Implementação da Janela de Parzen utilizando MATLAB

Gustavo Siebra Lopes[a]

[a]*Programa de Pós-Graduação em Ciências da Computação, Instituto Federal do Ceará, Fortaleza, CE, Brazil. Email: gustavosiebra@gmail.com*

Abstract

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

Keywords: Reconhecimento de Padrões, Segmentação, Janela de Parzen, Matlab.

1. Preparação da base

As bases utilizada nesse relatório estão disponibilizadas na UCI Machine Learning Repository. [11]

1.1. Base de dados da Flor de Íris

Nessa base são definidas 3 classes (Íris Setosa, Íris Versicolor, Íris Virgínica) 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.

1.1.1. Análise das características

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

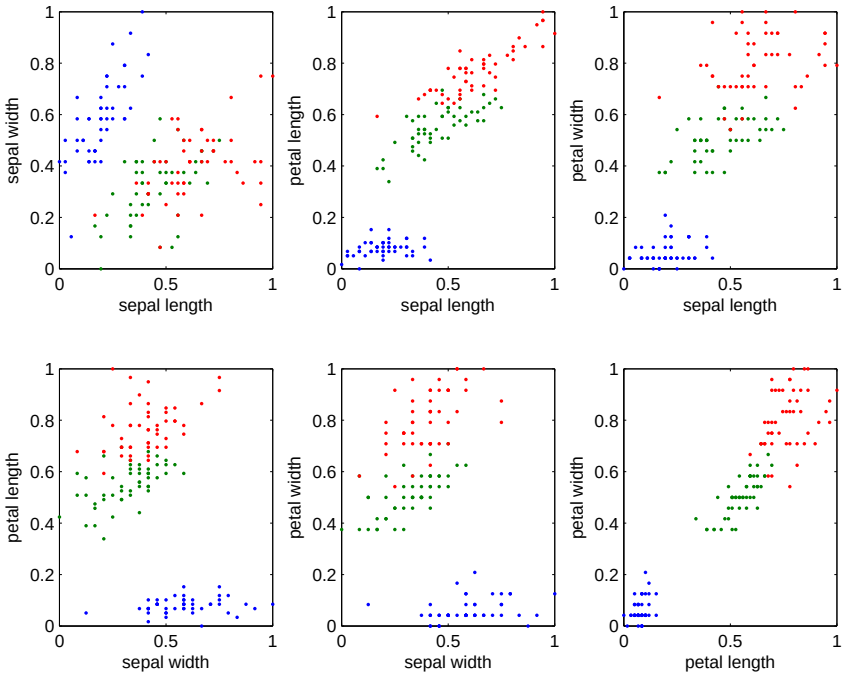


Figura 1: Matriz de características.

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.

1.2. 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 seborreica, líquen plano, pitiríase rósea, dermatite crônica, e pitiríase 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 levados 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. Janela de Parzen

Em estatística, a estimativa de densidade *Kernel* (EDK) é uma forma não-paramétrica para estimar a função de densidade de probabilidade de uma variável aleatória. Estimação da densidade *Kernel* é um problema fundamental de suavização de dados onde inferências sobre a população são feitas, com base numa amostra de dados finita.

Seja (x_1, x_2, ..., x_n) independentes e amostras identicamente distribuídos retirada de alguma distribuição com uma densidade desconhecida f. Estamos interessados em estimar a forma desta função f. O estimador de densidade de *kernel* é:

$$f_h(\mathrm{x}) = (\frac{1}{n})\sum_{i=1}^{n} K_h(\mathrm{x} - x_i) = (\frac{1}{nh})\sum_{i=1}^{n} \mathrm{K}(\frac{x - x_i}{h}), \tag{1}$$

onde **K** (•) é o núcleo - uma função não-negativa que integra a um e tem média zero - e **h** > 0 é um parâmetro de suavização chamado largura de banda. Um núcleo com índice **h** é chamado núcleo dimensionado e definido como:

$$K_h(\mathrm{x}) = (\frac{1}{h}) \times \mathrm{K}(\frac{x}{h}), \tag{2}$$

Intuitivamente se quer escolher **h** tão pequeno quanto os dados permitem, no entanto, há sempre um *trade-off* entre o bias do estimador e sua variância.

