APRENDIZADO DE MÁQUINA - 2 º semestre 2019

Relatório do Trabalho 1 de Machine Learning

Trabalho Tipo 1: Classificação Paramétrica (Gaussiana) Multivariada

Data: 24/10/2019

Prof.: Dr Flávio Henrique Teles Vieira
Aluno: Arthur Caetano Sabino Santos

Objetivo

Neste trabalho vamos estudar a **Classificação Paramétrica (Gaussiana) Multivariada,** implementando e testando no Octave.

Descrever os dados utilizados

Para esse trabalho usaremos os dados de classificação das diferentes espécies da flor íris encontrado no link https://archive.ics.uci.edu/ml/datasets/Iris. Nos dados, temos três espécies (classes):

- Iris Setosa
- Iris Versicolour
- Iris Virginica

Como variáveis temos:

- SepalLengthCm: Comprimento das sépalas da flor (fig1)
- SepalWidthCm: Largura das sépalas da flor (fig2)
- PetalLengthCm: Comprimento das pétalas da flor (fig3)
- PetalWidthCm: Largura das pétalas da flor (fig4)

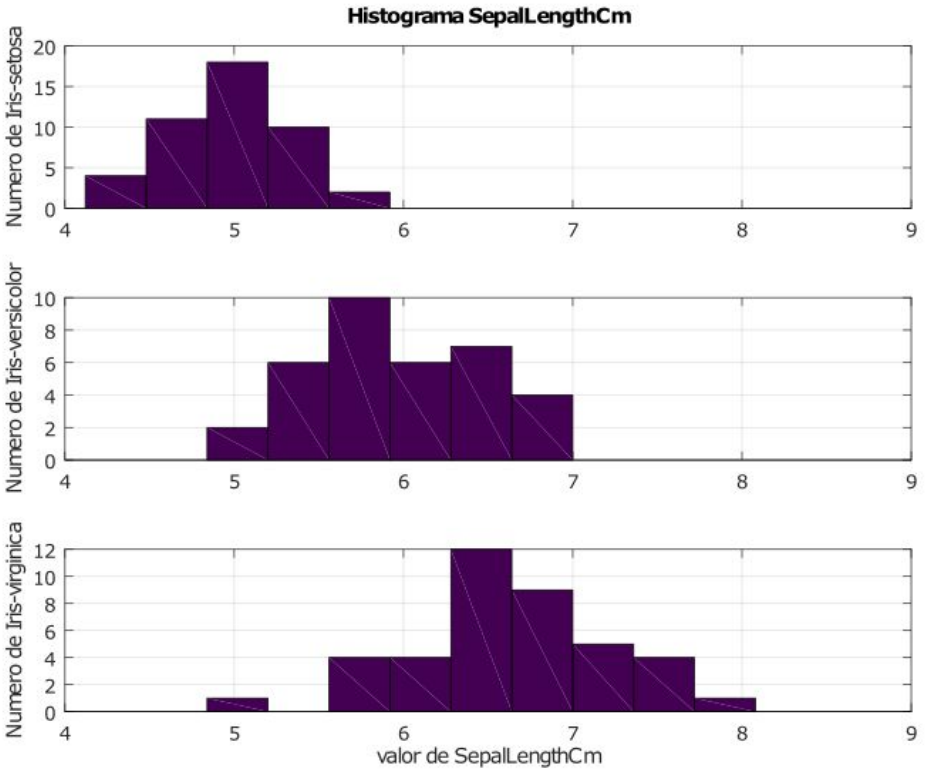

fig1: Histograma da variável SepalLengthCm para as classes, Iris Setosa, Iris Versicolour, Iris Virginica separadamente.

