

Classificação intervalar com aplicação em imagens coloridas

Rodrigo Augusto Dias Faria

TEXTO APRESENTADO
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO
PARA
O EXAME DE QUALIFICAÇÃO
DO
MESTRADO EM CIÊNCIAS

Programa: Ciência da Computação
Orientador: Prof. Dr. Roberto Hirata Junior

São Paulo, novembro de 2016

Resumo

FARIA, R. A. D. **Classificação intervalar com aplicação em imagens coloridas**. Dissertação (Mestrado) - Instituto de Matemática e Estatística, Universidade de São Paulo, São Paulo, 2016.

Há diversos fenômenos na natureza cuja quantificação mais adequada é representada por um intervalo, mais que por um número. Por exemplo, fenômenos cuja relação de ordem (pertinência) não é bem definida, ou quando queremos considerar a imprecisão, ou a incerteza das medidas na modelagem. Um exemplo mais concreto acontece em processamento de imagens de mais de uma cor cujo espaço não pode ser modelado naturalmente como um reticulado completo. Uma modelagem alternativa dos dados pode ser feita usando-se conjuntos *fuzzy*. A capacidade de conjuntos *fuzzy* expressarem transições graduais de pertinência e não pertinência tem uma grande utilidade. Ela fornece não só uma representação significativa e poderosa da medida de incerteza, mas também uma forma de expressar conceitos vagos em linguagem natural. Neste trabalho estudamos a classificação intervalar em imagens coloridas, em especial, na detecção de cor de pele, que tem um importante papel para uma ampla gama de aplicações de processamento de imagens, tais como, detecção e rastreamento de face, análise de gestos e outros domínios da interação humano computador.

Palavras-chave: conjuntos *fuzzy*, modelos de cores, classificação supervisionada.

Sumário

Lista de Abreviaturas	vii
Lista de Símbolos	ix
Lista de Figuras	xi
Lista de Tabelas	xiii
1 Introdução	1
1.1 Motivação	1
1.2 Trabalhos relacionados	1
1.3 Objetivos	4
1.4 Organização do texto	4
2 Fundamentação Teórica	5
2.1 Modelos de cores	5
2.1.1 Modelo de Munsell	5
2.1.2 Modelo CIE	6
2.1.3 Modelo RGB	8
2.1.4 Modelo CMY	8
2.1.5 Modelo de cores da família YUV	9
2.1.6 Modelo de cores da família HSI	10
2.2 Teoria <i>fuzzy</i>	12
2.2.1 Conjuntos <i>fuzzy</i>	13
2.2.2 Operações entre conjuntos <i>fuzzy</i>	14
2.2.3 Números <i>fuzzy</i>	17
2.2.4 Funções de pertinência	17
2.2.5 Variáveis linguísticas	18
2.2.6 Regras <i>fuzzy</i>	19
2.3 Classificadores	20
2.3.1 Máquinas de vetores suporte	22
2.3.2 k -vizinhos mais próximos	25
2.3.3 Árvores de decisão	26

3 Experimentos Preliminares	29
3.1 Conjuntos de dados	29
3.2 Primeiro experimento	30
3.3 Segundo experimento	32
3.4 Terceiro experimento	33
4 Plano de Trabalho	35
4.1 Atividades previstas	35
4.2 Cronograma proposto	36
Referências Bibliográficas	37

Lista de Abreviaturas

AR	Banco de imagens de faces de Aleix e Robert (<i>Aleix and Robert Face Database</i>)
CIE	Comissão Internacional de Iluminação (<i>Commission Internationale de l'Eclairage</i>)
CMY	Ciano, Magenta e Amarelo (<i>Cyan, Magenta and Yellow</i>)
FERET	Banco de imagens de faces Tecnologia de Reconhecimento Facial (<i>Face Recognition Technology Database</i>)
HSI	Matiz, Saturação e Intensidade (<i>Hue, Saturation, Intensity</i>)
HSL	Matiz, Saturação e Luminosidade (<i>Hue, Saturation, Lightness</i>)
HSV	Matiz, Saturação e Luminância (<i>Hue, Saturation, Value</i>)
ID3	Divisor Iterativo 3 (<i>Iterative Dichotomiser 3</i>)
k -NN	k -Vizinhos Mais Próximos (<i>k-Nearest Neighbors</i>)
NTSC	Comitê Nacional do Sistema de Televisão (<i>National Television System Committee</i>)
PAL	Linha de Fase Alternada (<i>Phase Alternation Line</i>)
RBF	Função de Base Radial (<i>Radial Basis Function</i>)
RGB	Vermelho, Verde e Azul (<i>Red, Green and Blue</i>)
SECAM	Cor Sequencial Com Memória (<i>Séquentiel Couleur à Mémoire</i>)
SFA	Banco de imagens de faces do FERET e AR (<i>Skin of FERET and AR</i>)
SVM	Máquinas de Vetores Suporte (<i>Support Vector Machines</i>)
UCS	Escala Uniforme de Cromaticidade (<i>Uniform Chromaticity Scale</i>)
UCI	Conjunto de dados de pele/não pele do repositório da Universidade da Califórnia em Irvine (<i>University of California at Irvine skin/non skin dataset</i>)
YIQ	Luminância, Matiz e Saturação (<i>Luma, Hue and Saturation</i>)
YUV	Luminância e Crominância (<i>Luma and Chrominance</i>)
WM	Método Wang-Mendel (<i>Wang-Mendel method</i>)

Lista de Símbolos

L^*	Luminância
a^*	Eixo verde/vermelho no modelo de cores $L^*a^*b^*$
b^*	Eixo azul/amarelo no modelo de cores $L^*a^*b^*$
u^*	Eixo verde/vermelho no modelo de cores $L^*u^*v^*$
v^*	Eixo azul/amarelo no modelo de cores $L^*u^*v^*$
θ	Ângulo do matiz no modelo de cores HSI
\max	Operador máximo
\min	Operador mínimo
$\mu_A(x)$	Função característica
\mathbb{R}	Conjunto dos números reais
σ	Desvio padrão da função de pertinência gaussiana
a_r	r -ésimo atributo de um vetor de características
D	Conjunto de dados
D'	Conjunto de dados transformado para o espaço de características
N	Cardinalidade do conjunto de dados
K	Número de partições do conjunto de dados na validação cruzada
α	Multiplicador de Lagrange
ξ	Variável de relaxamento
C	Constante de regularização
Φ	Função de mapeamento do conjunto de dados de entrada
γ	Parâmetro do <i>kernel</i> polinomial ou RBF em uma SVM não linear
$d(x_i, x_j)$	Função de distância entre dois vetores x_i e x_j
argmax	Conjunto de argumentos que maximizam uma função objetivo
$\delta(y, f(x_i))$	Função binária entre uma classe y e a correspondência $f(x_i)$ de uma amostra de treinamento x_i no k -NN
$H(D)$	Entropia do conjunto de dados D
y_{\oplus}	Proporção de amostras positivas
y_{\ominus}	Proporção de amostras negativas
$IG(D, a_r)$	Ganho de informação no particionamento do conjunto D pelo atributo a_r
$V(a_r)$	Conjunto de todos os possíveis valores do atributo a_r
D_v	Subconjunto de D no qual a_r tem o valor v

Lista de Figuras

2.1	Modelo de cores de Munsell.	6
2.2	Diagrama de cromaticidade CIE 1931	7
2.3	Cubo unitário representando as cores do modelo RGB	8
2.4	Modelo de cores subtrativo CMY	9
2.5	Representação gráfica do modelo HSI	10
2.6	Representação gráfica das principais propriedades dos conjuntos fuzzy	15
2.7	Operação de união entre dois conjuntos fuzzy	15
2.8	Operação de intersecção entre dois conjuntos fuzzy	16
2.9	Conjunto fuzzy e seu complemento	16
2.10	Gráfico da função de pertinência triangular	18
2.11	Gráfico da função de pertinência trapezoidal	18
2.12	Gráfico da função de pertinência gaussiana	19
2.13	Fronteira de decisão no espaço de características 2-dimensional	21
2.14	Exemplo de um problema de classes binárias separáveis num espaço 2-dimensional	23
2.15	Exemplo de aplicação do k -NN na tarefa de classificação em um espaço 2-dimensional	25
3.1	Visão 3-dimensional dos canais RGB do conjunto de dados UCI	30
3.2	Exemplos de imagens do banco de faces SFA	31
3.3	Estrutura das janelas que formam as amostras do SFA	32
3.4	Visão 3-dimensional dos canais RGB do conjunto de dados SFA	32

Lista de Tabelas

3.1	Trecho com amostras da base de dados UCI	30
3.2	Tabela de busca dos parâmetros do estimador ótimo na SVM	31
3.3	Tabela de busca dos parâmetros do estimador ótimo no k -NN	31
3.4	Resultados dos experimentos com k -NN e SVM nos conjuntos de dados UCI e SFA .	32
3.5	Resultados dos experimentos com k -NN e SVM no conjunto de dados SFA em espaços de cores distintos	33
3.6	Tabela de busca dos parâmetros do estimador ótimo no <i>FuzzyDT</i>	33
3.7	Resultados dos experimentos com árvore de decisão fuzzy no conjunto de dados SFA em espaços de cores distintos	34
4.1	Cronograma das atividades previstas	36

Capítulo 1

Introdução

1.1 Motivação

Há inúmeros indivíduos ou observações na natureza que não podem ser classificados com conjuntos clássicos pelo fato de que a relação de pertinência não é bem definida (Pedrycz e Gomide, 1998). Especificamente no campo da visão computacional, vários problemas reais de classificação só podem ser resolvidos quando da análise do contexto onde o problema está inserido.

Sendo assim, a motivação deste trabalho está em analisar problemas reais de visão computacional cujos conjuntos de dados possuem incerteza e imprecisão, modelando-os por meio de conjuntos *fuzzy*, explorando a capacidade desses conjuntos de expressarem transições graduais de pertinência e não pertinência.

1.2 Trabalhos relacionados

Em alguns problemas de classificação, o uso de dados intervalares pode ser uma ferramenta poderosa na obtenção de bons resultados. Intervalos representados por conjuntos *fuzzy* também têm grande valia nesta tarefa. Com base nessa característica, Umano *et al.* (1994) propuseram um novo algoritmo para gerar uma árvore de decisão *fuzzy*, a partir de dados numéricos, usando conjuntos *fuzzy* previamente definidos. Essa implementação é uma adaptação do algoritmo Divisor Iterativo 3 (ID3) clássico proposto por Quinlan (1986), brevemente discutido na seção 2.3.3. A diferença fundamental para o ID3 está no método de seleção do atributo que particiona o conjunto de dados. Assim como no ID3, aqui também é utilizado o ganho de informação como critério, porém, ao invés de aplicar o valor dos atributos no cálculo, o algoritmo computa pelo grau de pertinência dos conjuntos *fuzzy*. Outra propriedade particular da árvore *fuzzy* está no processo de inferência, pois permite que mais de um ramo seja ativado quando um atributo é testado na avaliação do grau de pertinência.

O *fuzzy* ID3 foi utilizado por Bhatt *et al.* (2009) para segmentação de regiões de pele. O conjunto de dados aplicado no experimento contém os três canais de cores do modelo Vermelho, Verde e Azul (RGB) e também foi utilizado nos experimentos deste trabalho, conforme pode ser visto no Capítulo 3. Antes da indução da árvore, os dados de treinamento foram aglomerados em cinco *clusters* por meio do algoritmo *fuzzy c-means*. Os *clusters* foram aproximados com função de pertinência gaussiana. A taxa de erro obtida foi de 94,1%.

Outra variante de árvore de decisão para conjuntos *fuzzy* é a proposta de Cintra *et al.* (2013), porém, baseada na versão clássica C4.5, uma versão estendida do ID3 também criada por Quinlan (1993). Chamado de *FuzzyDT*, o algoritmo igualmente usa a entropia e o ganho de informação do C4.5 como medidas para decidir sobre a importância dos atributos. A estratégia de indução da árvore também é a mesma, ou seja, os dados são particionados recursivamente criando ramos até que uma classe é atribuída a cada folha. Porém, antes da indução da árvore, os atributos contínuos são definidos em termos de conjuntos *fuzzy*. Em outras palavras, a árvore é induzida com os valores dos atributos "discretizados". O domínio dos conjuntos *fuzzy* devem ser estabelecidos

previamente pelo usuário. Experimentos foram realizados com 16 conjuntos de dados distintos obtidos de Lichman (2013), comparando o desempenho do algoritmo com o C4.5 em termos da taxa de erro, número médio de regras e condições geradas por ambas técnicas. Em resumo, o *FuzzyDT* produziu resultados satisfatórios na maioria dos conjuntos de dados avaliados. O algoritmo foi utilizado para experimentos preliminares que podem ser vistos no Capítulo 3.

Vale ressaltar que o *FuzzyDT* contém quatro métodos distintos de estimar o número de conjuntos *fuzzy* para cada atributo do conjunto de dados (Cintra *et al.*, 2011). Um deles define um número de conjuntos *fuzzy* uniforme para todos os atributos. Os demais são flexíveis, ou seja, permitem que o número de conjuntos *fuzzy* para cada atributo seja distinto. Um desses métodos é o ganho de informação, calculado e armazenado para cada atributo após o processo de "fuzzificação" dos valores contínuos. O processo é repetido 8 vezes, utilizando de 2 a 9 conjuntos *fuzzy* para cada atributo. A base de dados final é, então, formada com o número de conjuntos *fuzzy* que produziram o maior ganho de informação para cada atributo. O segundo método é oriundo do algoritmo RELIEF que mede a utilidade dos atributos com base em sua capacidade de distinguir entre amostras semelhantes pertencentes a classes distintas. Ele produz um ranking de mérito médio dos atributos e funciona tanto em dados discretos como contínuos (Kononenko, 1994). O terceiro método é a implementação da técnica proposta por Wang (2003) e conhecida como Wang-Mendel (WM), onde cada atributo é definido usando de 2 a 9 conjuntos *fuzzy* e, em seguida, usado individualmente para gerar um classificador WM. As taxas de erro obtidas do classificador são usadas como critério para selecionar o número de conjuntos *fuzzy*. Os métodos foram avaliados em 11 conjuntos de dados, sendo que os métodos flexíveis mostraram melhores taxas de erro.

Guevara *et al.* (2014) propõem uma formulação geral de *kernel* sobre conjuntos *fuzzy* e, além disso, define exemplos de construção de *kernels fuzzy* positivos definidos, tais como, polinomial e gaussiano, para resolver problemas de aprendizagem computacional, em particular de aprendizagem supervisionada, como os estudos presentes neste trabalho. Experimentos foram realizados com diferentes conjuntos de dados de baixa qualidade, ou seja, dados cujo valor dos atributos possuem um certo grau de incerteza, por possuírem amostras com valores ausentes e atributos com valores intervalares. O classificador utilizado foi Máquina de Vetores Suporte (SVM), sendo que a Função de Base Radial (RBF) é o *kernel* usado para efeito comparativo. Resultados mostraram que os *kernels* propostos, na maioria dos casos, superam o desempenho do RBF.

Como os experimentos preliminares do presente estudo baseiam-se na aplicação de classificadores em detecção de cor de pele baseado na informação de cor, uma série de trabalhos nesse campo também foram explorados. Alguns deles ocuparam-se de avaliar, comparativamente, diversos tipos de técnicas e classificadores, principalmente sob o ponto de vista de desempenho, modelos de cores, modelagem da cor da pele e diferentes conjuntos de dados (Kakumanu *et al.*, 2007; Mahmoodi e Sayedi, 2016; Vezhnevets *et al.*, 2003).

Jones e Rehg (2002) aplicaram a regra de decisão Bayesiana com um modelo de histograma 3-dimensional construído a partir de, aproximadamente, 2 bilhões de pixels coletados de 18.696 imagens da internet para realizar a detecção de pele. Jones e Rehg (2002) calcularam dois histogramas diferentes para pele e não pele no espaço de cores RGB. Tendo os histogramas com os dados de treinamento, o classificador foi derivado com a abordagem de proporção da probabilidade de que um pixel é pele para a probabilidade de que é não pele. Tal abordagem usa um limiar que pode ser ajustado para equilibrar a detecção correta e os falsos positivos. Os histogramas de tamanho 32 mostraram o melhor desempenho com uma taxa de erro de 88%.

Outro método para construir um classificador de pele é definir, explicitamente, através de um número de regras, as fronteiras que delimitam o agrupamento dos pixels de pele em algum espaço de cores (Vezhnevets *et al.*, 2003). Esta foi a abordagem adotada por Kovac *et al.* (2003) no espaço de cores YCbCr, tendo obtido uma taxa de verdadeiro positivo de 90,66%. Kovac *et al.* (2003) também realizaram experimentos apenas com os canais de cromaticidade Cb e Cr. Os resultados reduzem significativamente o desempenho do classificador. A vantagem óbvia deste método é a simplicidade das regras de detecção de pele que leva à construção de um classificador muito rápido. Por outro lado, a dificuldade em alcançar altas taxas de reconhecimento com este método é a necessidade de

encontrar um bom espaço de cores e regras de decisão adequadas empiricamente (Vezhnevets *et al.*, 2003).

