

UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

**Deteção de Manchas de Óleo em Imagens
SAR Através da Combinação de
Características e de Classificadores**

Geraldo Luis Bezerra Ramalho

FORTALEZA – CEARÁ
DEZEMBRO/2007



UNIVERSIDADE FEDERAL DO CEARÁ
DEPARTAMENTO DE ENGENHARIA DE TELEINFORMÁTICA
PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA DE TELEINFORMÁTICA

Detecção de Manchas de Óleo em Imagens SAR Através da Combinação de Características e de Classificadores

Autor

Geraldo Luis Bezerra Ramalho

Orientador

Profa. Dra. Fátima N. Sombra de Medeiros

*Dissertação de Mestrado apresentada
à Coordenação do Curso de
Pós-Graduação em Engenharia de
Teleinformática da Universidade
Federal do Ceará como parte dos
requisitos para a obtenção do grau
de Mestre em Engenharia de
Teleinformática.*

FORTALEZA – CEARÁ

DEZEMBRO/2007

GERALDO LUIS BEZERRA RAMALHO

**Deteção de Manchas de Óleo em Imagens SAR Através da Combinação
de Características e de Classificadores**

Esta Dissertação foi julgada adequada para a obtenção do título de Mestre em Engenharia de Teleinformática e aprovada em sua forma final pelo programa de Pós Graduação em Engenharia de Teleinformática da Universidade Federal do Ceará.

Geraldo Luis Bezerra Ramalho

Banca Examinadora:

Profa. Dra. Fátima N. Sombra de Medeiros
Orientador

Prof. Dr. Guilherme de Alencar Barreto
Universidade Federal do Ceará

Prof. Dr. Francisco Nivando Bezerra
Universidade de Fortaleza

Fortaleza, 14 de Dezembro de 2007

Resumo

O mapeamento da poluição de óleo no mar utilizando imagens de Radar de Abertura Sintética (SAR, do inglês *Synthetic Aperture Radar*) é uma importante área de interesse na área da vigilância ambiental. Pode-se utilizar imagens SAR para extrair características através de diferentes métodos com o objetivo de prever através de Redes Neurais Artificiais (RNAs) se uma região específica contém ou não uma mancha de óleo. O principal problema dessa abordagem é a ocorrência de excessivos alarmes falsos decorrentes de erros de classificação. Manchas de óleo são eventos raros e a pequena disponibilidade de imagens contendo manchas é um fator limitante do desempenho dos classificadores.

Este trabalho propõe a utilização de múltiplos conjuntos de características e métodos de combinação de classificadores para minimizar o número de alarmes falsos a fim de possibilitar a redução de custos operacionais de sistemas automáticos de detecção de manchas de óleo. As imagens SAR utilizadas neste trabalho não estão limitadas a um único sistema de imageamento e diferentes conjuntos de características baseados na geometria e textura das manchas foram testados. Os desempenhos de generalização de métodos de combinação de classificadores, como *boosting* e *bagging*, foram comparados com aqueles obtidos com classificadores individuais, como Perceptron Multi-Camadas (MLP, do inglês *Multi-Layer Perceptron*) e Máquina de Vetor de Suporte (SVM, do inglês *Support Vector Machine*). Os resultados experimentais sugerem que a caracterização das manchas de óleo pode ser significativamente melhorada através do uso do *boosting*, mesmo quando poucas imagens amostrais estão disponíveis e utilizando conjuntos de características de grande dimensionalidade.

Palavras-chave: redes neurais, comitê de classificadores, imagens SAR, manchas de óleo

Abstract

Mapping ocean oil pollution by using Synthetic Aperture Radar (SAR) images is an important area of interest for environmental surveillance. One can make use of the SAR images to extract features by using different methods in order to predict if a specific region contains an oil spill or not using Artificial Neural Networks (ANN). A major problem in this approach is the number of false alarms due to misclassification. Oil spills are rare events and the number of available images containing spills is statistically small which is a limitation for the classifier performance.

This work proposes the use of multiple feature sets and classifier combining methods to minimize the number of false alarms and thus, reduce the operational costs of automatic oil spill detection systems. The SAR images used in this work are not limited to a specific imaging system and different feature sets based on geometry and texture of the spills were tested. The generalization performances of classifier combination methods as boosting and bagging were compared with those resulting from single classifiers as Multilayer Perceptron (MLP) and Support Vector Machines (SVM). The experimental results suggest that oil spill characterization can be significantly improved using boosting even when few image samples are available and the feature sets have high dimensionality.

Keywords: neural networks, classifier committee, SAR images, oil spill

Agradecimentos

A todos os professores, colegas e amigos que contribuíram para que este trabalho pudesse ser concluído com qualidade.

Ao colega Cláudio Sá pela valiosa contribuição com suas sugestões e críticas.

Ao Professor Roberto Oscar Brasil pelo seu grande incentivo.

Em especial à Professora Fátima, pela sua paciência, pela confiança depositada em mim e pelo seu precioso apoio nos momentos mais complicados.

À CAPES pelo suporte financeiro.

"Um desdém pela prática varreu o mundo antigo.
Platão instigou os astrônomos a pensarem sobre os céus,
mas a não perder tempo observando-os...
A mácula antiempírica dos pitagóricos sobreviveu até hoje.
Por que? De onde surgiu esta aversão pela experiência?"

Carl Sagan, *Cosmos*, 1980

Dedico este trabalho à minha mãe, por sua garra; à minha esposa, pelo
companheirismo; e ao meus sogros pelo inestimável apoio.

Sumário

Lista de Figuras	viii
Lista de Tabelas	xi
Lista de Símbolos	xii
Lista de Siglas	xiv
1 Introdução	1
1.1 Motivação	4
1.2 Objetivo	7
1.3 Contribuições	9
1.4 Organização da Dissertação	10
2 Métodos de Detecção de Manchas de Óleo em Imagens SAR	11
2.1 Introdução	11
2.2 Estratégias Comumente Adotadas na Detecção de Manchas	12
2.2.1 Segmentação	16
2.2.2 Classificação pelas Propriedades Geométricas	18
2.2.3 Classificação pelas Propriedades de Textura	19
2.3 Conclusão	20
3 Detecção de Manchas Utilizando Classificadores Individuais	22
3.1 Introdução	22
3.2 Classificadores Paramétricos	24
3.2.1 Máxima Verossimilhança	25
3.2.2 Naïve Bayes	26
3.3 Classificadores Baseados em Distância	27
3.3.1 KNN (k-Nearest Neighbor)	27
3.4 Redes Neurais Artificiais	28
3.4.1 MLP	29
3.4.2 SVM (Support Vector Machine)	32
3.5 Combinação de Características na Detecção de Manchas	35

3.5.1	A Maldição da Dimensionalidade	36
3.6	Avaliação de Desempenho de Classificadores	37
3.7	Conclusão	39
4	Melhoria de Desempenho na Detecção de Manchas de Óleo Através da Combinação de Classificadores	41
4.1	Introdução	41
4.2	Bagging	43
4.3	Boosting	44
4.4	Logitboost	49
4.5	O AdaBoost Adaptado para a Combinação de Características	51
4.6	Conclusão	53
5	Resultados e Discussões	55
5.1	Conjuntos de Características	56
5.1.1	Limitações dos Conjuntos Individuais	57
5.1.2	Características Combinadas	57
5.2	Parametrização dos Classificadores Utilizados nos Experimentos	58
5.3	Resultados dos Classificadores Individuais	61
5.4	Resultados Finais	63
5.4.1	Combinação de Características	63
5.4.2	Combinação de Classificadores	66
5.5	Conclusão	70
6	Conclusões e Trabalhos Futuros	71
6.1	Contribuições Finais	72
6.2	Perspectiva de Trabalhos Futuros	73
	Apêndice A Imageamento SAR	77
	Apêndice B Imagens Utilizadas nos Experimentos	81
	Apêndice C Métodos de Extração de Características Utilizados	87
	Referências Bibliográficas	100

Lista de Figuras

1.1	Exemplo de imagens SAR obtidas de satélites exibindo manchas (a) na costa do Brasil, (b) no Mar do Norte, (c) na costa da Espanha e (d) no sul da China. As setas indicam as manchas mais evidentes.	3
1.2	Modelo da metodologia utilizada. As três áreas de abrangência da pesquisa aparecem destacadas em negrito.	8
2.1	A superfície do mar na presença de óleo torna-se mais viscosa e atenua o sinal retro-propagado.	12
2.2	Exemplo de um sistema de detecção de manchas de óleo (MARTINEZ; MORENO, 1996)	14
2.3	Exemplo de detecção semi-automática de manchas baseada em segmentação (MERCIER; GIRARD-ARDHUIN, 2005b). (a) Imagem com a região de interesse utilizada para definir a função de decisão. (b) Imagem segmentada.	17
2.4	Exemplo de uma (a) imagem SAR (b) segmentada pelo processo de crescimento de regiões (ARAÚJO, 2004).	17
2.5	Exemplos de imagens SAR contendo manchas (a,c) e suas correspondentes segmentadas (b,d) (TOPOUZELIS et al., 2004; MERCIER; GIRARD-ARDHUIN, 2005a).	18
3.1	Classificadores comumente encontrados na literatura.	24
3.2	Funções densidade de probabilidade das classes \mathcal{C}_1 e \mathcal{C}_2 . A seta indica o limiar de decisão de máxima verossimilhança. Adaptado de Haykin (2001).	26
3.3	Algoritmo KNN para uma vizinhança genérica.	28
3.4	Modelo matemático de um neurônio (perceptron).	29
3.5	Modelo de um MLP com uma camada escondida.	30
3.6	Funções de ativação típicas.	31
3.7	Modelo mostrando um hiperplano ótimo formado com os vetores de suporte de duas classes linearmente separáveis.	33
3.8	Modelo de um classificador com função núcleo representada por φ	34
3.9	Conjuntos de características e as possíveis combinações.	36
3.10	Representação gráfica dos métodos de validação cruzada mais comuns.	38

4.1	Modelo de combinação de classificadores. Os dados de entrada são submetidos a diversos classificadores. Um módulo combinador utiliza a predição individual dos especialistas para gerar uma única saída combinada, tornando a predição final mais apurada.	42
4.2	Modelo de combinação <i>bagging</i> . O combinador utiliza um método de votação de maioria.	43
4.3	Algoritmo <i>bagging</i>	44
4.4	Superfície discriminante do comitê (c) gerada pela combinação de duas redes neurais (a,b) utilizando o método <i>boosting</i>	46
4.5	Modelo de combinação <i>boosting</i> . O combinador utiliza o método de votação ponderada, baseada na taxa de erro de classificação dos classificadores intermediários.	47
4.6	Algoritmo do AdaBoost discreto binário (FREUND; SCHAPIRE, 1999).	48
4.7	Algoritmo LogitBoost binário.	50
4.8	Modelo de combinação <i>Variant AdaBoost</i> . Este método emprega uma fase intermediária de combinação para cada uma das T etapas. Os diferentes conjuntos de características são apresentados individualmente aos B classificadores. No restante do processo ele é similar ao AdaBoost.	52
4.9	Algoritmo do <i>Variant AdaBoost</i> (YIN; LIU; HAN, 2005).	53
5.1	Comparação de desempenho dos classificadores individuais sobre cada conjunto de características geométricas e texturais (espaciais e espectrais).	62
5.2	Variâncias do erro de classificação dos classificadores individuais sobre cada conjunto de características geométricas e texturais (espaciais e espectrais).	62
5.3	Comparação de desempenho dos classificadores individuais sobre os conjuntos de características combinadas.	64
5.4	Comparação do desempenho de classificação após a redução de espaço utilizando PCA.	65
5.5	Comparação de desempenho dos classificadores sobre os conjuntos de características combinados.	66
5.6	Comparação de desempenho entre os métodos de combinação de classificadores sobre os conjuntos de características combinados.	67
A.1	Modelo do sistema de imageamento SAR.	78
A.2	Histograma de intensidade de pixels de uma imagem SAR original (a,c) e após a filtragem do ruído (b,d) por um filtro não-linear adaptativo.	80
B.1	Exemplo de imagens SAR de diferentes resoluções e números de visadas. As manchas (a) a (j) foram causadas por derramamento de óleo e as restantes (k) a (t) são manchas similares.	83
B.2	Exemplo de imagem SAR do sistema ERS-2 contendo manchas delimitadas por ROIs.	84

B.3	Resultado da filtragem do ruído <i>speckle</i> utilizando a ondaleta <i>à trous</i> (ARAÚJO, 2004). (a) Imagem original. (b) Imagem filtrada. (c) e (d) Respective histogramas.	84
B.4	Máscaras utilizadas na extração de características. As máscaras marcadas com "*" foram criadas manualmente. As demais foram segmentadas utilizando o procedimento de Araújo (2004).	85
B.5	Resultado da multiplicação da imagem da mancha pela sua máscara.	86
C.1	(a) Mancha de óleo; (b) Histograma da imagem.	89
C.2	(a) Detalhe de uma mancha de óleo e (b) sua respectiva GLCM.	90
C.3	Função base da ondaleta Haar.	93
C.4	(a) Mancha de óleo segmentada. (b) Imagem vetorizada da mancha. (c) Esquema de decomposições. (d) Coeficientes de detalhe obtidos em 7 decomposições sucessivas.	94

Lista de Tabelas

2.1	Características gerais de diferentes métodos de detecção de manchas. .	15
5.1	Conjuntos de características combinadas.	58
5.2	Matriz de confusão obtida com o melhor desempenho de classificação das manchas.	63
5.3	Taxas de classificação e variâncias dos conjuntos transformados por PCA.	65
5.4	Comparação entre as taxas de acerto e variâncias do erro de classificação.	68
5.5	Matriz de confusão do <i>Variant AdaBoost</i>	69
A.1	Satélites equipados com sistema de imageamento SAR.	78

Lista de Símbolos

Reconhecimento de Padrões

\mathbf{x}	Vetor de entrada, padrão de entrada, vetor de características ou vetor de atributos
r	Rótulo que identifica a classe do vetor
\mathbf{w}	Vetor de pesos sinápticos
w	Peso sináptico
d	Peso da distribuição dos vetores de entrada
d	Distância entre vetores
y	Saída do neurônio ou classificador
\mathbf{y}	Vetor de saída do classificador
φ	Função de ativação do neurônio
α	Taxa de aprendizagem
η	Taxa de momento
δ	Erro ou gradiente (diferença entre o valor desejado e a saída do neurônio)
N	Número de amostras
J	Número de classes
M	Dimensão do espaço de características
B	Número de classificadores base
h	Hipótese ou função base ou saída do classificador base
\mathcal{H}	Hipótese final ou função de combinação dos classificadores base (saída combinada dos classificadores)

Estatística

\mathcal{C}	Classe de amostras
$p(\mathbf{x})$	Função densidade de probabilidade

$\hat{p}(\mathbf{x})$	Função densidade de probabilidade estimada
$P(\mathcal{C}_j)$	Probabilidade <i>a priori</i> da classe \mathcal{C}_j
$P(\mathcal{C}_j \mathbf{x})$	Probabilidade condicional <i>a posteriori</i> da classe \mathcal{C}_j , dado o vetor de características \mathbf{x}
λ	Autovalor
$E[.]$	Operador estatístico da esperança matemática ou valor esperado
\mathcal{D}	Conjunto de amostras
\mathcal{SD}	Subconjunto de amostras
μ	Média amostral
$\boldsymbol{\mu}$	Vetor média
σ	Desvio padrão
σ^2	Variância
$\boldsymbol{\Sigma}$	Matriz de covariância
$\boldsymbol{\Sigma}^{-1}$	Matriz de covariância inversa

Wavelets

Ψ	Função ondaleta mãe (<i>mother wavelet</i>)
$\Psi_{a,b}$	Função ondaleta dilatada de a e transladada de b

Identificadores

C_g	Conjunto de características geométricas
C_{te}	Conjunto de características espaciais de textura
C_{ts}	Conjunto de características espectrais de textura
C_{g+te}	Conjunto de características geométricas combinadas com características espaciais de textura
C_{g+ts}	Conjunto de características geométricas combinadas com características espectrais de textura
$C_{g+te+ts}$	Conjunto de características geométricas combinadas com características espaciais e espectrais de textura

Lista de Siglas

EQM	Erro Quadrático Médio
GLCM	Matriz de Coocorrência de Níveis de Cinza (<i>Gray Level Cooccurrence Matrix</i>)
KNN	k-Vizinhos mais próximos (<i>k-Nearest Neighbor</i>)
LMS	Mínimo Quadrado Médio (<i>Least Mean Square</i>)
MLP	Perceptron Multi-Camadas (<i>Multilayer Perceptron</i>)
NB	Naïve Bayes
PCA	Análise de Componentes Principais (<i>Principal Component Analysis</i>)
PDF	Função Densidade de Probabilidade (<i>Probability Density Function</i>)
RNA	Redes Neurais Artificiais
ROI	Região de Interesse (<i>Region of Interest</i>)
SAR	Radar de Abertura Sintética (<i>Synthetic Aperture Radar</i>)
SOM	Mapas Auto-Organizáveis (<i>Self-Organizing Maps</i>)
SVA	Sistema de Visão Artificial
SVM	Máquina de Vetor de Suporte (<i>Support Vector Machine</i>)

Capítulo 1

Introdução

O derramamento de óleo no mar causa grande transtorno ao meio-ambiente e, portanto, é uma permanente fonte de preocupação dos órgãos de vigilância ambiental, tanto no meio político quanto no meio científico. Embora não seja conhecido o impacto do não monitoramento de manchas de óleo, o impacto ambiental causado pelos derramamentos é facilmente observado pelos danos causados à ecologia costeira (BREKKE; SOLBERG, 2005).

Um derramamento de óleo pode ser proveniente de vazamentos causados, dentre outros, por acidentes com navios petroleiros e defeitos em tubulações de sistemas de transporte submarino. O derramamento também pode ter caráter criminoso incluindo a lavagem de porões de navios cargueiros e o descarte de óleo queimado. De 1974 até 2006 ocorreram 9328 derramamentos de óleo, a maioria resultante de problemas operacionais (ITOPF, 2006). Uma rápida ação é necessária no sentido de minimizar o efeito do derramamento e para isso são mobilizados equipamentos e pessoas numa operação que envolve um alto custo. Planejar corretamente essa operação requer informações precisas sobre o evento. Entretanto, se essas informações estiverem sujeitas a interpretações erradas, toda a operação terá sido em vão. Alarmes falsos acarretam custos desnecessários, daí a necessidade de se projetar sistemas que sejam capazes de generalizar muito bem as respostas, quando submetidos a dados de entrada desconhecidos.

Sistemas de sensoriamento remoto têm sido integrados a ferramentas de detecção de manchas de óleo para otimizar a capacidade dos sistemas de vigilância no auxílio à decisão e na formulação do plano de contingência na ocorrência de

um derramamento (ASSILZADEH; MANSOR, 2001). Imagens geradas através de sistemas de sensoriamento remoto, por exemplo sistemas de Radar de Abertura Sintética (SAR, do inglês *Synthetic Aperture Radar*), embarcados em aviões ou em satélites, são utilizadas para a geração de alarmes em resposta à detecção de manchas decorrentes de derramamento de óleo no mar.

Imagens SAR adaptam-se muito bem a esse tipo de aplicação, pois o sistema de imageamento SAR não sofre grande influência das condições atmosféricas ou da luz solar. A viscosidade do óleo proveniente de um derramamento produz um efeito atenuador sobre as ondas capilares na superfície do mar e resulta em uma diminuição do retroespalhamento do sinal do radar. Este efeito aparece na imagem SAR como regiões escuras formadas de pixels de baixa intensidade (MERCIER; GIRARD-ARDHUIN, 2005a; ARAÚJO, 2004; TOPOUZELIS et al., 2004).

Segundo Topouzelis et al. (2004), cardumes, algas, fitoplânctons e ausência de ventos são eventos naturais que produzem manchas similares as quais podem ser confundidas com as manchas de óleo. Aliado a este fato, o ruído *speckle*, intrínseco ao sistema de imageamento SAR, torna ainda mais complexo o projeto de um sistema automático de detecção de manchas de óleo. O ruído dificulta a segmentação da mancha através de métodos de detecção de bordas e, conseqüentemente, o desenvolvimento de um método automático para a sua localização na imagem (ARAÚJO, 2004).