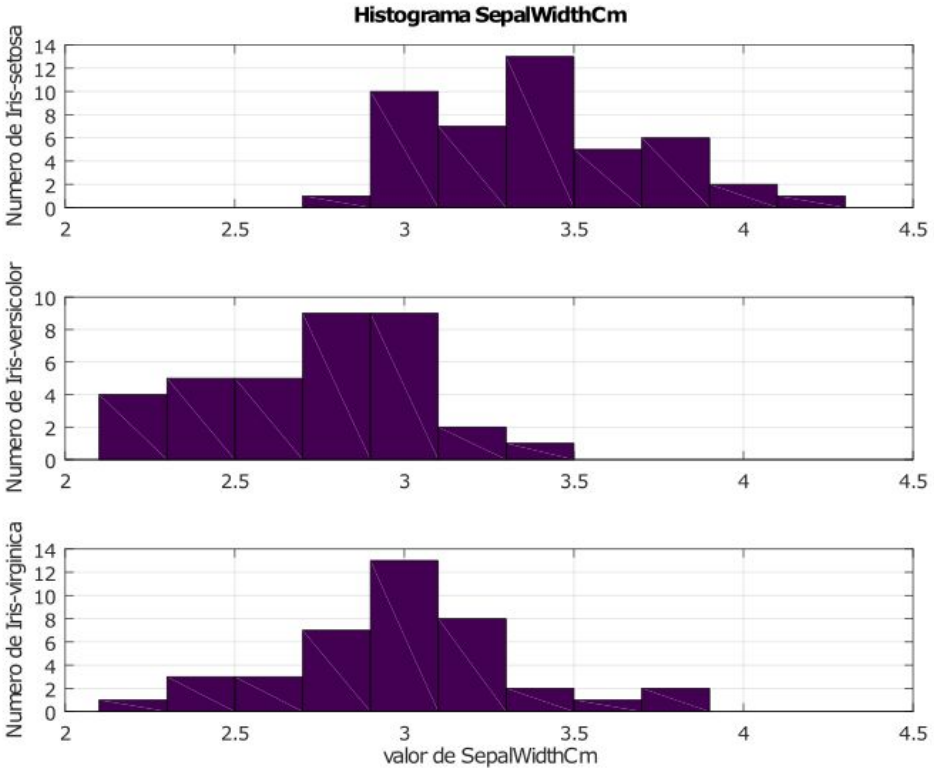

fig2: Histograma da variável SepalWidthCm para as classes, Iris Setosa, Iris Versicolour, Iris Virginica separadamente.

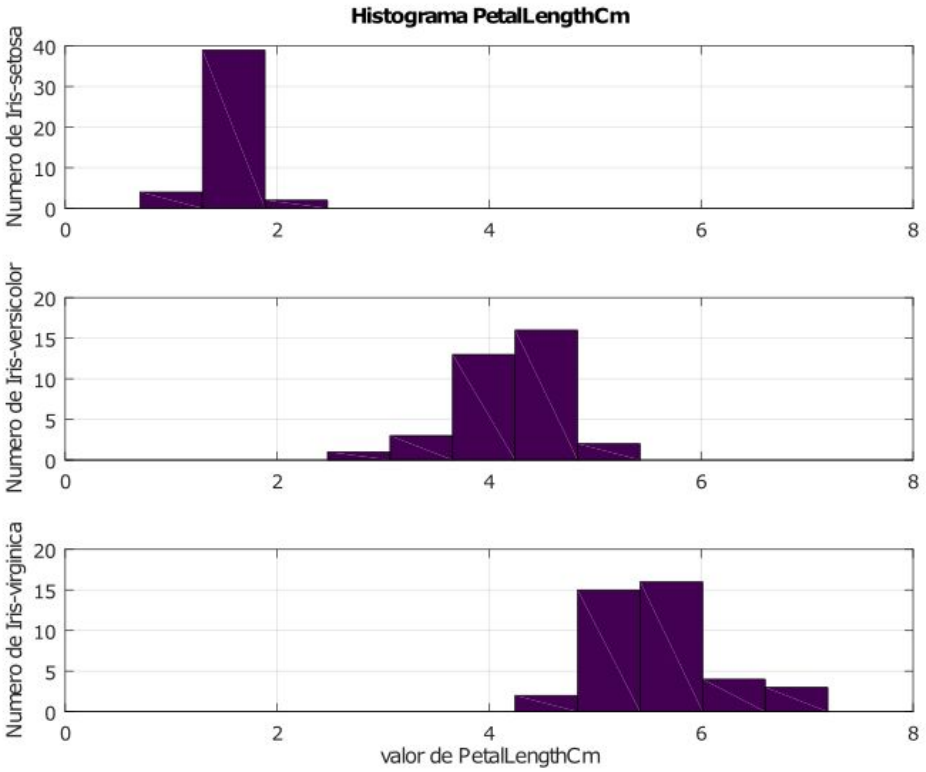

fig3: Histograma da variável PetalLengthCm para as classes, Iris Setosa, Iris Versicolour, Iris Virginica separadamente.