Diferentemente da proposta de Kovac *et al.* (2003), Yogarajah *et al.* (2011) desenvolveram uma técnica onde os limiares definidos nas regras são adaptados dinamicamente. O método consiste em detectar a região dos olhos e extrair uma região elíptica da face. Posteriormente, um filtro de Sobel é aplicado para detectar as bordas da região resultante que, por sua vez, é submetida a uma dilatação. A imagem resultante é subtraída da imagem elíptica da face original. Como resultado tem-se uma região de pele mais uniforme onde os limiares são calculados. A técnica foi utilizada como uma etapa de pré processamento por Tan *et al.* (2012) numa estratégia que combina um histograma de densidades 2-dimensional e um modelo Gaussiano para detecção de cor de pele. Os resultados mostraram uma acurácia de 90,39%.

Naji *et al.* (2012) construíram um classificador explícito no espaço de cores HSV para 4 diferentes etnias de pele em paralelo. Após a segmentação primitiva, é aplicado um algoritmo de crescimento de região baseado em regras, no qual a saída da primeira camada é usada como uma semente e, em seguida, a máscara final em outras camadas é construída iterativamente por meio de pixels de pele vizinhos. Uma das medidas de desempenho reportadas foi o número de pixels verdadeiros positivos no valor de 96,5%.

Kawulok *et al.* (2013) combinaram informações globais e locais da imagem para construir um mapa de probabilidade que é usado para gerar a semente inicial para análise espacial dos pixels de pele. As sementes extraídas utilizando um modelo local são altamente adaptadas à imagem, o que melhora muito o resultado da análise espacial.

Embora a cor não seja utilizada diretamente em algumas abordagens de detecção de pele, é uma das ferramentas mais decisivas que afetam o desempenho dos algoritmos (Mahmoodi e Sayedi, 2016). Apesar do desempenho da maioria dos detectores de pele estar diretamente relacionado com a escolha do espaço de cores, Albiol *et al.* (2001) provaram, matematicamente, que o desempenho ótimo dos classificadores de pele é independente do espaço de cores escolhido.

O RGB é o espaço de cores mais comumente utilizado para armazenar e representar imagens digitais, uma vez que as câmeras são habilitadas a fornecer as imagens nesse modelo. Para reduzir a influência da iluminação, os canais RGB podem ser normalizados, de modo que a soma dos componentes normalizados é unitária, ou seja, tem valor 1. Por conseguinte, o terceiro componente pode ser removido, já que não fornece informação significativa, o que reduz a dimensionalidade do problema (Kakumanu *et al.*, 2007). Essa característica levou Bergasa *et al.* (2000) a construírem um modelo Gaussiano adaptativo e não supervisionado para segmentar pele no espaço de cores RGB normalizado, usando apenas os canais r e g ¹.

Jayaram *et al.* (2004) desenvolveram um estudo comparativo usando abordagens Gaussiana e histograma em um conjunto de dados de 805 imagens coloridas em 9 espaços de cores distintos. Os resultados revelaram que a ausência do componente de luminância, ou seja, utilizando apenas dois canais do espaço de cor, impacta significativamente no desempenho, bem como a seleção do espaço de cores. Os melhores resultados foram obtidos nos espaços de cores SCT, HSI e CIELab com abordagem de histograma.

Chaves-González *et al.* (2010) compararam o desempenho de 10 espaços de cor com base no algoritmo de agrupamento k -means em 15 imagens do banco de imagens de faces de Aleix e Robert (AR) (Martínez e Benavente, 1998). De acordo com os resultados obtidos, os espaços de cores mais apropriados para detecção de cor de pele são YCgCr, YDbDr e HSV, sendo este último o que produziu o melhor desempenho.

Kaur e Kranthi (2012) implementaram um algoritmo similar ao proposto por Kovac *et al.* (2003) onde as fronteiras que delimitam o agrupamento dos pixels de pele são definidos por regras explícitas. Após a segmentação da imagem com as regras, o algoritmo também realiza operações morfológicas e de filtragem para melhorar a acurácia do método. Kaur e Kranthi (2012) aplicaram o algoritmo nos espaços de cor YCbCr e CIELab, ignorando o componente de luminância em ambos, que são Y

¹ r , g e b são normalmente utilizados para representar os canais do modelo RGB normalizado (Gonzalez e Woods, 2002).

e L, respectivamente. Os resultados foram mais satisfatórios quando o algoritmo foi aplicado sobre CIELab. Uma técnica similar foi implementada em Shaik *et al.* (2015) e Kumar e Malhotra (2015) nos espaços de cores HSV e YCbCr, sendo este último o que proporcionou os melhores resultados em ambos.

1.3 Objetivos

O principal objetivo deste projeto de mestrado está na avaliação de conjuntos *fuzzy* na aplicação em problemas de classificação em visão computacional, estabelecendo mecanismos comparativos, por meio de indicadores quantitativos e qualitativos, para mensurar seu desempenho em relação aos conjuntos clássicos.

Além disso, a escolha do espaço de cores para a modelagem dos dados como conjuntos *fuzzy* também será analisada com o intuito de compreender sua influência nos resultados obtidos pelo classificador.

1.4 Organização do texto

Quanto à organização deste trabalho, no Capítulo 2 são explicitados conceitos fundamentais sobre diferentes modelos de cores, teoria *fuzzy* e classificadores utilizados no desenvolvimento deste estudo. O Capítulo 3, são expostos os resultados de experimentos preliminares com conjuntos de dados de pele de seres humanos, bem como a avaliação da influência do modelo de cores escolhido para tratar este tipo de problema de classificação. Por fim, o Capítulo 4 apresenta o plano de trabalho com as atividades a serem realizadas no âmbito desta pesquisa.

Capítulo 2

Fundamentação Teórica

2.1 Modelos de cores

O uso de imagens coloridas em visão computacional ou no processamento de imagens pode ser motivado por dois fatores principais. O primeiro diz respeito a característica poderosa da cor de funcionar como um descritor que, frequentemente, simplifica a identificação e extração de um objeto em uma cena. O segundo está relacionado com a capacidade dos seres humanos de discernir milhares de tonalidades e intensidades, se comparado com apenas algumas dúzias de níveis de cinza (Gonzalez e Woods, 2002).

A percepção visual da cor pelo olho humano não deve variar conforme a distribuição espectral da luz natural incidente sobre um objeto. Em outras palavras, a aparência de cor dos objetos permanece estável sob condições de iluminação diferentes. Esse fenômeno é conhecido como constância de cor (Gevers *et al.*, 2012).

Como exemplo, o gramado de um estádio de futebol permanece verde durante todo o dia, inclusive ao entardecer quando, de um ponto de vista físico, a luz solar tem um aspecto mais avermelhado.

A percepção humana das cores se dá pela ativação de células nervosas que enviam mensagens ao cérebro sobre brilho (*brightness*), matiz (*hue*) e saturação (*saturation*) que, geralmente, são as características usadas para distinguir uma cor de outra (Gonzalez e Woods, 2002).

O brilho dá a noção de intensidade cromática. Matiz representa a cor dominante percebida por um observador. Já a saturação refere-se à pureza relativa ou quantidade de luz branca aplicada ao matiz. Combinados, matiz e saturação são conhecidos como cromaticidade e, portanto, uma cor deve ser caracterizada por seu brilho e cromaticidade (Gonzalez e Woods, 2002).

As cores podem ser especificadas por modelos matemáticos em tuplas de números em um sistema de coordenadas e um subespaço dentro desse sistema onde cada cor é representada por um único ponto. Tais modelos são conhecidos como modelo de cores (Gonzalez e Woods, 2002).

Esses modelos podem ser classificados como de dois tipos: os modelos aditivos em que as intensidades das cores primárias são adicionadas para produzir outras cores e subtrativos, onde as cores são geradas subtraindo-se o comprimento da onda dominante da luz branca.

As seções seguintes descrevem brevemente alguns dos principais modelos de cores, bem como seus variantes e principais áreas de aplicação.

2.1.1 Modelo de Munsell

Pioneiro na tentativa de organizar a percepção de cor em um espaço de cores, Albert H. Munsell conseguiu aliar a arte e a ciência das cores em uma única teoria (Plataniotis e Venetsanopoulos, 2000).

O princípio da igualdade de espaçamento entre os componentes do modelo é a ideia principal do modelo de cores de Munsell (Plataniotis e Venetsanopoulos, 2000). Esses componentes são matiz (*hue*), luminosidade (*value*) e saturação (*chroma*).

O modelo é representado por uma forma cilíndrica e pode ser visto na figura 2.1. O matiz está disposto no eixo circular que consiste de cinco cores de base e cinco secundárias, a saturação no eixo radial e a luminosidade no eixo vertical em uma escala variando de 0 a 10.

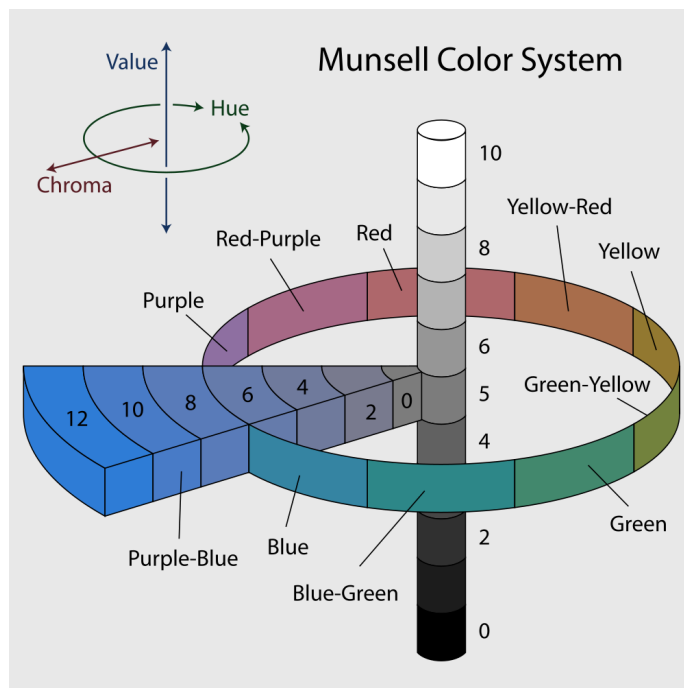


Figura 2.1: Modelo de cores de Munsell representado por uma forma cilíndrica. O matiz está disposto no eixo circular que consiste de cinco cores de base e cinco secundárias, a saturação no eixo radial e a luminosidade no eixo vertical em uma escala variando de 0 a 10. Fonte: *Rus (2007)*.

2.1.2 Modelo CIE

Em 1931, o CIE estabeleceu o primeiro modelo matemático de especificação numérica da cor, cujo objetivo era analisar a relação entre os aspectos físicos das cores no espectro eletromagnético e sua percepção pelo sistema visual humano para determinar como uma pessoa comum percebe a cor. Uma revisão desta especificação foi publicada em 1964 (*Gonzalez e Woods, 2002*).

O experimento que originou o padrão consistia em detectar as cores percebidas por um observador a partir de uma mistura de três cores primárias X, Y e Z chamadas de valores tristímulus. Estas coordenadas deram origem ao espaço de cores **CIE XYZ** que engloba todas as cores que podem ser percebidas por um ser humano comum e, por esta razão, é considerado uma representação independente de dispositivo (*Plataniotis e Venetsanopoulos, 2000*).

O sistema proposto pelo CIE XYZ para descrição de uma cor é baseado em um componente de luminância Y e outros dois componentes adicionais X e Z que dão a informação de cromaticidade. Esse sistema é formado por cores imaginárias que podem ser expressas como combinações das medidas normalizadas abaixo:

$$x = \frac{X}{X + Y + Z} \quad (2.1)$$

$$y = \frac{Y}{X + Y + Z} \quad (2.2)$$

$$z = \frac{Z}{X + Y + Z} \quad (2.3)$$

com $x + y + z = 1$.

As combinações de valores negativos e outros problemas relacionados à seleção de um conjunto de primárias reais são eliminados. As coordenadas de cromaticidade x e y permitem representar todas as cores num plano bidimensional, conhecido como diagrama de cromaticidade, que pode ser visto na figura 2.2.

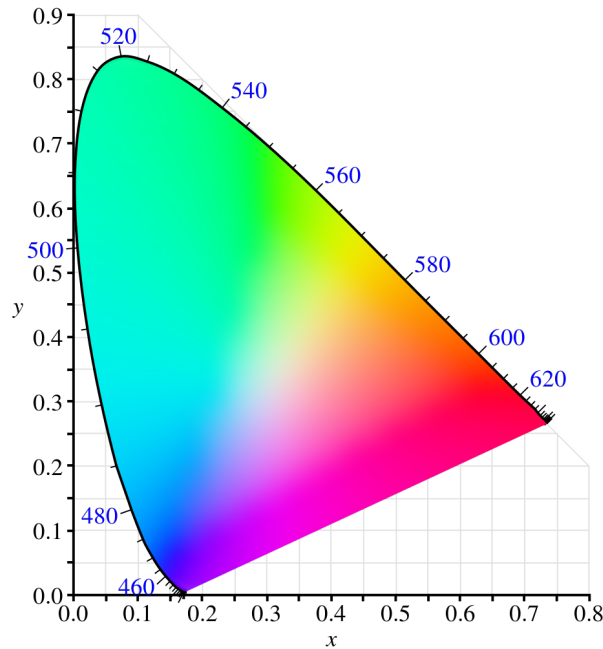


Figura 2.2: Diagrama de cromaticidade CIE 1931. Os pontos que representam as cores puras no espectro eletromagnético são rotulados de acordo com os seus comprimentos de onda e estão localizados ao longo da curva que vai da extremidade direita do eixo x , correspondente à cor vermelha, até a extremidade esquerda do mesmo eixo, correspondente à cor violeta, formando um polígono parecido com uma ferradura. Os pontos internos correspondem a todas as combinações possíveis de cores visíveis. Fonte: Ben (2009).

As coordenadas ($x = 1/3, y = 1/3$) correspondem à localização da luz branca, também conhecida como ponto branco, e servem de referência no processo de captura de imagem, codificação ou reprodução.

O CIE também derivou e padronizou outros dois modelos de cores a partir da especificação do CIE XYZ e, da mesma maneira, são independente de dispositivo. Ambos são perceptualmente uniformes, ou seja, distâncias perceptuais iguais separam todas as cores (Vezhnevets *et al.*, 2003). Como exemplo, a escala de cinzas do espaço deve permitir uma transição suave entre o preto e o branco.

O primeiro deles foi concebido para reduzir o problema de não uniformidade perceptual. Alguns diagramas de Escala Uniforme de Cromaticidade (UCS) foram propostos com base em equações matemáticas para transformar os valores XYZ ou as coordenadas x, y em um novo conjunto de valores (u, v), o que deu origem ao diagrama de cromaticidade 1960 CIE uv (Gevers *et al.*, 2012).

Ainda com resultados insatisfatórios, o CIE fez uma nova mudança multiplicando o componente v por um fator 1,5. Além disso, a escala de luminosidade dada pelo componente Y foi substituída por $L^* = [0, 100]$ para melhor representar as diferenças na luminosidade que são equivalentes. Esta revisão originou o modelo de cores **CIE 1976** $L^*u^*v^*$, comumente conhecido pela sigla CIELuv (Gevers *et al.*, 2012).

Em 1976 o CIE adotou um novo modelo de cores, baseado no modelo L, a, b , proposto por Richard Hunter em 1948, que melhor representava o espaçamento uniforme das cores. Denominado **1976 CIE** $L^*a^*b^*$ e conhecido pela sigla CIELab, é um espaço baseado em cores oponentes ¹ no

¹Teoria iniciada por volta do ano de 1500 quando Leonardo da Vinci concluiu que as cores são produzidas pela mistura de amarelo e azul, verde e vermelho, e branco e preto. Em 1950 houve a confirmação desta teoria quando sinais de cores oponentes foram detectados na conexão óptica entre a retina e o cérebro (Gevers *et al.*, 2012).

qual os estímulos de cor da retina são convertidos para distinções entre claro e escuro, vermelho e verde, e azul e amarelo, representados pelos eixos L^* , a^* , e b^* , respectivamente (Gevers *et al.*, 2012).

2.1.3 Modelo RGB

O modelo RGB, acrônimo do inglês *Red, Green, Blue*, é um modelo de cores aditivo no qual as três cores primárias vermelho, verde e azul são somadas para produzir as demais (Gonzalez e Woods, 2002).

Esse sistema foi baseado na teoria tricromática de Thomas Young e Hermann Helmholtz em meados do século 19 e pode ser representado graficamente através do cubo unitário definido sobre os eixos R, G e B, como ilustrado na figura 2.3 (Plataniotis e Venetsanopoulos, 2000).