A Figura 1.1 mostra imagens SAR contendo manchas de óleo e manchas similares de diversos formatos e tamanhos. A mancha maior da Figura 1.1 (a) tem uma forma bem definida, sendo claramente um rastro de derramamento de óleo causado por um navio em movimento, o qual aparece como um ponto branco na extremidade da mancha. Este é um tipo de mancha relativamente fácil de ser detectado, porque está no seu estágio inicial, antes de haver um espalhamento. Na mesma figura existe uma mancha menor suspeita, porém sem nenhuma evidência da sua natureza. No detalhe da Figura 1.1 (b) percebe-se a presença de ruído sob forma de uma granulação na imagem. No centro aparece uma mancha de óleo de origem similar à da Figura 1.1 (a), mas cuja aparência foi modificada pela ação dos ventos e marés. Na mesma imagem, a mancha superior tem formato similar, porém assemelha-se a uma mancha natural. A Figura 1.1 (c) mostra como um derramamento de óleo a centenas de quilômetros da costa pode facilmente atingir a costa pela ação dos ventos e das marés. As manchas escuras da Figura 1.1 (d) são causadas por derramamentos

de navios. Em poucas horas uma mancha reconhecida como um derramamento de óleo como a da Figura 1.1 (a) pode mudar completamente sua geometria como nas Figuras 1.1 (b) e (c). Isto pode provocar um erro de avaliação do especialista ao analisar a imagem em busca de manchas de óleo. No pior caso, a mancha pode ser identificada como sendo uma mancha similar, ou seja, a ocorrência de um falso negativo. Deixar de executar as ações necessárias poderia resultar num grande custo ambiental. Noutra circunstância, o especialista poderá identificar de forma errada uma mancha natural como sendo uma mancha de derramamento de óleo, ou seja, a ocorrência de um falso positivo. Neste caso, recursos materiais e humanos serão alocados para conter uma mancha de óleo que não existe.

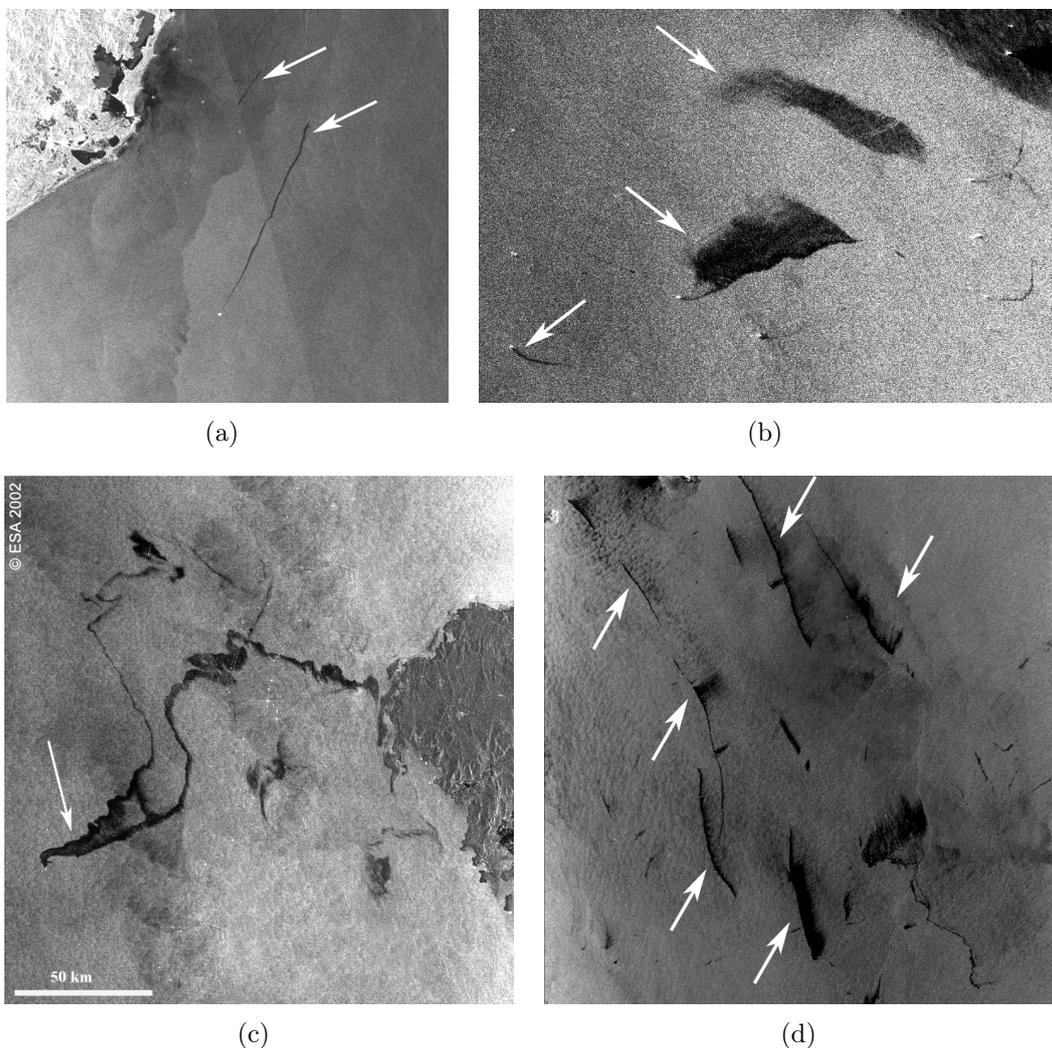


Figura 1.1: Exemplo de imagens SAR obtidas de satélites exibindo manchas (a) na costa do Brasil, (b) no Mar do Norte, (c) na costa da Espanha e (d) no sul da China. As setas indicam as manchas mais evidentes.

Alguns autores tratam a detecção de manchas como um problema de segmentação (MERCIER; GIRARD-ARDHUIN, 2005a; TOPOUZELIS et al., 2004), cujo objetivo é encontrar uma possível mancha de óleo numa imagem previamente selecionada. Dentro desta abordagem, alguns trabalhos apresentam métodos de segmentação utilizando uma filtragem prévia da imagem com base na modelagem do ruído *speckle* (ARAÚJO, 2004) ou utilizando o conhecimento prévio da análise das imagens para classificar os pixels (MERCIER; GIRARD-ARDHUIN, 2005a; TOPOUZELIS et al., 2004). Topouzelis et al. (2004) apresentam um método de segmentação de manchas com base em classificadores Perceptron Multi-Camadas (MLP, do inglês *Multi-Layer Perceptron*) e Funções de Base Radial (RBF, do inglês *Radial Basis Function*) com bons resultados. Uma Máquina de Vetor de Suporte (SVM, do inglês *Support Vector Machine*) é aplicada no trabalho de Mercier e Girard-Ardhuin (2005a) em que foram obtidos resultados satisfatórios na segmentação de manchas.

Utilizando outro tipo de abordagem, uma mancha previamente segmentada pode ser identificada como óleo numa imagem SAR através das suas características geométricas (FRATE et al., 2000; SOLBERG; BREKKE; SOLBERG, 2004), ou através da análise de características de textura (HARALICK, 1979; ASSILZADEH; MANSOR, 2001; MERCIER; GIRARD-ARDHUIN, 2005a; BREKKE; SOLBERG, 2005; SOLBERG; BREKKE; SOLBERG, 2004). O conceito por trás dessa abordagem é a extração de características da imagem que contém um determinado tipo de mancha para sua posterior classificação utilizando Redes Neurais Artificiais (RNAs). Mesmo sendo necessária a segmentação prévia da mancha para realizar a extração das características, o método de segmentação utilizado é menos complexo e pode ser feito através da escolha manual de uma pequena região de interesse (ROI, do inglês *region of interest*). Com a mancha segmentada, a classificação visa definir a sua natureza. Em Frate et al. (2000) as características extraídas das imagens segmentadas são utilizadas no treinamento de redes MLP para a predição do tipo de mancha.

1.1 Motivação

Uma mancha causada pelo derramamento de óleo recente de um navio em movimento possui um formato linear como as exibidas nas imagens da Figura 1.1 (a) e (d). Porém a ação de ventos e marés e a degradação com o tempo fazem com que sua geometria se modifique, tornando-se similar a uma mancha natural, como ilustrado na Figura 1.1 (b). Isto contribui para erros de interpretação visual do tipo

da mancha. Portanto, faz-se necessário adotar outro conjunto de características para descrever melhor as manchas, quando as suas características geométricas não contribuem isoladamente para uma identificação satisfatória.

Durante o desenvolvimento desta pesquisa, verificou-se que o projeto de uma solução confiável para a detecção automática de manchas de óleo através de imagens SAR deve contemplar simultaneamente três aspectos: a escassez de dados amostrais; a utilização da informação geométrica e de textura na extração de características; e a utilização da combinação de classificadores para a predição do tipo de mancha encontrada na imagem. Dentro dessa perspectiva, os principais fatores motivadores dessa dissertação são os seguintes:

- i.** os sistemas de detecção de manchas precisam ser confiáveis e, para isso, é necessário minimizar o número de alarmes falsos;
- ii.** a quantidade de registros de derramamento de óleo em imagens SAR é estatisticamente pouco representativa, porque as manchas de óleo são eventos raros do ponto de vista do registro em imagens;
- iii.** dependendo do seu tipo e tamanho, uma única imagem SAR pode custar em torno de 5 mil dólares¹ o que dificulta sua aquisição para fins de pesquisa;
- iv.** a geometria da mancha pode mudar rapidamente com a ação dos ventos, marés, mudança de temperatura e outros fatores, modificando as características da mancha de óleo;
- v.** é necessário que se disponha de um conjunto maior de características para descrever a natureza de uma mancha de óleo devido a sua complexidade e diversidade de formas;
- vi.** trabalhos recentes apontam para a melhoria de desempenho de classificação utilizando a combinação de classificadores em problemas de grande dimensionalidade dos dados;
- vii.** necessidade de uma metodologia capaz de detetar eficientemente manchas de óleo utilizando imagens SAR, sem a necessidade de se conhecer todos os parâmetros do sistema de imageamento e utilizando poucas amostras de treinamento.

¹Valor estimado de uma imagem RADARSAT obtido no site da ACRES (http://www.ga.gov.au/acres/prod_ser/rsatpri.jsp), acessado em 15-10-2007.

Este trabalho aborda o problema da detecção de manchas em imagens SAR do ponto de vista da classificação dessas manchas em dois grupos: manchas de óleo e manchas naturais ou similares. Um método automático de detecção de manchas tem como vantagens a repetibilidade na execução da tarefa e a capacidade de processamento de um grande número de informações. Além disso, métodos automáticos como o proposto por Fiscella et al. (2000) podem obter 80% de classificação correta das manchas quando comparado ao resultado da interpretação por especialistas humanos.

O método de Fiscella et al. (2000) utiliza informações estatísticas obtidas de medidas prévias das características físicas e geométricas de manchas de óleo e similares. Uma imagem amostral é avaliada utilizando dois procedimentos diferentes para determinar a probabilidade de que ela seja ou não uma mancha de óleo. O desempenho do algoritmo de classificação foi avaliado utilizando-se um conjunto de 80 imagens de manchas de óleo e 43 imagens de manchas naturais com características similares às manchas de óleo. Mais de 80% das amostras foram classificadas corretamente. A confiabilidade do método foi testada utilizando-se um conjunto de imagens diferentes, obtendo-se desempenho similar.

Utilizando as características geométricas das manchas, Fiscella et al. (2000) desenvolveram um método de classificação simples baseado na distância da média das classes para a amostra a ser classificada. No seu trabalho, 21 imagens foram testadas, obtendo 10% de falsos-negativos e 20% de falsos-positivos. Em Frate et al. (2000) foram utilizadas Perceptrons Multi-Camadas (MLPs, do inglês *Multi-Layer Perceptrons*) para classificar manchas utilizando um grande número de amostras de treinamento. Apesar dos resultados obtidos por Frate et al. (2000) terem sido considerados satisfatórios, 18% das manchas de óleo foram classificadas como manchas similares e 10% das manchas similares foram classificadas como manchas de óleo. Considerando que foram utilizadas 71 amostras de manchas de óleo e 68 de manchas similares, manchas classificadas de forma errada representam um número elevado de falsos-alarmes. Caso essa metodologia fosse aplicada em um sistema real de detecção de manchas de óleo, das 139 detecções, aproximadamente 7 teriam resultado em operações inúteis gerando um custo desnecessário. Entretanto, deixar de executar uma operação de contenção da mancha representa um custo ainda maior. Das 139 detecções, aproximadamente 12 teriam sido consideradas como manchas naturais quando, na verdade, eram manchas provenientes de derramamentos de

óleo. Numa aplicação real de detecção automática de manchas de óleo, a quantidade de falsos-positivos representa os custos operacionais desperdiçados e a quantidade de falsos-negativos representa o custo ambiental que poderia ter sido evitado. Para construir um sistema automático de detecção de manchas eficaz e confiável, é desejável que essas duas medidas sejam minimizadas ao mesmo tempo. Num sistema automático de detecção de manchas, este grande número de falsos-positivos e falsos-negativos contribuiria para a perda de confiabilidade ou poderia tornar inviável sua utilização prática.

Trabalhos recentes (PONTI-JUNIOR, 2004; BREVE; PONTI-JUNIOR; MASCARENHAS, 2007; PARK et al., 2007) demonstram que o uso da combinação de classificadores (ou comitês de classificadores), possibilita um aumento da taxa de acerto de classificação, comparados com classificadores individuais, sem perder o poder de generalização. Vários desses trabalhos têm resultado em sucesso na classificação de dados de grande dimensionalidade utilizando a combinação baseada do método *boosting* (SCHAPIRE, 1990). Feng, Cai e Chou (2005) apresentam um método bem sucedido na classificação de dados de alta dimensionalidade utilizando o *boosting* em problemas de predição da classe estrutural de proteínas. Yin, Liu e Han (2005) apresentam uma variante do método *boosting* para a combinação de alto nível de características e classificadores num problema de classificação de proteínas utilizando diferentes conjuntos de características. No mesmo tipo de aplicação, a combinação de classificadores foi experimentada com sucesso em Bittencourt (2005). Dettling e Bühlmann (2003) experimentam o uso de *boosting* em um problema de classificação de tumores a partir de bases de dados públicas, incluindo leucemia, câncer de colo e linfoma. Bailly, Arnaud e Puech (2007) apresentaram o uso do *boosting* no sensoriamento remoto aplicado à classificação de áreas geográficas em imagens de satélite. O método *boosting* também foi recentemente empregado na detecção de manchas de óleo em imagens SAR obtendo bons resultados na correta predição do tipo de mancha (RAMALHO; MEDEIROS, 2006, 2007).

1.2 Objetivo

Existem diversas etapas a serem consideradas no projeto de um sistema de sensoriamento remoto, dentre elas a seleção do sistema de imageamento, a seleção do sistema de classificação e das amostras de treinamento, o pré-processamento das imagens, a extração e seleção de características e a avaliação de desempenho de

classificação (LU; WENG, 2007).

O objetivo deste trabalho é desenvolver uma metodologia para a inferência confiável do tipo de mancha encontrada em uma imagem SAR a partir de um conjunto reduzido de imagens provenientes de sistemas de imageamento SAR com diferentes propriedades como o número de visadas, a resolução, a polarização e o azimute, dentre outras. Neste trabalho, a detecção de manchas é tratada como um problema de classificação em que se busca desenvolver uma metodologia baseada na combinação de classificadores para obtenção de um modelo capaz de lidar com um número reduzido de amostras e, ao mesmo tempo, atingir reduzidas taxas de erro, minimizando o número de falsos-alarmes. O modelo da Figura 1.2 apresenta uma visão geral da pesquisa, destacando as fases que constituem a contribuição principal deste trabalho.

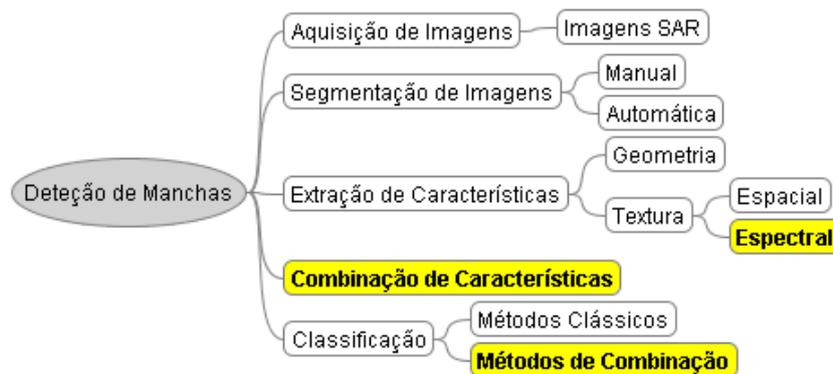


Figura 1.2: Modelo da metodologia utilizada. As três áreas de abrangência da pesquisa aparecem destacadas em negrito.

Para este trabalho, o ponto de partida é a imagem da mancha a qual foi previamente segmentada através de um procedimento manual ou automático. O processo de segmentação adotado foi desenvolvido e avaliado no trabalho de Araújo (2004). Na fase de extração de características buscou-se combinar diferentes características geradas por métodos distintos de extração a fim de aumentar o poder de discriminação das classes. Dois desses métodos de extração foram apresentados por Frate et al. (2000) e Haralick (1979). O primeiro método utiliza as características geométricas da mancha, partindo do princípio de que, em geral, as manchas de óleo têm geometria e espalhamento diferentes das manchas naturais. O segundo método utiliza informações extraídas das variações de tons de cinza de pixels adjacentes pertencentes à mancha, o que remete a uma interpretação da textura ou rugosidade.

Sabe-se que as manchas de óleo aparecem como áreas escuras e homogêneas numa imagem SAR e que possuem uma granulação diferente das áreas circundantes. Medidas como a entropia e a homogeneidade da vizinhança dos pixels pertencentes a essas regiões ajudam na identificação do tipo de mancha.

Um terceiro método foi desenvolvido ao longo da pesquisa para possibilitar a comparação de resultados obtidos com métodos comumente utilizados na literatura e a combinação de classificadores e de conjuntos de características. Este método de extração de características de textura utilizando ondaletas (*wavelets*) é apresentado no Apêndice C. Ao contrário do método de extração de características de textura proposto por Haralick (1979), as ondaletas possibilitam uma interpretação multiescalar da textura. Além disso, este conjunto adicional de características contribuiu para a construção do comitê conforme definido no Capítulo 4.

Os conjuntos iniciais de características foram combinados para construir novos conjuntos de maior dimensão a fim de aumentar o poder discriminatório dos dados. Para facilitar o projeto de classificadores neste ambiente de maior dimensão foi incluído um processo de transformação para a redução de dimensionalidade. A literatura especializada (DUDA; HART; STORK, 2000; HAYKIN, 2001; WEBB, 2002) disponibiliza um grande número de diferentes classificadores com propriedades diversas, aplicáveis na solução de problemas como o estudado nesta dissertação. O passo final deste trabalho foi a experimentação de vários classificadores individuais e diferentes algoritmos de comitê.

1.3 Contribuições

Este trabalho apresenta uma metodologia que possibilita, através da combinação de características e de classificadores, desenvolver um sistema automático de detecção de manchas de óleo no mar utilizando imagens SAR. Empregando a combinação da resposta de especialistas baseados em redes neurais, o método proposto permite determinar se a mancha encontrada na imagem é uma mancha de óleo ou uma mancha similar. A predição do tipo de mancha é feita com um número bastante reduzido de falsos alarmes. A pesquisa possibilitou, ainda:

- ▶ realizar o levantamento bibliográfico atualizado de métodos de combinação de classificadores baseados em *boosting*;

- ▶ desenvolver uma metodologia para a combinação de diversos conjuntos de características de alta dimensionalidade (LOPES et al., 2006; RAMALHO; MEDEIROS, 2007);
- ▶ criar um banco de dados de características espectrais de textura baseadas em *wavelets* (LOPES et al., 2006);
- ▶ avaliar os resultados da combinação de classificadores comparando-os com métodos de classificação clássicos (RAMALHO; MEDEIROS, 2006);
- ▶ prover uma solução de detecção de manchas através da utilização de um método de *boosting* para a combinação simultânea de características e de classificadores (RAMALHO; MEDEIROS, 2007).

