

Implementação do classificador Bayesiano utilizando MATLAB

Gustavo Siebra Lopes*

25 de Abril de 2015

1 Introdução

Esse trabalho é para compor a nota na disciplina de Aprendizagem de Máquina. Consiste na implementação do código para diferentes bases usando MATLAB, um breve relatório sobre as técnicas de reconhecimento de padrão utilizada e seus resultados.

2 Preparação da base

As bases utilizadas nesse relatório estão disponibilizadas na UCI Machine Learning Repository.[1]

2.1 Base de dados da Flor de Íris

Nessa base são definidas 3 classes (Iris Setosa, Iris Versicolour, Iris Virginica) e 4 parâmetros por classe (comprimento e largura da sépala e pétala). A base possui 150 padrões diferentes de Iris dividido em 50 para cada classe.

2.1.1 Análise das características

Na figura 1 é apresentada a matriz de características. Essa matriz consiste de gráficos formados pelos pares de características combinadas.

*gustavosiebra@gmail.com

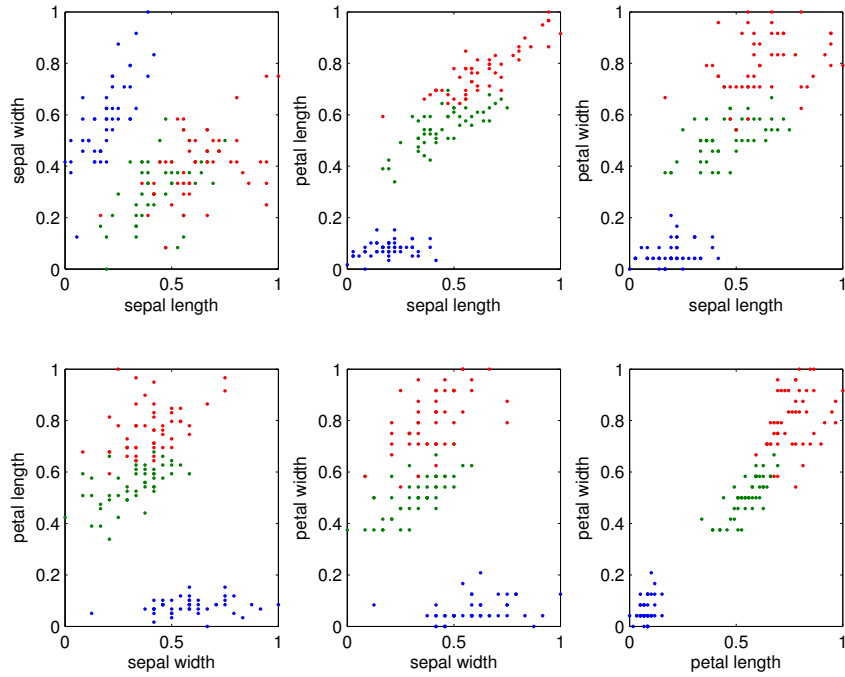


Figura 1: Matriz de características.

Podemos perceber que a classe setosa pode facilmente ser separada das outras utilizando a largura ou o comprimento da pétala como variável, já para as duas outras características essa separação não é tão simples, perceba a sobreposição dos histogramas em todos os atributos bem como a mistura das classes nos gráficos de dispersão.

2.2 Base de dados da Dermatologia

Esta base de dados contém 34 atributos, 33 dos quais são valorizados linear e um deles é nominal. O diagnóstico diferencial das doenças eritêmato-escamosas é um problema real em dermatologia. Todos eles compartilham as características clínicas de eritema e descamação, com poucas diferenças. As doenças deste grupo são a psoríase, dermatite seboréica, líquen plano, pitíriase rósea, dermatite crônica, e pitíriase rubra pilar. Geralmente é necessário para o diagnóstico de uma biópsia, mas infelizmente essas doenças compartilham muitas características histopatológicas. Uma outra dificuldade para o diagnóstico diferencial é uma doença que pode apresentar as características de uma outra doença na fase inicial e pode ter as características específicas nas fases seguintes. Os pacientes foram avaliados clinicamente primeiro com 12 recursos. Depois disso, amostras de pele foram levadas para a avaliação de 22 características histopatológicas. Os valores das características histopatológicas são determinadas por uma análise das amostras sob um microscópio.

No conjunto de dados construída para este domínio, o recurso história familiar tem o valor 1, se qualquer uma destas doenças tem sido observado na

família, e 0, caso contrário. A característica idade representa simplesmente a idade do paciente. Cada outro recurso (clínico e histopatológico) foi dado um grau na escala de 0 a 3. Aqui, 0 indica que o recurso não estava presente, 3 indica a maior quantidade possível, e 1, 2 indicam os valores intermediários relativos.

2.3 Base de dados da Coluna Vertebral

Cada paciente é representado no conjunto de dados por seis atributos biomecânicos decorrentes da forma e orientação da pelve e coluna lombar (nesta ordem): incidência pélvica, a inclinação da pelve, ângulo lordose lombar, inclinação sacral, raio pélvica e grau de espondilolistese. A seguinte convenção é utilizada para os rótulos de classe: DH (Disco de Hérnia), espondilolistese (SL), Normal (NO) e anormal (AB).