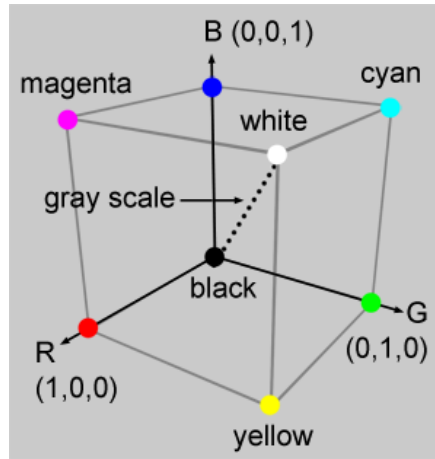


Figura 2.3: Cubo unitário representando as cores do modelo RGB. A origem, dada pelo vértice $(0,0,0)$, representa a cor preta. Já o vértice $(1,1,1)$, oposto à origem, representa a cor branca. Os vértices destacados sobre os eixos representam as cores primárias e os demais são o complemento de cada uma delas. Cada ponto no interior do cubo corresponde a uma cor que pode ser representada pela tripla (r, g, b) , onde $r, g, b \in [0, 1]$. Os tons de cinza são representados ao longo da diagonal principal do cubo, sendo que cada ponto ao longo dessa diagonal é formado por contribuições iguais de cada primária. Fonte: adaptado de Gonzalez e Woods (2002).

Vale ressaltar que existem duas formas de representar o espaço RGB: linear e não linear. O sistema supra citado mostra o modelo não linear, cuja sigla é $R'G'B'$, e é o mais utilizado por dispositivos e aplicações pela sua similaridade com o sistema visual humano. Na literatura, esse sistema é frequentemente citado com a sigla RGB, o que torna a nomenclatura dúbia, uma vez que o modelo linear também é denominado RGB e, portanto, a conversão entre os espaços de cores deve ser feita com certa cautela. Também é importante citar que os valores RGB lineares são raramente utilizados para representar uma imagem já que são, perceptualmente, altamente não uniformes (Plataniotis e Venetsanopoulos, 2000).

2.1.4 Modelo CMY

O modelo CMY é baseado nas cores primárias complementares Ciano, Magenta e Amarelo e, diferentemente do RGB, é um modelo de cores subtrativo no qual as cores são geradas subtraindo-se o comprimento da onda dominante da luz branca e, por isso, a cor resultante corresponde à luz que é refletida (Gonzalez e Woods, 2002).

Uma maneira de obter o sistema CMY é:

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} B \\ R \\ R \end{bmatrix} + \begin{bmatrix} G \\ B \\ G \end{bmatrix} \quad (2.4)$$

ou ainda, efetuando uma mudança de coordenadas subtraindo-se as cores primárias R, G e B da cor branca $W = (1, 1, 1)$ (Gonzalez e Woods, 2002):

$$\begin{bmatrix} C \\ M \\ Y \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} - \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.5)$$

Assim como o RGB, o CMY é dependente de dispositivo. O modelo é amplamente utilizado em equipamentos que depositam pigmentos coloridos sobre papel, tais como impressoras ou fotocopiadoras coloridas. A figura 2.4 mostra como os componentes do modelo são combinados para gerar as demais cores.

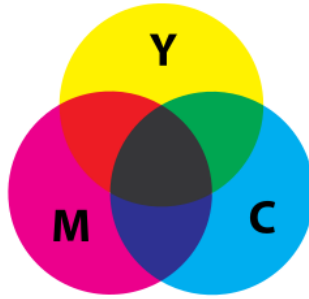


Figura 2.4: Modelo de cores subtrativo CMY. É interessante observar que a intersecção de amarelo com magenta gera a cor vermelha, magenta com ciano gera a cor azul e ciano com amarelo gera a cor verde. Fonte: Rus (2008).

A sobreposição das cores primárias CMY em iguais quantidades para gerar a cor preta tipicamente cria uma tonalidade próxima ao marrom ou verde escuro. Para evitar esse efeito indesejado, normalmente adiciona-se o componente de cor preta no sistema, representado pela letra K, obtendo-se um novo modelo conhecido como **CMYK** (Gonzalez e Woods, 2002).

2.1.5 Modelo de cores da família YUV

O termo YUV refere-se a uma família de espaços de cores dos quais a informação de luminância, representada pelo componente Y, é codificada separadamente da crominância, dada pelos componentes U e V. Os componentes U e V são representações dos sinais da diferença do azul subtraído da luminância (B-Y) e vermelho subtraído da luminância (R-Y). É utilizado para representar as cores em sistemas de transmissão analógica de televisão nos padrões Linha de Fase Alternada (PAL) e Cor Sequencial Com Memória (SECAM) (Pedrini e Schwartz, 2008).

A transformação do espaço RGB para YUV é dada por:

$$\begin{bmatrix} Y \\ U \\ V \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.147 & -0.289 & 0.436 \\ 0.615 & -0.515 & -0.100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.6)$$

onde $0 \leq R, G, B \leq 1$.

Análogo ao YUV, o modelo **YIQ** foi adotado em 1950 pelo Comitê Nacional do Sistema de Televisão (NTSC), um padrão americano para transmissão de sinal de televisão a cores. Nesse modelo, o componente Y corresponde à luminância e os componentes I (matiz) e Q (saturação) codificam a informação de crominância (Pedrini e Schwartz, 2008).

A transformação do espaço RGB para YIQ é dada por:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.275 & -0.321 \\ 0.212 & -0.523 & -0.311 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.7)$$

onde $0 \leq R, G, B \leq 1$.

Um outro modelo de cores da família YUV é o **YCbCr**, definido matematicamente por uma transformação de coordenadas em relação a algum espaço RGB (Pedrini e Schwartz, 2008).

O modelo YCbCr é largamente utilizado em vídeos digitais. Nesse modelo, o componente Y representa a luminância, o componente Cb dá a medida da diferença entre a cor azul e um valor de referência, análogo ao componente Cr que é a medida da diferença entre a cor vermelha e um valor de referência (Pedrini e Schwartz, 2008).

A conversão do espaço RGB para YCbCr é dada por:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.8)$$

2.1.6 Modelo de cores da família HSI

Modelos baseados no Matiz, Saturação e Intensidade (HSI) são mais adequados para aplicações de processamento de imagens, do ponto de vista do usuário, pela correlação com a percepção humana da cor (Plataniotis e Venetsanopoulos, 2000).

Nesse modelo, assim como no YIQ, a intensidade dada pelo componente I é decomposta da informação de cromaticidade, representada pelo matiz (H) e saturação (S) (Plataniotis e Venetsanopoulos, 2000). A combinação desses componentes resulta em uma estrutura piramidal que pode ser vista na figura 2.5.

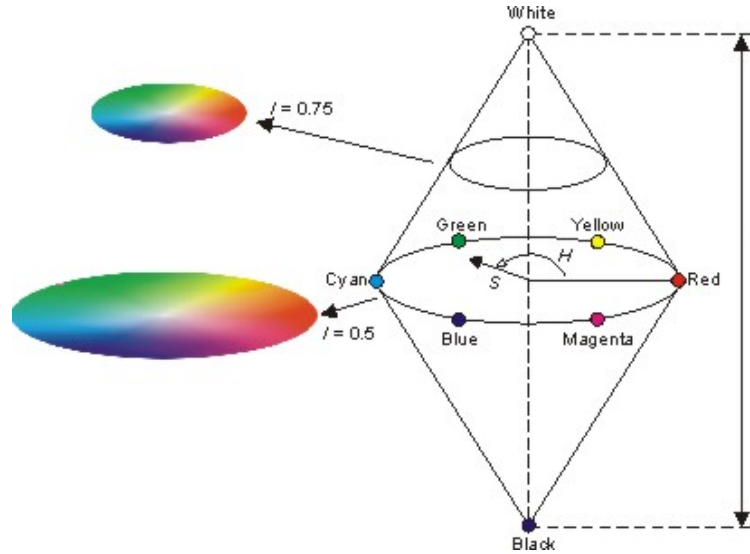


Figura 2.5: Representação gráfica do modelo HSI. O matiz descreve a cor em si, sob a forma de um ângulo θ , onde $\theta \in [0, 360]$. O vermelho está situado a 0 grau, amarelo a 60, verde a 120 e assim sucessivamente. O componente de saturação, que varia entre 0 e 1, indica o quanto a cor está poluída com a cor branca. A escala de intensidade está entre $[0, 1]$, onde 0 significa preto e 1, branco.

A transformação dos componentes do espaço RGB para HSI é dada pelas equações:

$$\begin{aligned}
 \theta &= \cos^{-1} \left(\frac{(R - G) + (R - B)}{2\sqrt{(R - G)^2 + (R - B)(G - B)}} \right) \\
 H &= \begin{cases} \theta, & \text{se } B \leq G \\ 360 - \theta, & \text{caso contrário} \end{cases} \\
 S &= 1 - \frac{3\min(R, G, B)}{R + G + B} \\
 I &= \frac{R + G + B}{3}
 \end{aligned} \tag{2.9}$$

É importante ressaltar que os valores R, G e B devem estar normalizados no intervalo entre 0 e 1. A intensidade I e a saturação S também estão normalizadas entre 0 e 1.

Outro modelo desta família é formado pelos componentes Matiz, Saturação e Luminância (**HSV**) e sua representação gráfica tridimensional é uma pirâmide hexagonal derivada do cubo RGB (Pedrini e Schwartz, 2008).

Os vários matizes estão representados na parte superior da pirâmide, a saturação é medida ao longo do eixo horizontal e a luminância é medida ao longo do eixo vertical, que passa pelo centro da pirâmide. O matiz, que corresponde às arestas ao redor do eixo vertical, varia de 0 (vermelho) a 360 graus e o ângulo entre os vértices é de 60 graus. A saturação varia de 0 a 1 e é representada como sendo a razão entre a pureza de um determinado matiz e a sua pureza máxima, ou seja, quando $S = 1$. A luminância varia de 0, no pico da pirâmide representando a cor preta, a 1 na base, onde as intensidades das cores são máximas (Pedrini e Schwartz, 2008).

A conversão do espaço RGB para HSV é dada pelas equações:

$$\begin{aligned}
 H &= \begin{cases} 60 \frac{(G - B)}{M - m}, & \text{se } M = R \\ 60 \frac{(B - R)}{M - m} + 120, & \text{se } M = G \\ 60 \frac{(R - G)}{M - m} + 240, & \text{se } M = B \end{cases} \\
 S &= \begin{cases} \frac{(M - m)}{M}, & \text{se } M \neq 0 \\ 0, & \text{caso contrário} \end{cases} \\
 V &= M
 \end{aligned} \tag{2.10}$$

onde $m = \min(R, G, B)$ e $M = \max(R, G, B)$. A luminância V e a saturação S estão normalizadas entre 0 e 1. Já o matiz H varia entre 0 e 360 graus.

Da mesma forma que o HSV, o modelo Matiz, Saturação e Luminosidade (**HSL**) é uma representação tridimensional e é formado por dois cones de altura 1, cujas bases são coincidentes (Pedrini e Schwartz, 2008).

O matiz é determinado pelos pontos no círculo das bases comuns aos cones. A saturação varia de 0 a 1, conforme a distância ao eixo do cone. A luminosidade está ao longo do eixo vertical comum aos dois cones e varia na escala $[0, 1]$, onde 0 significa preto e 1, branco (Pedrini e Schwartz, 2008).

A conversão do espaço RGB para HSL é dada pelas equações:

$$\begin{aligned}
 H &= \begin{cases} 60 \frac{(G - B)}{M - m}, & \text{se } M = R \\ 60 \frac{(B - R)}{M - m} + 120, & \text{se } M = G \\ 60 \frac{(R - G)}{M - m} + 240, & \text{se } M = B \end{cases} \\
 S &= \begin{cases} \frac{(M - m)}{M + m}, & \text{se } 0 < L \leq 0,5 \\ \frac{(M - m)}{2 - (M + m)}, & \text{se } L > 0,5 \\ 0, & \text{se } M = m \end{cases} \\
 L &= \frac{M + m}{2}
 \end{aligned} \tag{2.11}$$

onde $m = \min(R, G, B)$ e $M = \max(R, G, B)$. A luminância V e a saturação S estão normalizadas entre 0 e 1. Observe que a transformação do matiz H é a mesma utilizada na conversão do espaço RGB para HSV em 2.10 e varia entre 0 e 360 graus.

Todos os modelos de cores desta família têm a propriedade de se pensar em cores mais claras, obtidas pelo aumento do brilho ou luminosidade, e mais escuras, pela diminuição dos mesmos valores. As cores intermediárias são produzidas pela diminuição da saturação (Pedrini e Schwartz, 2008).

2.2 Teoria fuzzy

Em muitos problemas, não há dificuldade em determinar se um dado elemento é ou não parte de um grupo. Essa ideia vem da teoria clássica de conjuntos e é embasada no conceito fundamental de conjunto (Chen e Pham, 2000). Pode-se, por exemplo, afirmar que o número 7 pertence ao conjunto dos números naturais e, da mesma maneira, que o número -7 não pertence a esse mesmo conjunto. Este é um caso clássico em que a aplicação da lógica booleana é bem sucedida. Em outras palavras, quando questionado se o número 7 pertence ao conjunto dos naturais tem-se, apenas, duas respostas possíveis: sim ou não.

Entretanto, há inúmeros indivíduos ou observações na natureza que nem sempre podem ser classificados de tal forma, pelo fato de que a relação de pertinência não é bem definida (Pedrycz e Gomide, 1998). Como exemplo, o conjunto das pessoas altas, os números reais aproximadamente zero ou o grupo de alunos mais inteligentes da escola. Ocorre que, nesses casos, a aplicação da lógica booleana para classificar elementos como parte de um grupo ou não é imprecisa. Em outras palavras, se tomado $x = 1,70m$ como o limiar que separa as pessoas altas das baixas, fica evidente que a utilização da lógica booleana não resulta em interpretações realísticas, pois uma pessoa com $1,69m$ é, portanto, baixa ou, pode-se dizer ainda, que não pertence ao grupo das pessoas altas. Ora, apenas 1 centímetro separa tal pessoa de um grupo ou de outro. Analogamente, uma pessoa com $1,71m$ está muito próxima de pertencer ao grupo das pessoas baixas.

Claramente, a resposta a ser obtida quando da análise de possibilidades desse tipo é fortemente dependente do contexto, pois há um alto grau de incerteza inerente aos elementos e conjuntos sendo considerados.

O conceito de incerteza mudou gradativamente na ciência e na matemática. Na ciência, esta mudança foi manifestada por uma transição gradual da visão tradicional, que insiste que a incerteza é indesejável e, por esta razão, ela deve ser evitada por todos os meios possíveis, para uma visão alternativa, que é tolerante à incerteza. De acordo com a visão tradicional, a ciência deve esforçar-se de certeza em todas as suas manifestações, ou seja, com precisão, especificidade, nitidez, consistência e, por conseguinte, a incerteza é considerada como não científica. De acordo com a visão alternativa,

a dúvida é considerada essencial para a ciência; não é só uma praga inevitável, mas tem, de fato, uma grande utilidade (Klir e Yuan, 1995).

E com base na ideia moderna de que a incerteza é algo útil na ciência é que Zadeh convergiu para um de seus mais importantes trabalhos: a teoria de conjuntos *fuzzy*, cuja abordagem sobre conjuntos dá-se sob a ótica de que os limites entre eles não são precisos e, portanto, é possível estabelecer funções que forneçam um certo grau de pertinência de um elemento a um dado conjunto (Zadeh, 1965).

A capacidade de conjuntos *fuzzy* expressarem transições graduais de pertinência e não pertinência tem uma grande utilidade. Ela fornece não só uma representação significativa e poderosa da medida de incerteza, mas também uma forma de expressar conceitos vagos em linguagem natural (Klir e Yuan, 1995). Voltando ao exemplo do conjunto de pessoas altas, em vez de descrever a altura de uma pessoa com exatidão em metros ou centímetros, pode-se apenas dizer que ela é alta, baixa ou mediana. Cada um desses conjuntos pode ser um conjunto *fuzzy*, onde um valor 1 é atribuído a um membro que está totalmente incluído em um deles, ou 0 caso contrário. Valores intermediários nessa escala indicam que o elemento está parcialmente em um dos conjuntos.

A seção 2.2.1 explicita algumas das definições formais e conceitos básicos da teoria dos conjuntos *fuzzy* proposta por Zadeh.

2.2.1 Conjuntos *fuzzy*

A teoria de conjuntos clássicos está baseada em uma função, geralmente conhecida como função característica, dada por:

$$\mu_A(x) = \begin{cases} 1 & \text{se } x \in A \\ 0 & \text{se } x \notin A \end{cases} \quad (2.12)$$

onde U é o conjunto Universo, A é um subconjunto de U e x é um elemento de U (Klir e Yuan, 1995).

Portanto, a função característica é um mapeamento dos elementos de U no conjunto binário $\{0, 1\}$, formalmente definida como:

$$\mu_A = U \rightarrow \{0, 1\} \quad (2.13)$$

Sendo assim, a função característica determina que, de acordo com algum critério, $\forall x \in U$, se $\mu_A(x) = 1$, então x é um membro de A , por outro lado, quando $\mu_A(x) = 0$, x não é um membro de A .

Quanto aos conjuntos *fuzzy*, basta generalizar a função característica aplicada nos conjuntos clássicos no intervalo $[0, 1]$ para obter-se um conjunto "fuzzificado", ou seja, o grau de pertinência de um elemento x passa, agora, a ser expresso em termos contínuos. Em outras palavras, o elemento x pertence ao subconjunto A de U com algum grau de pertinência obtido do intervalo $[0, 1]$. Formalmente, tem-se:

$$\mu_A = U \rightarrow [0, 1] \quad (2.14)$$

Cabe aqui citar a definição de conjuntos *fuzzy* dada por Zadeh que designa um conjunto *fuzzy* como uma classe de objetos com um grau de pertinência contínuo. Tal conjunto é caracterizado por uma função de pertinência (característica), que atribui a cada objeto um grau de pertinência que varia entre zero e um. As noções de inclusão, união, intersecção, complemento, relação, convexidade, etc., oriundas da teoria de conjuntos clássica, são estendidas a esses conjuntos (Zadeh, 1965).