1.4 Organização da Dissertação

Este trabalho está organizado em seis capítulos. No Capítulo 2 é apresentada uma revisão sobre as técnicas atualmente empregadas em sistemas de detecção de manchas que utilizam imagens SAR. No Capítulo 3 é apresentada uma visão geral dos algoritmos de classificação comumente utilizados em reconhecimento de padrões e que são aplicados nesse tipo de sistema de vigilância ambiental. Nesse capítulo também são apresentados alguns fundamentos teóricos do reconhecimento de padrões. No Capítulo 4 é apresentada a abordagem proposta para melhorar o desempenho de detecção de manchas através da adoção de algoritmos de combinação de classificadores. Uma discussão sobre os diferentes classificadores e os resultados experimentais obtidos com a classificação das manchas são apresentados no Capítulo 5. A conclusão e possíveis trabalhos futuros são comentados no Capítulo 6.

Capítulo 2

Métodos de Detecção de Manchas de Óleo em Imagens SAR

Neste capítulo são descritas as técnicas comumente utilizadas em sistemas de detecção de manchas de óleo em imagens SAR. O foco principal deste capítulo é o relato das técnicas utilizadas na extração de características e das diferentes abordagens de detecção encontradas na literatura.

2.1 Introdução

Equipamentos de sensoriamento remoto montados em aviões ou em satélites têm sido utilizados para identificar descargas ilegais no mar e no monitoramento de derramamento acidental de óleo. Os dispositivos de sensoriamento remoto geralmente utilizam sensores óticos, infravermelhos, térmicos ou radares em uma variedade de espectros diferentes. Os sistemas imageadores utilizam sensores que detetam diferentes propriedades da superfície do mar como cor, refletância, temperatura e rugosidade. A presença de óleo na superfície do mar modifica algumas dessas propriedades e essas mudanças podem ser detetadas através da análise visual das imagens. Em geral, também faz-se uso de informações contextuais como dados meteorológicos, localização geográfica, velocidade e direção de ventos, localização de plataformas de petróleo e navios petroleiros, etc. (BREKKE; SOLBERG, 2005). O conjunto dessas informações fornece meios de distinguir entre manchas de óleo e manchas similares evitando falsos alarmes.

Os sistemas SAR têm uma grande vantagem sobre os sistemas de imageamento

ótico, pois o sinal emitido pelo radar não sofre influência das condições atmosféricas (tem boa penetração) e o registro das imagens independe da luz solar. A presença de óleo na superfície do mar desvia a energia emitida pelo radar, o que reduz drasticamente o sinal retro-propagado, como ilustrado na Figura 2.1. Dessa forma, as manchas de óleo aparecem como regiões escuras dentro da imagem SAR e isto possibilita desenvolver diferentes métodos para sua deteção.

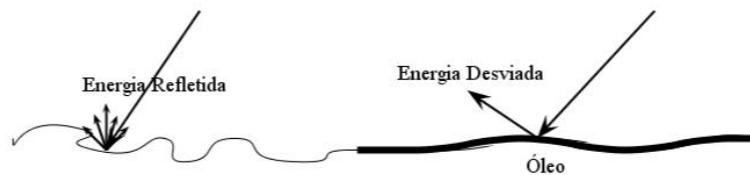


Figura 2.1: A superfície do mar na presença de óleo torna-se mais viscosa e atenua o sinal retro-propagado.

Entretanto, as imagens geradas por sistemas de imageamento SAR são contaminadas pelo ruído *speckle*, fato este inerente ao sistema físico do sensor. Mesmo em áreas homogêneas da imagem, os pixels possuem distribuição estatística com grande desvio padrão (BREKKE; SOLBERG, 2005). A presença desse tipo de ruído dificulta o processo de segmentação da mancha e exige uma fase de pré-processamento para que a imagem possa ser utilizada em algoritmos de segmentação automática (ARAÚJO, 2004).

2.2 Estratégias Comumente Adotadas na Detecção de Manchas

Segundo Brekke e Solberg (2005), sistemas de deteção de óleo através de imagens SAR podem ser manuais, semi-automáticos ou automáticos. O sistema KSAT¹ (Kongsberg Satellite Service) do Centro de Computação da Noruega é um sistema manual, no qual as imagens são analisadas por um técnico especialista que se baseia também em informações externas contextuais (como velocidade e direção do vento, localização de navios e tubulações de transporte de óleo) para decidir a ação a ser tomada. Já o sistema MaST² (Marine Surveillance Tool) da empresa britânica QinetiQ é semi-automático, porque cobre apenas a fase de deteção automática de manchas sem, contudo, extrair características ou classificá-las de acordo com essas

¹<http://www.ksat.no>

²<http://mast.space.qinetiq.com>

características. O sistema MaST, porém, auxilia os especialistas ao indicar uma possível presença de mancha de óleo nas imagens SAR de áreas monitoradas. Brekke e Solberg (2005) também relatam sucessos com o sistema automático da NR³ (Norsk Regnesentral), também do Centro de Computação da Noruega e faz uma comparação de desempenho de ambos, mostrando que esse sistema apresenta desempenho similar ao manual e semi-automático e ainda tem a vantagem de repetir sempre a mesma resposta para os mesmos dados de entrada.

No âmbito nacional, em cumprimento à Convenção Internacional sobre Preparo, Resposta e Cooperação para Derramamentos de Óleo no Mar, da qual o Brasil é um dos países signatários, a ANP (Agência Nacional de Petróleo) e Coppe-UFRJ (Universidade Federal do Rio de Janeiro) em cooperação com Ibama CCSivam iniciaram em 2002 o desenvolvimento de um sistema de monitoramento de derramamento de óleo em ambiente marinho envolvendo também a UFRGS (Universidade Federal do Rio Grande do Sul) e UFRN (Universidade Federal do Rio Grande do Norte). O sistema piloto entrou em operação no Centro de Monitoramento de Derramamento de Óleo no Mar (CeMOM) da COPPE/UFRJ/ANP monitorando a bacia de Campos com objetivo de realizar a detecção, identificação, classificação e divulgação de eventos de derramamento de óleo no mar, com base na integração de dados de sensoriamento remoto por satélite e produtos de modelagem numérica atmosférica e oceanográfica.

Sistemas automáticos de detecção de manchas de óleo no mar funcionam através da análise de imagens SAR como mostrado na Figura 2.2 os quais, em geral, possuem três etapas: segmentação da mancha, extração de características e classificação do tipo de mancha. As imagens capturadas pelo satélite são transmitidas para estações de processamento de dados. Estas imagens são analisadas por especialistas que, auxiliados por Sistemas de Visão Artificial (SVAs), geram relatórios informando aos centros de monitoramento ou de vigilância ambiental a ocorrência de algum possível derramamento de óleo.

Na literatura, o termo detecção de manchas é empregado em diferentes abordagens. Em seus trabalhos, Araújo (2004), Topouzelis et al. (2004), Mercier e Girard-Ardhuin (2005a) tratam a detecção de manchas como um problema de segmentação, através da qual busca-se encontrar possíveis manchas causadas por derramamentos de óleo utilizando a filtragem ou a classificação de pixels. Frate et

³<http://www.nr.no>

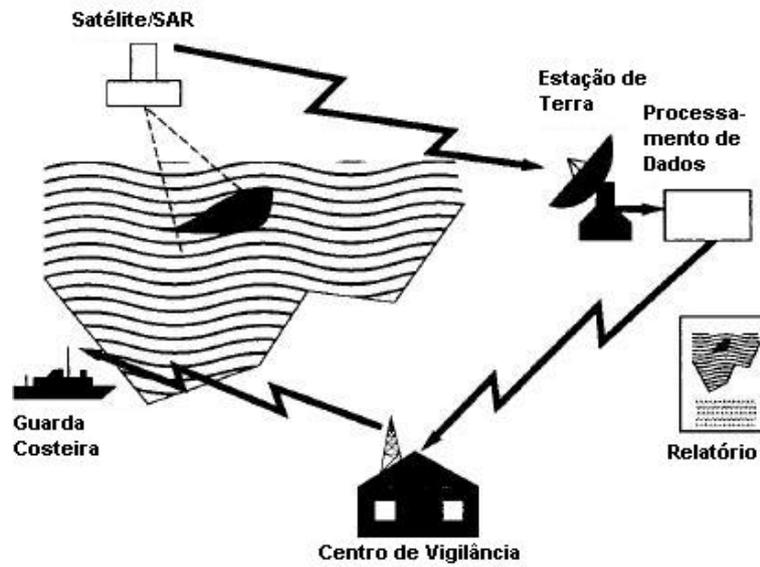


Figura 2.2: Exemplo de um sistema de detecção de manchas de óleo (MARTINEZ; MORENO, 1996)

al. (2000), Lopes et al. (2006), Ramalho e Medeiros (2006) tratam a detecção como um problema de classificação, em que se propõe a classificação de manchas com base nas características extraídas das imagens previamente segmentadas. Esta última é a abordagem adotada nesta dissertação.

A Tabela 2.1 resume alguns dos métodos de detecção de manchas estudados.

Tabela 2.1: Características gerais de diferentes métodos de detecção de manchas.

Método	Descritor	Objetivo	Classificador	Radar
Barni, Betti e Mecocci (1995)	Intensidade dos pixels	Segmentação	Fuzzy	SEASAT e ERS
Gade e Redondo (1999)	Dimensão Fractal	Classificação	-x-	ERS
Fiscella et al. (2000)	Geometria	Classificação	Distância e Probabilidade	ERS
Frate et al. (2000)	Geometria	Classificação	MLP	ERS
Assilzadeh e Mansor (2001)	Textura	Segmentação	-x-	Radarsat
Solberg, Brekke e Solberg (2004)	Geometria	Segmentação e Classificação	Estatístico	Radarsat e ENVISAT
Topouzelis et al. (2004)	Geometria e Textura	Segmentação	RBF e MLP	ERS
Araújo (2004)	Geometria	Segmentação e Classificação	KNN e MLP	ERS e RADARSAT
Mercier e Girard-Ardhuin (2005b)	Textura	Segmentação	SVM	ERS

2.2.1 Segmentação

A segmentação é o processo responsável por separar uma mancha do restante da cena. Diferentes métodos de segmentação podem ser empregados, dentre eles um bastante simples é a limiarização de histograma (GONZALEZ; WOODS, 1992). Brekke e Solberg (2005) relatam diversas técnicas de segmentação baseadas neste princípio. O método mais simples de deteção da mancha através da segmentação é feito através da limiarização sobre o histograma em nível de cinza da imagem SAR (MERCIER; GIRARD-ARDHUIN, 2005b). Porém, em imagens ruidosas como as geradas pelos radares SAR, um procedimento prévio de filtragem do ruído *speckle* é fundamental. A limiarização de histograma consiste em determinar um limiar ótimo L no intervalo $0 < L < 255$, utilizado para rotular os pixels de intensidade $f(x, y)$ da imagem, cujas coordenadas são x e y . O valor de L pode ser definido por

$$L = F[x, y, p(x, y), f(x, y)], \quad (2.1)$$

em que $p(x, y)$ é uma função que denota alguma propriedade local em torno do pixel (GONZALEZ; WOODS, 1992). O resultado final obtido será uma imagem segmentada definida por

$$g(x, y) = \begin{cases} 1 & \text{se } f(x, y) > L \\ 0 & \text{se } f(x, y) \leq L \end{cases}, \quad (2.2)$$

na qual será possível identificar a presença da mancha na cena.

Em geral, o tamanho original das imagens SAR é da ordem de 1000 pixels de largura. O processamento de uma imagem dessa dimensão pode ser muito lento, de tal forma que o passo inicial para deteção da mancha é a seleção de uma ROI dentro da imagem SAR. Este procedimento é manual e deve ser feito de tal forma a limitar a área de busca da mancha. A ROI pode representar uma área de busca da mancha ou uma área a ser tomada como padrão de comparação. A Figura 2.3 exemplifica a utilização de uma ROI na escolha de uma área utilizada para segmentar a mancha. A região selecionada define uma área não poluída da imagem que foi utilizada para calcular as propriedades dos pixels que a representam e, assim, utilizá-las para segmentar os pixels que não seguem as mesmas propriedades.

Entretanto, a presença do ruído *speckle* torna imprecisa a segmentação pela

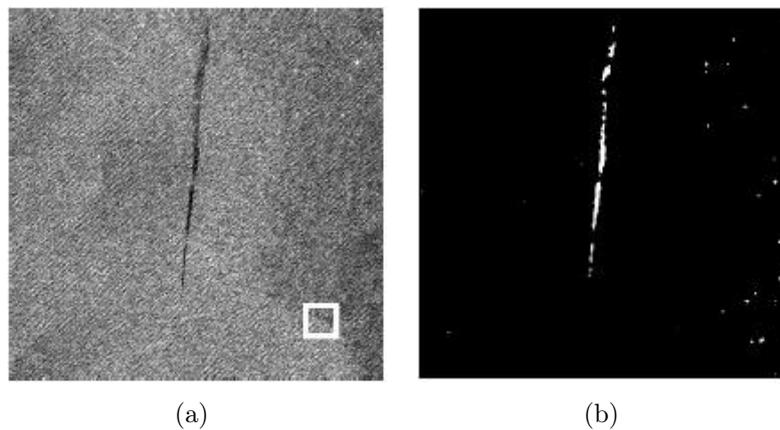


Figura 2.3: Exemplo de detecção semi-automática de manchas baseada em segmentação (MERCIER; GIRARD-ARDHUIN, 2005b). (a) Imagem com a região de interesse utilizada para definir a função de decisão. (b) Imagem segmentada.

limiarização. Um procedimento de segmentação por crescimento de regiões exibido na Figura 2.4 foi apresentado por Araújo (2004). Através desse método é possível segmentar as manchas (ver máscaras no Apêndice B) minimizando a influência do ruído.

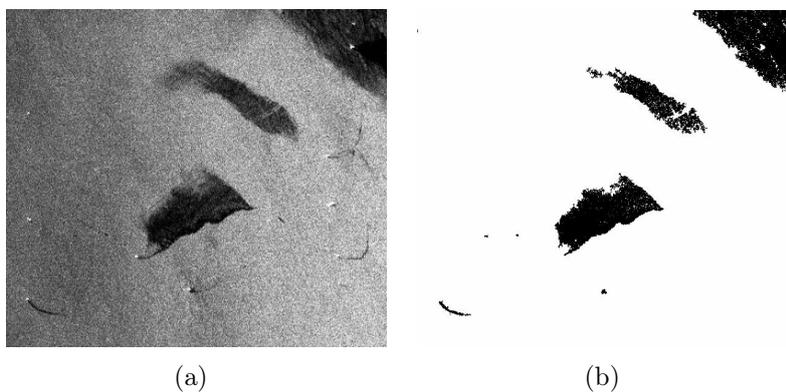


Figura 2.4: Exemplo de uma (a) imagem SAR (b) segmentada pelo processo de crescimento de regiões (ARAÚJO, 2004).

A mudança de direção ou de velocidade dos ventos e as mudanças de estado da superfície do mar (e.g. temperatura) podem modificar as características da mancha, principalmente sua geometria e rugosidade ou textura. Além disso, óleos com propriedades diferentes (e.g. composição e grau de degradação) ou a presença de fitoplânctons e algas, dentre outras manifestações naturais, também produzirão manchas diferentes (MERCIER; GIRARD-ARDHUIN, 2005a). Alguns trabalhos utilizam RNAs para classificar os pixels com base em características extraídas da imagem. Topouzelis et al. (2004) realizaram testes de segmentação

com RBF e MLP atingindo excelentes resultados como mostrado na Figura 2.5. Em seu trabalho, foram utilizadas como entradas das RNAs imagens geradas pelo filtro da média e pelo desvio padrão dos pixels vizinhos dentre outras. Mercier e Girard-Ardhuin (2005a) utilizaram uma SVM com núcleo polinomial como classificador. Os dados utilizados no projeto do classificador foram obtidos através de uma transformada ondaleta não-decimada em três níveis de decomposição. Cada pixel da imagem foi classificado com base nos descritores obtidos a partir do coeficientes da transformação, gerando a imagem temática da Figura 2.5 (d).

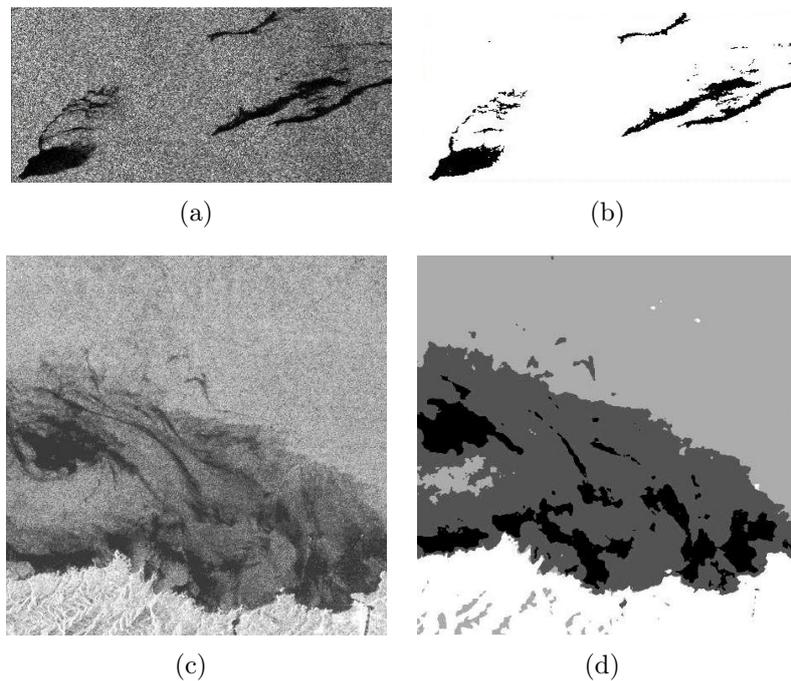


Figura 2.5: Exemplos de imagens SAR contendo manchas (a,c) e suas correspondentes segmentadas (b,d) (TOPOUZELIS et al., 2004; MERCIER; GIRARD-ARDHUIN, 2005a).

2.2.2 Classificação pelas Propriedades Geométricas

Em Frate et al. (2000) foram definidas 11 características para descrever uma mancha previamente segmentada tomando por base suas propriedades geométricas como área, perímetro, complexidade, espalhamento, desvio padrão da mancha e da cena, contraste máximo e médio, gradiente máximo e médio e desvio padrão do gradiente. Detalhes do método de extração destas características de Frate et al. (2000) são descritos no Apêndice C.

O objetivo do trabalho de Frate et al. (2000) foi classificar as manchas utilizando

uma MLP projetada com base num banco de dados contendo 139 imagens. Depois de treinado, o classificador fornece uma saída proporcional à probabilidade de uma mancha pertencer à classe de manchas de óleo ou de manchas similares.

Solberg, Brekke e Solberg (2004) descrevem uma metodologia similar com base nas características geométricas de aproximadamente 100 manchas, porém, utilizam modelos estatísticos e regras de decisão para classificá-las.

2.2.3 Classificação pelas Propriedades de Textura

A viscosidade do óleo influencia diretamente no retroespalhamento do sinal no sistema de imageamento SAR mudando não somente a intensidade dos pixels na imagem, mas também a variação de intensidade entre os pixels adjacentes. Isto pode ser interpretado, em uma abordagem estatística, como sendo a textura da mancha (GONZALEZ; WOODS, 1992).

Haralick (1979) definiu textura como uniformidade, densidade, aspereza, regularidade e intensidade dentre outras características da imagem. No trabalho citado, o autor caracteriza a textura como um conceito *bi*-dimensional, em que uma dimensão contém as propriedades primitivas da tonalidade dos pixels e a outra corresponde aos relacionamentos espaciais entre elas. Eles indicaram que os conceitos de tonalidade e textura não são independentes, de tal forma que em algumas imagens a tonalidade é dominante e, em outras, a textura é dominante.

A textura de uma imagem está relacionada à distribuição de probabilidade conjunta da intensidade dos pixels, a qual pode ser aproximada pela Matriz de Coocorrência de Níveis de Cinza (GLCM, do inglês *Gray Level Cooccurrence Matrix*). Haralick, Shanmugam e Dinstein (1973) e Haralick (1979) descrevem uma metodologia de classificação de imagens a partir do uso da abordagem estatística de segunda ordem calculada sobre a GLCM. Valores locais de textura podem ser obtidos tomando-se por base uma pequena janela (e.g. 9 x 9 pixels) centralizada em cada pixel da imagem (ver Apêndice C). Utilizando essa metodologia de extração de características, Assilzadeh e Mansor (2001) definiram um sistema onde utilizam a homogeneidade e o momento angular de segunda ordem calculados sobre a GLCM para deteção e classificação das manchas de óleo.

