J(\alpha_1, \ldots, \alpha_n) = \sum_{k=1}^{m} \left(\alpha_1 g_1(Xtr(:,k)) + \ldots + \alpha_n g_n(Xtr(:,k)) - Ytr(k)\right)^2$$

Sabemos que o mínimo de $J(\alpha_1, \ldots, \alpha_n)$ se dá quando:

$$\frac{\partial J}{\partial \alpha_j} = 0, \ \forall j = 1, \ldots, n$$

Logo,

$$\sum_{k=1}^{m} \left(\alpha_1 g_1(Xtr(:,k)) + \ldots + \alpha_n g_n(Xtr(:,k)) - Ytr(k)\right) g_j(Xtr(:,k)) = 0 \ , \forall j = 1, \ldots, n$$

Note que a expressão acima pode ser expressa pelo sistema linear

$$\mathbf{A}\boldsymbol{\alpha} = \boldsymbol{C} \quad (4)$$

Onde $\mathbf{A} = (a_{ij}) \in \mathrm{R}\ n \times n$, $\boldsymbol{\alpha} = (\alpha_j) \in R^n$ e $\mathbf{C} = (c_i) \in R^n$, com

$$a_{ij} = \sum_{k=1}^{m} \left(g_i(Xtr(:,k))\, g_j(Xtr(:,k)) \right), \quad \forall i,j = 1, \ldots, n \tag{5}$$

e

$$c_i = \sum_{k=1}^{m} \left(Ytr(k)\, g_i(Xtr(:,k)) \right), \qquad \forall i = 1, \ldots, n \tag{6}$$

Assim, basta encontrarmos as matrizes **A** e **C** conforme descrito acima e resolver o sistema linear para encontrar o vetor ***α.***

Feito isso, podemos agora reescrever a equação (3) em termos matriciais, tal que:

$$s_k = \varphi(Xtr(:,k))$$

$$s_k = \boldsymbol{\alpha}^{\mathrm{T}} G, \qquad \text{para todo } k = 1, \ldots, m \tag{7}$$

Seja $\mathbf{S_k} = \alpha^T G$ o vetor contendo informações sobre a biópsia da k-ésima paciente, o diagnóstico é efetuado da seguinte maneira:

$$\begin{cases} \text{se } L < S_k \rightarrow Yt_h = 1 \text{ (câncer maligno)} \\ \text{senão } Yt_h = -1 \text{ (câncer benigno)} \end{cases} \tag{8}$$

Onde L é uma limiar de decisão e $yt_h \in \{-1, 1\}^m$, onde *h* é um número natural que identifica o limiar. Utilizaremos a princípio limiares L = {-2, 0, 2}, portanto, teremos seus valores salvos nos vetores yt1, yt2 e yt3 respectivamente.

Portanto, conseguimos ter resultados em $Yt_h \in \{-1, 1\}^m$ para comparar com $Ytr \in \{-1, 1\}^m$ e analisar a eficácia do nosso modelo.

O desempenho do sistema pode ser medido quantitativamente, por exemplo, calculando a acurácia (AC) ou a taxa de falsos negativos (TFN) definidos respectivamente pelas equações:

$$\mathrm{AC} = \frac{\text{Número de pacientes diagnosticados corretamente pelo sistema}}{\text{Número total de pacientes}}, \tag{9}$$

$$\mathrm{TFN} = \frac{\text{Número de pacientes com câncer maligno diagnosticados como benigno pelo sistema}}{\text{Número de pacientes que possuem câncer maligno}}. \tag{10}$$

2.c Comando GNU Octave

Conjunto de Treinamento

```
1. >> load DadosTreinamento.mat;
```

Carrega a matriz Xtr e o vetor Ytr no workspace

```
2. >> n = 20;
3. >> d = 30;
4. >> m = 393;
5. >> W = randn(n,d);
6. >> b = randn(n,1);
7. >> G = tanh(W*Xtr+b);
```

Obtemos as funções g_is utilizadas na construção da função φ /matriz s

```
8. >> [A,C] = minimos_quadrados(n, G, ytr);
```