Os dados foram organizados em duas tarefas de classificação diferentes, mas relacionadas. A primeira tarefa consiste em classificar os pacientes como pertencente a uma das três categorias: Normal (100 doentes), Disco de Hérnia (60 pacientes) ou Espondilolistese (150 pacientes). Para a segunda tarefa, as categorias de disco hérnia e espondilolistese foram fundidas em uma única categoria rotulada como "anormal". Assim, a segunda tarefa consiste em classificar os pacientes como pertencendo a uma de duas categorias: Normal (100 doentes) ou anormal (210 pacientes).

3 Classificador de Bayes

Classificadores Bayesianos são classificadores estatísticos que classificam um objeto em uma determinada classe baseando-se na probabilidade deste objeto pertencer a esta classe. Produz resultados rapidamente e ótimo no tocante a minimização da probabilidade de erro de classificação.

3.1 Teorema de Bayes

Sejam padrões $x = \{x_1, x_2, \dots, x_m\} \in \mathbb{R}^d$ e classes $\{w_1, w_2, \dots, w_m\}$. Em reconhecimento de padrões ou classificação, o interesse está em associar a um padrão uma determinada classe.

A abordagem Bayesiana supõe que as probabilidades de cada classe $P(w_i)$ e as densidades de probabilidade condicionais $p(x | w_i)$ de x com respeito a cada uma das classes $w_i = \{1, 2, \dots, c\}$ são conhecidas.

O classificador Bayesiano decide pela classe com maior probabilidade $P(w_i|x)$ (probabilidade a posteriori). Esta probabilidade é calculada pela regra de Bayes.

$$P(w_i | x) = \frac{P(w_i) \times p(x | w_i)}{p(x)} \quad (1)$$

$$p(x) = \sum_{j=1}^c P(w_j) p(x | w_j) \quad (2)$$

na qual $P(w_i)$ é a priori, o $p(x)$ é a pdf (função densidade de probabilidade misturada das classes) e $p(x | w_i)$ é a função densidade de probabilidade de cada classe w_i . No entanto, o $p(x)$ é a evidência sendo o mesmo para todas as classes. Portanto, o classificador pode selecionar a classe com maior valor $g_i(x) = P(w_i)p(x | w_i)$ (função critério).

Esse classificador pode ser paramétrico e não-paramétrico, no sentido que os paramétricos buscam estimar as densidades de probabilidade da Equação 1 e o não-paramétrico estima a probabilidade a posteriori a partir das amostras em uma dada região R de \mathbb{R}^n .

$$p(x) = \frac{1}{(2\pi)^{\frac{d}{2}} \Sigma^{\frac{1}{2}}} \cdot \exp\left\{-\frac{1}{2}(x - \mu)^T \Sigma^{-1}(x - \mu)\right\} \quad (3)$$

A função densidade multivariada Gaussiana contínua é dada na Equação 3 em que \mathbf{x} é um vetor coluna com d componentes, μ é o vetor de médias também com d componentes e Σ é a matriz de covariância de tamanho $d \times d$. Logo, $|\Sigma|$ e Σ^{-1} são o determinante e a inversa da matriz de covariância, respectivamente.

Nesse relatório é apresentado 4 Funções Discriminantes dado por:

1. **Quadrática:** Matriz de covariância diferentes e probabilidades a priori diferentes.
2. **Linear:** Matriz de covariância diagonal com atributos da mesma variância e probabilidades priori diferentes.
3. **Distância Euclidiana:** Matriz de covariância diagonal com atributos da mesma variância e probabilidades a priori iguais.
4. **Mahalanobis:** Matriz de covariância diferentes e probabilidades a priori iguais.

3.2 Algoritmo

A base foi dividida usando o modelo holdout. Este método consiste em dividir o conjunto total de dados em dois subconjuntos mutuamente exclusivos, um para treinamento (estimação dos parâmetros) e outro para teste (validação). O conjunto de dados pode ser separado em quantidades iguais ou não. Uma proporção muito comum é considerar 2/3 dos dados para treinamento e o 1/3 restante para teste. [3]

Após o carregamento da base foi realizado a normalização dos dados separadamente para cada atributo. Identificando o mínimo e o máximo que foram

normalizados na faixa $[0,1]$.

O algoritmo tem como objetivo calcular a probabilidade que uma amostra desconhecida pertença a cada uma das classes possíveis, ou seja, prever a classe mais provável. Este tipo de predição é chamada de classificação estatística, pois é completamente baseada em probabilidades.

Esse algoritmo requer um conjunto de dados prévio que já esteja classificado, ou seja, um conjunto que já estejam separadas em classes (ou clusters). Baseado neste conjunto de dados prévio, que também é chamado de conjunto de treinamento, o algoritmo recebe como entrada uma nova amostra desconhecida, ou seja, que não possui classificação, e retorna como saída a classe mais provável para esta amostra de acordo com cálculos probabilísticos. O algoritmo deve seguir os seguintes passos:

Passo 01: Cálculos das probabilidades das classes.

Neste passo, cada classe do conjunto de treinamento possui sua probabilidade calculada. O cálculo é feito dividindo-se o número de instâncias de determinada classe pelo número total de instâncias do conjunto de treinamento.

Passo 02: Cálculo das probabilidades da amostra desconhecida.