Uma outra interpretação do efeito da viscosidade do óleo na superfície do mar é discutida em Mercier e Girard-Ardhuin (2005a), Liu e MacGregor (2007) e Liew et al. (1995). Nestes trabalhos, é utilizada uma abordagem espectral da textura a qual

baseia-se em propriedades do espectro de Fourier, sendo principalmente utilizadas na detecção de periodicidade global em uma imagem através da identificação de picos de alta energia no espectro. Em Mercier e Girard-Ardhuin (2005a) um conjunto de características baseado no espectro do sinal de retroespalhamento foi utilizado como entrada de um classificador para fins de segmentação da mancha de óleo. Mercier e Girard-Ardhuin (2005a) utilizaram ondaletas para gerar um conjunto de descritores espectrais de textura. Seguindo uma técnica semelhante, apresentada no Apêndice C, um novo conjunto de características de textura é proposto neste trabalho para reforçar a separabilidade entre as classes de manchas. O resultado da classificação de manchas com base neste conjunto de características espectrais de textura e nos demais conjuntos de características são apresentados no Capítulo 5.

2.3 Conclusão

O processo de detecção de manchas pela segmentação possibilita sua visualização, localização e a análise da sua geometria. Entretanto, a correta interpretação das características geométricas da mancha (forma, área, perímetro, etc.) é dependente da resolução da imagem e, conseqüentemente, de um método de segmentação bastante robusto ao ruído presente na imagem.

Os métodos de detecção baseados em RNAs para a classificação de características extraídas das manchas previamente segmentadas apresentam bons resultados. Contudo, existe a possibilidade da predição falhar caso os descritores estejam baseados apenas em informações geométricas, porque as manchas podem apresentar formas e tamanhos modificados pela ação dos ventos. Mesmo quando utilizadas características espaciais de textura, estas também podem introduzir imprecisões devido às diferentes resoluções das imagens ou devido à degradação do óleo e à sua mudança de viscosidade.

Para minimizar esses problemas, diferentes abordagens podem ser testadas como o aperfeiçoamento da técnica de segmentação ou da filtragem do ruído. Outra solução consiste em adotar um conjunto de características espectrais de textura baseado em ondaletas. Isto proporciona uma análise multiescala da textura, o que sugere incorporar significativa informação independente da resolução da imagem ou da forma da mancha. Como complemento, pode-se investigar algoritmos de classificação modernos cujas características permitam agregar ao mesmo tempo uma maior capacidade de generalização e a utilização de diferentes conjuntos de

características.

No próximo capítulo são apresentados classificadores clássicos e RNAs cujos resultados na classificação das manchas são tomados em comparação com a metodologia proposta de combinação de classificadores e de características.

Deteção de Manchas Utilizando Classificadores Individuais

Neste capítulo são apresentados alguns algoritmos de aprendizagem clássicos utilizados em reconhecimento de padrões e outros baseados em redes neurais, comumente utilizados em sistemas de deteção de manchas de óleo. Estes algoritmos de aprendizagem serão chamados de classificadores individuais para diferenciar daqueles gerados a partir de métodos de combinação. Os classificadores individuais apresentados neste capítulo foram utilizados como referência para comparação de desempenho com a metodologia proposta.

3.1 Introdução

Os dados utilizados neste trabalho foram obtidos a partir de imagens SAR separadas em duas classes equiprováveis de manchas de óleo e manchas similares. Para representar um determinado tipo de mancha, são necessários diversos descritores geométricos ou de textura, podendo ser combinados entre si. Os vetores de entrada gerados com esses descritores possuem grande dimensionalidade, variando de 7 a 15 elementos, o que dificulta uma interpretação visual desses dados.

Predizer a classe à qual pertence uma mancha é uma tarefa dos classificadores supervisionados, em que são conhecidas as classes a que pertencem as amostras de treinamento. Uma determinada mancha dentro do conjunto de amostras é representada por um vetor de características $\mathbf{x} = (x_1, \dots, x_M)$, em que x_m ($m = 1, \dots, M$) representa um determinado descritor dentro do conjunto de M

descritores. Quando este vetor de características é apresentado ao classificador ele é chamado de vetor de entrada ou padrão de entrada. A saída \mathbf{y} do classificador, em certos casos, pode ser interpretada como a probabilidade daquele padrão de entrada pertencer a uma determinada classe \mathcal{C}_j , em que $j = 1, \dots, J$, sendo J o número de classes previamente conhecida.

A habilidade de generalização de um classificador está diretamente relacionada ao seu desempenho na classificação dos padrões de teste, ou seja, aqueles vetores de entrada que não foram utilizados na etapa de treinamento. Uma generalização pobre pode ser atribuída a três motivos (JAIN; DUIN; MAO, 2000; HAYKIN, 2001) listados a seguir:

- i. a quantidade de características (descritores) é muito grande com relação ao número de amostras de treinamento;
- ii. o número de parâmetros ajustáveis do classificador é muito grande em relação ao número de amostras de treinamento;
- iii. o classificador foi excessivamente otimizado sobre o conjunto de treinamento; ocorre o excesso de treinamento (*overtraining*), que é similar ao fenômeno de sobreajustamento (*overfitting*) da regressão quando existem muitos parâmetros livres.

Nos trabalhos comentados no Capítulo 2, os autores construíram classificadores baseados em RNAs obtendo bons resultados tanto na segmentação quanto na classificação das manchas. Entretanto, apesar do MLP ter boa capacidade de generalização mesmo em condições de grande dimensionalidade (BARRON, 1993), seu projeto é prejudicado quando apenas uma pequena quantidade de amostras de treinamento está disponível (JAIN; DUIN; MAO, 2000; HAYKIN, 2001). Os métodos de combinação baseados no *boosting* (FREUND; SCHAPIRE, 1999) e no *bagging* (BREIMAN, 1996), apresentados no Capítulo 4, são propostos como solução para esse problema.

A fim de obter modelos de referência e, ao mesmo tempo, avaliar a qualidade dos dados quanto à capacidade de separação das classes, foram experimentados classificadores com diferentes propriedades. Classificadores clássicos, como o *k*-Vizinhos mais Próximos (KNN, do inglês *k-Nearest Neighbor*) e o *Naiive Bayes*, foram utilizados como referência na comparação com os métodos de combinação.

Além desses classificadores, foram experimentadas uma rede neural (MLP) e uma máquina de vetor de suporte (SVM), pois a literatura apresenta esses dois classificadores como solução de vários problemas de detecção de manchas. O gráfico da Figura 3.1 exibe classificadores agrupados em categorias diferentes, a saber, clássicos, redes neurais e os métodos de combinação utilizados no desenvolvimento desta dissertação.



Figura 3.1: Classificadores comumente encontrados na literatura.

3.2 Classificadores Paramétricos

O projeto de um classificador paramétrico começa a partir de uma análise da probabilidade de ocorrência de um determinado grupo de amostras dentro de uma população. Dentro dessa perspectiva encontram-se os classificadores paramétricos que se baseiam no conhecimento prévio de informações que possam ser utilizadas para modelar o perfil probabilístico dos dados.

Conforme Duda, Hart e Stork (2000), pode-se projetar um classificador com base nas probabilidades *a priori* $P(\mathcal{C}_j)$ e funções de densidade de probabilidade condicional à classe $p(\mathbf{x}|\mathcal{C}_j)$, também chamada função de verossimilhança da classe \mathcal{C}_j , segundo a regra de Bayes definida como

$$P(\mathcal{C}_j|\mathbf{x}) = \frac{p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j)}{p(\mathbf{x})}, \quad (3.1)$$

em que $P(\mathcal{C}_j|\mathbf{x})$ é a probabilidade condicional *a posteriori* da classe \mathcal{C}_j e $p(\mathbf{x})$ é a função densidade de probabilidade (PDF, do inglês *probability density function*) dada por

$$p(\mathbf{x}) = \sum_{j=1}^J p(\mathbf{x}|\mathcal{C}_j)P(\mathcal{C}_j). \quad (3.2)$$

Os classificadores paramétricos são considerados classificadores estatísticos ótimos ou sub-ótimos e a equação (3.1) é chamada de regra de Bayes para o erro mínimo (WEBB, 2002). Conhecidos os parâmetros dos modelos de cada classe, é possível projetar um classificador capaz de obter a melhor taxa de classificação possível para um determinado problema de reconhecimento de padrões. Porém, em geral, não estão disponíveis informações *a priori* suficientes para uma modelagem satisfatória. Na prática, os parâmetros necessitam ser estimados a partir do conjunto de amostras. A equação (3.3) é uma estimativa da probabilidade *a priori*, determinada pela frequência de ocorrência $f(\mathbf{x})$ de um determinado vetor de entrada pertencente a uma classe \mathcal{C}_j , dentro de um conjunto de dados limitado contendo N amostras. Outros parâmetros também podem ser estimados a partir dos dados, como a média (equação 3.4) e a variância não-enviesada (equação 3.5). Entretanto, se o conjunto amostral for muito pequeno, a estimativa dos parâmetros estatísticos não será confiável. Em outras palavras, o valor de N deve ser grande o suficiente para obter uma estimativa estatisticamente representativa.

$$P(\mathcal{C}_j|\mathbf{x}) = \frac{f(\mathbf{x})}{N}. \quad (3.3)$$

$$\boldsymbol{\mu} = \frac{1}{N} \sum_{n=1}^N \mathbf{x}_n. \quad (3.4)$$

$$\sigma^2 = \frac{1}{N-1} \sum_{n=1}^N (\mathbf{x}_n - \boldsymbol{\mu})^2. \quad (3.5)$$

3.2.1 Máxima Verossimilhança

O classificador de máxima verossimilhança (*maximum likelihood*) baseia-se na equação (3.1), assumindo que as amostras são representativas, para gerar a regra de decisão da equação (3.6), para todo $j \neq i$. A Figura 3.2 exibe um exemplo de classificação de máxima verossimilhança para duas classes \mathcal{C}_1 e \mathcal{C}_2 .

$$P(\mathcal{C}_i|\mathbf{x}) \underset{\mathbf{x} \in \mathcal{C}_j}{\overset{\mathbf{x} \in \mathcal{C}_i}{\geq}} P(\mathcal{C}_j|\mathbf{x}). \quad (3.6)$$

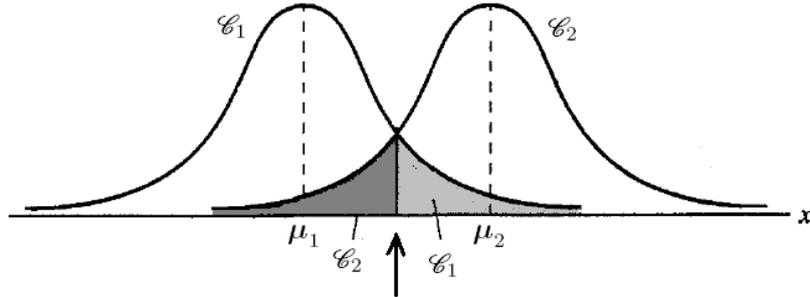


Figura 3.2: Funções densidade de probabilidade das classes \mathcal{C}_1 e \mathcal{C}_2 . A seta indica o limiar de decisão de máxima verossimilhança. Adaptado de Haykin (2001).

Na implementação multi-classe de um classificador de máxima verossimilhança, a PDF de uma determinada classe \mathcal{C}_j pode ser estimada a partir da equação (3.7), assumindo-se que a distribuição das amostras é normal. A PDF é expressa em termos do vetor de médias $\boldsymbol{\mu}$ da classe e da matriz de covariância $\boldsymbol{\Sigma}$ dos dados, considerando as relações inter-classes. Para que a equação (3.7) seja válida, a matriz $\boldsymbol{\Sigma}$ estimada a partir dos dados deve ser inversível. Em aplicações práticas, quando o número de amostras é pequeno, a estimativa da matriz de covariância $\boldsymbol{\Sigma}$ pode resultar numa matriz singular e um procedimento de regularização devem ser adotado (WEBB, 2002).

$$p_{\boldsymbol{\mu}, \boldsymbol{\Sigma}}(\mathbf{x}|\mathcal{C}) = \frac{1}{\det(\boldsymbol{\Sigma})^{1/2}(2\pi)^{M/2}} \exp \left[-\frac{(\mathbf{x} - \boldsymbol{\mu})^T \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu})}{2} \right]. \quad (3.7)$$

3.2.2 Naïve Bayes

Existem diversas técnicas para regularização da matriz de covariância $\boldsymbol{\Sigma}$ da equação (3.7), a fim de torná-la inversível, conforme descreve Webb (2002). Uma delas consiste em forçar a independência estatística dos dados atribuindo uma matriz diagonal a $\boldsymbol{\Sigma}$, com variâncias unitárias para cada classe. Neste caso, obtém-se o princípio do classificador *Naïve Bayes*. A superfície de decisão é formada com base na mesma regra do classificador de máxima verossimilhança.

O classificador *Naïve Bayes* pode ser utilizado em problemas onde se ignora a natureza estatística dos dados de treinamento.

3.3 Classificadores Baseados em Distância

Uma determinada classe \mathcal{C}_j , em que $j = 1, \dots, J$, pode ser representada por uma média amostral dos seus N_j vetores \mathbf{x} , que define seu vetor protótipo $\boldsymbol{\mu}_j$ dado por

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{\mathbf{x} \in \mathcal{C}_j} \mathbf{x}. \quad (3.8)$$

O procedimento de classificação implica em atribuir a um vetor \mathbf{x} desconhecido o rótulo da classe do protótipo mais próximo utilizando, por exemplo, a distância euclidiana. Esta regra de classificação utilizando distância pode ser expressa por

$$H = \arg \min_{j=1, \dots, J} \|\mathbf{x} - \boldsymbol{\mu}_j\|. \quad (3.9)$$

3.3.1 KNN (k-Nearest Neighbor)

O algoritmo k-Vizinhos mais Próximos (KNN, do inglês *k-Nearest Neighbor*) é um método clássico de uso da distância para a estimativa da PDF das classes a partir do conjunto de amostras de treinamento (WEBB, 2002). A PDF de uma determinada amostra \mathbf{x} pode ser aproximada por

$$\hat{p}(\mathbf{x}) = \frac{k}{NV}, \quad (3.10)$$

em que k é um número pré-determinado de vizinhos, N é o número de vetores de entrada e V representa um volume ocupado pela vizinhança. Para um volume V pequeno, $\hat{p}(\mathbf{x})$ define a probabilidade estimada de que uma amostra \mathbf{x} pertença a uma determinada classe contendo k vizinhos.

O algoritmo da Figura 3.3 possibilita uma melhor compreensão do classificador KNN, considerando uma vizinhança genérica. Em geral, escolhe-se uma vizinhança ímpar com $k > 1$, de tal forma que um possível empate seja resolvido por uma regra de maioria (SÁ, 2001). A classificação de uma amostra \mathbf{x}^* desconhecida é feita atribuindo-se a ela o rótulo r_i mais freqüente dentre os k vetores mais próximos. Para o cálculo desta distância, é comum a utilização da norma euclidiana ou norma L_2 , definida como

$$d(\mathbf{x}^*, \mathbf{x}_n) = \|\mathbf{x}^* - \mathbf{x}_n\|^2 = \sqrt{\sum_{m=1}^M (x_m^* - x_m^n)^2}, \quad (3.11)$$

em que $n = 1, \dots, N$ e M é a dimensão dos vetores \mathbf{x}^* e \mathbf{x}_n .

<p>Entrada: Vetor de atributos $\mathbf{x}^* = (x_1, \dots, x_M)$ desconhecido e $\mathbf{X} = [(\mathbf{x}_1, r_1), \dots, (\mathbf{x}_N, r_N)]$ as amostras conhecidas.</p> <p>Saída: H</p> <p>1 início</p> <p>2 Calcular a distância euclidiana do vetor de entrada \mathbf{x}^* para as N amostras de treinamento, obtendo $d_n(\mathbf{x}^*, \mathbf{x}_n)$, $n = 1, \dots, N$.</p> <p>3 Obter a predição final H escolhendo o rótulo r_n mais freqüente dentre os k vizinhos mais próximos dada a distância d_n;</p> <p>4 fim</p>
--

Figura 3.3: Algoritmo KNN para uma vizinhança genérica.

Segundo Theodoridis e Koutroumbas (2003), para um grande número de amostras, espera-se que o KNN tenha um desempenho similar ao classificador bayesiano com probabilidade de erro pequeno. Por outro lado, complementam os autores, o KNN necessita de $O(kN^2)$ operações, o que é particularmente severo em espaços de características de grande dimensão.

3.4 Redes Neurais Artificiais

Uma Rede Neural Artificial (RNA) fornece uma forma de estimativa direta das probabilidades *a posteriori* sem a necessidade de conhecer os parâmetros estatísticos das classes. A RNA pode ser vista como uma função de mapeamento

$$F : \mathbb{R}^M \rightarrow \mathbb{R}^J,$$

em que um vetor de entrada \mathbf{x} de dimensão M é submetido à rede para produzir uma saída \mathbf{y} no espaço de dimensão J que, em geral, representa o número de classes.

As RNAs utilizam uma modelagem baseada no processamento da informação dos neurônios biológicos, como ilustra a Figura 3.4. Os neurônios das RNAs possuem entradas x_i e uma saída y_j que são análogos aos dendritos e axônios de um neurônio biológico. As sinapses, conexões entre o axônio de um neurônio e o dendrito de outro, são representadas nas RNAs pelos pesos sinápticos w_{ji} . Os índices i e j

identificam, respectivamente, o peso sináptico de uma entrada (i) pertencente a um determinado neurônio (j). Os pesos sinápticos são modificadores dos vetores de entrada submetidos aos neurônios. Quanto maior o peso, maior a influência de uma determinada característica ou neurônio dentro da rede.

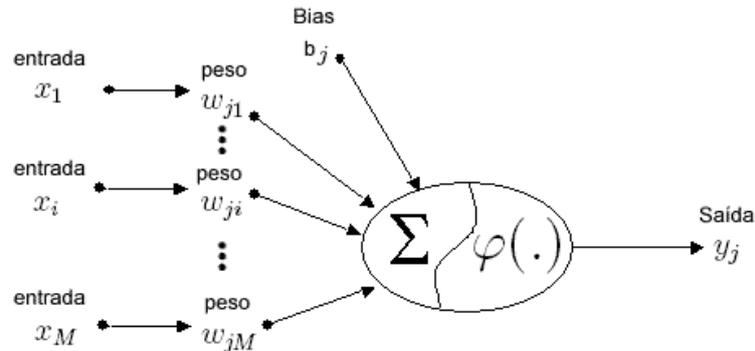


Figura 3.4: Modelo matemático de um neurônio (perceptron).

A saída de um neurônio j é dada por $y_j = \varphi(v_j)$, em que $v_j = \sum w_{ji}x_i + b_j$ para $i = 1, \dots, M$ é chamado de potencial de ativação (*net*) e $\varphi(\cdot)$ é a função de ativação do neurônio. A função de ativação (ver Figura 3.6) confere ao neurônio a conveniente propriedade de limitar assintoticamente o valor da saída entre um limite máximo e mínimo conhecidos de forma linear ou não-linear, dependendo da função escolhida.

Segundo Baum e Haussler (1989), para obter boa generalização com uma RNA, o tamanho N do conjunto de treinamento deve satisfazer ao critério

$$N = O\left(\frac{W}{\varepsilon}\right), \quad (3.12)$$

em que W é o número total de parâmetros livres, ou seja o número de pesos sinápticos, ε representa o erro admitido e $O(\cdot)$ representa a ordem da quantidade entre parênteses. Por exemplo, para um erro admissível de 10 por cento, o número de amostras de treinamento deve ser pelo menos 10 vezes a quantidade de parâmetros ajustáveis (HAYKIN, 2001).

3.4.1 MLP

A rede Perceptron Multi-Camadas (MLP, do inglês *Multi-Layer Perceptron*) é um classificador que aprende a partir de exemplos e possui uma memória (vetor de pesos) para armazenamento das regras de geração de discriminantes não-lineares. Depois

de treinada, a rede MLP pode produzir respostas para uma amostra desconhecida e se adaptar a ela, incorporando mais conhecimento.