Definição 2.1. Um conjunto *fuzzy* A é um subconjunto do conjunto universo U formado por pares ordenados de um elemento qualquer x e seu grau de pertinência dado por $\mu_A(x)$, da forma:

$$A = \{(x, \mu_A(x)) \mid x \in U\} \quad (2.15)$$

O universo de discurso U pode ser composto por elementos discretos ou ser um espaço contínuo. A mesma implicação vale para o subconjunto A . No caso em que U é um conjunto discreto e finito, tal que $U = \{x_1, x_2, x_3, \dots, x_n\}$, pode-se simplesmente enumerar os seus elementos, juntamente com seus graus de pertinência, como segue:

$$A = \frac{\mu_A(x_1)}{x_1} + \frac{\mu_A(x_2)}{x_2} + \dots + \frac{\mu_A(x_n)}{x_n} = \sum_{i=1}^n \frac{\mu_A(x_i)}{x_i} \quad (2.16)$$

É importante ressaltar que o somatório em 2.16 não significa uma adição algébrica, mas sim a união de todos os pares ordenados de x e $\mu_A(x)$ que formam o conjunto A (Klir e Yuan, 1995).

Exemplo 2.1. Colocar um exemplo aqui de conjunto fuzzy que tenha sentido com o restante do texto, posteriormente. Pode ser pessoas altas, para ligar com o exemplo supra citado.

Definição 2.2. A altura de A , denotado por $altura(A)$, corresponde ao limite superior do codomínio da sua função de pertinência, da forma:

$$altura(A) = \{\mu_A(x) \mid x \in U\} \quad (2.17)$$

Definição 2.3. Um conjunto fuzzy A é dito normalizado se existe pelo menos um elemento $x \in U$, tal que $\mu_A(x) = 1$.

A definição de conjunto fuzzy normalizado implica que pelo menos um membro de A alcança o grau de pertinência máximo possível. Note que a definição em 2.3 considera que os valores dos graus de pertinência variam no intervalo fechado entre 0 e 1. Logo, no mínimo um elemento deve ter um grau de pertinência 1 para que A possa ser considerado, de fato, normalizado. Observe que, decorrente dessa afirmação, imediatamente tem-se que $altura(A) = 1$.

Definição 2.4. O suporte de um conjunto fuzzy A em U , denotado por $suporte(A)$, é o conjunto dado por:

$$suporte(A) = \{x \in U \mid \mu_A(x) > 0\} \quad (2.18)$$

Em outras palavras, o conjunto suporte é um conjunto clássico ou conjunto ordinário que contém todos os elementos de U cujo grau de pertinência é maior do que zero em A . Denomina-se por conjunto unitário ou *singleton* um conjunto fuzzy cujo suporte é um único ponto, por exemplo w , tal que $\mu_A(w) = 1$.

Definição 2.5. Um α -corte, também conhecido por α -nível, é o subconjunto clássico de elementos cujo grau de pertinência é maior ou igual a um valor α , formalmente:

$$\alpha\text{-corte}(A) = \{x \in U \mid \mu_A(x) \geq \alpha\} \quad (2.19)$$

Definição 2.6. O núcleo ou *kernel* de um conjunto fuzzy A em U , é o conjunto de elementos pertencentes inteiramente à A , da forma:

$$núcleo(A) = \{x \in U \mid \mu_A(x) = 1\} \quad (2.20)$$

Vale destacar que, por construção, o núcleo do conjunto A está contido no conjunto suporte, o que pode ser representado formalmente como $núcleo(A) \subseteq suporte(A)$.

A figura 2.6 mostra graficamente algumas das propriedades definidas acima.

2.2.2 Operações entre conjuntos fuzzy

Assim como na teoria de conjuntos clássica, há basicamente três operações que podem ser estendidas aos conjuntos fuzzy. Essas operações são a união, a intersecção e o complemento, sendo representadas pelos operadores lógicos de conjunção (**OU**), disjunção (**E**) e complemento (**NÃO**),

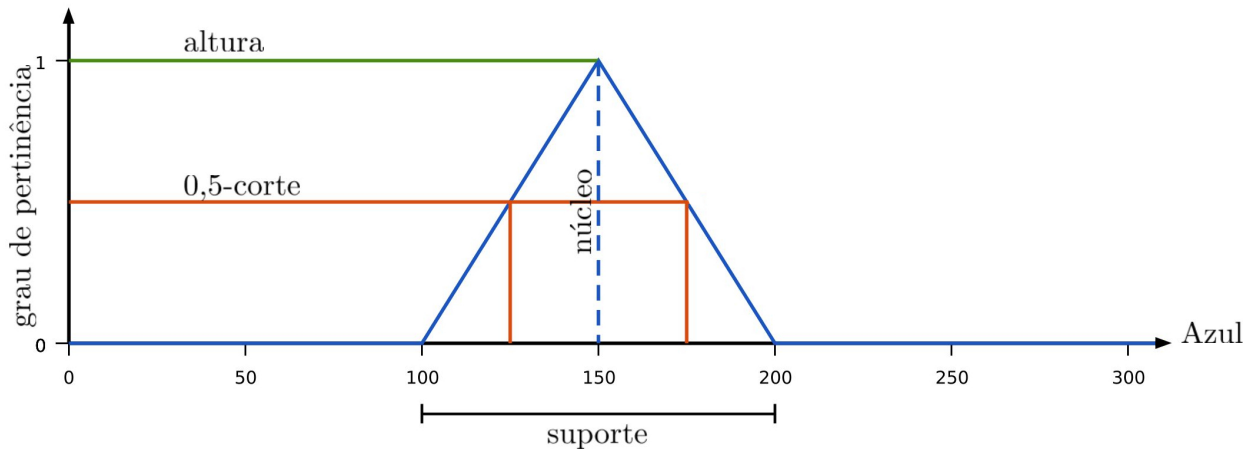


Figura 2.6: Representação gráfica das principais propriedades dos conjuntos fuzzy. *Detalhar melhor e corrigir a imagem.*

respectivamente. As operações básicas formam o que é conhecido como operações padrão em conjuntos fuzzy (Klir e Yuan, 1995).

Tais operações são realizadas sobre suas respectivas funções de pertinência e há muitas formulações matemáticas possíveis para os operadores que representam-nas. O par de operadores mais amplamente utilizado é o operador de mínimo (\min) e máximo (\max), que são semelhantes aos operadores produto e soma da álgebra elementar e utilizados para expressar a conjunção e disjunção, respectivamente (Thole *et al.*, 1979).

Seja dois conjuntos fuzzy A e B , definidos sobre um mesmo universo de discurso U . Utilizando as funções \max e \min , tem-se que (Klir e Yuan, 1995):

Definição 2.7. O conjunto união é formado por todos os valores máximos entre dois conjuntos fuzzy A e B , onde $A, B \in U$, da forma:

$$\mu_A(x) \cup \mu_B(x) = \max\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in U \quad (2.21)$$

A figura 2.7 exhibe graficamente dois conjuntos fuzzy A e B quaisquer, bem como a área pertinente à união entre eles.

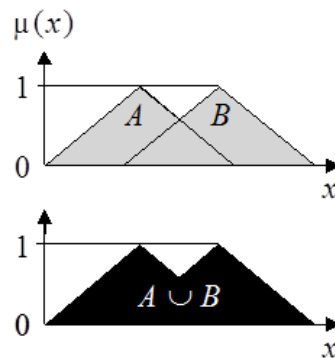


Figura 2.7: Representação gráfica da operação de união entre dois conjuntos fuzzy. A parte superior em cinza claro mostra os dois conjuntos fuzzy A e B em sobreposição. A união é destacada em preto na parte inferior. Adaptado de Vrusias (2006).

Definição 2.8. O conjunto intersecção é formado por todos os valores mínimos entre dois conjuntos fuzzy A e B , onde $A, B \in U$, da forma:

$$\mu_A(x) \cap \mu_B(x) = \min\{\mu_A(x), \mu_B(x)\}, \quad \forall x \in U \quad (2.22)$$

A figura 2.8 exibe graficamente dois conjuntos *fuzzy* A e B quaisquer, bem como a área pertinente à intersecção.

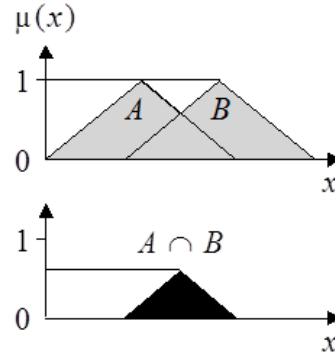


Figura 2.8: Representação gráfica da operação de intersecção entre dois conjuntos *fuzzy*. A parte superior em cinza claro mostra os dois conjuntos *fuzzy* A e B em sobreposição. A intersecção é destacada em preto na parte inferior. Adaptado de *Vrusias (2006)*.

Definição 2.9. O complemento de um conjunto *fuzzy* A , denotado por $\mu_{\bar{A}}(x)$, onde $A \in U$, é formado pela subtração entre o valor unitário e $\mu_A(x)$, da forma:

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x), \quad \forall x \in U \quad (2.23)$$

A figura 2.9 exibe graficamente um conjunto *fuzzy* A qualquer, bem como seu complemento.

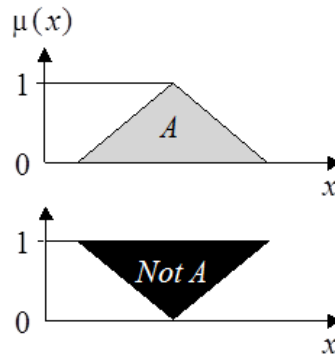


Figura 2.9: Representação gráfica de um conjunto *fuzzy* e seu complemento. A parte superior em cinza claro mostra o conjunto *fuzzy* A . Seu complemento é destacado em preto na parte inferior. Adaptado de *Vrusias (2006)*.

Além desses operadores, existem outros que podem ser utilizados para efetuar as operações lógicas. Um deles é a norma triangular, também denominada *norma-t*. Outro é a co-norma triangular, usualmente denominada por *co-norma-t* ou *norma-s*. Os operadores utilizados para a intersecção de conjuntos *fuzzy* pertencem à classe *norma-t*. Os operadores utilizados para a união de conjuntos *fuzzy* pertencem à classe *co-norma-t* ou *norma-s* (*Zimmermann, 2001*).

Ambas *norma-t* e *norma-s* são funções binárias do tipo $[0, 1] \times [0, 1] \rightarrow [0, 1]$, que devem satisfazer algumas propriedades, tais como, monotonicidade, comutatividade, associatividade e condições de contorno ². Os operadores *max* e *min* satisfazem tais propriedades e, por essa razão, podem ser utilizados nas operações lógicas *fuzzy* (*Zimmermann, 2001*).

²As propriedades não serão expressas neste trabalho, embora sejam amplamente conhecidas, exceto pelas condições de contorno. Mais detalhes podem ser vistos em *Zimmermann (2001)*.

2.2.3 Números *fuzzy*

Conjuntos *fuzzy* são composições de números *fuzzy*, definidos em um universo discreto ou contínuo, de forma tal que a incerteza ou imprecisão associada a uma dada informação possa ser quantificada. Portanto, grandezas do tipo *em torno de 7, aproximadamente -7, mais ou menos 30* podem ser mapeadas e, conseqüentemente, interpretadas por números *fuzzy*. Dessa forma, a concepção intuitiva sobre números aproximados ou intervalos pode ser capturada (Klir e Yuan, 1995).

Klir e Yuan (1995) caracterizam um número *fuzzy* como um tipo especial de conjunto *fuzzy* definido no conjunto \mathbb{R} dos números reais, cuja função de pertinência tem a forma:

$$A = \mathbb{R} \rightarrow [0, 1] \quad (2.24)$$

Para que A seja, de fato, um número *fuzzy*, o conjunto universo no qual μ_A está definida deve ser \mathbb{R} e as seguintes propriedades devem ser satisfeitas (Barros e Bassanezi, 2006):

- (i) todos os α -corte de A são não vazios, com $0 \leq \alpha \leq 1$;
- (ii) todos os α -corte são intervalos fechados de \mathbb{R} ;
- (iii) $\text{suporte}(A) = \{x \in U \mid \mu_A(x) > 0\}$.

De modo geral, números *fuzzy* podem ser representados por funções parametrizadas. Algumas dessas funções serão apresentadas na seção 2.2.4.

2.2.4 Funções de pertinência

As funções de pertinência são responsáveis por um aspecto fundamental na teoria de conjuntos *fuzzy*. Como supra citado, elas permitem, por exemplo, que um número *fuzzy* seja representado de forma paramétrica.

As funções lineares por partes são as mais populares devido à sua baixa complexidade e eficiência e, por conseguinte, baixo custo computacional necessário para o seu processamento, o que não ocorre com outros tipos de funções já que, em geral, não resultam em uma melhoria significativa na qualidade dos valores de saída dos sistemas (Yen e Langari, 1999). Nesse contexto, as funções de pertinência mais comumente utilizadas são as triangulares e trapezoidais. Vale ressaltar que não é necessário que as funções sejam simétricas ou igualmente espaçadas. Além disso, cada variável do problema pode ser modelada com funções de pertinência diferentes, com formas e distribuições próprias, definidas de acordo com as suas características e do contexto onde estão sendo aplicadas.

Em aplicações onde é necessário transições mais suaves, pode-se usar funções gaussianas, sigmoidais, dentre outras, desde que estejam definidas no intervalo $[0, 1]$. Algumas dessas funções serão apresentadas em 2.10, 2.11 e 2.12, com base nas definições de Sumathi e Paneerselvam (2010).

Definição 2.10. Um número *fuzzy* A é dito triangular se sua função de pertinência, denotada por $\mu_A(x)$, é da forma:

$$\mu_A(x) = \begin{cases} \frac{x-a}{m-a}, & \text{se } a \leq x \leq m \\ \frac{b-x}{b-m}, & \text{se } m \leq x \leq b \\ 0, & \text{caso contrário} \end{cases} \quad (2.25)$$

onde m é o máximo grau de pertinência ou $\text{altura}(A)$, em geral $\mu_A(x) = 1$, e a e b os limites inferior e superior do intervalo, respectivamente, onde $\mu_A(x)$ é não nula.

A figura 2.10 exhibe o gráfico da função $\mu_A(x)$ de um número *fuzzy* triangular A .

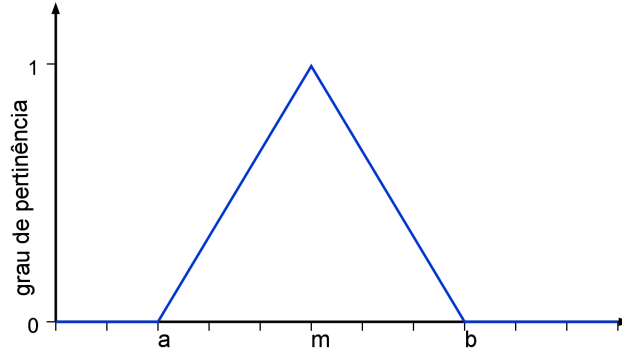


Figura 2.10: Gráfico da função de pertinência triangular. Note que a forma é de um triângulo, que tem como base o intervalo $[a, b]$ e como $\text{núcleo}(A)$ o vértice $(m, 1)$. Fonte: proposto pelo autor.

Definição 2.11. Um número fuzzy A é dito trapezoidal se sua função de pertinência, denotada por $\mu_A(x)$, é da forma:

$$\mu_A(x) = \begin{cases} \frac{x-a}{b-a}, & \text{se } a \leq x \leq b \\ 1, & \text{se } b \leq x \leq c \\ \frac{d-x}{d-c}, & \text{se } c \leq x \leq d \\ 0, & \text{caso contrário} \end{cases} \quad (2.26)$$

A figura 2.11 exibe o gráfico da função $\mu_A(x)$ de um número fuzzy trapezoidal A .

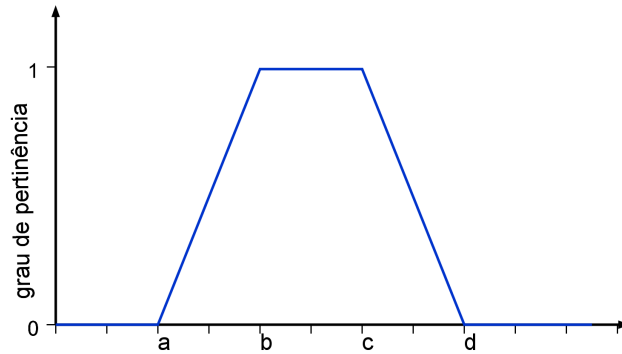


Figura 2.11: Gráfico da função de pertinência trapezoidal. Aqui a forma é de um trapézio, sendo que $\mu_A(x)$ é nula para todo $x \notin [a, d]$, linearmente crescente até altura(A), geralmente, para $x = b = c$, constante nesse valor para x entre b e c , e, finalmente, linearmente decrescente até zero em $x = d$. Fonte: proposto pelo autor.