Agora, o valor de cada atributo da amostra desconhecida possui sua probabilidade calculada para cada possível classe. Este passo é onde o processamento mais ‘pesado’ do algoritmo ocorre, pois, dependendo do número de atributos, classes e instâncias do conjunto de treinamento, é possível que muitos cálculos sejam necessários para se obter as probabilidades.

É importante notar que este cálculo depende inteiramente dos valores dos atributos da amostra desconhecida, ou seja, da amostra que se deseja prever a classes. Supondo que existam k classes no conjunto de testes e m atributos no conjunto de testes será necessário calcular $k \times m$ probabilidades.

Passo 03: Calcular as probabilidades da amostra desconhecida.

Neste passo, as probabilidades calculadas para os valores da amostra desconhecida de uma mesma classe são multiplicadas. Em seguida, o valor obtido é multiplicado pela probabilidade da classe calculada no Passo 01.

Com as probabilidades de cada classe calculadas, verifica-se qual é a classe que possui maior probabilidade para a amostra desconhecida. Com isso, o algoritmo termina retornando a classe que possui maior probabilidade de conter a amostra desconhecida.

4 Resultados

4.1 Acurácia

Acurácia é a proporção de acertos, ou seja, o total de verdadeiramente positivos e verdadeiramente negativos, em relação a amostra estudada. O resultado da acurácia foi obtido através de 25 iterações, realizadas para 4 tipos de funções discriminante.

4.1.1 Flor de Íris

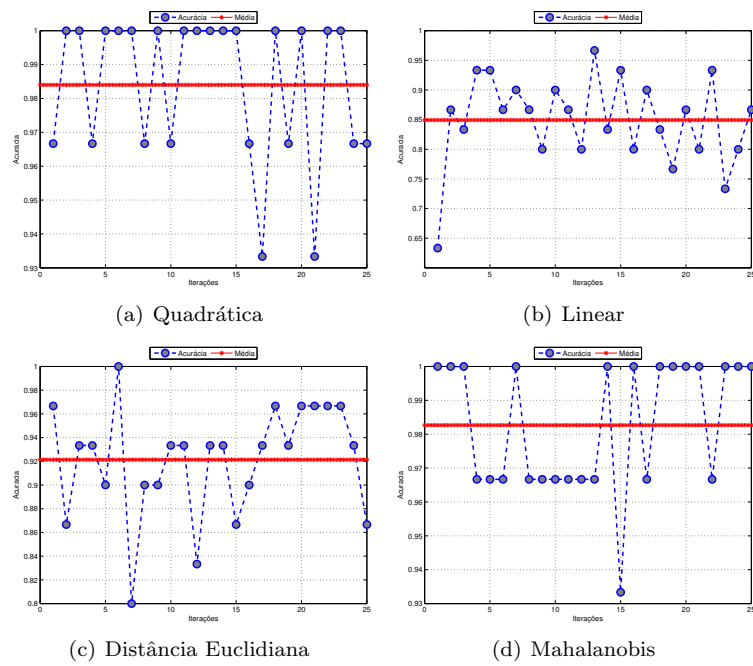


Figura 2: Resultados da Acurácia para Flor de Íris.

4.1.2 Dermatologia

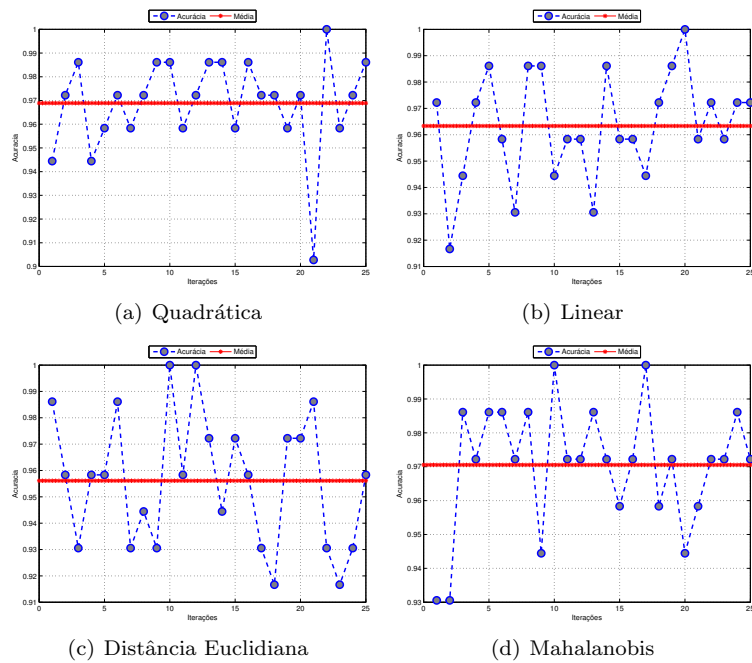


Figura 3: Resultados da Acurácia para Dermatologia.