Utilizamos uma função minimos_quadrados.m para computar o método dos quadrados mínimos pela equação (5) e (6) e, assim descobrir os valores de A e C, formando a equação (4).

```
9. >> a = A/C;
```

Computamos a equação (4) utilizando a resolução do sistema linear feito diretamente pelo GNU Octave. Note que poderíamos ter utilizado o método de resolução de Gauss implementado no Projeto 1.

```
10.  >>  S = a'*G;
```

Agora, temos os valores de $\mathbf{S}(\mathrm{k}) = \varphi(Xtr(:,k))$ definidos, ou seja, temos os valores de φ calculado em cada uma das pacientes k, como dito na equação (7).

```
11.  >> yt1 = limiar(S, -2, m)
```

Utilizamos a função limiar.m para computar a comparação mostrada em (8), onde definimos os valores {-1,1} para o vetor yt1, yt2 e yt3, conforme o limiar L = {-2,0,2} respectivamente.

```
12.  >> yt2 = limiar(S, 0, m)
13.  >> yt3 = limiar(S, 2, m)
14.  >>  [qtdCorretos1, ac1, tfn1] = compara_testes(yt1,
     ytr, m)
```

Tendo o vetor yt1, podemos calcular, chamando a função compara_testes.m, o número de pacientes diagnosticados corretamente pelo sistema (qtdCorretos1), a acurácia (ac1) e a taxa de falso negativo (tfn1). O mesmo foi feito para os vetores yt2 e yt3.

```
15.  >>  [qtdCorretos2, ac2, tfn2] = compara_testes(yt2,
     ytr, m)
16.  >>  [qtdCorretos3, ac3, tfn3] = compara_testes(yt3,
     ytr, m)

17.  >> yt4 = limiar(S, -4.5, m);
18.  >> [qtdCorretos4, ac4, tfn4] = compara_testes(yt4,
     ytr, m);
```

Conjunto de Teste

Tendo definido nosso modelo, obtido pelas funções g_is e os valores de alpha,vamos utilizá-lo em dados teste para verificar se o modelo realmente está adequado.

```
19.  >> load DadosTeste.mat
```

Obtemos a matriz Xte e o vetor yte, que são os conjuntos de teste.

```
20.  >> Ge = tanh(W*Xte+b);
21.  >> Se = a'*Ge;
22.  >> yteste = limiar(Se, -4.5, 176);
23.  >> [qtdCorretosteste, ace, tfne] =
     compara_testes(yteste, yte, 176)
```

3. Questões

1) **Sintetize a aplicação φ resolvendo o problema de quadrados mínimos em (4) com respeito ao conjunto de treinamento considerando n = 20.**

Podemos sintetizar a aplicação de φ por meio do método de quadrados mínimos, como descrito na seção 2.b, com isso obtemos os respectivos valores do parâmetro α. Após isso, como já havíamos definido a matriz G, basta aplicarmos os resultados encontrados na equação (7) e com base no limiar atribuir o valor 1, caso seja diagnosticado câncer maligno e -1 caso contrário, assim obtendo o diagnóstico realizado pelo sistema. Todos os comandos de como sintetizamos foram descritos na seção 2.c nos comandos de 1 até 10.

2) **Ainda usando o conjunto de treinamento, isto é, Xtr e ytr, determine a acurácia e a taxa de falsos negativos considerando os limiares L = −2, L = 0 e L = 2.**

Pelos comandos de 11 até 16, descritos na seção 2.c, temos que os valores da acurácia e taxa de falsos negativos são:

Limiar	Acurácia	Taxa Falsos Negativos
-2	0.87786	0.020548
0	0.91094	0.047945
2	0.93639	0.12329

Tabela 1 - Valores de acurácia e taxa de falsos negativos para limiares dados

3) **Interprete o limiar e comente sobre os valores da acurácia e a taxa de falsos negativos obtidos no item anterior.**

Atribuindo os limiares dados no item anterior, percebemos que quanto menor o limiar menor a taxa de falsos negativos, porém a acurácia também diminui. Com essa observações devemos ponderar a escolha do limiar fazendo com que a taxa de resultados corretos seja alta mas com valores baixos para falsos negativos, já que esses representam erros graves e danos irreversíveis as pacientes.