Definição 2.12. Um número fuzzy A é dito gaussiano se sua função de pertinência, denotada por $\mu_A(x)$, é da forma:

$$\mu_A(x) = \begin{cases} \exp\left(-\frac{(x-m)^2}{\sigma}\right), & \text{se } m - \sigma \leq x \leq m + \sigma \\ 0, & \text{caso contrário} \end{cases} \quad (2.27)$$

onde m é o $\text{núcleo}(A)$ e σ é o desvio padrão da função gaussiana. Quanto menor σ , mais estreita a abertura horizontal da curva. A figura 2.12 exibe o gráfico da função $\mu_A(x)$ de um número fuzzy gaussiano A .

2.2.5 Variáveis linguísticas

Uma variável linguística é aquela cujo valor pode ser expresso por um termo qualitativo capaz de fornecer um significado conceitual à variável, tais como muito pequeno, pequeno, médio,

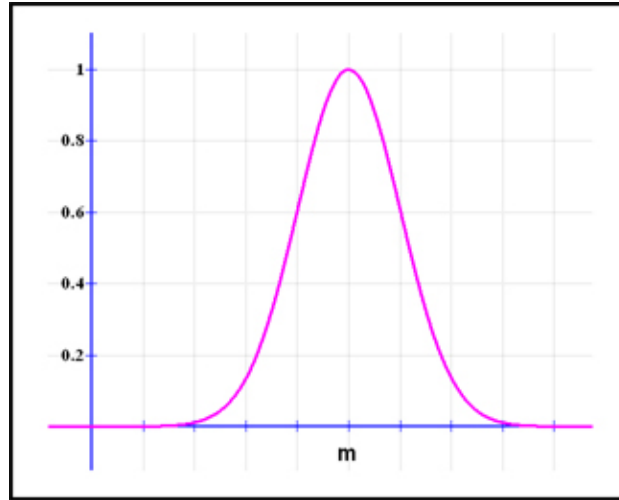


Figura 2.12: Gráfico da função de pertinência gaussiana. Aqui a forma é de um sino, sendo que $\mu_A(x)$ é nula para todo $x \notin [m - \sigma, m + \sigma]$, crescente até altura(A), onde $x = m$ e decrescente até zero em $x = m + \sigma$. Fonte: proposto pelo autor.

baixo, alto e assim por diante, e que podem fornecer uma interpretação num contexto específico (Pedrycz e Gomide, 1998).

Para Zadeh (1975), uma variável linguística é aquela cujo valor é uma palavra ou uma sentença numa linguagem natural ou artificial.

A motivação para o uso de variáveis linguísticas é que elas são, em geral, menos específicas do que variáveis numéricas (Zadeh, 1975). Essa motivação tem um propósito particular quando os subconjuntos *fuzzy* são determinados em termos de regras, já que o objetivo não é definir exaustivamente as variáveis linguísticas.

Cada variável linguística tem seus estados expressos por meio de termos de uma variável base, cujos valores são números reais dentro de um intervalo específico. Uma variável de base é uma variável no sentido clássico, por exemplo, temperatura, pressão, velocidade, tensão, umidade, idade, desempenho, salário, etc (Klir e Yuan, 1995).

Intuitivamente, as variáveis linguísticas podem ser vistas como substantivos, por exemplo altura ou nota e os seus valores são adjetivos que caracterizam-nas, tais como alto, baixo, ruim, bom, dentre outros. A definição formal é dada em 2.13 (Klir e Yuan, 1995; Zadeh, 1975).

Definição 2.13. Uma variável linguística é caracterizada por uma quintupla (v, T, U, G, M) , onde:

- (i) v é o nome da variável.
- (ii) U é o universo de discurso.
- (iii) T é o conjunto de termos linguísticos de v relacionado a uma variável base cujos valores variam ao longo de U .
- (iv) G é uma regra sintática, ou seja, a gramática para a geração de termos linguísticos em T .
- (v) M é uma regra semântica que atribui para cada termo linguístico $t \in T$ seu significado, que é um conjunto *fuzzy* em U .

2.2.6 Regras fuzzy

Os seres humanos tomam decisões baseados em regras, do tipo *se* a temperatura está baixa, *então* coloque um casaco, *se* o combustível está acabando, *então* abasteça o veículo, etc.

As regras estão entre as técnicas mais utilizadas para representar o conhecimento. Para Zimmermann (2001), as regras ainda são amplamente utilizados devido ao fato de que tornam possível modelar

o conhecimento em um sistema *fuzzy* capturando-o por meio de especialistas. As regras também têm a vantagem de poderem ser expressas por uma linguagem natural ou artificial que, em geral, é facilmente compreensível.

De maneira similar a uma afirmação clássica, uma regra *fuzzy* é composta por duas partes, resultando em uma estrutura do tipo (Dubois e Prade, 1996):

Se {antecedentes}, então {consequentes}.

Os antecedentes caracterizam premissas, enquanto a parte consequente são ações tomadas quando as premissas são atendidas. Diferentemente de uma regra clássica, um ou mais antecedentes podem ser parcialmente atendidos para que as premissas sejam aferidas e, portanto, as regras *fuzzy* são mais relaxadas do que as regras clássicas que devem ser verificadas na sua totalidade.

A forma de interpretação mais comum e amplamente usada de uma regra *fuzzy* é (Dubois e Prade, 1996):

Se x é A , então y é B .

onde A e B são variáveis linguísticas definidas por conjuntos *fuzzy* em universos de discurso U e V , respectivamente. Um exemplo da aplicação da regra pode ser visto em 2.2.

Exemplo 2.2. *Se verde é baixo, então pixel é pele.*

É importante observar que **verde** é uma variável linguística e o adjetivo **baixo** é um termo linguístico representado como um número no intervalo $[0, 1]$ e, portanto, o antecedente é uma interpretação que retorna um único número desse intervalo. Por outro lado, **pele** é a representação de um conjunto *fuzzy* e, por isso, o consequente é uma atribuição de todo o conjunto *fuzzy* B para a variável de saída y , o pixel no exemplo 2.2. Porém, o consequente pode assumir outros valores que não só um conjunto *fuzzy*, como também, um valor numérico clássico.

Também é válido destacar a possibilidade de utilizar conectivos na formação da regra para que ela seja estendida e tenha múltiplas partes, o que possibilita a interpretação de diversas variáveis linguísticas ao mesmo tempo em um contexto específico, como pode ser visto em 2.3. Nesse caso, todas as partes dos antecedentes são calculadas simultaneamente em um único resultado, com base nos operadores destacados na seção 2.2.2.

Exemplo 2.3. *Se verde é médio e vermelho é baixo e azul é alto, então pixel é não pele.*

Apesar de ser possível empregar múltiplas partes também para os consequentes, tal capacidade não será utilizada e, consequentemente, não abordada neste trabalho.

2.3 Classificadores

A aprendizagem de máquina é uma área que está preocupada com o desenvolvimento de programas de computador capazes de melhorar automaticamente com a experiência (Mitchell, 1997). Essa definição está intimamente ligada ao modo com que seres humanos aprendem. Os esforços de pesquisa nessa área têm sido realizados com o propósito de aproximar esta relação.

Como exemplo, ao mostrar a imagem de uma árvore para uma criança de três anos de idade, muito provavelmente ela saberá reconhecê-la, pois deve ter sido exposta a situações em que tenha visto imagens semelhantes e, sendo assim, foi treinada para dar tal resposta. Logo, ela aprendeu o que é uma árvore apenas olhando para elas, não necessariamente por meio de definições matemáticas precisas. Em outras palavras, o aprendizado foi feito com base em dados que, muitas vezes, são utilizados para obter-se soluções empíricas para determinados problemas onde não há a possibilidade de se criar uma solução analítica (Abu-Mostafa et al., 2012).

Analogamente, um algoritmo pode ser treinado para diferenciar uma árvore de outros objetos com base em um conjunto de dados que possui descrições sobre as árvores, tais como, altura, cores, espessura, comprimento, etc. Tais descrições também são chamadas de atributos, propriedades ou

características e são submetidas a um classificador que avalia as evidências apresentadas e toma uma decisão sobre o objeto sendo analisado (Duda *et al.*, 2012). Essa tarefa começa com a definição de um vetor de características em um espaço d -dimensional, da forma (Duda *et al.*, 2012):

$$x = \begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_d \end{bmatrix} \quad (2.28)$$

onde $x \in X$ e X é o espaço de entrada, ou seja, todos os x vetores possíveis.

O conjunto de dados é formado por N destes vetores e, sendo assim, o problema agora está em particionar o espaço de características de tal maneira que uma fronteira de decisão é formada (Duda *et al.*, 2012). Pode-se, então, atribuir uma classe ou rótulo y específico para um dado vetor, onde $y \in Y$ e Y é um conjunto finito de classes que, no caso binário, é da forma $Y = \{+1, -1\}$. A figura 2.13 mostra exemplos de particionamento de um espaço de características 2-dimensional.

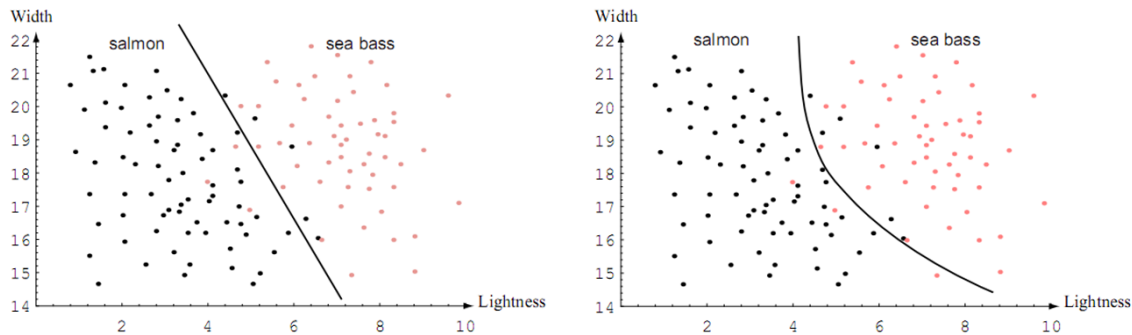


Figura 2.13: Fronteira de decisão no espaço de características 2-dimensional. As figuras à esquerda e à direita representam a separação do espaço de características por funções linear e não linear, respectivamente. Vale ressaltar que, a complexidade computacional aumenta significativamente nos casos em que existem muitas características. O ideal aqui é trocar essa distribuição pelo dataset de pele. A fonte dessas imagens é Duda *et al.* (2012).

Note que a fronteira de decisão é uma função hipotética que o algoritmo de aprendizado deve produzir com base no treinamento dos dados. Ela deve ser mais próxima possível de uma função alvo ou função objetivo $f : X \mapsto Y$ que é desconhecida (Abu-Mostafa *et al.*, 2012).

Este tipo de abordagem é caracterizada como aprendizagem supervisionada, ou seja, os dados de treinamento contêm amostras explícitas de como deve ser a saída correta sobre um vetor de entrada (Abu-Mostafa *et al.*, 2012).

Na aprendizagem não supervisionada, também conhecida como clusterização, não se sabe a que classes pertencem os dados de treinamento. A missão do classificador, neste caso, é aglomerar os dados de entrada em agrupamentos naturais (Duda *et al.*, 2012). Logo, a aprendizagem não supervisionada pode ser vista como uma tarefa de encontrar padrões ou estruturas espontaneamente a partir dos dados de entrada (Abu-Mostafa *et al.*, 2012).

Uma outra abordagem é a aprendizagem reforçada que, da mesma forma que na aprendizagem não supervisionada, não usa dados de entrada rotulados, em vez disso, a saída proposta pelo treinamento é utilizada juntamente com uma medida de sua qualidade para melhorar os resultados do classificador (Abu-Mostafa *et al.*, 2012).

Ainda sobre os resultados do classificador, é importante salientar que aprender os parâmetros de uma função alvo e testá-la nos mesmos dados é um erro fundamental, pois o modelo repetiria os rótulos das amostras tão logo treinadas implicando em um ajuste perfeito dos dados, porém, não seria suficiente para prever qualquer coisa útil sobre novos dados de entrada. Este fenômeno é chamado de sobreajuste e para evitá-lo, o conjunto de dados é, então, particionado de forma a

armazenar parte dos dados para uma etapa subsequente de teste, no intuito de encontrar o estimador com a melhor performance (Abu-Mostafa *et al.*, 2012).

A divisão dos dados em subconjuntos disjuntos de treinamento e de teste é uma abordagem eficaz quando uma grande quantidade de dados está disponível. Entretanto, quando os dados são limitados, a retenção de parte deles para o conjunto de teste reduz ainda mais o número de amostras disponíveis para treinamento (Mitchell, 1997). Logo, uma outra abordagem, conhecida como validação cruzada, pode ser aplicada de modo que todo o conjunto de dados é usado no treinamento e teste.

A validação cruzada consiste em dividir o conjunto de dados D em K subconjuntos disjuntos D_1, D_2, \dots, D_K , onde cada subconjunto tem tamanho aproximado N/K . O modelo é, então, treinado em cada um desses subconjuntos, exceto um que é mantido como um conjunto de validação, no qual a medida de erro é realizada. Este processo é repetido K vezes, de tal forma que cada um dos subconjuntos tem a oportunidade de agir como o conjunto de teste e, por isso, essa abordagem também é conhecida como *K-fold*. Ao final de todas as K iterações, a média do erro obtido por cada estimador é usado como a medida de performance do classificador (Abu-Mostafa *et al.*, 2012).

Alguns dos classificadores, que foram usados nos experimentos preliminares do capítulo 3, serão brevemente tratados nas seções 2.3.1 e 2.3.2.

2.3.1 Máquinas de vetores suporte

As Máquinas de Vetores Suporte (SVM) constituem uma técnica de aprendizagem computacional baseada na Teoria de Aprendizado Estatístico (Vapnik, 2013), cujo objetivo era de resolver problemas de classificação de padrões. Na prática, uma SVM tem a habilidade de gerar um hiperplano ou conjunto de hiperplanos num espaço de alta ou infinita dimensionalidade, que pode ser usado para tarefas de classificação, regressão ou outras (Duda *et al.*, 2012).

Derivado do próprio nome da técnica, os vetores de suporte são as amostras do conjunto de treinamento que definem o hiperplano ótimo e são os padrões mais difíceis de classificar (Duda *et al.*, 2012). Intuitivamente, eles são os padrões mais relevantes para a tarefa de classificação, pois uma mudança nestes vetores implicam diretamente no resultado do hiperplano ótimo ³.

Seja o conjunto de dados de treinamento com N amostras da forma:

$$D = (x_1, y_1), (x_2, y_2), \dots, (x_N, y_N) \quad (2.29)$$

onde cada x_i é um vetor d -dimensional da forma dada em 2.28, $i = 1, 2, \dots, N$, $y_i \in Y$ e $Y = \{+1, -1\}$. Portanto, o conjunto de treinamento contém N observações com suas respectivas classes.

Assumindo que D é linearmente separável, pode-se separar os dados por meio de um hiperplano usando um classificador, também linear, definido pela equação (Lorena e Carvalho, 2003):

$$w \cdot x + b = 0 \quad (2.30)$$

onde $w \cdot x$ é um produto escalar, w é o vetor normal ao hiperplano e b é um termo compensador. O parâmetro $\frac{b}{\|w\|}$ determina o deslocamento do hiperplano em relação à origem.

A partir dessa definição, outros dois hiperplanos paralelos ao hiperplano ótimo podem ser obtidos, conforme as equações em 2.31, de forma que uma região delimitada, conhecida como margem, se forma entre eles.

$$\begin{cases} w \cdot x + b = +1 \\ w \cdot x + b = -1 \end{cases} \quad (2.31)$$

Além disso, algumas restrições são definidas para evitar que não existam pontos entre $w \cdot x + b = 0$

³O hiperplano de separação ótimo de uma SVM é aquele de uma classe de hiperplanos com a maior margem de separação entre os dois conjuntos de treinamento (Cortes e Vapnik, 1995).

e $w \cdot x + b = \pm 1$, formalmente tem-se (Lorena e Carvalho, 2003):

$$\begin{cases} w \cdot x_i + b \geq +1, & \text{se } y_i = +1 \\ w \cdot x_i + b \leq -1, & \text{se } y_i = -1 \end{cases} \quad (2.32)$$

ou, equivalentemente:

$$y_i(w \cdot x_i + b) \geq 1 \quad (2.33)$$

Segundo Campbell (2000), no sistema dado na equação 2.32, supõe-se que a margem é sempre maior que a distância entre $w \cdot x + b = 0$ e $|w \cdot x + b| = 1$ e, por essa razão, SVMs desta natureza são usualmente denominadas de SVMs com margens rígidas. A figura 2.14 mostra a representação gráfica destes conceitos.