4.1.3 Coluna Vertebral

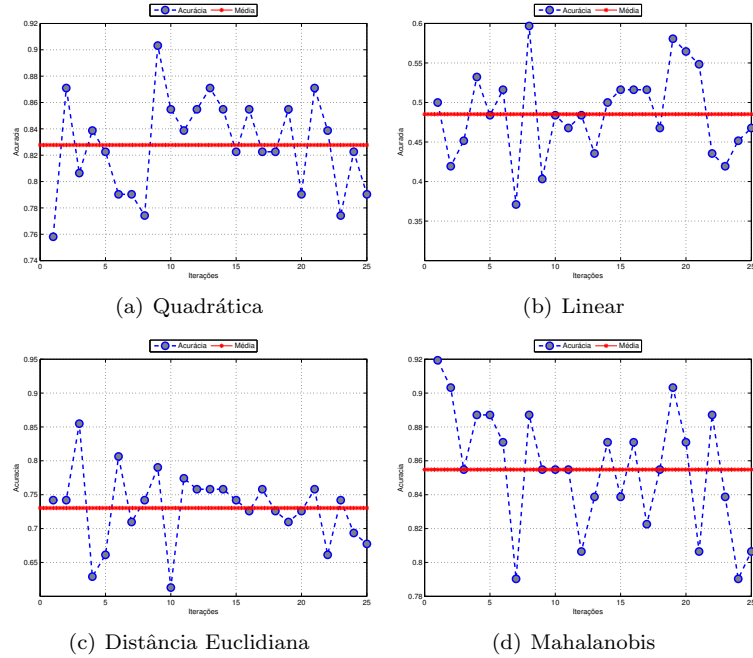


Figura 4: Resultados da Acurácia para Coluna Vertebral.

Na Tabela 1, mostra os resultados da acurácia média e desvio padrão para cada função discriminante de acordo com a base.

Tabela 1: Acurácia e Desvio Padrão para os classificadores Bayesiano.

		Flor de Íris	Dermatologia	Coluna Vertebral
Quadrática	Média	0.9840	0.9688	0.8277
	Desvio Padrão	0.0217	0.0197	0.0365
Linear	Média	0.8493	0.9633	0.4851
	Desvio Padrão	0.0739	0.0203	0.0558
Distância Euclidiana	Média	0.9253	0.9561	0.7303
	Desvio Padrão	0.0433	0.0249	0.0540
Mahalanobis	Média	0.9826	0.9705	0.8548
	Desvio Padrão	0.0195	0.0185	0.0360

4.2 Distribuição Gaussiana para Flor de Íris

Levando em consideração a questão prática e teórica, da pra dizer que a distribuição Normal conhecida também como distribuição Gaussiana, é sem dúvida, a mais relevante distribuição estatística. Essa distribuição apresenta-se em formato de sino, unimodal, simétrica em relação a sua média, considerando a probabilidade de ocorrência, a área sob sua curva soma 100%. Isso quer dizer que a probabilidade de uma observação assumir um valor entre dois pontos quaisquer é igual à área compreendida entre esses dois pontos. A seguir as imagens das Gaussianas para as 4 Funções Discriminantes.

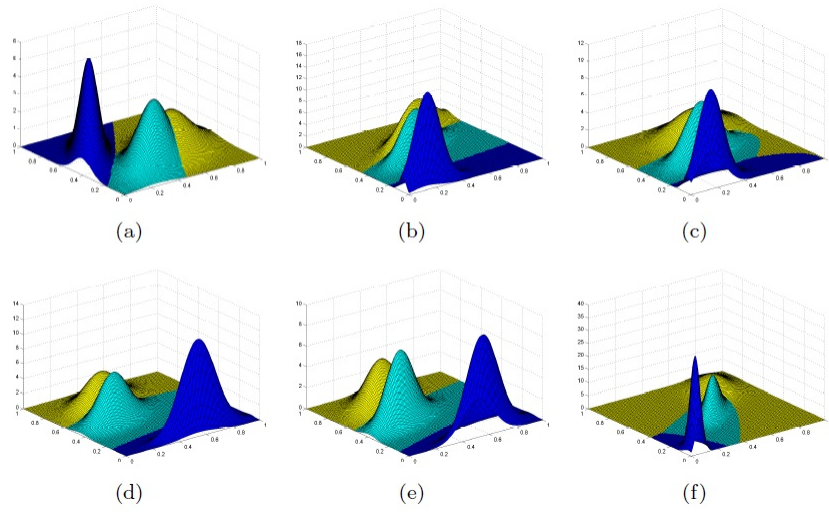


Figura 5: Gaussiana Quadrática.

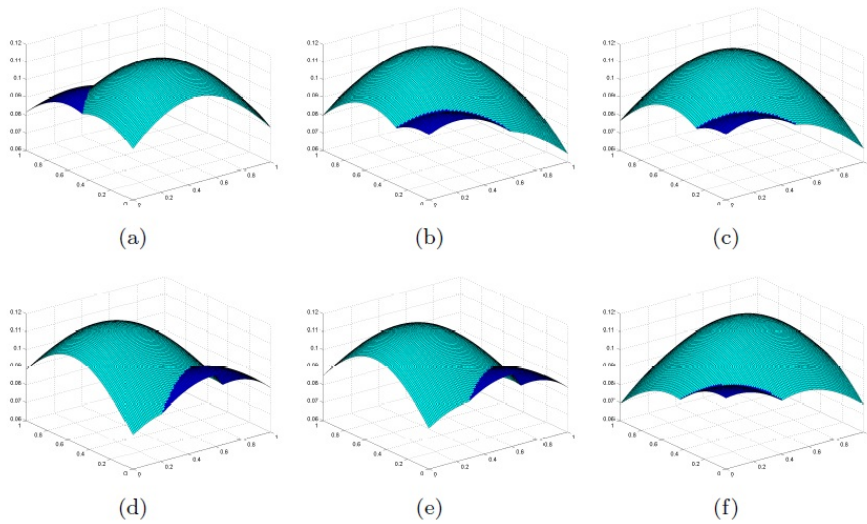


Figura 6: Gaussiana Linear.