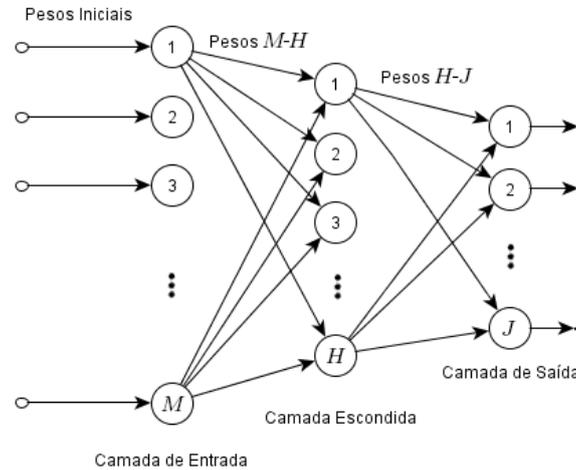


Figura 3.5: Modelo de um MLP com uma camada escondida.

Um MLP utiliza uma rede de neurônios interconectados em camadas como mostra o modelo da Figura 3.5. Cada neurônio da camada oculta gera um discriminante com base numa função de ativação $\varphi(\cdot)$, em geral, sigmoideal. Duas funções de ativação (ilustradas na Figura 3.6) freqüentemente utilizadas são a função logística

$$\varphi(x) = \frac{1}{1 + \exp(-x)} \quad (3.13)$$

e a função tangente hiperbólica \tanh

$$\varphi(x) = \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}. \quad (3.14)$$

O tipo de função de ativação é escolhido conforme deseja-se obter na saída dos neurônios intervalos entre $[0,1]$ ou $[-1,1]$, respectivamente obtidos com a função logística e \tanh . Foi demonstrado empiricamente que, por ser uma função ímpar (anti-simétrica) $\varphi(-x) = -\varphi(x)$, a \tanh exibida na Figura 3.6 (b), propicia uma aprendizagem mais rápida (HAYKIN, 2001; SÁ, 2001). Por outro lado, a função logística aplicada num neurônio da camada de saída de um MLP propicia uma interpretação probabilística da sua resposta a um determinado vetor de entrada. As

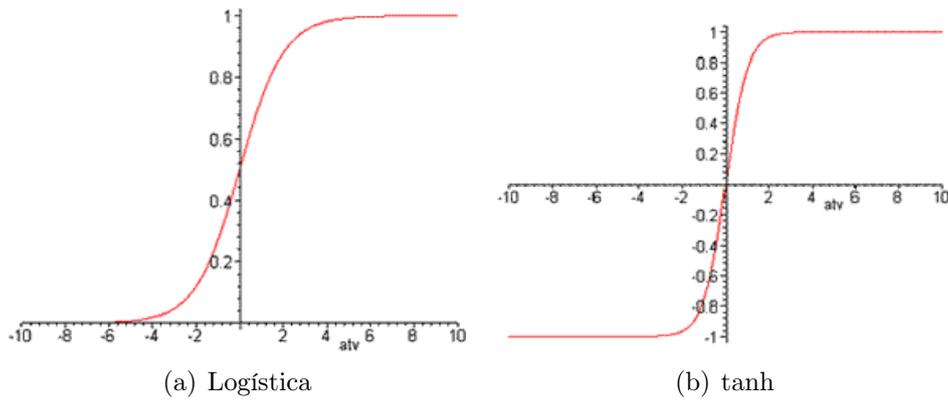


Figura 3.6: Funções de ativação típicas.

funções de ativação citadas são responsáveis pela capacidade da rede MLPs produzir discriminantes não-lineares.

O treinamento de uma rede MLP consiste, basicamente, na minimização do Erro Quadrático Médio (EQM), que é calculado sobre a diferença entre o valor conhecido da classe de um vetor de entrada e o valor de saída gerado pela rede. O treinamento é feito em épocas $t = 1, \dots, T$ e baseia-se no algoritmo de retro-propagação (*backpropagation*) do erro, que é uma generalização do algoritmo mínimo quadrado médio (LMS, do inglês *Least Mean Square*). Uma época de treinamento corresponde à atualização das saídas dos neurônios das diversas camadas no sentido direto (*forward*), ou seja, da entrada para a saída da rede. Em seguida, é calculado o gradiente $\delta_j(t) = \varphi'[v_j(t)]e_j(t)$, em que $e_j(t)$ representa o erro de um neurônio j em uma etapa t do treinamento da rede e $\varphi'(\cdot)$ é a derivada da função de ativação. Posteriormente são atualizados os pesos sinápticos no sentido reverso (*backward*), daí o nome do algoritmo. A regra de aprendizagem (regra delta generalizada) é utilizada para atualização dos pesos sinápticos w dos neurônios:

$$w(t+1) = w(t) + \alpha[\Delta w(t)] + \eta\delta(t)y(t), \quad (3.15)$$

em que η é a taxa de aprendizagem e α é uma constante de momento (HAYKIN, 2001). Esse modelo de aprendizagem leva em consideração a velocidade de aprendizagem $0 \leq \eta \leq 1$ e um termo de momento $0 \leq \alpha \leq 1$ que ajuda a minimizar oscilações para grandes valores de η .

Os neurônios da camada oculta desempenham um papel fundamental na operação de um MLP com aprendizagem por retro-propagação, porque agem

como detetores de características. Conforme o processo de aprendizagem avança, os neurônios ocultos começam gradualmente a descobrir as características que descrevem os dados de treinamento (HAYKIN, 2001). Em aplicações prática, é raro utilizar mais do que duas camadas escondidas (SÁ, 2001).

Embora a rede MLP seja conhecida por sua capacidade de generalização (DUDA; HART; STORK, 2000; HAYKIN, 2001), existem algumas limitações:

- ▶ o algoritmo de treinamento dos MLPs garante a convergência para um erro mínimo de treinamento, entretanto não garante que este erro seja um mínimo global. Além disso, a convergência depende fortemente dos parâmetros iniciais;
- ▶ para obter uma boa estimativa dos parâmetros livres (os pesos sinápticos), cuja quantidade é influenciada diretamente pela dimensionalidade dos dados, é necessário que o número de amostras de treinamento seja representativo.
- ▶ ao contrário dos classificadores paramétricos, que permitem uma análise inversa da sua resposta a um determinado vetor de entrada, para os MLPs não existe uma caminho inverso que responda à pergunta: "dada a classe, quais possíveis vetores de entrada podem gerar uma resposta equivalente?".

3.4.2 SVM (Support Vector Machine)

A Máquina de Vetor de Suporte (SVM, do inglês *Support Vector Machine*), inicialmente proposta por Vapnik (HAYKIN, 2001), é um processo de otimização quadrática que utiliza o mapeamento de vetores de entrada num espaço de características de maior dimensão. No novo espaço é construído um hiperplano ótimo de separação utilizando apenas os vetores de suporte, ou seja, aqueles que maximizam a distância entre os hiperplanos, ou margem. A idéia central é ajustar uma função discriminante linear para uso otimizado da informação de separabilidade dos padrões da fronteira.

A SVM pode ser utilizada na classificação de padrões e na inferência de relações não-lineares entre variáveis. O objetivo da SVM é encontrar uma superfície de decisão ótima entre duas classes através da identificação das amostras de treinamento mais representativas de cada classe (HAYKIN, 2001) como mostra a Figura 3.7. Ao contrário do MLP, a SVM pode lidar muito bem com poucas amostras, porque no seu processo de otimização são utilizadas apenas as amostras na vizinhança da fronteira entre as classes.

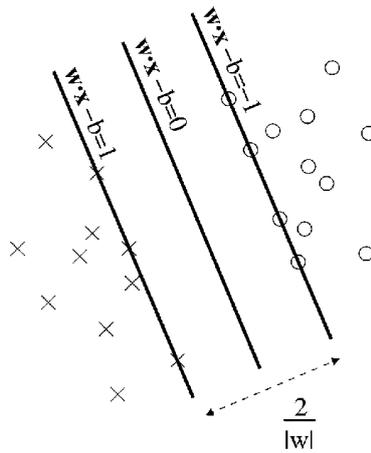


Figura 3.7: Modelo mostrando um hiperplano ótimo formado com os vetores de suporte de duas classes linearmente separáveis.

Considere um conjunto de amostras de treinamento $\mathcal{D} = \{(\mathbf{x}_1, r_1), \dots, (\mathbf{x}_N, r_N)\}$, em que $\mathbf{x}_n \in \mathbb{R}^M$ é o vetor de entrada (ou padrão de entrada) e $r_n \in \{-1, 1\}$ é o rótulo da classe a que pertence aquele vetor, para $n = 1, \dots, N$. A SVM busca encontrar um hiperplano ótimo

$$\mathbf{w}_o \cdot \mathbf{x} - b_o = 0, \quad (3.16)$$

sendo \mathbf{w}_o o vetor peso e b_o o viés ou limiar de classificação ótimos, para gerar um classificador binário linear $f(\mathbf{x}) \in \{-1, 1\}$ definido como

$$f(\mathbf{x}) = \text{sgn}(\mathbf{w}_o \cdot \mathbf{x} - b_o), \quad (3.17)$$

capaz de minimizar o erro de classificação.

Encontrar o classificador $f(\mathbf{x})$ ótimo em SVM é equivalente a solucionar o problema de otimização quadrática

$$\min_{\mathbf{w}, \xi} \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{n=1}^N \xi_n, \quad (3.18)$$

em que C é um parâmetro de regularização e ξ_n são chamadas variáveis "frouxas" que medem o desvio da condição de separabilidade ideal entre as classes (HAYKIN, 2001). O termo $C \sum_n \xi_n$ pode ser considerado como uma medida de parte do erro de classificação (WEBB, 2002).

Haykin (2001) demonstra que a maximização de margem consiste na minimização da norma euclidiana (L_2) do vetor de pesos ideal \mathbf{w}_0 para uma margem ótima, expressada por

$$\rho = \frac{2}{\|\mathbf{w}_0\|}. \quad (3.19)$$

Problemas práticos, em geral, envolvem margens não-linearmente separáveis. Quando se considera um problema de separação não-linear, faz-se uso de funções núcleo para converter a SVM num classificador capaz de gerar superfícies de separação não-lineares. Uma máquina de aprendizagem baseada em núcleo como a da Figura 3.8 é obtida através da implementação de um mapeamento não-linear das entradas $\varphi(\mathbf{x}) = \{\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \dots, \varphi_H(\mathbf{x})\}$ para um espaço diferente, obtendo uma superfície de decisão definida por

$$\sum_{h=0}^H w_h \cdot \varphi_h(\mathbf{x}) = 0, \quad (3.20)$$

sendo que, por definição, $\varphi_0 = 1$ para todo \mathbf{x} , de forma que o respectivo peso w_0 representa o viés b da equação (3.16).

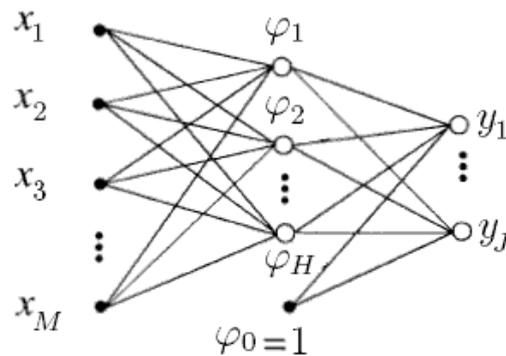


Figura 3.8: Modelo de um classificador com função núcleo representada por φ .

A função núcleo é utilizada para converter um classificador linear em um classificador não-linear. A base teórica, como mostra Haykin (2001), é o teorema de Mercer, o qual define que qualquer função $k(\mathbf{x}, \mathbf{y})$ simétrica, contínua e semi-definida pode ser expressa como o produto interno em um espaço de maior dimensão de dois vetores \mathbf{x} e \mathbf{y} , expressado por

$$k(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}), \quad (3.21)$$

em que φ é uma função para a qual está definido o produto interno.

Diferentes tipos de funções podem ser utilizados como núcleo da SVM:

- i. Polinomial: $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^p$;
- ii. Função de base radial: $k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}\right)$;
- iii. Sigmoidal: $k(\mathbf{x}, \mathbf{y}) = \tanh(\kappa \mathbf{x} \cdot \mathbf{y} + c)$, para $\kappa > 0$ e $c < 0$

O núcleo mais simples é o polinomial. Existe apenas um parâmetro a ser definido que é a ordem do polinômio. Se o núcleo escolhido for uma função de base radial, então a arquitetura da SVM é muito similar à de uma rede RBF. A diferença entre a SVM e a RBF está no procedimento de treinamento. Na rede RBF é feito um mapeamento em um espaço M -dimensional e depois os centros das funções precisam ser estimados através dos dados de treinamento. Na SVM, o número de nós e os centros são resultados de um procedimento de otimização. Finalmente, se a função núcleo escolhida for a tangente hiperbólica, a arquitetura resultante torna-se um caso especial do MLP de duas camadas. Infelizmente, não existe um método estabelecido para a seleção da melhor função núcleo (THEODORIDIS; KOUTROUMBAS, 2003).

3.5 Combinação de Características na Detecção de Manchas

Jain, Duin e Mao (2000) afirmam que a probabilidade de erro de classificação de uma regra de decisão não cresce com o aumento do número de características se o número de amostras de treinamento é suficientemente grande. Levando isso em consideração, podemos aumentar arbitrariamente o número de características utilizando métodos diferentes de extração de características para verificar a hipótese de geração de um discriminante mais eficaz. Diferentes conjuntos de características possibilitam a combinação mostrada na Figura 3.9. Lopes et al. (2006) investigaram esta hipótese utilizando uma combinação de dois conjuntos de características, a saber, características geométricas e características espaciais de textura das manchas de óleo. Os resultados obtidos demonstraram que, para o problema específico, a utilização de características de textura combinadas com as características geométricas contribui para uma melhoria de desempenho de classificação, embora o

aumento o número de características tenha tornado o projeto do classificador mais complexo.

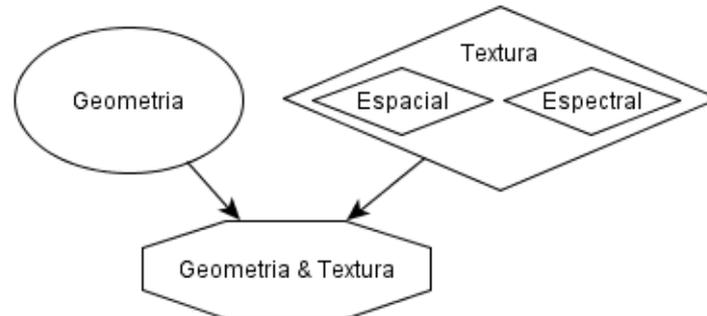


Figura 3.9: Conjuntos de características e as possíveis combinações.

3.5.1 A Maldição da Dimensionalidade

Na prática, dependendo do número de amostras disponíveis, o aumento de dimensionalidade do problema, através da inclusão de novos descritores, pode fazer com que o desempenho do classificador seja degradado. De fato, quando se utiliza um MLP, por exemplo, é necessário aumentar o número de neurônios na camada escondida para tentar melhorar o desempenho da rede. Este fenômeno é comentado em Duda, Hart e Stork (2000), que afirmam que para um determinado problema de classificação deve haver um compromisso entre a taxa de classificação e o poder de generalização do classificador. Em outras palavras, quanto maior a dimensionalidade do problema, maior deverá ser a complexidade do mesmo em termos de número de parâmetros a serem estimados. Por um lado, pode-se projetar uma rede neural, por exemplo, com tantos parâmetros quanto sejam necessários para obter altas taxas de classificação correta. A contrapartida é que o classificador pode, na pior das hipóteses, ser levado ao sobreajustamento que é responsável pela perda de generalização do classificador. Barron (1993) demonstra que as RNAs levam vantagem em problemas de grande dimensionalidade quando são treinadas com parcimônia e de forma otimizada. Em outras palavras, mantendo-se relativamente pequeno o número de parâmetros livres, o conjunto de funções ajustáveis geradas durante o treinamento de uma RNA consegue obter erros de aproximação menores do que os métodos de regressão tradicionais, baseados em funções fixas.

3.6 Avaliação de Desempenho de Classificadores

Um critério comumente utilizado na avaliação de desempenho de um classificador é a análise do erro médio de classificação em sucessivos procedimentos de treinamento e teste (WEBB, 2002). Entretanto, outros critérios devem ser levados em consideração como, por exemplo, a variância do erro, o tamanho do conjunto de dados utilizado no treinamento, o tamanho do conjunto de dados utilizado nos testes e a quantidade de falsos-positivos e falsos-negativos.

Quando o conjunto de dados é pequeno, como é o caso do utilizado neste trabalho, um método de validação cruzada (*cross-validation*) possibilita numa estimativa mais confiável da taxa de erro (DUDA; HART; STORK, 2000). Na validação cruzada são criados conjuntos de treinamento e de teste a partir de repetidas subamostragens do conjunto original. O conjunto de teste é utilizado para obter uma estimativa do erro de generalização do classificador. O resultado dessa estimativa é freqüentemente usado na escolha de um dentre vários modelos, por exemplo, de arquiteturas diferentes de redes neurais. Algumas das técnicas de validação cruzada mais utilizadas são o *leave-one-out* e o *bootstrap*.

Na técnica *leave-one-out*, a cada rodada de treinamento e teste uma amostra é retirada do conjunto, conforme ilustrado na Figura 3.10 (a). Essa amostra de teste é utilizada para estimar a generalização do classificador que foi treinado com o restante das $N - 1$ amostras.

O *bootstrap* (EFRON; TIBSHIRANI, 1993) é semelhante ao *leave-one-out*, porém os subconjuntos são gerados de forma aleatória e não circularmente. A amostragem é feita com reposição, isto é, um determinado exemplo pode estar repetido quer no conjunto de treino quer no de dados. Como pode ser observado na Figura 3.10 (b), o conjunto de teste é formado por vetores aleatoriamente retirados do conjunto de amostras em cada etapa de treinamento e teste. É uma prática comum utilizar de 10% a 30% das amostras para formar o conjunto de validação do classificador. A estimativa de erro pelo método *bootstrap* produz uma variância menor que no *leave-one-out*. Além disso, o método *bootstrap* é muito útil quando o número de amostras é pequeno (JAIN; DUIN; MAO, 2000).

Segundo Duda, Hart e Stork (2000), devido à variação dos elementos dos conjuntos de treinamento, selecionados aleatoriamente, o classificador poderá ser favorecido ou penalizado na estimativa do erro. O valor esperado do erro é definido

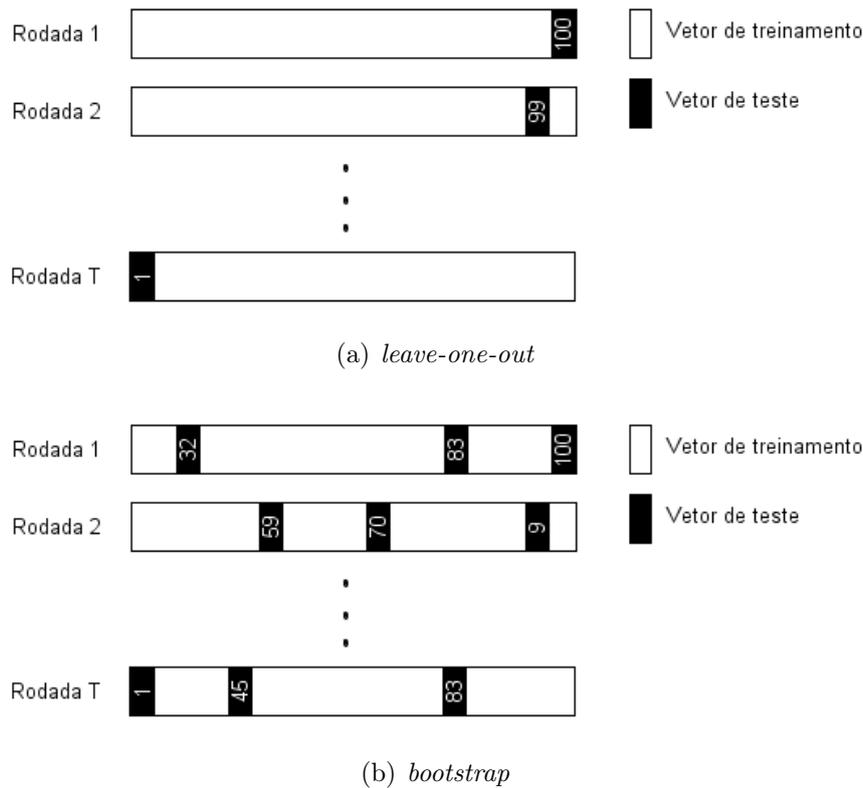


Figura 3.10: Representação gráfica dos métodos de validação cruzada mais comuns.

como

$$E\{[g(\mathbf{x}, \mathcal{D}) - F(\mathbf{x})]^2\} = \underbrace{E\{[g(\mathbf{x}, \mathcal{D}) - F(\mathbf{x})]\}^2}_{(\text{viés})^2} + \underbrace{E\{(g(\mathbf{x}, \mathcal{D}) - E[g(\mathbf{x}, \mathcal{D})])^2\}}_{\text{variância}}, \quad (3.22)$$

em que $g(\mathbf{x}, \mathcal{D})$ é predição de uma função de aprendizagem treinada sobre um determinado conjunto \mathcal{D} e $F(\mathbf{x})$ é a saída previamente conhecida para o padrão de entrada \mathbf{x} . Este erro possui dois componentes. O primeiro termo da soma representa o viés e o segundo a variância. Um viés pequeno significa que houve um bom ajustamento da função $F(\mathbf{x})$ pela função $g(\mathbf{x}, \mathcal{D})$. Uma variância pequena significa que a predição obtida com $g(\mathbf{x}, \mathcal{D})$ varia pouco para diferentes conjuntos de treinamento.