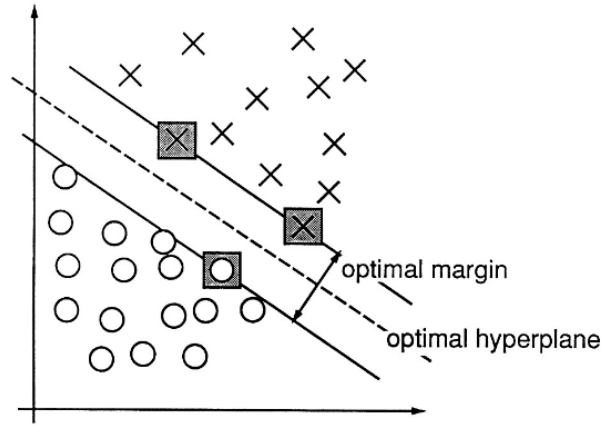


Figura 2.14: Exemplo de um problema de classes binárias separáveis num espaço 2-dimensional. Os vetores de suporte, marcados com quadrados cinza, definem a margem de maior separação entre as duas classes. Aqui também é interessante colocar as amostras de pele e não pele, talvez somente com duas características. Essa imagem é de (Cortes e Vapnik, 1995)

Geometricamente, a distância entre os dois hiperplanos paralelos ao hiperplano ótimo é $\frac{2}{\|w\|}$. Consequentemente, pode-se deduzir que a distância entre o hiperplano ótimo e $w \cdot x + b = \pm 1$ é $\frac{1}{\|w\|}$. Portanto, a minimização de $\|w\|$ maximiza a margem e, sendo assim, tem-se um problema de otimização no qual deseja-se minimizar $\|w\|^2$ sujeito às restrições em 2.33 (Lorena e Carvalho, 2003). Logo, w e b ótimos que resolvem este problema definem o classificador e podem ser obtidos por multiplicadores de Lagrange (Campbell, 2000):

$$\text{Maximizar: } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j x_i \cdot x_j \quad (2.34)$$

$$\text{Sujeito a: } \begin{cases} \alpha_i \geq 0 \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.35)$$

onde α são os multiplicadores de Lagrange.

É importante ressaltar que este tipo de SVM tem êxito em conjuntos de dados de treinamento linearmente separáveis. Nos casos em que os dados são não linearmente separáveis, é admitida a ocorrência de alguns erros de classificação para o conjunto de treinamento, pela inclusão de variáveis de relaxamento (Lorena e Carvalho, 2003). Essas alterações foram produzidas por Cortes e Vapnik (1995) e fazem parte de uma técnica denominada suavização de margens e, por conseguinte, SVMs dessa família também são chamadas de SVM com margens suaves. Logo, o problema de otimização

torna-se (Lorena e Carvalho, 2003):

$$\text{Minimizar: } ||w||^2 + C \sum_{i=1}^N \xi_i \quad (2.36)$$

$$\text{Sujeito a: } \begin{cases} \xi_i \geq 0 \\ y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{cases} \quad (2.37)$$

onde C é uma constante de regularização determinada empiricamente que impõe um peso diferente para o treinamento em relação à generalização. Sendo assim, o problema pode ser resolvido da mesma maneira com a equação em 2.34, porém, com algumas alterações nas restrições (Lorena e Carvalho, 2003):

$$\text{Sujeito a: } \begin{cases} 0 \leq \alpha_i \leq C \\ \sum_{i=1}^N \alpha_i y_i = 0 \end{cases} \quad (2.38)$$

Em geral, a maioria dos problemas de classificação impedem que classificadores lineares sejam utilizados por não ser possível obter resultados satisfatórios na partição dos dados de treinamento por um hiperplano. Entretanto, as SVMs lineares podem ser estendidas para lidar com essa situação mapeando o espaço de entrada em um espaço de características de dimensão mais alta, tipicamente muito maior que o espaço original (Duda *et al.*, 2012), da forma:

$$D' = (\Phi(x_1), y_1), (\Phi(x_2), y_2), \dots, (\Phi(x_N), y_N) \quad (2.39)$$

A escolha apropriada de uma função Φ torna o conjunto de treinamento linearmente separável no espaço transformado (Lorena e Carvalho, 2003). A forma do hiperplano ótimo agora é definida por:

$$w \cdot \Phi(x) + b = 0 \quad (2.40)$$

Logo, os conceitos de vetores de suporte, margem, hiperplanos paralelos e, conseqüentemente, de problema de otimização para encontrar a solução para os parâmetros w e b se aplicam aqui de forma similar. Formalmente, o problema de otimização pode ser resolvido como (Lorena e Carvalho, 2003):

$$\text{Maximizar: } \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j \Phi(x_i) \cdot \Phi(x_j) \quad (2.41)$$

sujeito às mesmas restrições definidas em 2.38.

Agora, há a necessidade de se definir como o produto interno $\Phi(x_i) \cdot \Phi(x_j)$ entre dois vetores quaisquer $x_i, x_j \in D$ é realizado, cuja resposta está na introdução do conceito de *kernels* (Lorena e Carvalho, 2003).

Kernels são funções que têm a finalidade de projetar os vetores de entrada num espaço de características com número de dimensões exponencial ou infinito (Shawe-Taylor e Cristianini, 2004), da forma:

$$k(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j) \quad (2.42)$$

Logo, é possível aplicar um *kernel* como descrito na equação 2.42 na etapa de otimização da SVM, calculando de maneira eficiente o produto interno a partir dos dados de entrada, sem nem mesmo computar explicitamente o mapeamento da função Φ (Shawe-Taylor e Cristianini, 2004).

Alguns dos *kernels* mais utilizados são (Lorena e Carvalho, 2003):

(i) *Kernel* linear

$$k(x_i, x_j) = (x_i \cdot x_j) \quad (2.43)$$

O *kernel* linear é o mais simples das funções *kernel* e se dá pelo produto interno de dois vetores x_i e x_j .

(ii) *Kernel* polinomial

$$k(x_i, x_j) = (\gamma x_i \cdot y_i + r)^g \quad (2.44)$$

onde g é o grau do polinômio e r é um termo constante. Para este *kernel*, os mapeamentos Φ também são funções polinomiais e, portanto, a complexidade é proporcional à escolha de g (Lorena e Carvalho, 2003).

(iii) *Kernel* gaussiano

$$k(x_i, x_j) = \exp(-\gamma \|x_i - y_i\|^2) \quad (2.45)$$

Também conhecido como Função de Base Radial (RBF), o *kernel* gaussiano corresponde a um espaço de características de dimensão infinita (Lorena e Carvalho, 2003).

No *kernel* RBF e polinomial, o parâmetro γ pode ser visto como o inverso do raio de influência das amostras selecionadas pelo modelo como vetores de suporte (Pedregosa *et al.*, 2011).

2.3.2 k -vizinhos mais próximos

O k -Vizinhos Mais Próximos (k -NN) é um algoritmo baseado em instâncias, o que significa que a função alvo não é aprendida com base em amostras de treinamento; elas simplesmente são mantidas, de maneira que a relação entre uma nova amostra com as amostras de treinamento armazenadas é avaliada e uma função alvo é, então, atribuída a ela (Mitchell, 1997).

Como o próprio nome sugere, o k -NN classifica um novo ponto x atribuindo-lhe a classe de maior frequência dentre as k amostras mais próximas, ou seja, a decisão da classe de x é feita por maioria de votos dos k vizinhos mais próximos e, por essa razão, é interessante que a escolha de k seja um número ímpar para evitar empates (Duda *et al.*, 2012).

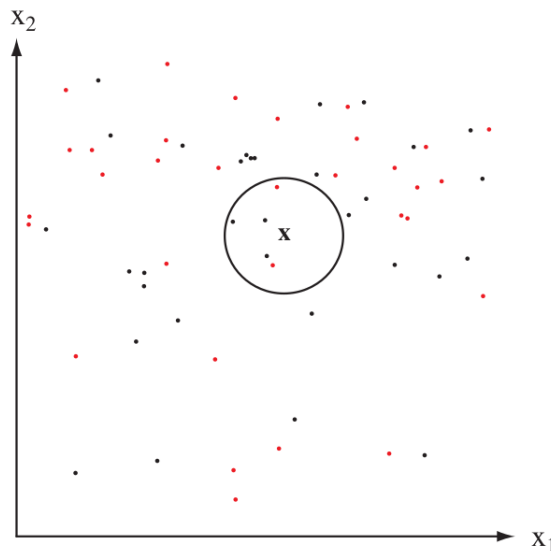


Figura 2.15: Exemplo de aplicação do k -NN na tarefa de classificação em um espaço 2-dimensional. O algoritmo avalia as k amostras próximas de x , criando uma região esférica, e rotula x com a classe de maior frequência. Neste caso, $k = 5$ e a classe atribuída a x deve ser a mesma dos pontos em preto. *Aqui também é interessante colocar as amostras de pele e não pele, talvez somente com duas características. Essa imagem é de (Duda et al., 2012)*

Seja D o conjunto de dados de treinamento com N amostras da forma apresentada em 2.29. Logo, a distância entre duas amostras x_i e x_j quaisquer pode ser obtida em termos da distância

Euclidiana, definida como:

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^d (a_r(x_i) - a_r(x_j))^2} \quad (2.46)$$

onde a_r refere-se ao r -ésimo atributo do vetor de entrada x e $i, j = 1, 2, \dots, N$. Outras métricas de distância podem ser atribuídas a $d(x_i, x_j)$, tais como, Manhattan, Chebyshev e Minkowski (Duda *et al.*, 2012).

Para, então, classificar uma nova amostra x_q , toma-se x_1, \dots, x_k instâncias do conjunto de treinamento, cujas distâncias dadas por 2.46, são dos k pontos mais próximos de x_q . Sendo assim, a função que estima a classe de x_q é dada por (Mitchell, 1997):

$$g(x_q) = \operatorname{argmax}_{y \in Y} \sum_{i=1}^k \delta(y, f(x_i)) \quad (2.47)$$

onde

$$\delta(y, f(x_i)) = \begin{cases} 1, & \text{se } y = f(x_i) \\ 0, & \text{caso contrário} \end{cases} \quad (2.48)$$

É importante ressaltar que $f(x_i)$ é conhecida, ou seja, é a classe da amostra x_i do conjunto de treinamento.

Uma variação óbvia da função dada na equação 2.47 é a atribuição de pesos de cada um dos k vizinhos, conforme sua distância, para um ponto x_q sendo classificado (Mitchell, 1997). Essa variação implica que pontos mais próximos de x_q têm maior influência na sua rotulação. Formalmente, tem-se que (Mitchell, 1997):

$$g(x_q) = \operatorname{argmax}_{y \in Y} \sum_{i=1}^k w_i \delta(y, f(x_i)) \quad (2.49)$$

onde

$$w_i = \frac{1}{d(x_q, x_i)^2} \quad (2.50)$$

No caso em que $d(x_q, x_i) = 0$, ou seja, x_q e x_i estão exatamente nas mesmas coordenadas, $g(x_q)$ pode assumir o mesmo valor de $f(x_i)$. Se há outras amostras de treinamento x_i com a mesma característica, então x_q pode assumir a classe da maioria delas (Mitchell, 1997).

2.3.3 Árvores de decisão

Árvore de decisão é um método para aproximação de funções alvo discretas, nas quais a função aprendida é representada por uma árvore de decisão ou, ainda, por um conjunto de regras do tipo *Se-Então* que são de fácil interpretação. É uma das técnicas de aprendizagem mais populares de inferência indutiva⁴ (Mitchell, 1997).

As amostras de um conjunto de dados são classificadas por uma árvore de decisão por um processo iterativo onde um atributo (nó) é escolhido como raiz até algum nó folha, onde a classe é atribuída à amostra. Cada ramo partindo de um nó representa um dos valores possíveis de um dado atributo (Mitchell, 1997).

Há uma família de algoritmos de árvore de decisão. Um deles é o Divisor Iterativo 3 (ID3) proposto por Quinlan (1986). O ID3 avalia cada atributo através de um teste estatístico para determinar o quão bem ele, por si só, classifica as amostras de treinamento. O melhor atributo é selecionado como o nó raiz da árvore. Um ramo descendente do nó raiz é criado para cada valor possível deste atributo e as amostras de treinamento são classificadas para o nó descendente

⁴A tarefa de indução é desenvolver uma regra de classificação que pode determinar a classe de qualquer objeto a partir dos valores de seus atributos (Quinlan, 1986).

adequado. Este processo é então repetido recursivamente utilizando as amostras associadas a cada nó descendente. Esse é um algoritmo de busca guloso, já que ele não retrocede para reconsiderar escolhas anteriores (Mitchell, 1997). O algoritmo para quando o subconjunto de amostras associado a um nó é da mesma classe ou quando não há relevância estatística para uma nova partição (Quinlan, 1986).

Para medir a impureza de uma partição, o ID3 usa o conceito de entropia, formalmente (Quinlan, 1986):

$$H(D) = -y_{\oplus} \log_2 y_{\oplus} - y_{\ominus} \log_2 y_{\ominus} \quad (2.51)$$

onde D é o conjunto de dados de treinamento com N amostras da forma apresentada em 2.29, y_{\oplus} e y_{\ominus} é a proporção de amostras positivas e negativas de D , respectivamente. É importante notar que a entropia é 0 quando todas as amostras de D pertencem à mesma classe, 1 quando D contém um número igual de amostras positivas e negativas e um valor entre 0 e 1 nos demais casos (Mitchell, 1997).

Dada a entropia como uma medida de impureza de um conjunto de amostras de treinamento, pode-se definir o teste estatístico, conhecido como ganho de informação, que mede a efetividade de um atributo na classificação dos dados de treinamento, formalmente (Quinlan, 1986):

$$IG(D, a_r) = H(D) - \sum_{v \in V(a_r)} \frac{|D_v|}{|D|} H(D_v) \quad (2.52)$$

onde $V(a_r)$ é o conjunto de todos os possíveis valores do atributo a_r , D_v é o subconjunto de D no qual o atributo a_r tem o valor v .

Vale enfatizar que o primeiro termo da equação dado em 2.52 é a entropia do conjunto original D e o segundo termo é o valor esperado da entropia depois que D é particionado usando o atributo a_r (Mitchell, 1997). Outro aspecto importante é que o algoritmo inicia com o conjunto D original, que é substituído por D_v à medida que a recursão se aprofunda.

Batizado de C4.5, Quinlan (1993) estendeu o ID3 para possibilitar o uso de atributos contínuos, atributos com dados ausentes e melhoria na eficiência computacional. Além disso, essa versão cuida de questões como quão profunda a árvore deve ser para evitar que os dados de treinamento sejam perfeitamente classificados, ou seja, quando o conjunto de treinamento é particionado até que cada subconjunto contenha apenas amostras de uma única classe. A estratégia adotada por Quinlan (1993) foi podar a árvore posteriormente à geração da árvore ajustada. Esse processo, apesar de ser mais lento que a proposta anterior do ID3, tornou o algoritmo mais confiável e com maior capacidade de generalização.

Capítulo 3

Experimentos Preliminares

Uma série de experimentos foram realizados com dois conjuntos de dados de pele de seres humanos. Os classificadores utilizados foram SVM, k -NN e a árvore de decisão *fuzzy* proposta por Cintra *et al.* (2013). Outro aspecto estudado é o efeito da escolha do modelo de cores no classificador. A formação dos conjuntos de dados é detalhada na seção 3.1 e os resultados dos experimentos podem ser vistos nas seções 3.2, 3.3 e 3.4.

3.1 Conjuntos de dados

Para os experimentos foram utilizadas duas bases de dados distintas. A primeira delas, denominada UCI neste trabalho, foi proposta por Bhatt e Dhall (2012) e obtida no repositório de aprendizagem de máquina da Universidade da Califórnia em Irvine (Lichman, 2013). A base é constituída por imagens de várias texturas de pele e não pele obtidas a partir de milhares de imagens arbitrárias de faces de diferentes idades, sexo e raças (Minear e Park, 2004; Phillips *et al.*, 1996).

A UCI contém 245.057 amostras, compostas por 3 atributos que constituem o vetor de entradas $x = [x_1, x_2, x_3]$, $x \in \mathbb{R}^d$, sendo d a dimensão do espaço, e que representam, respectivamente, os canais azul (B), verde (G) e vermelho (R) do modelo de cores RGB, além de uma quarta coluna que determina a classe a qual a amostra x pertence, denotada por y , sendo $y \in Y$ e $Y = \{+1, -1\}$.

Em outras palavras, cada amostra é um pixel RGB com um determinado rótulo. Como tem-se os dados rotulados com exemplos de qual é a saída correta para uma dada entrada, os experimentos subsequentes são categorizados como sendo de aprendizado supervisionado (Abu-Mostafa *et al.*, 2012).

A tabela 3.1 exemplifica um pequeno trecho da base de dados UCI. Vale ressaltar que do total das 245.057 amostras, 194.198 são de pixels não pele e 50.859 de pixels com diferentes tons de pele. Além disso, as imagens que foram utilizadas na extração do conjunto de dados não foi disponibilizada pelos autores.

Uma vez que a dimensão do espaço do problema é $d = 3$, é possível plotar os dados para melhor interpretação dos mesmos, conforme mostra a figura 3.1.