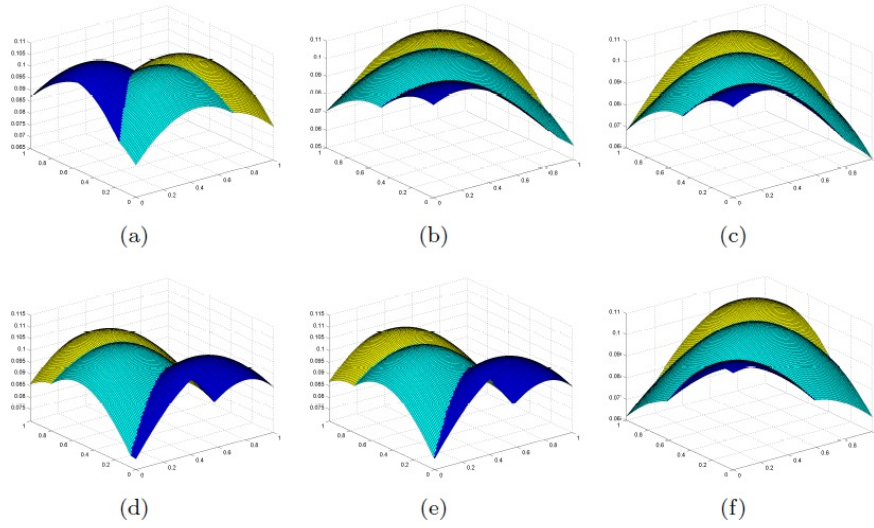


Figura 7: Gaussiana da Distância Euclidiana.

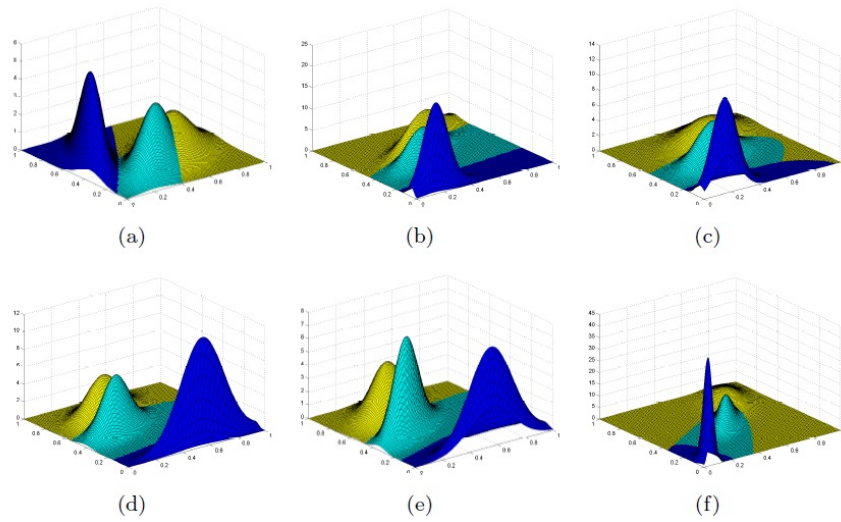


Figura 8: Gaussiana de Mahalanobis.

4.3 Região de Decisão para Flor de Íris

Em geral, um classificador particiona o espaço de características em volumes designados regiões de decisão. Todos os vectores de características no interior de uma região de decisão são atribuídos à mesma categoria.

O efeito de qualquer regra de decisão é dividir o espaço de características em c regiões de decisão R_1, R_2, \dots, R_c . Se $g_i(x) > g_j(x) \forall j \neq i$, então x está em R_i .

As regiões são separadas por superfícies de decisão, isto é, as superfícies formadas pelos pontos que pertencem a mais de uma função discriminante (interseção entre as superfícies).

Para as superfícies de decisão Quadrática, assumem a forma de elipsóides, parábolas e hipérbolas no espaço de separação, ver figura 9.

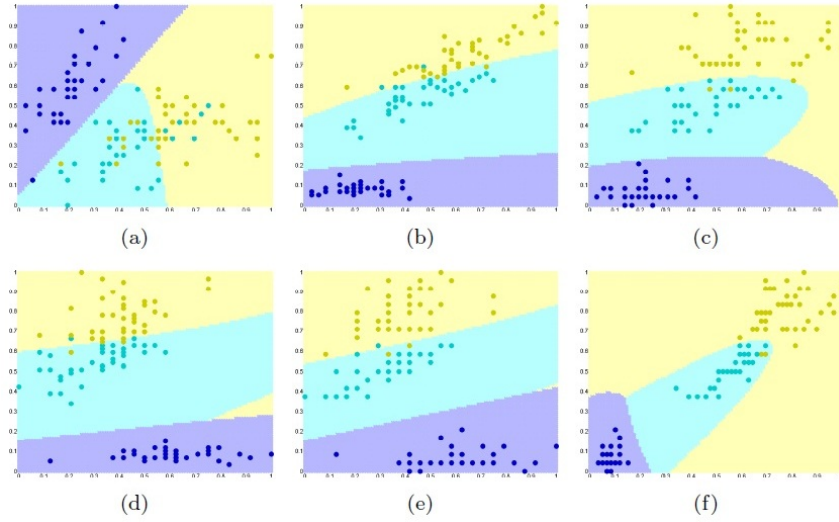


Figura 9: Região de Decisão Quadrática.