É interessante notar que o limiar está diretamente relacionado com o quão conservador queremos ser, sendo que limiares mais altos representam maiores valores de acurácia porém maior espaço para erros graves como o de falso negativo, enquanto escolhas de limiares menores demonstram escolhas conservadoras, visto que a taxa de acertos é menor, porém não se comete muitos erros sobre a taxa de falsos negativos.

4) **Um falso negativo pode incorrer danos irreversíveis para a paciente uma vez que ela tem câncer maligno que não foi detectado pelo sistema. Em vista disso, determine o melhor valor para o limiar de decisão L que assegura uma taxa de falsos negativos menor que 1%. Justifique sua resposta.**

Conforme visto na questão anterior, quanto menor o limiar menor será a taxa de falso negativo, assim para termos uma taxa menor que 1% vamos pegar valores menores que -2, mas que ainda tenha uma assertividade significativa. Testando alguns valores, encontramos como melhor limiar o valor de -4.5, que nos retorna os seguintes valores para a acurácia e para a taxa de falsos negativos - obtidos pelos comandos 17 e 18, em 2.c:

Conjunto de Treinamento:

Limiar	Acurácia	Taxa Falsos Negativos
-4.5	0.75827	0.0068493

Tabela 2 - Valores de acurácia e taxa de falsos negativos para limiar -4.5 em dados de treinamento

5) **Usando o conjunto de teste, isto é, Xte e yte, calcule a acurácia e a taxa de falsos negativos com o limiar obtido no item anterior.**

Utilizando o limiar obtido no item anterior de -4.5, agora conseguimos utilizar nosso modelo para testar seu desempenho com os dados testes.

Obtemos os seguintes valores para o conjunto teste, seguindo os comandos de 19 à 23 em 2.c:

Conjunto de Teste:

Limiar	Acurácia	Taxa Falsos Negativos
-4.5	0.77273	0

Tabela 3 - Valores de acurácia e taxa de falsos negativos para limiar -4.5 em dados de teste

Perceba que a taxa de falso negativo é praticamente nula mas ainda mantemos uma acurácia acima de 70%, o que representa resultados mais conservadores porém eficientes.

6) **O desempenho no conjunto de teste é consistente com o esperado, isto é, eles são semelhantes aos valores obtidos considerando o conjunto de treinamento?**

Com base no limiar apresentado no item 4, percebemos que os resultados dos dados de testes são consistentes com os resultados dos dados de treinamento, pois mantivemos uma boa taxa de acertos no diagnósticos cometendo poucos erros graves(falsos negativos).

Note que além disso, os resultados obtidos em ambos os caso são bem próximos, mostrando que o modelo construído não é viesado e nos traz boas conclusões sobre o tema.

4. Conclusão

Com os estudos realizados por meio deste projeto, vimos como o uso da tecnologia aplicada a conceitos matemáticos podem auxiliar situações do cotidiano como um diagnóstico médico.

Em particular, vimos que a rede neural *perceptron de múltiplas camadas*, associada com o método de resolução de quadrados mínimos foi eficiente no auxílio ao diagnóstico de câncer de mama. É importante ressaltar que o modelo apresentado não substitui a avaliação de um especialista mas sim agrega conhecimentos e torna os resultados mais precisos e eficientes.

5. Anexos

D: > Documents > numerico > numerico_projeto_4 > C limiar.m

Unsaved changes (cannot determine recent change or authors)

```
1  function Yteste = limiar(S, L, m)
2    Yteste = linspace(0,0,m);
3    for i = 1:m
4      if (L < S(i))
5        Yteste(i) = 1;
6      else
7        Yteste(i) = -1;
8      end
9    end
10 end
```

Figura 1 - Função para atribuir {-1,-1} para o yth correspondente, em função de um limiar