O segundo conjunto de dados utilizado nos experimentos foi proposto por Casati *et al.* (2013), intitulado pelos autores como banco de imagens de pele humana baseado no FERET e AR (SFA), cuja sigla também será utilizada neste trabalho. O SFA possui um conjunto de imagens de faces obtidas a partir de outros dois bancos de imagens coloridas: o FERET, criado por Phillips *et al.* (1996) e o AR, proposto por Martínez e Benavente (1998), tendo sido utilizadas 876 e 242 imagens de cada um, respectivamente. É importante ressaltar que as imagens do AR têm fundo branco e pequenas variações de cor da pele e, portanto, o ambiente é mais controlado que as imagens do FERET (Casati *et al.*, 2013). A figura 3.2 mostra um exemplo das 1.118 imagens do banco.

As amostras de pele e não pele foram geradas aleatoriamente considerando a máscara *ground truth*¹ de cada imagem, sendo três amostras de pele e cinco de não pele. Cada amostra é uma

¹ *Ground truth* é o termo utilizado para denotar uma imagem cujo objeto de interesse está devidamente segmentado e ressaltado, descartando os pixels remanescentes atribuindo-lhes cores uniformes.

B	G	R	Classe
74	85	123	1
207	215	255	1
74	82	122	1
202	211	255	1
54	72	125	1
...
166	164	116	-1
148	150	91	-1
29	26	5	-1
167	166	115	-1
180	177	133	-1

Tabela 3.1: Trecho com amostras da base de dados UCI. Cada uma das três primeiras colunas representam um canal do pixel do espaço de cores RGB e variam entre 0 e 255. A quarta coluna é a classe atribuída à amostra, que pode assumir +1 se for pele e -1, caso contrário. Originalmente, a classe que representa um pixel de não pele tinha valor 2, substituída por -1 para conformidade com os experimentos.

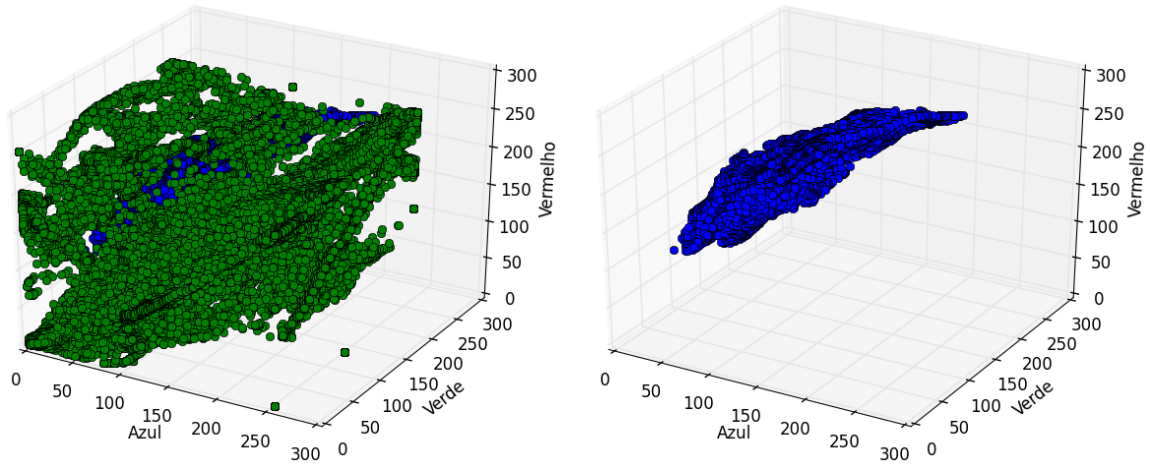


Figura 3.1: Visão 3-dimensional dos canais RGB do conjunto de dados UCI. Os pontos em azul são amostras de pele e os verdes de não pele. À esquerda tem-se todas as amostras do conjunto de dados; à direita apenas as amostras de pele. Fonte: proposto pelo autor.

janela de tamanho $n \times n$, sendo n ímpar, com um pixel central, a partir do qual outros tamanhos de amostra foram criados, que variam de 1×1 até 35×35 , como pode ser visto na figura 3.3 (Casati *et al.*, 2013).

A partir das amostras criadas no SFA, um conjunto de dados foi extraído gerando uma base de dados de pele e não pele similar ao UCI na tabela 3.1. O tamanho da janela utilizado foi 9×9 e o espaço de cores também foi o RGB. Portanto, o total de amostras empregado nos experimentos é 724.464, sendo 271.674 de pele e 452.790 de não pele. A classe atribuída à cada amostra tem o mesmo valor que no UCI, ou seja, +1 se for pele e -1, caso contrário. A figura 3.4 mostra a representação gráfica dos dados plotados em 3 dimensões.

3.2 Primeiro experimento

O primeiro experimento foi realizado com k -NN e SVM, disponíveis no pacote *scikit-learn* (Pedregosa *et al.*, 2011). O espaço de cores usado originalmente foi o RGB, como citado na descrição dos conjuntos de dados em 3.1. Em ambos os casos, a estratégia escolhida de validação cruzada foi a *10-fold*, que é uma escolha comum desta abordagem na prática (Abu-Mostafa *et al.*, 2012). Além disso, a técnica de tabela de busca do *scikit-learn* também foi utilizada com o objetivo de encontrar os parâmetros mais adequados para cada classificador.

A tabela de busca é usada no *scikit-learn* para encontrar os parâmetros ótimos de um classifica-



Figura 3.2: Exemplos de imagens do banco de faces SFA. Na parte superior tem-se a imagem original e na inferior o ground truth com os pixels de cor de pele anotados manualmente, sendo que a cor preta RGB = (0, 0, 0) foi atribuída a todos os pixels no fundo. Fonte: *Casati et al. (2013)*.

dor quando eles não podem ser aprendidos pelo estimador, tais como, *kernel* e γ na SVM ou número de vizinhos no *k*-NN (*Pedregosa et al., 2011*). As tabelas de parâmetros utilizadas no treinamento da SVM e *k*-NN podem ser vistas em 3.2 e 3.3. Os parâmetros de cada linha são combinados na tentativa de encontrar o estimador ótimo. Por exemplo, uma escolha no treinamento da SVM seria *kernel*=rbf, $C=100$, *gamma*= $1e-4$. Todas as combinações possíveis são exploradas, sendo que a melhor delas é retornada (*Pedregosa et al., 2011*).

<i>kernel</i>	<i>C</i>				<i>gamma</i>			<i>degree</i>	
rbf	1	10	100	1000	1e-3	1e-4	1e-5		
poly	1	10	100	1000		1e-4	1e-5	3	4
linear	1	10	100	1000					

Tabela 3.2: Tabela de busca dos parâmetros do estimador ótimo na SVM. A coluna *kernel* refere-se aos kernels usados no treinamento que são gaussiano, polinomial e linear, respectivamente. *C* é um parâmetro de regularização que diz à SVM a quantidade de erro admitido no treinamento. *Gamma* é um parâmetro usado somente nos kernels gaussiano e polinomial, conforme citado em 2.3.1. *Degree* é o grau do polinômio; usado somente no kernel polinomial (*Pedregosa et al., 2011*).

<i>n_neighbors</i>										<i>weights</i>		<i>algorithm</i>
3	5	9	15	25	50	100	200	400	800	distance	uniform	auto

Tabela 3.3: Tabela de busca dos parâmetros do estimador ótimo no *k*-NN. A coluna *n_neighbors* refere-se ao número de vizinhos considerado no treinamento. *Weights* é a função peso usada na predição, onde *uniform* indica que os pontos têm pesos iguais e *distance* indica que o inverso da distância é aplicado na classificação, conforme citado em 2.50. A terceira coluna indica qual algoritmo deve ser utilizado; *auto* significa que o algoritmo será decidido com base nos dados (*Pedregosa et al., 2011*).

Os resultados deste experimento podem ser vistos na tabela 3.4. Vale ressaltar que o treina-

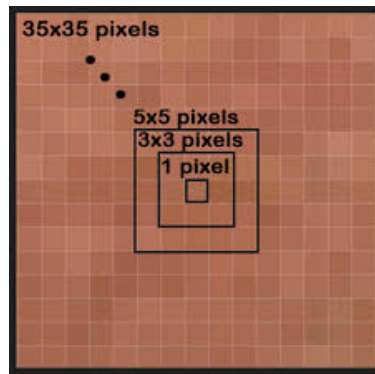


Figura 3.3: Estrutura das janelas que formam as amostras do SFA. No total, são 3.354 amostras de pele e outras 5.590 de não pele para cada tamanho de janela. Fonte: *Casati et al. (2013)*.

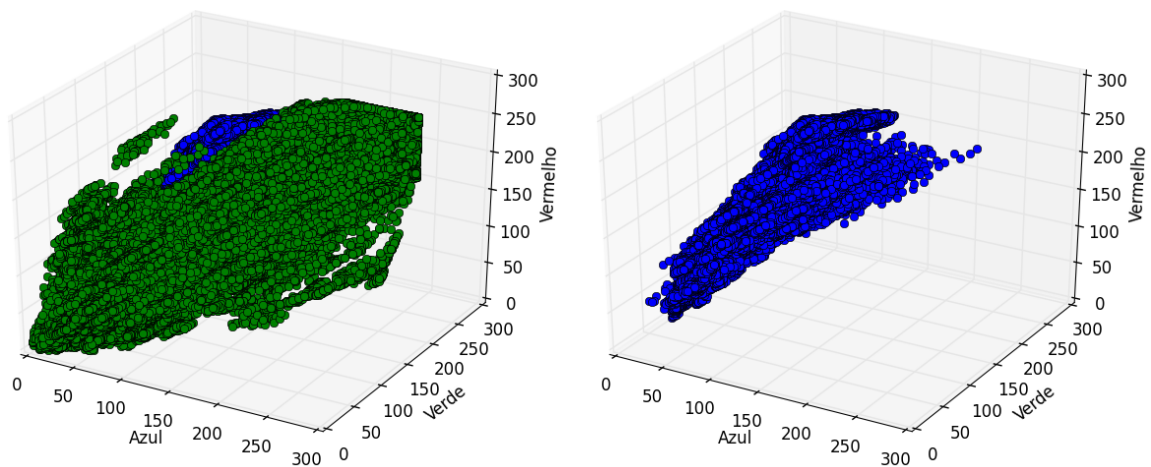


Figura 3.4: Visão 3-dimensional dos canais RGB do conjunto de dados SFA. Os pontos em azul são amostras de pele e os verdes de não pele. À esquerda tem-se todas as amostras do conjunto de dados gerado; à direita apenas as amostras de pele. Fonte: proposto pelo autor.

mento foi executado com 10 tarefas em paralelo em ambos os classificadores e 30% dos dados, aleatoriamente, foram separados para teste.

Base de dados	Classificador	Modelo de cores	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
UCI	<i>k</i> -NN	RGB	1,00	1,00	1,00
	SVM	RGB	0,98	0,98	0,98
SFA	<i>k</i> -NN	RGB	0,9672	0,9669	0,9670
	SVM	RGB	todo		

Tabela 3.4: Resultados dos experimentos com *k*-NN e SVM nos conjuntos de dados UCI e SFA. Os parâmetros ótimos do *k*-NN no treinamento com UCI foram $n_neighbors=3$, $weights=uniform$ e com SFA $n_neighbors=15$, $weights=uniform$. Os parâmetros ótimos da SVM no treinamento com UCI foram *completar*.

3.3 Segundo experimento

Este experimento, de maneira similar ao realizado na seção 3.2, também foi realizado com *k*-NN e SVM. Porém, o conjunto de dados SFA foi transformado para os espaços de cor HSV, Lab e YCbCr com o objetivo de compreender a influência nos resultados. Além disso, o atributo referente ao componente de luminância foi ignorado para que um teste somente com os componentes de crominância fosse realizado. Novamente, a estratégia escolhida de validação cruzada foi a 10-fold.

Os resultados podem ser vistos na tabela 3.5. Vale ressaltar que o treinamento foi executado com 10 tarefas em paralelo em ambos os classificadores e 30% dos dados, aleatoriamente, foram separados para teste.

Modelo de cores	Classificador	<i>Precision</i>	<i>Recall</i>	<i>F1-score</i>
RGB	<i>k</i> -NN	0,9672	0,9669	0,9670
	SVM	0,98	0,98	0,98
HSV	<i>k</i> -NN	0,9676	0,9673	0,9674
	SVM	todo		
HS	<i>k</i> -NN	0,9215	0,9194	0,9199
	SVM	todo		
Lab	<i>k</i> -NN	0,9671	0,9660	0,9670
	SVM	todo		
ab	<i>k</i> -NN	0,9444	0,9439	0,9440
	SVM	todo		
YCbCr	<i>k</i> -NN	0,9679	0,9677	0,9677
	SVM	todo		
CbCr	<i>k</i> -NN	0,9487	0,9482	0,9483
	SVM	todo		

Tabela 3.5: Resultados dos experimentos com *k*-NN e SVM no conjunto de dados SFA em espaços de cores distintos. As linhas com os modelos de cores HS, ab e CbCr indicam que o componente de luminância não foi utilizado no treinamento.

3.4 Terceiro experimento

O último experimento foi realizado com a árvore de decisão *fuzzy* proposta por Cintra *et al.* (2013). O conjunto de dados SFA transformado para os espaços de cor HSV, Lab e YCbCr também foi utilizado aqui. Além disso, há um parâmetro que define o nível de confiança que deve ser aplicado no processo de poda da árvore. Por padrão, esse parâmetro assume o valor de 25%, o qual não compromete o desempenho da árvore e evita o fenômeno de sobreajuste. Outra questão relevante neste experimento é a escolha do método que estima a quantidade de conjuntos *fuzzy* por atributo, conforme citado na seção 1.2. A tabela de busca dos parâmetros do estimador ótimo no *FuzzyDT* pode ser vista em 3.6.

Conjunto de dados	Nível de confiança				Método				# conjuntos <i>fuzzy</i>
RGB	10	15	20	25	infogain	wm	rf	fixed	2-9
HSV	10	15	20	25	infogain	wm	rf	fixed	2-9
Lab	10	15	20	25	infogain	wm	rf	fixed	2-9
YCbCr	10	15	20	25	infogain	wm	rf	fixed	2-9

Tabela 3.6: Tabela de busca dos parâmetros do estimador ótimo no *FuzzyDT*. A coluna conjunto de dados, conforme o nome sugere, refere-se aos conjuntos de dados oriundos do SFA em diferentes espaços de cores. A segunda coluna é o nível de confiança aplicado na poda da árvore. A coluna método indica o método de estimativa do número de conjuntos *fuzzy* para treinamento, seguida do número de conjuntos *fuzzy* que devem ser gerados por atributo. Este último parâmetro foi testado em um intervalo entre 2 e 9, conforme citado em Cintra *et al.* (2011) e é aplicável somente quando o método é *fixed*.

Para que os melhores parâmetros pudessem ser introduzidos no treinamento da árvore, foi desenvolvido um algoritmo que testa diversas combinações deles, avaliando a performance do modelo ajustado, de maneira similar à tabela de busca do *scikit-learn*. O algoritmo também tem a capacidade de executar várias instâncias em paralelo. A tabela 3.7 mostra os resultados com a taxa de erro em cada conjunto de dados para o respectivo espaço de cor, bem como os parâmetros que proporcionaram o melhor desempenho.

Os resultados mostram taxas de erros competitivas, do ponto de vista de classificação usando dados intervalares, em relação à modelagem com conjuntos de dados clássicos. Um ponto relevante é

que o método fixo de geração dos conjuntos *fuzzy* foi absolutamente unânime dentre todas as quatro técnicas propostas por Cintra *et al.* (2011), sendo o tamanho dos conjuntos a única peculiaridade, independente do espaço de cor. Além disso, a taxa de erro obtida foi a mesma para todos os níveis de confiança testados para poda da árvore. Isso também ocorreu nas demais combinações de parâmetros, ou seja, naquelas onde a taxa de erro obtida foi maior. Portanto, é possível afirmar que, de fato, o nível de confiança pode ser mantido em 25% para o treinamento do *FuzzyDT*.

Modelo de cores	Taxa de erro	Desvio padrão	Método	# conjuntos <i>fuzzy</i>
RGB	3,00	0,00	fixed	4
HSV	3,23	0,00	fixed	3
Lab	2,84	0,00	fixed	6
YCbCr	2,72	0,00	fixed	8

Tabela 3.7: Resultados dos experimentos com árvore de decisão *fuzzy* no conjunto de dados SFA em espaços de cores distintos.

Além das taxas de erros produzidas pelo *FuzzyDT*, outra saída são as regras induzidas pelo algoritmo que ficam disponíveis para a inferência de novas amostras. Por exemplo, para o conjunto de dados no modelo de cores HSV, cuja taxa de erro foi de 3,23%, foram geradas 9 regras:

1. Se saturação é baixa **então** classe é -1
2. Se saturação é media e intensidade é baixa **então** classe é -1
3. Se matiz é baixa e saturação é media e intensidade é media **então** classe é 1
4. Se matiz é media e saturação é media e intensidade é media **então** classe é -1
5. Se matiz é alta e saturação é media e intensidade é media **então** classe é -1
6. Se matiz é baixa e saturação é media e intensidade é alta **então** classe é 1
7. Se matiz é media e saturação é media e intensidade é alta **então** classe é -1
8. Se matiz é alta e saturação é media e intensidade é alta **então** classe é -1
9. Se saturação é alta **então** classe é -1

Como citado na tabela 3.7, no espaço de cores HSV, o algoritmo gerou um total de 3 conjuntos *fuzzy* por atributo, que são: baixa, media e alta. As regras são arranjadas de tal forma que o usuário pode facilmente interpretá-las por meio de conjunções. O *FuzzyDT* demonstrou robustez e bom desempenho quando comparado com classificadores clássicos, o que o credencia como um bom candidato na classificação de conjuntos de dados intervalares.