Para as superfícies de decisão Linear, assumem a forma de hiperplano no espaço de separação, ver figura 10.

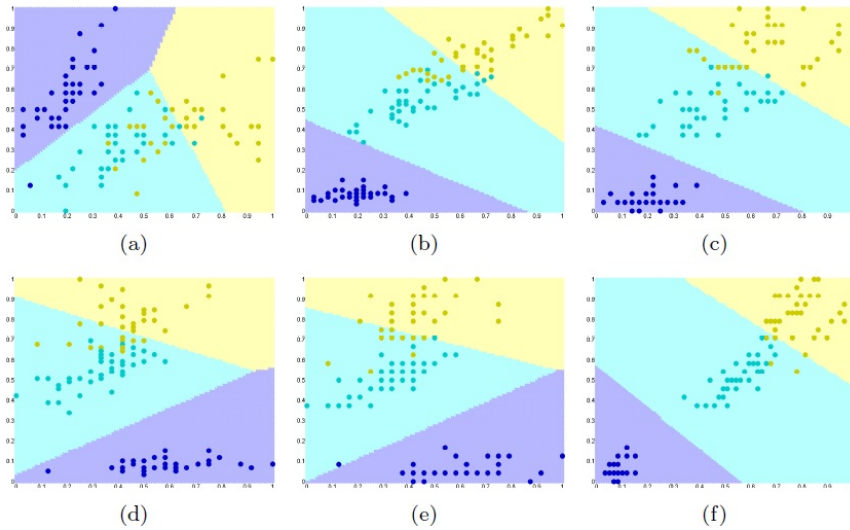


Figura 10: Região de Decisão Linear.

Para as superfícies de decisão Distância Euclidiana, assumem a forma de hiperplano, com pequenas variações em relação superfícies de separação linear, ver figura 11.

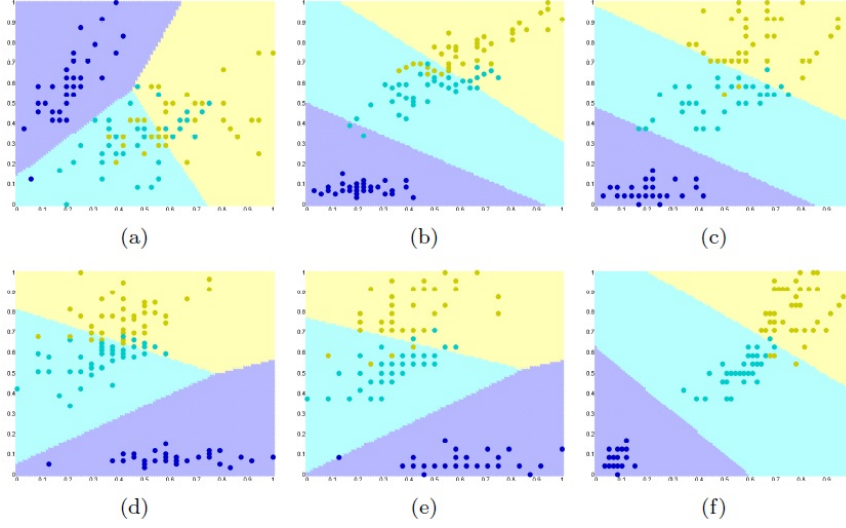


Figura 11: Região de Decisão Distância Euclidiana.

Para as superfícies de decisão de Mahalanobis, são formadas por hiper-elipsóides, com pequenas variações em relação superfícies de decisão Quadrática, ver figura 12.

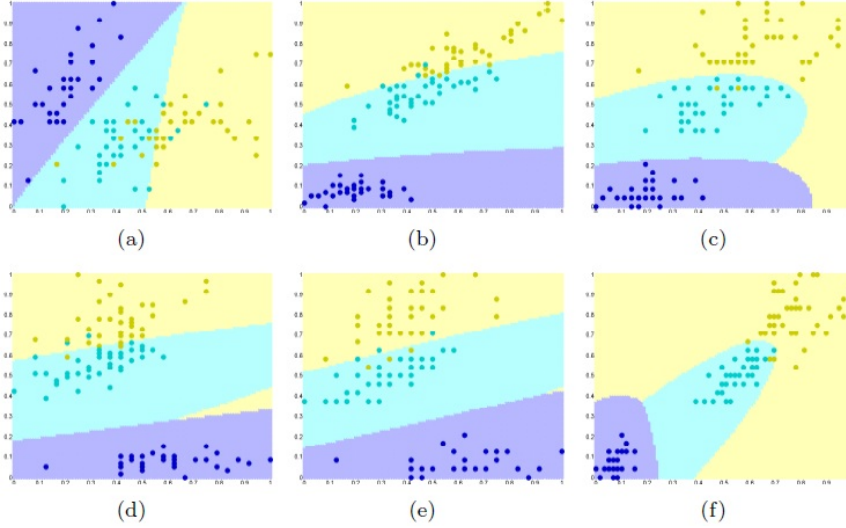


Figura 12: Região de Decisão de Mahalanobis.