As funções de *kernel* que podem ser usadas são: uniforme, triangular, *biweight, triweight, Epanechnikov*, normais, e outros. O *kernel Epanechnikov* é o ideal em um sentido erro médio quadrado, [3] embora a perda de eficiência seja pequena para os *kernels* listadas anteriormente, [12], devido às suas propriedades matemáticas convenientes, o *kernel* normal é muitas vezes usado $\mathrm{K(x)} = \phi(\mathrm{x})$, onde ϕ é o padrão normal da função de densidade.

3. 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. [5]

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, predizer 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 passo:

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 conjunto de testes será necessário calcular k x m probabilidades.

Passo 03: Calcular a 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

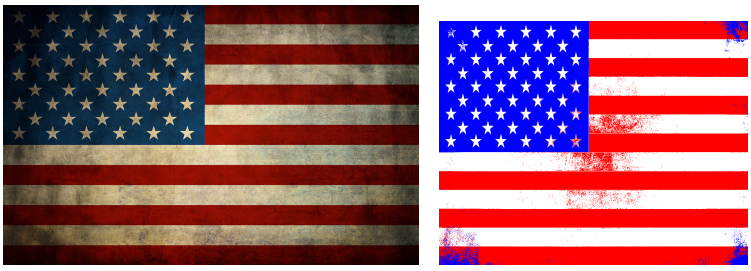
4.1. Segmentação

Em visão computacional, segmentação se refere ao processo de dividir uma imagem digital em múltiplas regiões (conjunto de *pixels*) ou objetos, com o objetivo de simplificar e/ou mudar a representação de uma imagem para facilitar a sua análise. Segmentação de imagens é tipicamente usada para localizar objetos e formas (linhas, curvas, etc) em imagens.

O resultado da segmentação de imagens é um conjunto de regiões/objetos. Com o resultado, cada um dos *pixels* em uma mesma região é similar com referência a alguma característica ou propriedade computacional, tais como cor, intensidade, textura ou continuidade. Regiões adjacentes devem possuir diferenças significativas com respeito a mesma característica(s). Esse trabalho a segmentação é feita pela extração das características das cores RGB para o tamanho de janela igual a 3.

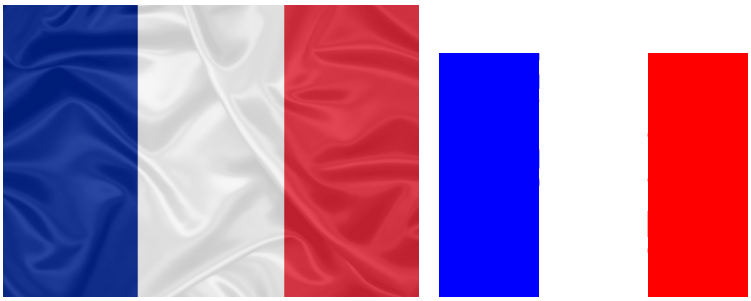
(a) Imagem Original (b) Imagem Segmentada

Figura 2: Resultado da Segmentação da bandeira do Japão.



(a) Imagem Original (b) Imagem Segmentada

Figura 3: Resultado da Segmentação da bandeira dos EUA.