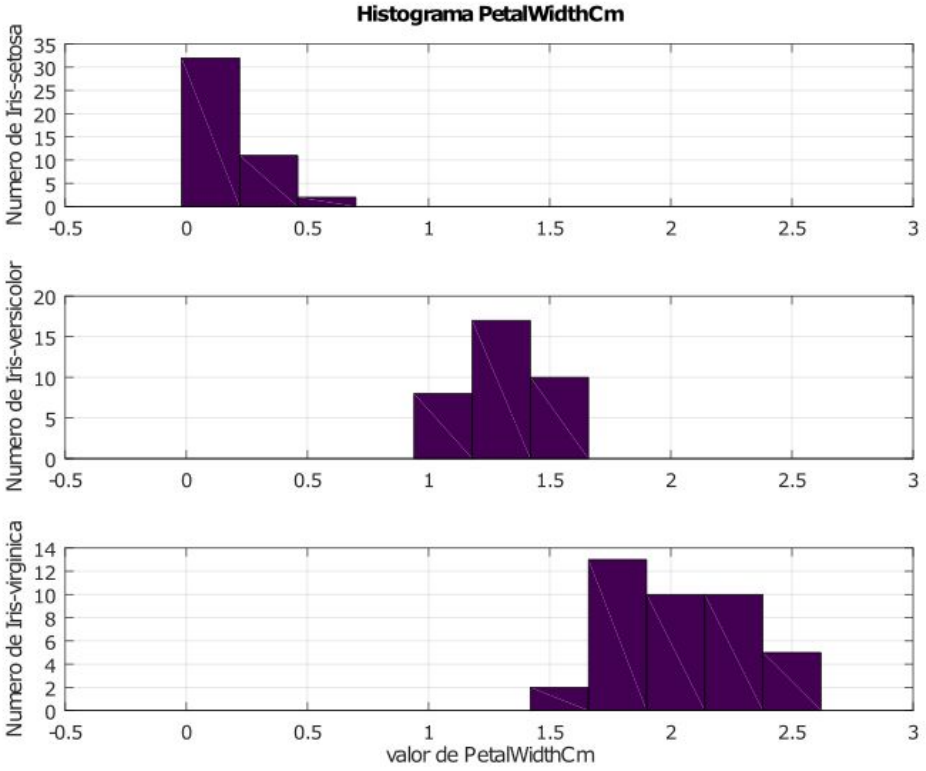

fig4: Histograma da variável PetalWidthCm para as classes, Iris Setosa, Iris Versicolour, Iris Virginica separadamente.

Separação de dado de treino e teste

Para garantir que o modelo não seja treinados com dados enviesados, eles foram previamente embaralhados aleatoriamente, e só depois separados. Os dados embaralhados foram salvos como “iris_rand.csv”

```
%% Split train and test
slice = floor(length(x)*0.8) % 120
y_train = x(1:slice,end);
x_train = x(1:slice,1:end-1)';

y_test = x(slice+1:end,end);
x_test = x(slice+1:end,1:end-1)';
```

Classificador (Gaussiana) Multivariada

Estimando os parâmetros

```
function model = estimator(x,y)
	c = unique(y);
	mu = cell(length(c),1);
	S = cell(length(c),1);
	P = cell(length(c),1);
	d = size(x)(2);

	for i=1:length(c)
		ri = y==c(i);
		m = x*ri/sum(ri);
		mu(i)= m;
		S(i) = (x(:,ri) - m)*(x(:,ri)- m)'/sum(ri);
		P(i) = sum(y==c(i))/length(y);
	endfor

	model = struct('mu',mu,'S',S,'P',P);

endfunction
```

```
function pred = predict(model, x)

	mu = struct2cell(model)(1,:);
	S = struct2cell(model)(2,:);
	P = struct2cell(model)(3,:);
```

```
        p = zeros(length(x),length(mu));

        for j=1:length(mu)
                mi = cell2mat(mu(j));
                Si = cell2mat(S(j));
                Pi = cell2mat(P(j));
                Si_inv = inv(Si);
                for i = 1:length(x)
                        xi = x(:,i);
                        p(i,j) = xi' * (-1*Si_inv/2) * xi + (Si_inv*mi)'*xi -  mi' * (-1*Si_inv/2) *
                        mi - log(norm(Si))/2 + log(Pi);
                endfor
        endfor

        [w, pred] = max(p,[],2);

endfunction
```