5 Conclusões

As técnicas de Aprendizagem de Máquina tem sido cada vez mais usadas para resolver todos os tipos de problemas da computação. São vários os motivos pelo seu uso, tendo destaque a sua maior flexibilidade, adaptabilidade e bons resultados gerado.

Neste relatório, foi apresentado e implementado o classificador de Bayes demonstrando que o aprendizado bayesiano é uma abordagem bastante simples e robusta, com valores de acurácia bem satisfatório e minimizando as taxas de erro.

Referências

- [1] [UCI15] Uci machine learning repository, 2015. Disponível em: [http:// archive.ics.uci.edu/ml/](http://archive.ics.uci.edu/ml/).
- [2] Frutuoso, R. L. Identificação de Órgãos foliares utilizando as wavelets de daubechies. XIV Workshop de Informática Médica, FCT/U-NESP, v. 1, n. 1, p. 211–126, 2013. ISSN none. Disponível em: <http://www.lbd.dcc.ufmg.br/colecoes/wvc/2010/0037.pdf>. Citado na página 8.
- [3] Kohavi, R. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: International joint Conference on artificial intelligence. [S.l.: s.n.], 1995. v. 14, p. 1137–1145.
- [4] Scaranti, A. ; Bernardi, R. ; Plotze, R. O.. (2010) “Identificação de Órgãos Foliares Utilizando as Wavelets de Daubechies”. In: WVC’2010 - VI Workshop de Visão Computacional.
- [5] [DeGroot, 1989] DeGroot, M. H. (1989). Probability and Statistics. Addison-Wesley, 2nd edition.
- [6] [Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern Classification. John Wiley and Sons.
- [7] [Kohn, 1999] Kohn, A. F. (1999). Reconhecimento de Padrões - Uma Abordagem Estatística. EPUSP.
- [8] [Meyer, 1969] Meyer, P. L. (1969). Probabilidade - Aplicações à Estatística. Ao Livro Técnico S.A. Versão traduzida do original em inglês.
- [9] Conversando Sobre Banco De Dados, Pichiliani, M.C., Clube de Autores, 2008

A Apêndice

Código para implementação em MATLAB

Bayes

```
1 %% @autor: Gustavo Siebra
2 % IFCE - Campus Fortaleza
3 % Programa de Pos-Graduacao em Ciencias da Computacao - PPGCC
4 % Disciplina: Machine Learning
5
6 %% Variaveis de limpeza
7 clc;
8 clear all;
9 close all;
10
11 %% Carregando arquivo
12 data = load('iris.txt');
13
14 %% Iteracoes
15 R = 25;
16 % Alocando as matrizes das metricas.
17 acc = zeros(1,R);
18 confMat = zeros(1,R);
19
20 for r = 1:R
21     %% Normalizacao dos dados.
22     dataset = normalizeData(data);
23
24     %% Embaralhar base de dados.
25     dataset = randomizeData(dataset);
26
27     %% Separacao dos dados.
28     [dataTr,dataTe,xTr,dTr,xTe,dTe] = separateData(dataset);
29
30     %% Bayes Treino
31     [ndTr, prior, media, sigma] = bayesTraining(xTr, dTr);
32
33     %% Bayes Teste
34     [id, F] = bayesTest(xTe, dTe, ndTr, prior, media, sigma,4);
35
36     %% Acuracia
37     acc(r) = acuracia(id, dTe);
38 end
39
40 % Plot Caracteristicas
41 tmp = plotLabels(dataset);
42
43 % Plot da Acuracia
44 [accMedia, desvioPadrao] = plotAccuracy( r, acc );
45
46 %% Plot mvnpdf
47 [XY] = plotmvnpdf(dataset, 1);
48
49 %% Regiao de Decisao
50 [ XY] = decisionregion(dataset, 1);
```

Treino Bayes

```
1 function [ndTr, prior, media, sigma] = bayesTraining(xTr, dTr)
2     %% Parametros:
3     % Entrada:
4     %   xTr      - Atributos dos padroes de treinamento.
5     %   dTr      - Classes de treinamento.
6     %   dTe      - Classes de teste.
7     %
8     % Saida:
9     %   ndTr     - Quantidade de classes
10    %   prior    - Probabilidade da classe
11    %   media    - Media das classes
12    %   sigma    - Matriz de covariancia
13
14    %% Quantidade de classes
15    ndTr = unique(dTr);
16    nC = length(ndTr);
17
18    %% Computar a probabilidade da classe
19    for i=1:nC
20        prior(i) = sum(double(dTr == ndTr(i)))/length(dTr);
21    end
22
23    %% Calculo da Media e Desvio Padrao
24    % parameters from training set
25    for i=1:nC
26        xi = xTr((dTr == ndTr(i)),:);
27        media(i,:) = mean(xi,1);
28        sigma(:, :, i) = cov(xi,1);
29    end
30
31 end
```