Em outras palavras, uma predição ruim está relacionada a um viés alto, possivelmente causado por um sobreajustamento. A variância mede a precisão do

classificador, ou seja, quanto maior ela for, menos confiável será a predição. Ambos são afetados pelo número de parâmetros livres a serem estimados e pelo número de amostras disponíveis, porém de maneira inversa um do outro (dilema viés-variância).

3.7 Conclusão

Neste capítulo foram apresentados classificadores que possuem características determinantes do seu sucesso na solução de um determinado tipo de problema. Para classes linearmente separáveis, o classificador *Naïve Bayes* é o mais indicado, por sua simplicidade e baixo custo computacional na geração de discriminantes lineares. Entretanto, as classes de manchas possuem grande superposição, o que limita assintoticamente o desempenho do *Naïve Bayes*.

O classificador KNN possui grande habilidade para estimar PDFs, o que em princípio possibilita um bom desempenho de classificação. Entretanto, em sistemas com muitos dados amostrais o custo computacional é muito elevado.

Os MLPs são considerados excelentes generalizadores. No entanto, para que possam generalizar bem, é necessário que a fase de treinamento seja cuidadosamente planejada. Um dos cuidados para treinar um MLP consiste em escolher o número de neurônios e estimar de forma confiável os parâmetros da rede, ou seja, os pesos sinápticos desses neurônios. Infelizmente, na prática, existe pouca possibilidade de controle sobre os dados. Portanto, é fundamental realizar diversos testes de classificação com diferentes arquiteturas de rede.

A principal vantagem da SVM é a sua capacidade de maximização de margem do discriminante utilizando poucas amostras. Entretanto, o conjunto de amostras de treinamento deve ser bastante representativo. Infelizmente, em aplicações práticas, não existe uma maneira de mensurar essa representatividade a menos que se tenha algum conhecimento prévio do problema.

O processo de validação cruzada *bootstrap* produz T classificadores, cada um com sua taxa de erro de classificação individual. Se o conjunto de dados é pequeno, a quantidade de testes T deve ser significativamente grande (SÁ, 2001). Numa aplicação prática, é necessário escolher qual desses T classificadores será utilizado para prever a classe da mancha. O classificador com melhor desempenho obtido durante os testes não é necessariamente a melhor escolha. A estabilização dos pesos durante o treinamento de um MLP garante que o EQM durante o treinamento

atingiu um mínimo, mas não garante que esse mínimo seja global (HAYKIN, 2001). Por outro lado, simplesmente escolher um classificador com erro de treinamento muito pequeno pode mascarar o sobreajustamento, ou seja, a incapacidade de generalização. Na prática, a escolha do classificador com desempenho próximo ao da média é uma hipótese a ser considerada, porém o classificador escolhido pode ter sido beneficiado por uma escolha favorável dos dados de testes utilizados na sua validação. Além disso, a escolha do classificador torna-se uma tarefa complexa e tediosa, pois várias arquiteturas e condições iniciais devem ser testadas (SÁ, 2001).

No próximo capítulo será apresentada a metodologia empregada na melhoria de desempenho da detecção de manchas. São discutidos algoritmos de combinação que possibilitam obter melhor desempenho de classificação com variância reduzida, mesmo em condições de grande dimensionalidade dos dados.

Melhoria de Desempenho na Detecção de Manchas de Óleo Através da Combinação de Classificadores

A combinação de classificadores (ou comitê) é um recurso utilizado para melhorar arbitrariamente o desempenho de classificação através da combinação da resposta de múltiplos classificadores base (HAYKIN, 2001). Os métodos de combinação apresentados neste capítulo baseiam-se nos algoritmos de combinação *boosting* (FREUND; SCHAPIRE, 1999) e *bagging* (BREIMAN, 1996). Estes métodos foram escolhidos por apresentarem excelentes resultados com algoritmos relativamente simples.

4.1 Introdução

O processo de validação cruzada *bootstrap*, conforme comentado no capítulo anterior, produz T classificadores. Os algoritmos de combinação de classificadores tiram proveito de todo o esforço computacional empregado no treinamento desses T classificadores sem descartar nenhum deles. Cada classificador, ou função base, é um especialista treinado com um subconjunto dos dados de entrada. Pode-se combinar as saídas desses classificadores utilizando, por exemplo, uma votação de maioria. O método *bagging*, apresentado por Breiman (1996), baseia-se neste princípio.

A melhoria de desempenho de um classificador pode ser obtida através da reutilização ou seleção dos dados utilizando o método combinação e ponderação

adaptativa (ARCING, do inglês *adaptive reweighting and combining*) (DUDA; HART; STORK, 2000). Um modelo simples da metodologia de combinação de classificadores explorada pelo ARCING é mostrada na Figura 4.1. Classificadores base são treinados, cada um sobre uma distribuição diferente de um mesmo conjunto de dados. Os algoritmos dos classificadores base do comitê podem ser diferentes (comitê heterogêneo), porém a utilização de um mesmo algoritmo para todos os classificadores base (comitê homogêneo) torna o processo mais simples. KUNCHEVA e WHITAKER (2003) realizaram estudos para medir como a diversidade em comitês de classificadores afeta o resultado da predição. Neste trabalho, dez métricas são apresentadas e os resultados da pesquisa sugerem uma relação direta entre a diversidade de classificadores base e a precisão dos comitês sobre os conjuntos de dados experimentais estudados. Os autores, porém, lançam dúvidas quanto à utilidade prática das métricas estudadas. Em outras palavras, embora a diversidade seja importante para a construção de comitês de classificadores, ainda não existe uma solução eficaz para medir essa diversidade.

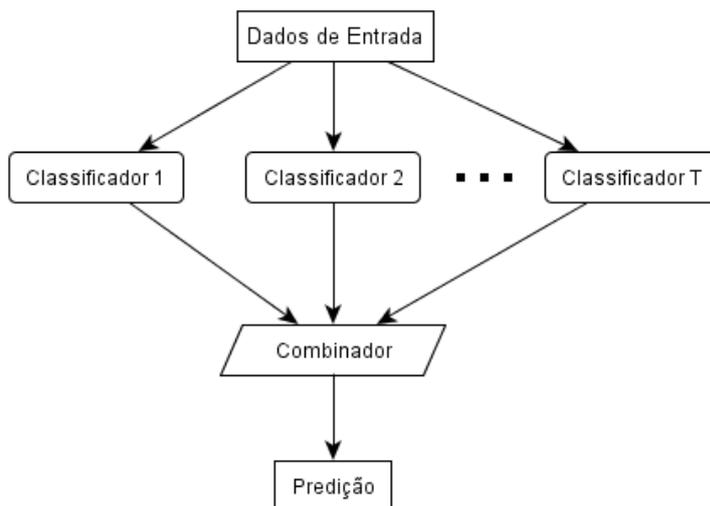


Figura 4.1: Modelo de combinação de classificadores. Os dados de entrada são submetidos a diversos classificadores. Um módulo combinador utiliza a predição individual dos especialistas para gerar uma única saída combinada, tornando a predição final mais apurada.

Neste trabalho é enfatizado o método de combinação de classificadores por reforço (*boosting*) que surgiu na década de 90 (SCHAPIRE, 1990; FREUND; SCHAPIRE, 1997). Este método e suas diversas variações são utilizados na identificação de proteínas (BITTENCOURT, 2005; FENG; CAI; CHOU, 2005; YIN; LIU; HAN, 2005; CAI et al., 2006); na classificação de tumores (DETTLING; BÜHLMANN, 2003); na classificação

de regiões em imagens de sensoriamento remoto (BAILLY; ARNAUD; PUECH, 2007); e na detecção de manchas de óleo no mar (RAMALHO; MEDEIROS, 2006, 2007).

4.2 Bagging

O método de combinação de classificadores *bagging* (BREIMAN, 1996), um nome derivado de *bootstrap aggregation* (agregação de vários *bootstrap*), confere maior estabilidade e menor erro de classificação a algoritmos de treinamento instáveis. São considerados instáveis os algoritmos de treinamento cuja resposta varia significativamente com pequenas mudanças nas condições iniciais, como ocorre com os MLPs. O viés final do comitê é o mesmo relativo ao do classificador base. Entretanto, a variância do erro de classificação é significativamente reduzida. Um modelo do algoritmo é ilustrado na Figura 4.2.

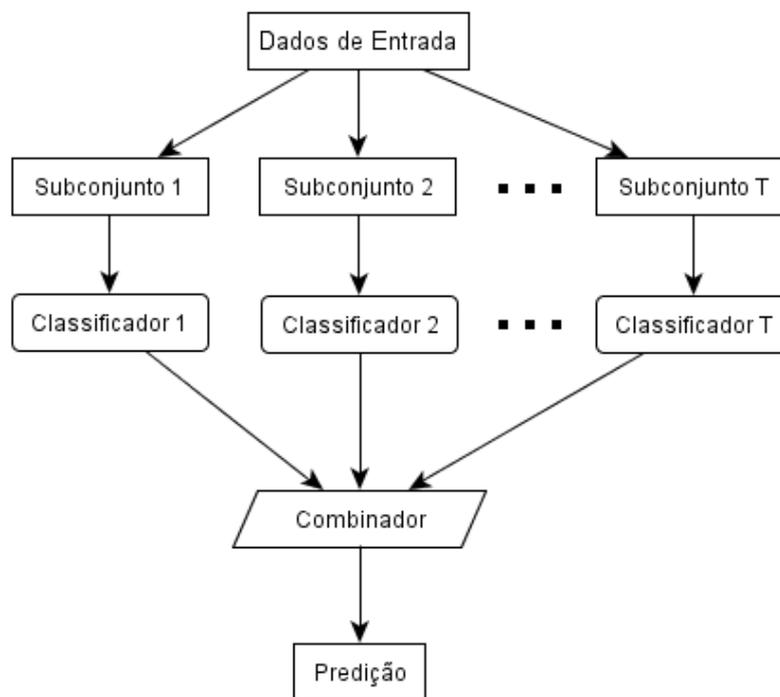


Figura 4.2: Modelo de combinação *bagging*. O combinador utiliza um método de votação de maioria.

O algoritmo do *bagging* gera conjuntos de treinamento aleatoriamente e permite gerar classificadores em paralelo (WEBB, 2002). O mesmo classificador base é executado T vezes e, em cada etapa, é utilizada uma amostragem diferente do conjunto de treinamento. Cada um desses subconjuntos \mathcal{SD} é criado a partir da subamostragem aleatória (*bootstrap*) com reposição de $N' < N$ amostras do conjunto

\mathcal{D} de amostras de treinamento (DUDA; HART; STORK, 2000). Em geral, este método de combinação melhora o desempenho de um algoritmo de classificação instável, porque o erro da predição final é diluído entre os classificadores base.

O resultado final de classificação no *bagging* é obtido através de votação de maioria. A cada execução do algoritmo, para cada elemento do conjunto de treinamento, é armazenado o resultado de classificação, ou hipótese, obtido na etapa t . No final, para cada elemento, escolhe-se o resultado de classificação mais freqüente nas T etapas, gerando a hipótese \mathcal{H} ou predição final. Uma visão geral do algoritmo do *bagging* é apresentada na Figura 4.3.

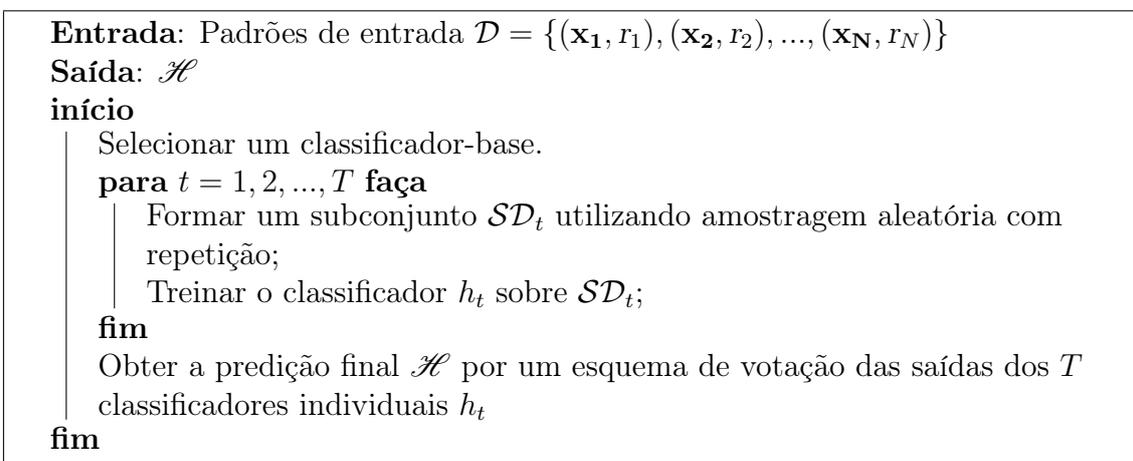


Figura 4.3: Algoritmo *bagging*.

A predição pelo *bagging* é a forma mais elementar de combinação ou integração das saídas de um conjunto de classificadores (DUDA; HART; STORK, 2000). Variações do algoritmo podem ser obtidas simplesmente modificando o esquema de votação para uma combinação das saídas através de produto ou soma, por exemplo.

4.3 Boosting

O método de combinação de classificadores por reforço (*boosting*) foi apresentado pela primeira vez na década de 90 por Schapire (1990). Este método baseia-se num modelo de aprendizagem fraca (*weak learning*), conceito introduzido por Kearns e Valiant (1989), o qual define que o classificador base necessita gerar uma hipótese com taxa de erro ligeiramente inferior a 50%, ou seja, com desempenho apenas um pouco melhor do que uma estimativa aleatória (HAYKIN, 2001). O *boosting* é um procedimento determinístico que gera conjuntos de treinamento seqüencialmente baseado no resultado da iteração anterior (WEBB, 2002). Ao contrário do *bagging*,

no *boosting* cada especialista é treinado com uma distribuição estatística diferente dos dados. Embora utilize um modelo de aprendizagem fraca, este método pode ser utilizado para melhorar o desempenho de qualquer algoritmo de aprendizagem (HAYKIN, 2001), através da maximização de margem que ele proporciona e que é similar ao executado pelo SVM. O objetivo de ambos é encontrar um classificador linear em um espaço de maior dimensionalidade, conforme afirma Schapire et al. (1998).

Segundo Webb (2002), assim como a SVM, o *boosting* é um classificador que focaliza os vetores que mais contribuem para o erro de classificação, correspondentes àqueles padrões mais próximos da fronteira de decisão entre as classes. Entretanto, existem diferenças significativas entre os dois classificadores. Para a minimização da margem da fronteira de decisão, a SVM baseia-se em uma norma euclideana (L_2) enquanto o *boosting* baseia-se em uma norma de ordem infinita (L_∞) (FREUND; SCHAPIRE, 1999). Conforme explicam Freund e Schapire (1999), diferentes normas podem resultar em diferentes margens, dando ao *boosting* alguma vantagem sobre o SVM principalmente em espaços de grande dimensão. Os autores também mostram que o custo computacional da SVM corresponde a uma programação quadrática, enquanto no *boosting* trata-se de uma programação linear. Outra diferença básica relatada, é a utilização de funções núcleo na SVM para permitir lidar com espaços não-lineares. Já no *boosting* esse trabalho é conseguido pela redistribuição das amostras de treinamento. O algoritmo *boosting* produz discriminantes complexos como ilustrado na Figura 4.4.

O *boosting* adaptativo (AdaBoost, do inglês *Adaptive Boosting*) (FREUND; SCHAPIRE, 1999) surgiu para resolver alguns problemas práticos do algoritmo *boosting* original de Schapire (1990). O novo algoritmo possui a propriedade de se adaptar aos vetores de treinamento, o que confere ao *boosting* maior estabilidade. O algoritmo AdaBoost é um método de reforço por ponderação, no qual os exemplos são amostrados de acordo com uma determinada distribuição de probabilidade. A probabilidade de uma determinada amostra ser utilizada num passo posterior do treinamento é diretamente proporcional à sua contribuição para o erro de classificação.

O algoritmo original do *boosting* (SCHAPIRE, 1990) constrói a hipótese final utilizando uma votação de maioria. Já o algoritmo de Freund e Schapire (1999) busca aproximar uma combinação linear de classificadores através de ponderadores

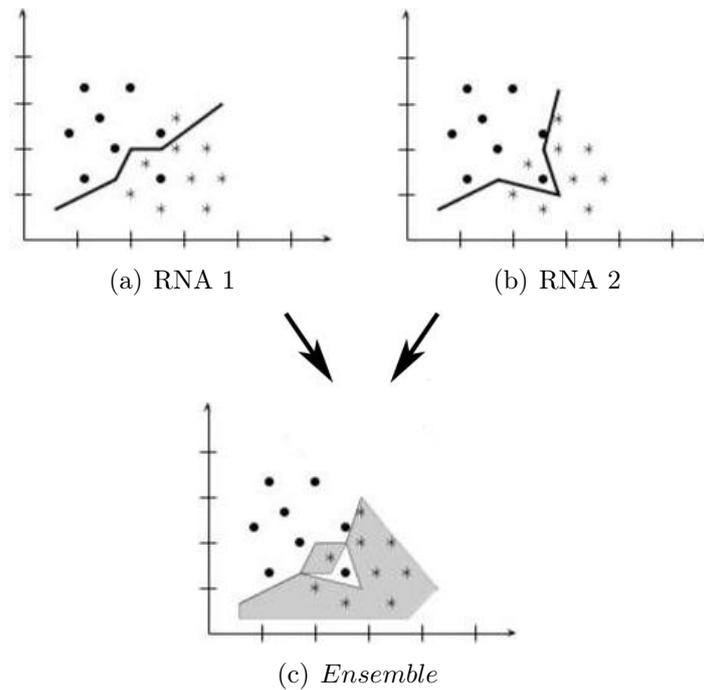


Figura 4.4: Superfície discriminante do comitê (c) gerada pela combinação de duas redes neurais (a,b) utilizando o método *boosting*.

c_t proporcionais aos erros de classificação das hipóteses h_t , ou classificadores base, ou seja

$$\mathcal{H}(\mathbf{x}) = \sum_{t=1}^T c_t h_t(\mathbf{x}). \quad (4.1)$$

A idéia básica do AdaBoost (ver Figura 4.5) consiste em executar o mesmo algoritmo de treinamento T vezes sobre diferentes distribuições dos vetores de entrada \mathbf{x} . Na primeira etapa os pesos $d_t(n)$ da distribuição são iguais para todos os vetores de treinamento \mathbf{x}_n , ou seja, o procedimento é iniciado com uma distribuição uniforme $d_t(n) = 1/N$, em que $n = 1, \dots, N$. No final de cada etapa, somente aqueles vetores \mathbf{x}_n que contribuem para um maior erro de classificação têm seus pesos $d_t(n)$ aumentados. Um vetor \mathbf{x}_n nas proximidades da fronteira entre as classes possui a tendência de ter seu peso $d_t(n)$ aumentado rapidamente devido a sua maior contribuição para o erro do classificador base. Um novo conjunto de treinamento de mesmo tamanho do conjunto original é então gerado, porém com uma ocorrência maior dos vetores \mathbf{x}_n de maior peso $d_t(n)$. Para cada etapa t do algoritmo, é armazenado o erro ponderado de classificação e_t e c_t que é uma normalização do

erro utilizada como fator ponderador na geração da hipótese final. No final do processo de treinamento, para cada vetor de entrada, escolhe-se o rótulo decorrente da combinação linear das hipóteses, ou classificadores base, ponderadas pelo peso c_t . O algoritmo AdaBoost discreto binário (considera apenas duas classes) é mostrado na Figura 4.6.