Resultados

Nos dados de treino:

Erro de 5 em 120: 16.6667%
Acurácia de 115 em 120: 95.8333%

Nos dados de teste:

Erro de 2 em 30: 6.66667%
Acurácia de 28 em 30: 93.3333%

Naive Bayes

Para usarmos o classificador **Naive Bayes** vamos categorizar as variáveis. Dividindo cada variável em três classes igualmente espaçadas: **BIG**, **MED**, **SMALL**. Essa nova configuração das variáveis é melhor descrita abaixo. Os dados categorizados foram salvos como “iris_rand_cat.csv”.

Prioris da base de treino

Espécie	Frequência	Prob.
Iris-virginica	40	33,3%
Iris-versicolor	35	29,2%
Iris-setosa	45	37,5%
Total	120	100,0%

Verossimilhanças da base de treino

SepalLengthCm	Iris-virginica	Iris-versicolor	Iris-setosa	P(Iris-virginica)	P(Iris-versicolor)	P(Iris-setosa)
BIG (4,3 - 5,5)	15	2	0	37,5%	5,7%	0,0%
MED (5,5 - 6,7)	24	25	2	60,0%	71,4%	4,4%
SMALL (6,7 - 7,9)	1	8	43	2,5%	22,9%	95,6%
Total	40	35	45	100,0%	100,0%	100,0%

SepalWidthCm	Iris-virginica	Iris-versicolor	Iris-setosa	P(Iris-virginica)	P(Iris-versicolor)	P(Iris-setosa)
BIG (2 - 2,8)	2	0	12	5,0%	0,0%	26,7%
MED (2,8 - 3,6)	25	15	33	62,5%	42,9%	73,3%
SMALL (3,6 - 4,4)	13	20	0	32,5%	57,1%	0,0%
Total	40	35	45	100,0%	100,0%	100,0%

PetalLengthCm	Iris-virginica	Iris-versicolor	Iris-setosa	P(Iris-virginica)	P(Iris-versicolor)	P(Iris-setosa)
BIG (1,0 - 3,0)	35	1	0	87,5%	2,9%	0,0%
MED (3,0 - 4,9)	5	34	0	12,5%	97,1%	0,0%
SMALL (4,9 - 6,9)	0	0	45	0,0%	0,0%	100,0%
Total	40	35	45	100,0%	100,0%	100,0%

PetalWidthCm	Iris-virginica	Iris-versicolor	Iris-setosa	P(Iris-virginica)	P(Iris-versicolor)	P(Iris-setosa)
BIG (0,1 - 0,9)	37	0	0	92,5%	0,0%	0,0%
MED (0,9 - 1,7)	3	35	0	7,5%	100,0%	0,0%
SMALL (1,7 - 2,5)	0	0	45	0,0%	0,0%	100,0%
Total	40	35	45	100,0%	100,0%	100,0%

Resultados

Nos dados de treino:

Erro de 2 em 120: 1.66667%
Acurácia de 118 em 120: 98.3333%

Nos dados de teste:

Erro de 4 em 30: 13.3333%
Acurácia de 26 em 30: 86.6667%

Comparação

Para essa base de dados que tem uma distribuição que pode ser aproximada para a gaussiana, a **Classificação Paramétrica (Gaussiana) Multivariada** obteve melhor desempenho que seu classificador pai **Naive Bayes**.
Isso porque o **Naive Bayes** é um classificador mais simples, e que trabalha melhor com variáveis categóricas já que trabalha com o conceito mais básico de probabilidade de eventos. Sendo assim, ele tem dificuldades em classificar eventos que não foram previamente observados.
Já a **Classificação Paramétrica (Gaussiana)** apoia-se na distribuição gaussiana para estimar a probabilidade de eventos não observados o que garante um modelo que generaliza melhor em casos de variáveis não categóricas.