Teste Bayes

```
1 function [id,F] = bayesTest(xTe, dTe, ndTr, prior, media,sigma, opc)
2
3     %% Tamanho da base de teste
4     [numLinhas, numAttr] = size(xTe);
5     nC = length(ndTr); % number of classes
6     %ns = length(dTe); % test set
7
8     %% Funcoes de verossimilhanca e probabilidades a posteriori
9     switch opc
10         % caso geral, matriz de covariancia ...
11         %diferentes e prioris diferentes
12         case 1
13             for i = 1:nC
14                 if (rcond(sigma(:,:,i)) < 1e-12)
15                     sigma(:,:,i)=sigma(:,:,i)+(0.01*eye(numAttr));
16                 end
17                 fu = mvnpdf(xTe,media(i,:),sigma(:,:,i));
18                 F(i,:) = prior(:,i) .* fu;
19             end
20         % matriz de covariancia diagonal com ...
21         %mesma variancia e prioris diferentes
22         case 2
23             for i = 1:nC
24                 fu = mvnpdf(xTe,media(i,:),...
25                     diag(diag(ones(numAttr)).*.5));
26                 F(i,:) = prior(:,i) .* fu;
27             end
28         % matriz de covariancia diagonal ...
29         % com mesma variancia e equiprovavel
30         case 3
31             prior = ones(1, nC)./nC;
32             for i = 1:nC
33                 fu = mvnpdf(xTe,media(i,:),...
34                     diag(diag(ones(numAttr)).*.5));
35                 F(i,:) = prior(:,i) .* fu;
36             end
37         case 4 % equiprovavel e matriz de covariancia diferentes
38             prior = ones(1, nC)./nC;
39             for i = 1:nC
40                 if (rcond(sigma(:,:,i)) < 1e-12)
41                     sigma(:,:,i)=sigma(:,:,i)+(0.01*eye(numAttr));
42                 end
43                 fu = mvnpdf(xTe,media(i,:),sigma(:,:,i));
44                 F(i,:) = prior(:,i) .* fu;
45             end
46         otherwise
47             ...
48     end
49
50     %% Get predicted output for test set
51     [pv0,id] = max(F);
52     for i=1:length(numLinhas)
53         pv(i,1) = ndTr(id(i));
54     end
55 end
```


Plot das Características

```
1 function tmp = plotLabels(dataset)
2     %% Parametros:
3     % Entrada:
4     %     dataset - Base de dados.
5
6     % Saida:
7     %     tmp     - variavel aleatoria
8
9     %% Classes
10    setosa = dataset((dataset(:,end)==1),:); % data for setosa
11    versicolor = dataset((dataset(:,end)==2),:); % data for versicolor
12    virginica = dataset((dataset(:,end)==3),:); % data for virginica
13
14    %% plot caracteristicas
15    Characteristics = {'sepal length','sepal width',...
16                      'petal length','petal width'};
17    %     pairs = [1 2; 1 3; 1 4; 2 3; 2 4; 3 4];
18
19    %% Separacao dos dados de treino.
20
21    xTr = dataset(:, 1: end-1);
22
23    p = combnk(1:size(xTr,2),2);
24
25    col = round(size(p,1)/2);
26
27    lin = round((col)/2);
28
29    h = figure;
30
31    for k = 1:size(p,1)
32
33        %         for j = 1:size(p,1),
34            x = p(k,1);
35            y = p(k,2);
36            subplot(lin,col,k);
37            plot([setosa(:,x) versicolor(:,x) virginica(:,x)],...
38                [setosa(:,y) versicolor(:,y) virginica(:,y)], '.');
39            xlabel(Characteristics{x},'FontSize',10);
40            ylabel(Characteristics{y},'FontSize',10);
41        %     end
42
43    end
44
45    tmp = 1;
46 end
```

Plot Região de Decisão

```
1 function [X, Y, XY, P] = deciosionregion(dataset)
2     %% Cores
3     cmap2 = jet(3);
4     cmap2 = cmap2*0.8;
5     cmap1 = (cmap2 + 0.6) * 1.2;
6     cmap1(cmap1 > 1) = 1;
7
8     %% Tamanho da base de teste
9     minx = min(dataset(:, 1));
10    maxx = max(dataset(:, 1));
11    miny = min(dataset(:, 2));
12    maxy = max(dataset(:, 2));
13
14    %% Passo
15    stepx = (maxx - minx) *.01;
16    stepy = (maxy - miny) *.01;
17
18    x1 = minx:stepx:maxx;
19    x2 = miny:stepy:maxy;
20
21    [X, Y] = meshgrid(x1, x2);
22
23    %% Concatenacao
24    XY = [X(:), Y(:)];
25
26    %% Separacao dos dados de treino.
27    xTr = dataset(:, 1: end-1);
28    dTr = dataset(:, end);
29
30    p = combnk(1:size(xTr,2),2);
31    p = sort(p);
32
33    for k = 1:size(p,1)
34        nxTr = xTr(:,p(k,:));
35        %% Bayes Treino
36        [ndTr, prior, media, sigma] = bayesTraining(nxTr, dTr);
37
38        %% Bayes Teste
39        [id] = bayesTest(xTe, dTe, ndTr, prior, media, sigma);
40
41        %% Probability for test set
42        nC = length(ndTr); % number of classes
43
44        for i = 1:nC
45            fu = mvnpdf(XY,media(i,:),sigma(:, :,i));
46            P(i,:) = prior(i) * fu;
47        end
48
49        %% Get predicted output for test set
50        [~,id] = max(P);
51
52        %% Plot Superficie de Decisao
53        figure;
54        scatter(XY(:,1), XY(:,2), 40, cmap1(id,:), 'filled');
55        hold on;
56        scatter(nxTr(:,1), nxTr(:,2), 80, cmap2(dTr',:), 'filled');
57
58    end
59 end
```