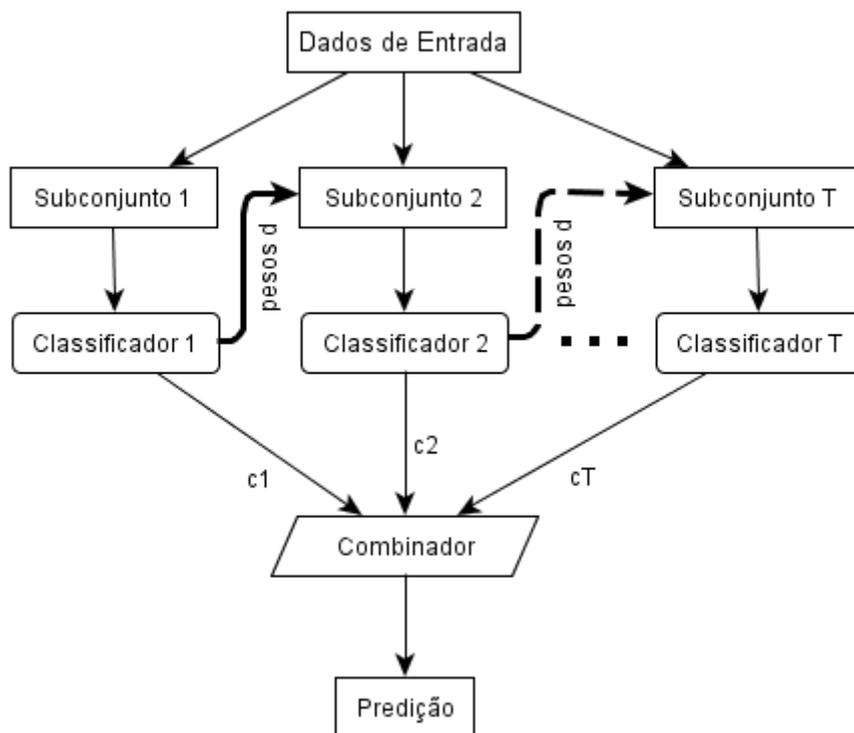


Figura 4.5: Modelo de combinação *boosting*. O combinador utiliza o método de votação ponderada, baseada na taxa de erro de classificação dos classificadores intermediários.

Uma das grandes dificuldades na implementação desse algoritmo consiste em definir uma estratégia adequada para modificar a probabilidade de reamostragem de um dado vetor de entrada com base na sua contribuição para o erro de classificação. Quando o classificador base é uma RNA, pode-se utilizar a distribuição d do AdaBoost para alterar na mesma proporção as entradas dos neurônios da RNA, como se fossem pesos de uma camada adicional de neurônios. Isto, porém, requer uma mudança significativa no algoritmo do classificador base. Uma opção mais simples consiste em realizar uma reamostragem baseada nos pesos d_t (FREUND; SCHAPIRE, 1999). Vetores de entrada com pesos mais altos terão mais chance de serem utilizados na próxima etapa do algoritmo. Qualquer que seja a estratégia adotada, o objetivo final é fazer com que o classificador base da etapa seguinte seja

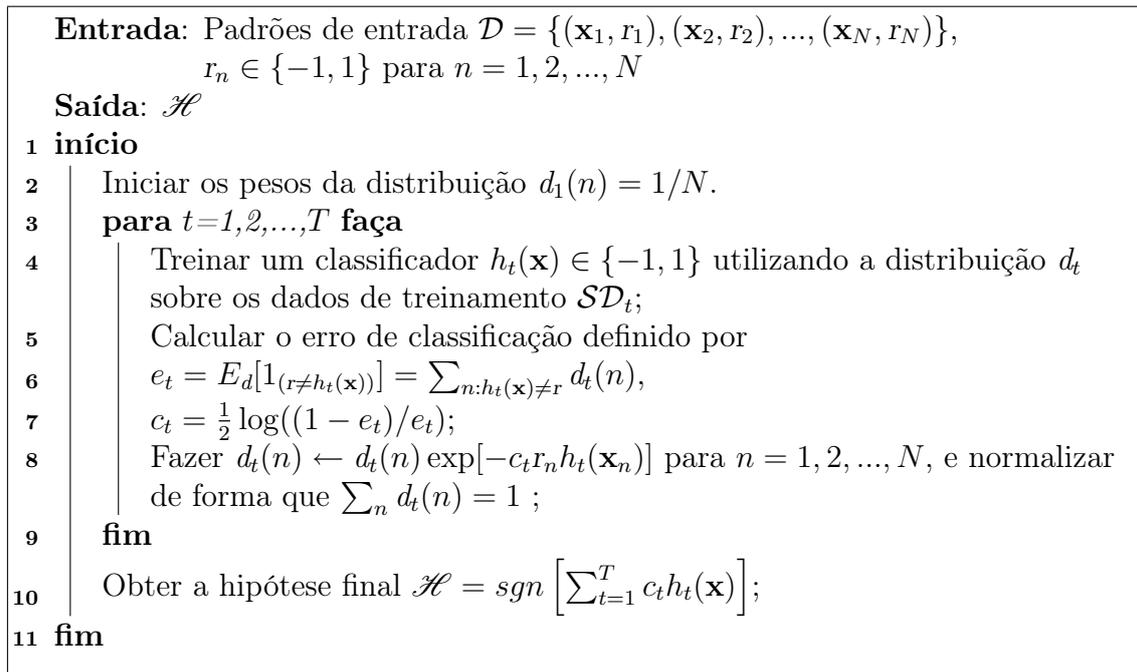


Figura 4.6: Algoritmo do AdaBoost discreto binário (FREUND; SCHAPIRE, 1999).

treinado com ênfase nos vetores que mais contribuem para o erro de classificação.

Em geral, um número limitado de classificadores base ($T \cong 10$) é suficiente para gerar uma hipótese confiável, mas este número depende diretamente da complexidade do conjunto de dados. A definição do número de classificadores base, ou seja, do número de etapas de treinamento pode ser feita através de métodos empíricos (DETTING; BÜHLMANN, 2003). Freund e Schapire (1999) relatam que mesmo sendo o número de classificadores base muito grande ($T \gg 10$), não se corre o risco de prejudicar a capacidade de generalização do comitê devido ao sobreajustamento.

Percebe-se que o erro do classificador base é medido com base na distribuição definida pelos pesos d_t dos elementos não corretamente classificados, definido por Freund e Schapire (1999) através da equação (4.2). Dessa forma, o erro é uma estimativa obtida através da soma das probabilidades dos vetores que foram classificados de forma errada, ou seja

$$e_t = E_d[1_{(r \neq h_t(\mathbf{x}))}] = \sum_{n: h_t(\mathbf{x}) \neq r} d_t(n). \quad (4.2)$$

Este erro pode ser utilizado como critério de parada do treinamento quando

atinge um valor mínimo desejado ou quando estabiliza. Assim, pode-se diminuir o número de etapas necessárias e, conseqüentemente, diminuir a quantidade de classificadores base, tempo e memória necessária para o treinamento.

Quando comparado a outros métodos de classificação como RNA ou SVM, o *boosting* apresenta uma clara vantagem quanto à sua parametrização. Pode ser utilizado como classificador base qualquer algoritmo de classificação que produza uma taxa de classificação ligeiramente superior a 50%. Entretanto, foi provado experimentalmente que classificadores base com melhor desempenho também podem tirar vantagem do *boosting* (FREUND; SCHAPIRE, 1999). O algoritmo do *boosting* não possui tantos parâmetros livres quanto os MLPs e não necessita de uma sofisticada otimização não-linear presente na SVM (DETTLING; BÜHLMANN, 2003). No *boosting*, depois de escolhido o classificador base, existe apenas um parâmetro de ajuste que é o número T de classificadores a serem treinados. Como o *boosting* é, em geral, resistente ao sobreajustamento, a determinação de T não é muito crítica. Uma abordagem empírica para a escolha desse parâmetro é apresentada por Bühlmann (2003), Dettling e Bühlmann (2003). Ao contrário do treinamento de MLPs, no *boosting* não é necessário escolher um dos classificadores treinados e descartar os outros, já que todo o esforço de treinamento dos T classificadores é combinado para ser utilizado na predição final. Essas características fazem do *boosting* uma ferramenta útil em muitas aplicações práticas em que não existe a preocupação com memória e tempo de processamento.

4.4 Logitboost

O algoritmo *boosting* aditivo logístico (LogitBoost, do inglês *Additive Logistic Boosting*), foi proposto por Friedman, Hastie e Tibshirani (2000) para reduzir linearmente o erro de treinamento. O LogitBoost possibilita uma diminuição do viés (KOTSIANTIS; PANAYIOTIS, 2005), promovendo ainda uma melhoria na capacidade de generalização do algoritmo. Este algoritmo baseia-se na observação de que o AdaBoost é, em essência, um modelo de ajuste de uma regressão logística aditiva (F) aos dados de treinamento (KOTSIANTIS; PANAYIOTIS, 2005). Friedman, Hastie e Tibshirani (2000) fizeram uma interpretação estatística do algoritmo AdaBoost introduzindo uma probabilidade conjunta (*log-likelihood*)

$$p(\mathbf{x}) = P(\mathcal{C}_j|\mathbf{x}) = \frac{e^{F(\mathbf{x})}}{e^{F(\mathbf{x})} + e^{-F(\mathbf{x})}}, \quad (4.3)$$

que representa a probabilidade $p(\mathbf{x})$ de um vetor de entrada pertencer a uma determinada classe \mathcal{C}_j . Esta probabilidade é utilizada para modificar os pesos d_t da distribuição em cada etapa do algoritmo.

O algoritmo AdaBoost real binário minimiza o critério de custo exponencial

$$J(h_t) = E(e^{-rh_t(\mathbf{x})}), \quad (4.4)$$

em que $r \in \{-1, 1\}$ é o rótulo conhecido da classe e $h_t \in \{-1, 1\}$ é a hipótese numa dada etapa de treinamento para o vetor de entrada \mathbf{x} . No LogitBoost o critério de custo a ser minimizado utiliza um modelo de regressão logístico aditivo (*log-likelihood*)

$$J(h_t) = E(-\log(1 + e^{-2rh_t(\mathbf{x})})). \quad (4.5)$$

A suavização do erro no LogitBoost proporciona uma melhora significativa no desempenho de classificação e na diminuição da variância como evidenciam os resultados experimentais exibidos no Capítulo 5. O algoritmo exibido na Figura 4.7 resume o procedimento do LogitBoost.

	Entrada: Padrões de entrada $\mathcal{D} = \{(\mathbf{x}_1, r_1), (\mathbf{x}_2, r_2), \dots, (\mathbf{x}_N, r_N)\}$, $r_n \in \{0, 1\}$ para $n = 1, 2, \dots, N$
	Saída: Predição \mathcal{H}
1	início
2	Iniciar os pesos da distribuição $d_1(n) = 1/N$,
3	$\mathcal{H}(\mathbf{x}) = 0$ e as probabilidades $p(\mathbf{x}_n) = 1/2$.
4	para $t = 1, 2, \dots, T$ faça
5	Calcular a resposta probabilística e os pesos:
6	$z_n = \frac{r_n - p(\mathbf{x}_n)}{p(\mathbf{x}_n)(1 - p(\mathbf{x}_n))}$,
7	$d_t(n) = p(\mathbf{x}_n)(1 - p(\mathbf{x}_n))$;
8	Treinar um classificador $h_t(\mathbf{x})$ utilizando a distribuição d_t para gerar os dados de treinamento \mathcal{SD}_t com resposta z_n ;
9	Atualizar $p(\mathbf{x}) = \frac{e^{\mathcal{H}(\mathbf{x})}}{e^{\mathcal{H}(\mathbf{x})} + e^{-\mathcal{H}(\mathbf{x})}}$ e $\mathcal{H}(\mathbf{x}) \leftarrow \mathcal{H}(\mathbf{x}) + \frac{1}{2}h_t(\mathbf{x})$;
10	fim
11	Obter a hipótese final $\text{sgn}[\mathcal{H}(\mathbf{x})] = \text{sgn}\left[\sum_{t=1}^T h_t(\mathbf{x})\right]$;
12	fim

Figura 4.7: Algoritmo LogitBoost binário.

4.5 O AdaBoost Adaptado para a Combinação de Características

Yin, Liu e Han (2005) propuseram uma variante do AdaBoost, a partir da necessidade de combinar ao mesmo tempo classificadores e características obtidas por diferentes métodos de extração. Os autores não especificaram um nome para seu algoritmo, portanto, neste trabalho, o método é chamado de *Variant AdaBoost*. Segundo Yin, Liu e Han (2005), através do *Variant AdaBoost* é possível utilizar características geradas por diferentes métodos de extração para produzir um conjunto de classificadores especializados em cada conjunto. Como uma amostra pode ser melhor representada por um determinado conjunto de características, o algoritmo *Variant AdaBoost* proporciona, além da combinação de classificadores, uma forma de combinação das características mais representativas de cada amostra.

O método, ilustrado na Figura 4.8, consiste na modificação da linha 4 do algoritmo AdaBoost (ver Figura 4.6). Em cada etapa de treinamento, é feito um procedimento intermediário de combinação de classificadores utilizando diferentes subconjuntos de características num processo semelhante ao *bagging*. Por meio de votação ponderada desses classificadores intermediários, é selecionado o conjunto de características que apresenta o menor erro de classificação neste passo. A contrapartida é um expressivo aumento no custo computacional e no uso de memória. A implementação do algoritmo é mais complicada do que no AdaBoost. Esses fatores, entretanto, representam uma desvantagem mínima dada a grande utilidade prática do método.

Em cada uma das T etapas de treinamento do *Variant AdaBoost*, são construídos B classificadores h_t^b , cada um sobre um subconjunto de características formado por um método diferente de extração. Através de uma votação ponderada definida por

$$h_t(\mathbf{x}) = \frac{1}{B} \sum_{b=1}^B h_t^b(\mathbf{x}), \quad (4.6)$$

é escolhido o resultado da classificação h_t desta etapa, sendo que o conjunto de características com menor poder de separabilidade exercerá maior influência sobre a taxa de erro. Um peso inversamente proporcional ao erro de classificação dessa etapa é atribuído ao vetor de entrada. No final, para cada elemento do vetor de entrada, escolhe-se o resultado decorrente da maior soma dos pesos. Na Figura

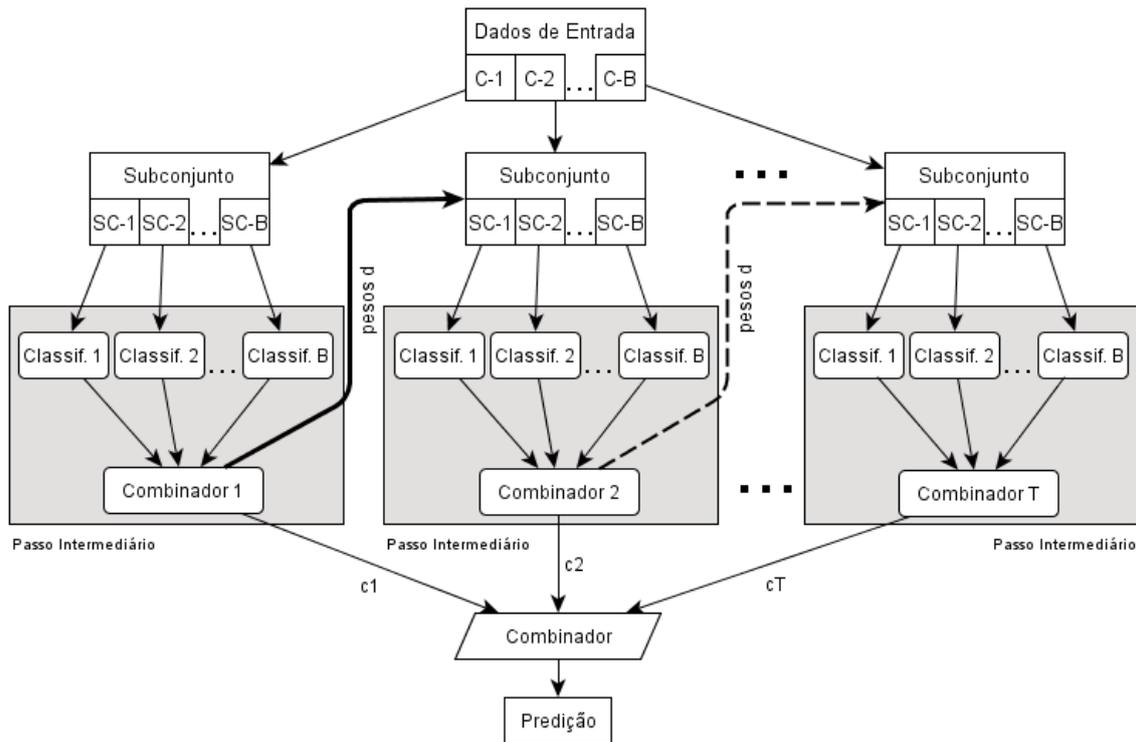


Figura 4.8: Modelo de combinação *Variant AdaBoost*. Este método emprega uma fase intermediária de combinação para cada uma das T etapas. Os diferentes conjuntos de características são apresentados individualmente aos B classificadores. No restante do processo ele é similar ao AdaBoost.

4.9 é apresentado o algoritmo do *Variant AdaBoost* para melhor compreensão. As mudanças no algoritmo AdaBoost original aparecem nas linhas 4 e 5.

A implementação desse algoritmo é mais complexa e exige mais memória e tempo de treinamento do que os demais algoritmos de *boosting* citados. São necessários pelo menos dois conjuntos diferentes de características para gerar o classificador intermediário. Caso contrário, o método recai sobre o algoritmo AdaBoost. Yin, Liu e Han (2005) não especificaram um número mínimo de conjuntos de características. Porém, no seu trabalho, foram realizados experimentos com 3, 4 e 5 conjuntos diferentes. Os autores sugerem que estes conjuntos sejam gerados por métodos de extração de características distintos. Para os classificadores base, foram utilizados MLPs de três camadas, utilizando a equação

$$NN = \frac{2}{3} \max(M, J) \quad (4.7)$$

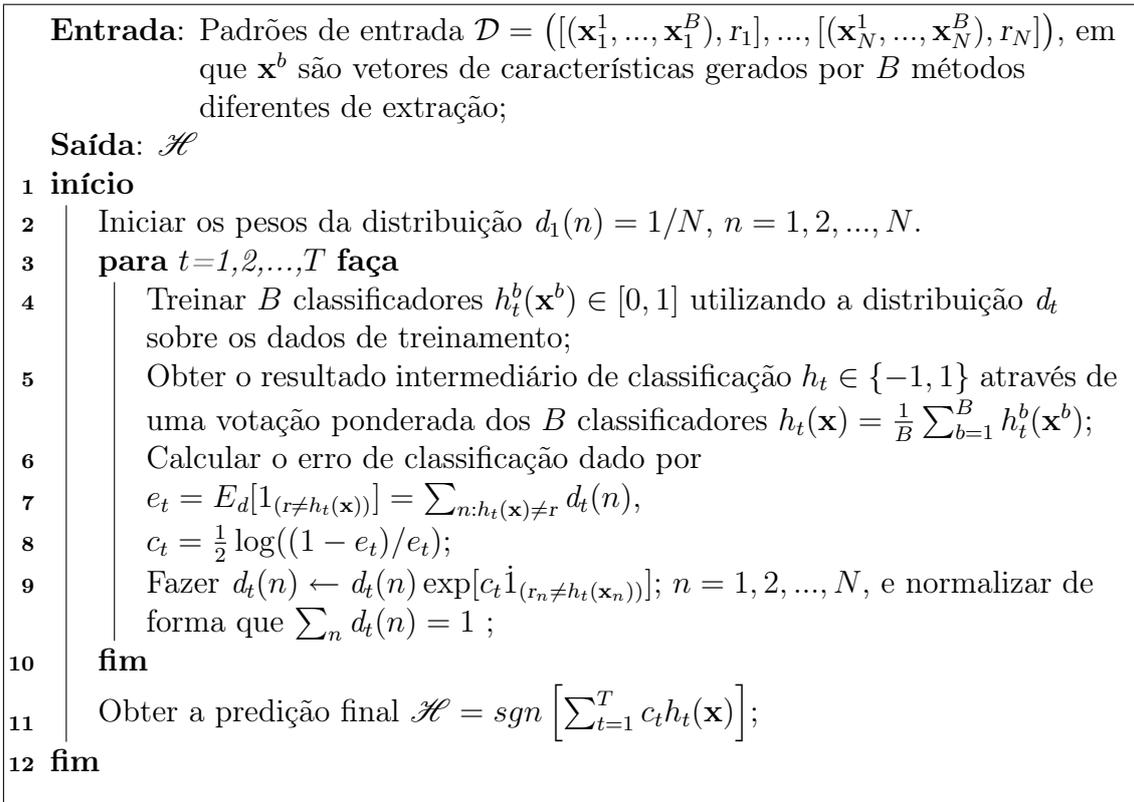


Figura 4.9: Algoritmo do *Variant AdaBoost* (YIN; LIU; HAN, 2005).

como critério para definir o número de neurônios da camada escondida, sendo M a dimensão do espaço de características e J é o número de classes conhecidas.