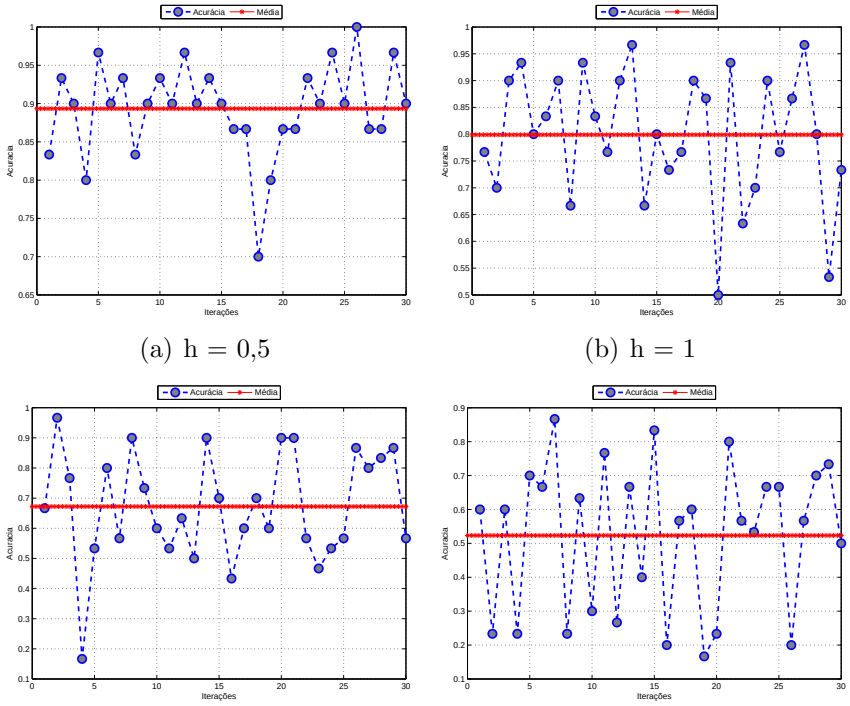
(a) Imagem Original (b) Imagem Segmentada

Figura 4: Resultado da Segmentação da bandeira da França.

4.2. Acurácia

Acurácia é a proporção de acertos, ou seja, o total de verdadeiramente positivos e verdadeiramente negativos, em relação a amostra estudada. O

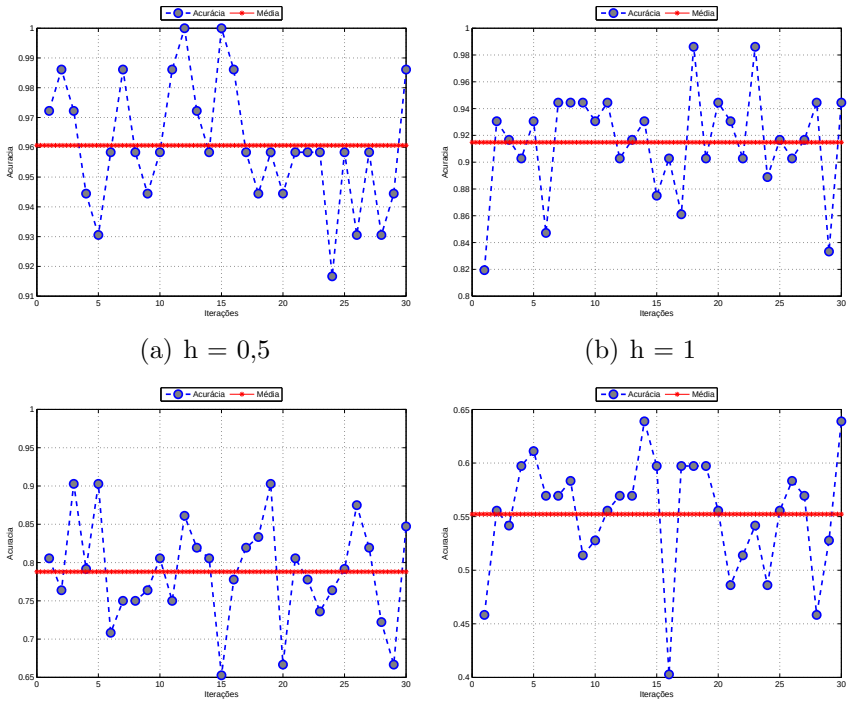
resultado da acurácia foi obtido através de 30 iterações para o tamanho da janela com valor de 0.5, 1, 1.5 e 2. Percebemos nas Figuras 5 e 6 que quanto maior for o valor da janela menor será a taxa de acurácia.



(a) h = 0,5 (b) h = 1

(c) h = 1,5 (d) h = 2

Figura 5: Resultados da Acurácia para flor da Íris.