D: > Documents > numerico > numerico_projeto_4 > C minimos_quadrados.m

Unsaved changes (cannot determine recent change or authors)

```
1  function [A,C] = minimos_quadrados(n, G, ytr)
2    A = zeros(n,n);
3    C = linspace(0,0,n);
4
5    # Encontrando os valores da matriz A
6    for i = 1:n
7      for j = 1:n
8        #printf("(%d,%d,%d)\n", i,j,k);
9        A(i,j) = dot(G(i,:), G(j,:));
10       #print("%d\n", R)
11       # printf("%d %d\n", i, j)
12     end
13     C(i) = dot(G(i,:), ytr);
14   end
15 end
```

Figura 2 - Função para calcular os valores a_{ij} e c_i utilizados no método dos mínimos quadrados

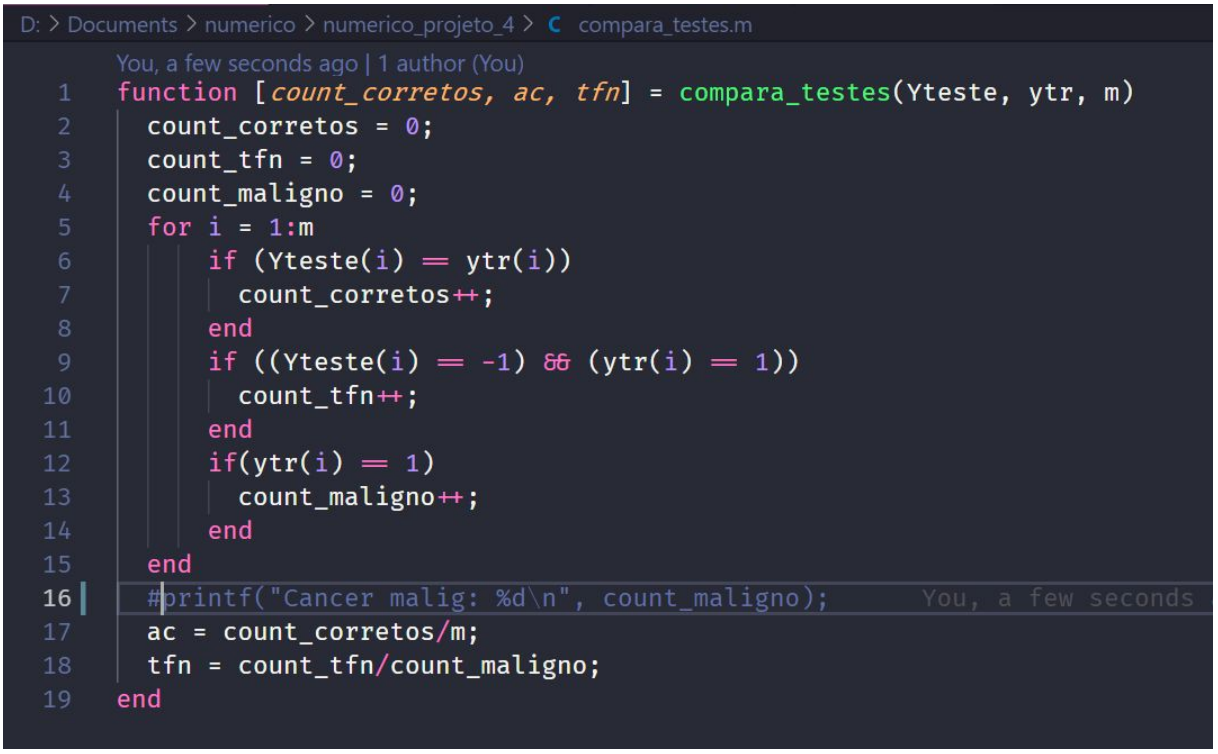

Figura 3 - Função para medir o desempenho do modelo, retornando valores de acertos, acurácia e taxa de falsos negativos

6. Referências

[1] HAYKIN, S. Neural Networks and Learning Machines, 3rd edition ed. Prentice-Hall, Upper Saddle River, NJ, 2009.

[2] HUANG, G.-B., WANG, D., AND LAN, Y. Extreme learning machines: a survey. Int. J. Machine Learning & Cybernetics 2, 2 (2011), 107–122.