Capítulo 4

Plano de Trabalho

Os créditos em disciplinas necessários para o programa de mestrado em Ciência da Computação no IME-USP foram cumpridos de Fevereiro de 2015 até Julho de 2016. Em meados de 2015, iniciou-se a pesquisa bibliográfica para o desenvolvimento deste projeto. Durante esse período, até o presente momento, foram realizados os experimentos preliminares que estão incorporados neste texto para a etapa de qualificação. Após apresentação, as recomendações da comissão julgadora serão ponderadas e devem ser refletidas nas atividades subsequentes listadas na seção 4.1.

4.1 Atividades previstas

1. **Revisar leituras adicionais:** outras referências deverão ser adicionadas ao presente trabalho, principalmente, conteúdo relacionado a métodos de *kernel*, que servirão de embasamento para compreender e estudar a técnica de *kernels* em conjuntos *fuzzy* proposta por Guevara *et al.* (2014).
2. **Incorporar novos conjuntos de dados:** essa atividade consiste em analisar e integrar novos conjuntos de dados ao projeto para utilização nos experimentos, tanto de bancos de imagens de cor de pele, como outros conjuntos de imagens coloridas onde a classificação de dados intervalares possa ser aplicada.
3. **Investigar características passíveis de uso em classificadores de dados intervalares:** os classificadores treinados durante a etapa de experimentos preliminares consistiam apenas da informação de cor da imagem, independente do espaço de cores escolhido. Portanto, essa atividade visa compreender o papel de outros atributos, tais como textura e características locais, durante o treinamento e se a inclusão de tais atributos, de fato, pode acarretar em melhoria de performance.
4. **Desenvolver novas ferramentas:** novas ferramentas devem ser desenvolvidas para dar suporte aos experimentos subsequentes, tais como a tabela de busca implementada para otimização de parâmetros do *FuzzyDT*. Todo software criado por este projeto de pesquisa será disponibilizado para toda a comunidade científica em formato de código livre.
5. **Elaborar novos experimentos:** com base nas ferramentas desenvolvidas e conjuntos de dados estabelecidos, elaborar e executar novos experimentos.
6. **Analisar os resultados:** os resultados obtidos deverão ser avaliados e reportados no projeto de pesquisa e em publicações a serem realizadas. Eventualmente, correções nas ferramentas desenvolvidas e/ou nos experimentos também serão agregadas nesta etapa.
7. **Publicar resultados:** artigos científicos com os resultados obtidos devem ser publicados em revistas ou conferências sobre processamento de imagens, classificação ou outras áreas relacionadas com o contexto do projeto de pesquisa.

8. **Escrever a dissertação:** a dissertação deverá ser escrita assim que as demais atividades tiverem sido concluídas, principalmente em relação aos experimentos e análise de resultados, o que fornecerá dados consolidados para a conclusão do projeto de pesquisa.

4.2 Cronograma proposto

As atividades listadas na seção 4.1 serão realizadas de acordo com o cronograma disposto na tabela 4.1. As tarefas foram divididas dentro de um período de onze meses de execução e a previsão de defesa da dissertação é outubro de 2017.

Atividade	Meses 2016/2017										
	Dez	Jan	Fev	Mar	Abr	Mai	Jun	Jul	Ago	Set	Out
1	X	X	X								
2	X	X									
3		X	X								
4			X	X	X	X					
5					X	X	X				
6						X	X	X			
7							X	X			
8								X	X	X	X

Tabela 4.1: *Cronograma das atividades previstas*

Referências Bibliográficas

- Abu-Mostafa et al.(2012)** Yaser S. Abu-Mostafa, Malik Magdon-Ismael e Hsuan-Tien Lin. *Learning from data : a short course*. AMLBook.com. Citado na pág. 20, 21, 22, 29, 30
- Albiol et al.(2001)** Alberto Albiol, Luis Torres e Edward J. Delp. Optimum color spaces for skin detection. Em *International Conference on Image Processing*, páginas 122–124. Citado na pág. 3
- Barros e Bassanezi(2006)** Laécio Carvalho Barros e Rodney Carlos Bassanezi. *Tópicos de lógica fuzzy e biomatemática*. Grupo de Biomatemática, Instituto de Matemática, Estatística e Computação Científica (IMECC), Universidade Estadual de Campinas (UNICAMP). Citado na pág. 17
- Ben(2009)** R. G. Ben. CIE 1931 xy color space diagram. https://en.wikipedia.org/wiki/File:CIE1931xy_blank.svg, 2009. Último acesso em 12/10/2016. Citado na pág. 7
- Bergasa et al.(2000)** Luis Miguel Bergasa, Manuel Mazo, Alfredo Gardel, Miguel A. Sotelo e Luciano Boquete. Unsupervised and adaptive gaussian skin-color model. *Image and Vision Computing*, 18(12):987–1003. Citado na pág. 3
- Bhatt e Dhall(2012)** Rajen B. Bhatt e Abhinav Dhall. Skin segmentation dataset. <https://archive.ics.uci.edu/ml/datasets/Skin+Segmentation>, 2012. Último acesso em 05/06/2016. Citado na pág. 29
- Bhatt et al.(2009)** Rajen B. Bhatt, Abhinav Dhall, Gaurav Sharma e Santanu Chaudhury. Efficient skin region segmentation using low complexity fuzzy decision tree model. Em *2009 Annual IEEE India Conference*, páginas 1–4. IEEE. Citado na pág. 1
- Campbell(2000)** Colin Campbell. *An Introduction to Kernel Methods*, páginas 155–192. Springer Verlag, Berlin. Citado na pág. 23
- Casati et al.(2013)** João Paulo Brognoni Casati, Diego Rafael Moraes e Evandro Luis Linhari Rodrigues. SFA: A human skin image database based on FERET and AR facial images. Em *IX workshop de Visão Computacional*. Citado na pág. 29, 30, 31, 32
- Chaves-González et al.(2010)** Jose M. Chaves-González, Miguel A. Vega-Rodríguez, Juan A. Gómez-Pulido e Juan M. Sánchez-Pérez. Detecting skin in face recognition systems: A colour spaces study. *Digital Signal Processing*, 20(3):806–823. Citado na pág. 3
- Chen e Pham(2000)** Guanrong Chen e Trung Tat Pham. *Introduction to Fuzzy Sets, Fuzzy Logic, and Fuzzy Control Systems*. CRC Press. Citado na pág. 12
- Cintra et al.(2011)** Marcos E. Cintra, Maria C. Monard, Everton A. Cherman e Heloisa de Arruda Camargo. On the estimation of the number of fuzzy sets for fuzzy rule-based classification systems. Em *11th International Conference on Hybrid Intelligent Systems*, páginas 211–216. IEEE. Citado na pág. 2, 33, 34
- Cintra et al.(2013)** Marcos E. Cintra, Maria C. Monard e Heloisa de Arruda Camargo. A fuzzy decision tree algorithm based on C4.5. *Mathware & Soft Computing*, 20:56–62. Citado na pág. 1, 29, 33

- Cortes e Vapnik(1995)** Corinna Cortes e Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297. Citado na pág. 22, 23
- Dubois e Prade(1996)** Didier Dubois e Henri Prade. What are fuzzy rules and how to use them. *Fuzzy Sets and Systems*, 84(2):169–185. Citado na pág. 20
- Duda et al.(2012)** Richard O. Duda, Peter E. Hart e David G. Stork. *Pattern Classification*. John Wiley & Sons, segunda edição. Citado na pág. 21, 22, 24, 25, 26
- Gevers et al.(2012)** Theo Gevers, Arjan Gijsenij, Joost van de Weijer e Jan-Mark Geusebroek. *Color in Computer Vision: Fundamentals and Applications*. Wiley. Citado na pág. 5, 7, 8
- Gonzalez e Woods(2002)** Rafael C. Gonzalez e Richard E. Woods. *Digital Image Processing*. Prentice Hall, segunda edição. Citado na pág. 3, 5, 6, 8, 9
- Guevara et al.(2014)** Jorge Guevara, Roberto Hirata Jr e Stephane Canu. Positive definite kernel functions on fuzzy sets. Em *IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, páginas 439–446. doi: 10.1109/FUZZ-IEEE.2014.6891628. Último acesso em 14/09/2016. Citado na pág. 2, 35
- Jayaram et al.(2004)** Sriram Jayaram, Stephen Schmugge, Min C. Shin e Leonid V. Tsap. Effect of colorspace transformation, the illuminance component, and color modeling on skin detection. Em *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, páginas 813–818. IEEE. Citado na pág. 3
- Jones e Rehg(2002)** Michael J. Jones e James M. Rehg. Statistical color models with application to skin detection. *International Journal of Computer Vision*, 46(1):81–96. Citado na pág. 2
- Kakumanu et al.(2007)** Praveen Kakumanu, Sokratis Makrogiannis e Nikolaos Bourbakis. A survey of skin-color modeling and detection methods. *Pattern recognition*, 40(3):1106–1122. Citado na pág. 2, 3
- Kaur e Kranthi(2012)** Amanpreet Kaur e B. V. Kranthi. Comparison between YCbCr color space and CIELab color space for skin color segmentation. *International Journal of Applied Information Systems*, 3(4):30–33. Citado na pág. 3
- Kawulok et al.(2013)** Michal Kawulok, Jolanta Kawulok, Jakub Nalepa e Maciej Papiez. Skin detection using spatial analysis with adaptive seed. Em *2013 IEEE International Conference on Image Processing*, páginas 3720–3724. IEEE. Citado na pág. 3
- Klir e Yuan(1995)** George J. Klir e Bo Yuan. *Fuzzy Sets and Fuzzy Logic: Theory and Applications*. Prentice Hall. Citado na pág. 13, 14, 15, 17, 19
- Kononenko(1994)** Igor Kononenko. Estimating attributes: analysis and extensions of RELIEF. Em *European conference on machine learning*, páginas 171–182. Springer. Citado na pág. 2
- Kovac et al.(2003)** Jure Kovac, Peter Peer e Franc Solina. *Human skin color clustering for face detection*, volume 2. IEEE. Citado na pág. 2, 3
- Kumar e Malhotra(2015)** Amit Kumar e Shivani Malhotra. Performance analysis of color space for optimum skin color detection. Em *2015 Fifth International Conference on Communication Systems and Network Technologies*, páginas 554–558. IEEE. Citado na pág. 4
- Lichman(2013)** M. Lichman. UCI machine learning repository. <http://archive.ics.uci.edu/ml>, 2013. Citado na pág. 2, 29
- Lorena e Carvalho(2003)** Ana Carolina Lorena e André C. P. L. F. Carvalho. Introdução às Máquinas de Vetores Suporte. Relatório Técnico 192, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, Brasil. Citado na pág. 22, 23, 24, 25

- Mahmoodi e Sayedi(2016)** Mohammad Reza Mahmoodi e Sayed Masoud Sayedi. A comprehensive survey on human skin detection. *International Journal of Image, Graphics and Signal Processing*, 8(5):1–35. Último acesso em 23/04/2016. Citado na pág. 2, 3
- Martínez e Benavente(1998)** Aleix Martínez e Robert Benavente. The AR face database. Relatório técnico, Purdue University. Citado na pág. 3, 29
- Miner e Park(2004)** Meredith Miner e Denise Park. Productive aging lab face database. <https://pal.utdallas.edu/facedb/>, 2004. Citado na pág. 29
- Mitchell(1997)** Tom M. Mitchell. *Machine Learning*. McGraw-Hill Education. Citado na pág. 20, 22, 25, 26, 27
- Naji et al.(2012)** Sinan A. Naji, Roziati Zainuddin e Hamid A. Jalab. Skin segmentation based on multi pixel color clustering models. *Digital Signal Processing*, 22(6):933–940. Citado na pág. 3
- Pedregosa et al.(2011)** F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot e E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830. Citado na pág. 25, 30, 31
- Pedrini e Schwartz(2008)** Hélio Pedrini e William Robson Schwartz. *Análise de imagens digitais: princípios, algoritmos e aplicações*. Thomson Learning, São Paulo. Citado na pág. 9, 10, 11, 12
- Pedrycz e Gomide(1998)** Witold Pedrycz e Fernando Gomide. *An Introduction to Fuzzy Sets: Analysis and Design*. A Bradford Book. Citado na pág. 1, 12, 19
- Phillips et al.(1996)** P. Jonathon Phillips, Harry Wechsler, Jeffrey Huang e Patrick J. Rauss. The facial recognition technology (FERET) database. <https://www.nist.gov/programs-projects/face-recognition-technology-feret>, 1996. Último acesso em 23/06/2016. Citado na pág. 29
- Plataniotis e Venetsanopoulos(2000)** Konstantinos N. Plataniotis e Anastasios N. Venetsanopoulos. *Color Image Processing and Applications*. Springer, primeira edição. Citado na pág. 5, 6, 8, 10
- Quinlan(1986)** John Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106. Citado na pág. 1, 26, 27
- Quinlan(1993)** John Ross Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann. Citado na pág. 1, 27
- Rus(2007)** Jacob Rus. The Munsell color system. <https://commons.wikimedia.org/wiki/File:Munsell-system.svg>, 2007. Último acesso em 12/10/2016. Citado na pág. 6
- Rus(2008)** Jacob Rus. The "primary" and "secondary" colors in a four-color print process. <https://en.wikipedia.org/wiki/File:SubtractiveColor.svg>, 2008. Último acesso em 12/10/2016. Citado na pág. 9
- Shaik et al.(2015)** Khamar Basha Shaik, Ganesan P., V. Kalist, B. S. Sathish e J. Merlin Mary Jenitha. Comparative study of skin color detection and segmentation in HSV and YCbCr color space. *Procedia Computer Science*, 57:41–48. Citado na pág. 4
- Shawe-Taylor e Cristianini(2004)** John Shawe-Taylor e Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press. Citado na pág. 24
- Sumathi e Paneerselvam(2010)** Sai Sumathi e Surekha Paneerselvam. *Computational Intelligence Paradigms: Theory & Applications Using MATLAB*. CRC Press. Citado na pág. 17

- Tan et al.(2012)** Wei Ren Tan, Chee Seng Chan, Pratheepan Yogarajah e Joan Condell. A fusion approach for efficient human skin detection. *IEEE Transactions on Industrial Informatics*, 8(1): 138–147. Citado na pág. [3](#)
- Thole et al.(1979)** U. Thole, H.-J. Zimmermann e P. Zysno. On the suitability of minimum and product operators for the intersection of fuzzy sets. *Fuzzy sets and Systems*, 2(2):167–180. Citado na pág. [15](#)
- Umano et al.(1994)** Motohide Umano, Hirotaka Okamoto, Itsuo Hatono, Hiroyuki Tamura, Fumio Kawachi, Sukehisa Umedzu e Junichi Kinoshita. Fuzzy decision trees by fuzzy ID3 algorithm and its application to diagnosis systems. Em *Proceedings of the Third IEEE Conference on Fuzzy Systems. IEEE World Congress on Computational Intelligence.*, páginas 2113–2118. IEEE. Citado na pág. [1](#)
- Vapnik(2013)** Vladimir N. Vapnik. *The Nature of Statistical Learning Theory*. Springer Science & Business Media. Citado na pág. [22](#)
- Vezhnevets et al.(2003)** Vladimir Vezhnevets, Vassili Sazonov e Alla Andreeva. A survey on pixel-based skin color detection techniques. Em *In Proc. GRAPHICON-2003*, páginas 85–92. Citado na pág. [2](#), [3](#), [7](#)
- Vrusias(2006)** Bogdan L. Vrusias. Fuzzy logic 2. http://www.computing.surrey.ac.uk/teaching/2006-07/cs364/lecturenotes/week4_Fuzzy_Logic_2.ppt, 2006. Último acesso em 10/08/2016. Citado na pág. [15](#), [16](#)
- Wang(2003)** Li-Xin Wang. The WM method completed: a flexible fuzzy system approach to data mining. *IEEE Transactions on fuzzy systems*, 11(6):768–782. Citado na pág. [2](#)
- Yen e Langari(1999)** John Yen e Reza Langari. *Fuzzy Logic: Intelligence, Control, and Information*. Prentice Hall Upper Saddle River. Citado na pág. [17](#)
- Yogarajah et al.(2011)** Pratheepan Yogarajah, Joan Condell, Kevin Curran, Paul McKeivitt e Abbas Cheddad. A dynamic threshold approach for skin tone detection in colour images. *International Journal of Biometrics*, 4(1):38–55. Citado na pág. [3](#)
- Zadeh(1965)** Lotfi A. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353. Citado na pág. [13](#)
- Zadeh(1975)** Lotfi A. Zadeh. The concept of a linguistic variable and its application to approximate reasoning—i. *Information Sciences*, 8(3):199–249. Citado na pág. [19](#)
- Zimmermann(2001)** H.-J. Zimmermann. *Fuzzy Set Theory and Its Applications*. Springer, quarta edição. Citado na pág. [16](#), [19](#)