(a) h = 0,5 (b) h = 1

(c) h = 1,5 (d) h = 2

Figura 6: Resultados da Acurácia para Derme.

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 vetores 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 R1, R2, $\cdots$, Rc . 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).

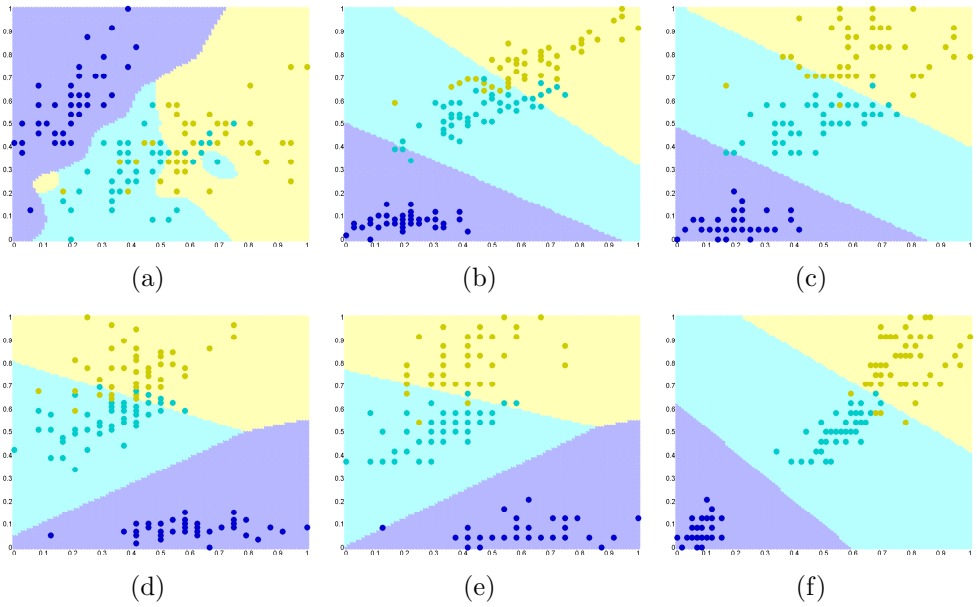


Figura 7: Região de Decisão para flor da Íris.

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 Parzen usado para segmentar imagens digitais e classificar dois tipos de bases, demonstrando que o aprendizado é uma abordagem bastante simples e robusta, com valores de acurácia bem satisfatório e minimizando as taxas de erro.

Referências

[1] [DeGroot, 1989] DeGroot, M. H. (1989). Probability and Statistics. Addison-Wesley, 2nd edition.

[2] [Duda et al., 2001] Duda, R. O., Hart, P. E., and Stork, D. G. (2001). Pattern Classification. John Wiley and Sons.

[3] Epanechnikov, V.A. (1969). "Non-parametric estimation of a multivariate probability density". Theory of Probability and its Applications 14: 153–158.

[4] Frutuoso, R. L. Identificação de Órgãos foliares utilizando as wavelets de daubechies. XIV Workshop de Informática Médica, FCT/UNESP, 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.

[5] 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.

[6] [Kohn, 1999] Kohn, A. F. (1999). Reconhecimento de Padrões - Uma Abordagem Estatística. EPUSP.

[7] [Meyer, 1969] Meyer, P. L. (1969). Probabilidade - Aplicações à Estaística. Ao Livro Técnico S.A. Versão traduzida do original em inglês.

[8] Parzen, E. (1962). "On Estimation of a Probability Density Function and Mode". The Annals of Mathematical Statistics 33 (3): 1065.

[9] Rosenblatt, M. (1956). "Remarks on Some Nonparametric Estimates of a Density Function". The Annals of Mathematical Statistics 27 (3): 832.

[10] 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.

[11] [UCI15] Uci machine learning repository, 2015. Disponível em: ¡http://archive.ics.uci.edu/ml/¿.

[12] Wand, M.P; Jones, M.C. (1995). Kernel Smoothing. London: Chapman & Hall/CRC. ISBN 0-412-55270-1.