4.6 Conclusão

Neste capítulo foram descritos algoritmos de combinação de classificadores que possibilitam aumentar o desempenho na detecção de manchas através da classificação de características. O objetivo principal desses algoritmos é a redução da variância e da taxa de erro de classificação. O algoritmo *bagging* atinge esse objetivo através da diluição do erro entre os especialistas. Por sua vez, os algoritmos *boosting* reduzem a taxa de erro e variância adaptando-se a diferentes distribuições dos padrões de treinamento. No algoritmo de *boosting* a distribuição da subamostragem dos dados é atualizada antes de cada treinamento do classificador base. Assim, o conjunto de classificadores do comitê cresce de forma progressivamente diversa. Conforme afirmam KUNCHEVA e WHITAKER (2003), não existe nenhuma forma explícita de medição dessa diversidade no processo de formação do comitê, porém assume-se que a diversidade obtida com essa estratégia é um fator de sucesso do algoritmo.

O Capítulo 5 apresenta os resultados finais obtidos com os métodos de combinação de classificadores e características e uma discussão sobre o projeto dos classificadores.

Capítulo 5

Resultados e Discussões

Neste capítulo são apresentados os resultados e discussões de vários experimentos de classificação de manchas utilizando diferentes conjuntos de características e diferentes métodos de classificação.

Os experimentos realizados permitiram avaliar os classificadores e posteriormente definir aqueles mais adequados ao problema de detecção de manchas sob o aspecto da redução de alarmes falsos. Foram utilizadas 20 amostras de imagens SAR contendo 10 manchas de óleo e 10 manchas similares. Exemplos dessas imagens aparecem na Figura B.1 do Apêndice B.

A partir das imagens segmentadas foram extraídos 3 conjuntos de características, cada um deles construído por um método diferente de extração de características. A metodologia de extração de características é explicada no Apêndice C. Todos os conjuntos foram normalizados de tal forma que os novos dados \mathbf{x}' possuam média zero e variância unitária, através da equação

$$\mathbf{x}' = \frac{\mathbf{x} - \boldsymbol{\mu}}{\sigma^2}, \quad (5.1)$$

em que $\boldsymbol{\mu}$ é a média e σ^2 a variância das amostras. A normalização é importante para uniformizar a ordem de grandeza das características e ajuda a minimizar o tempo de treinamento das RNAs (HAYKIN, 2001).

Além da normalização dos dados de entrada, medidas específicas para melhoria de desempenho do algoritmo de retro-propagação foram adotadas. Estes cuidados, como a utilização de função de ativação antissimétrica, a forma de iniciar os pesos

e os cuidados com a taxa de aprendizagem, estão relacionadas em Haykin (2001).

Os experimentos foram realizados em três etapas distintas. Inicialmente foram testados diferentes algoritmos de classificação sobre diferentes conjuntos contendo características geométricas, características espaciais de textura e características espectrais de textura. Em seguida, os conjuntos de características originais foram combinados em três novos conjuntos. Os resultados de classificação sobre os novos conjuntos combinados e seus equivalentes com dimensão reduzida foram verificados e comparados.

Para cada procedimento de classificação foi utilizado o método *bootstrap* de validação cruzada. A estimativa de erro de classificação e uma matriz de confusão foram calculadas para cada classificador e para cada conjunto de características. Dado o pequeno número de amostras, o método *bootstrap* foi repetido diversas vezes para aumentar a probabilidade de uma determinada amostra ser utilizada pelo menos 1 vez em cada etapa de treinamento e teste.

Diversos testes foram realizados para definir os valores do número de rodadas ou etapas T e o tamanho do conjunto de treinamento. Através dos resultados preliminares, chegou-se à conclusão de que a utilização de 70% dos dados para treinamento dos classificadores seria suficiente para a análise de desempenho dos classificadores. As amostras restantes foram utilizadas para estimar o erro de generalização dos classificadores. Uma matriz de confusão também foi utilizada na análise dos resultados. Através da matriz de confusão é possível avaliar a quantidade de falsos-positivos e falsos-negativos, o que é fundamental no desenvolvimento de uma metodologia para detecção confiável de manchas.

Os algoritmos dos classificadores foram escritos para o programa MATLAB[®]. Exceto a SVM, para o qual foi utilizado um conjunto de *scripts* desenvolvido por Gunn (1998), os demais algoritmos foram implementados durante a execução deste trabalho e fazem parte das contribuições desta pesquisa.

5.1 Conjuntos de Características

O conjunto de imagens, os conjuntos de características geométricas (C_g) e características espaciais de textura (C_{te}) foram obtidos do trabalho de Araújo (2004). Um terceiro conjunto de características espectrais de textura (C_{ts}) foi desenvolvido durante esta pesquisa para ser utilizado nos testes com o algoritmo *Variant*

AdaBoost. Mais detalhes da extração de características podem ser encontrados no Apêndice C.

5.1.1 Limitações dos Conjuntos Individuais

Quando utilizados individualmente na classificação das manchas, esses conjuntos proporcionaram uma taxa média de acerto em torno de 80%. Em parte, esse limite máximo pode ser atribuído às características do conjunto das imagens disponíveis. Através da análise visual da geometria das manchas, percebe-se que não existe um padrão visual específico que defina claramente se a mancha é decorrente de um derramamento de óleo ou de fenômenos naturais. Em outras palavras, dentro do conjunto de imagens existem manchas similares cuja área, perímetro ou espalhamento são semelhantes aos das manchas de óleo e vice-versa. Portanto, somente a informação da geometria da mancha não apresenta poder discriminatório suficiente para obter uma taxa de classificação correta desejada.

Para agregar informação adicional à geometria da mancha, dois conjuntos de características de textura foram gerados através de métodos diferentes de extração. Um conjunto de características espaciais de textura foi obtido a partir da GLCM das manchas segmentadas. Outro conjunto de características espaciais de textura foi obtido através da análise por ondaletas. Verificou-se, em alguns casos, que as características de textura contribuem para melhor generalização do classificador.

De modo geral, ocorre que quando duas manchas de classes distintas apresentam descritores similares pertencentes a um mesmo conjunto de características, os descritores do outro conjunto de características acrescentam detalhes que permitem que estas manchas sejam corretamente separadas em classes diferentes. Assim, quando as características geométricas de uma mancha não permitem determinar a que classe ela pertence, a comparação das características de textura adiciona informação complementar para classificá-la corretamente.

5.1.2 Características Combinadas

Durante estudos preliminares observou-se que, através da combinação de dois ou mais conjuntos de características, o desempenho de classificação apresentou melhora significativa, com média da taxa de acerto superior a 80%. Assim, ao combinar as características geométricas e de textura das manchas, foi possível descrever com mais detalhes cada tipo de mancha e melhorar a separação entre os conjuntos de

manchas de óleo e de manchas similares. Este resultado, porém, não foi confirmado para todos os classificadores. Para a realização desses experimentos, foram definidos seis conjuntos de características, três dos quais foram gerados através de diferentes técnicas de extração e o restante foi construído através da combinação dos três primeiros conforme descrito na Tabela 5.1.

Tabela 5.1: Conjuntos de características combinadas.

Conjuntos	Combinação de características			Dimensão
	Geometria	Textura Espacial	Textura Espectral	
C_g	✓			8
C_{te}		✓		15
C_{ts}			✓	7
C_{g+te}	✓	✓		23
C_{g+ts}	✓		✓	15
$C_{g+te+ts}$	✓	✓	✓	30

Para os conjuntos combinados, devido ao aumento de dimensionalidade, realizou-se um teste de transformação de espaço, reduzindo essa dimensionalidade através da Análise por Componentes Principais (PCA, do inglês *Principal Component Analysis*) (RICHARDS; JIA, 2006). Neste procedimento, foram escolhidos os autovetores para que a dispersão dos dados no novo espaço ficasse em torno de 99%, preservando a maior parte da informação. Em alguns casos testados, os novos conjuntos de treinamento, com sua dimensão reduzida, possibilitaram uma pequena melhoria no desempenho de classificação, verificado pelos resultados exibidos nas próximas seções. Entretanto, essa melhoria não foi significativa quando foram utilizados os algoritmos de combinação de classificadores. Em parte, isto era esperado, pois a transformação linear obtida com a PCA não garante a melhoria do poder discriminatório dos dados se as classes não são linearmente separáveis.

5.2 Parametrização dos Classificadores Utilizados nos Experimentos

Os algoritmos de classificação utilizados nos experimentos foram escolhidos por serem algoritmos bastante utilizados em diversos trabalhos científicos que abrangem a classificação de padrões. Pelas suas diferentes propriedades, sete classificadores

foram selecionados e são discriminados a seguir:

- i. Naïve Bayes (NB): classificador paramétrico, baseia-se em suposições de independência estatística e produz um discriminante linear;
- ii. KNN: estimador de função de densidade de probabilidade;
- iii. MLP: robustez a ruído nos dados e boa capacidade de generalização;
- iv. SVM: foco na fronteira (vetores de suporte) entre as classes e necessidade de poucas amostras de treinamento, mesmo com grande dimensão do vetor de características;
- v. *bagging*: algoritmo simples de combinação com diminuição do erro de classificação;
- vi. AdaBoost: realiza a combinação de classificadores com características similares à SVM;
- vii. LogitBoost: uso da função logística no *boosting* para minimização da variância;
- viii. *Variant AdaBoost* (VB): realiza a combinação simultânea de características e de classificadores.

Exceto para o classificador *Variant AdaBoost*, todos os conjuntos de características foram utilizados nos experimentos de classificação. Devido à forma como o *Variant AdaBoost* utiliza separadamente os conjuntos de características para fazer uma seleção de alto nível, conforme explicado no Capítulo 4, somente se empregam os conjuntos de características C_{g+te} , C_{g+ts} e $C_{g+te+ts}$ neste algoritmo.

Os parâmetros dos classificadores foram ajustados durante a realização dos experimentos para a obtenção do melhor desempenho médio de classificação. Para o KNN, foi utilizada uma vizinhança de 3 em todos os conjuntos de características. A utilização de uma vizinhança de ordem ímpar garante uma melhor solução de "impasses" quando um determinado vetor está a uma distância equiprovável de duas classes diferentes. Uma vizinhança maior do que 3, por exemplo 5, aumenta demasiadamente o custo computacional e não produz um ganho significativo no desempenho do classificador.

Todas as MLPs foram projetadas com 3 camadas, sendo 1 camada escondida. Para os classificadores individuais, a quantidade de neurônios da camada escondida

foi ajustada de acordo com a dimensão do conjunto de características. Por causa do pequeno conjunto de amostras de treinamento disponível, foi utilizado o menor número possível de neurônios na camada escondida, que permitisse um equilíbrio entre reduzida taxa de erro e boa capacidade de generalização. A quantidade de neurônios da camada escondida foi escolhida inicialmente como sendo a metade da dimensão dos dados de entrada e aumentada gradativamente em diversos testes até encontrar um ponto de desempenho similar ou superior ao obtido com os classificadores clássicos *Naïve Bayes* e KNN. O treinamento dos MLPs foi feito com parada antecipada, considerando a estabilização do EQM em um valor mínimo arbitrário em torno de 0,001. Quando esse erro mínimo não foi atingido, o treinamento foi limitado em 400 épocas. Observou-se em geral que, quando o EQM não convergiu para o valor mínimo dentro das 400 épocas, não foi possível diminuir o EQM aumentando o número de épocas de treinamento.

Os classificadores SVM foram parametrizados com um núcleo polinomial de terceiro grau. O grau do polinômio também foi ajustado gradativamente para o mesmo objetivo do MLP. Este procedimento foi utilizado para que os classificadores pudessem ser avaliados em termos de desempenho e custo computacional, sendo comparados com métodos clássicos de reconhecimento de padrões. Caso não fosse possível ajustar um determinado classificador para igualar ou superar o desempenho dos métodos clássicos, não seria justificada a sua utilização.

Nos métodos de combinação de classificadores, foram utilizadas redes MLP como classificadores base. No *bagging*, os MLPs foram ajustados como nos classificadores individuais. Para os algoritmos de *boosting*, os MLPs foram construídos com um número mínimo de neurônios. Ao utilizar MLPs como classificadores base, Yin, Liu e Han (2005) sugerem definir o número de neurônios NN da camada escondida através da equação (4.7). Já que apenas duas classes ($J = 2$) estão definidas para os dados utilizados neste trabalho, essa equação pode ser simplificada para $NN = \frac{2}{3}M$. O restante da arquitetura do MLP no *boosting* é similar àquelas utilizadas como classificadores individuais, com apenas uma camada escondida. No LogitBoost, foi necessário efetuar uma pequena modificação no algoritmo a fim de substituir a regressão originalmente definida por uma regressão aproximada pela rede MLP. Para a modificação da distribuição dos dados durante a fase treinamento foi adotada a estratégia da reamostragem com reposição dos vetores de entrada com base nos pesos d_t . Assim, para pesos maiores, um mesmo vetor de entrada é repetido

mais de uma vez no subconjunto de treinamento. Embora para cada etapa de treinamento o número de neurônios do MLP não varie, as diferentes subamostragens e condições iniciais geram classificadores base bastante distintos, o que contribui para a diversidade do comitê.

No *Variant AdaBoost*, além de usar o MLP como classificador base, foi utilizado também o *Naïve Bayes*. Esperava-se com essa substituição, diminuir o custo computacional o qual, de fato, ficou em torno de cinco a dez vezes menor, dado que o custo computacional de treinamento de um MLP é muito superior ao do treinamento de um *Naïve Bayes*. Entretanto, não se observa grande diferença no tempo de resposta na fase de teste para ambos. A estratégia adotada para definir o classificador intermediário do *Variant AdaBoost* foi a votação ponderada.

Todos os algoritmos baseados em *boosting* foram iniciados com 10 classificadores base. Durante o treinamento foi utilizada a estratégia de verificação da estabilização do erro. Dessa forma o treinamento foi acelerado e o número de classificadores base durante a fase de treinamento variou entre 3 e 10.

5.3 Resultados dos Classificadores Individuais

Numa primeira etapa, os conjuntos de características foram classificados individualmente utilizando-se 4 algoritmos: *Naïve Bayes*, KNN, MLP e SVM. Os resultados obtidos com estes classificadores foram utilizados para analisar o grau de separabilidade dos conjuntos de características e como referência na comparação de desempenho com os métodos de combinação.

Como mostra a Figura 5.1, os classificadores *Naïve Bayes* e KNN foram os mais sensíveis ao conjunto C_{ts} , obtendo um significativo aumento de desempenho. Para o MLP e a SVM, o conjunto C_{te} foi classificado com o menor erro de generalização. De modo geral, a variância do erro foi reduzida significativamente quando utilizados os conjunto C_{ts} , exceto para o MLP e a SVM, como pode ser observado na Figura 5.2.

A matriz de confusão da Tabela 5.2 exibe uma estimativa da quantidade de falsos-positivos e falsos-negativos do melhor resultado dentre os três classificadores individuais. Os números foram obtidos com o KNN treinado sobre o conjunto de características espectrais de textura (C_{ts}). Considerando-se apenas os 343 grandes derramamentos acima de 700 toneladas ocorridos no período de 1974 a 2006

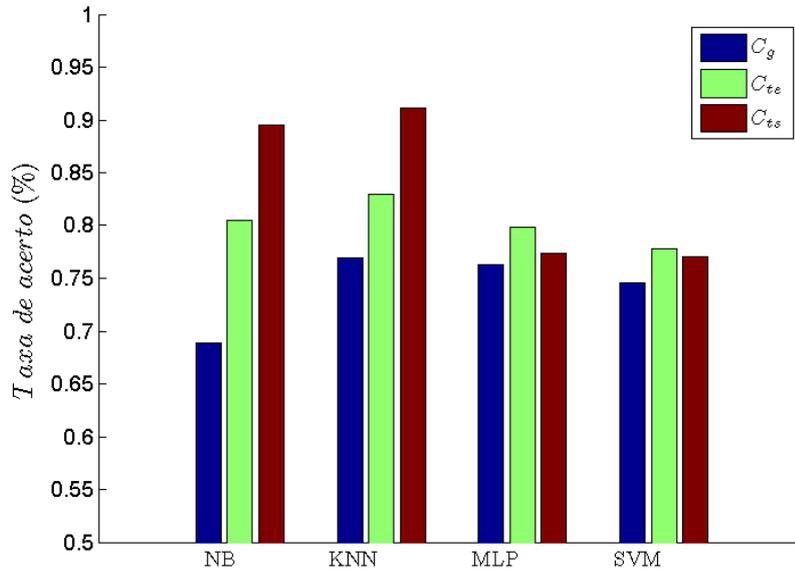


Figura 5.1: Comparação de desempenho dos classificadores individuais sobre cada conjunto de características geométricas e texturais (espaciais e espectrais).

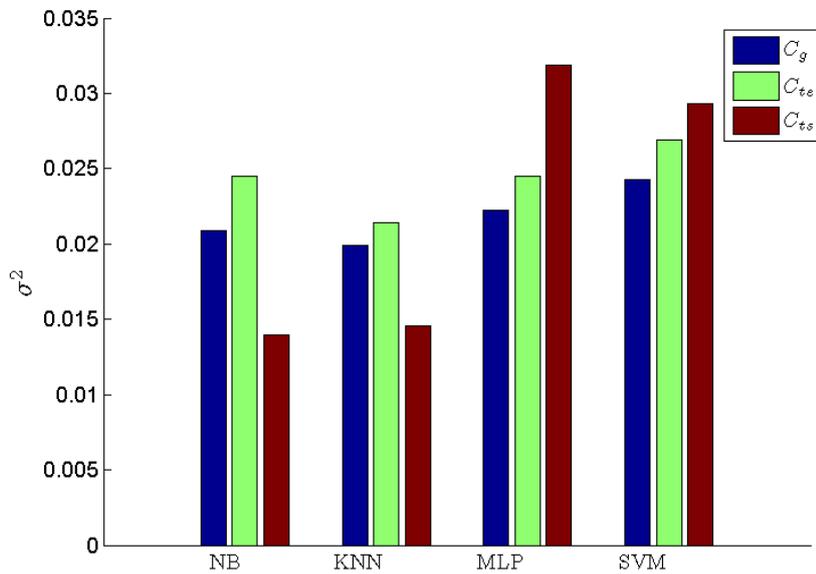


Figura 5.2: Variâncias do erro de classificação dos classificadores individuais sobre cada conjunto de características geométricas e texturais (espaciais e espectrais).

(ITOPF, 2006), a Tabela 5.2 expõe que, pelo menos 13 manchas naturais teriam sido interpretadas como manchas de óleo, representando um grande desperdício em ações

de contenção. Por outro lado, o custo maior está relacionado ao impacto ambiental, representado pelas 17 manchas de óleo que seriam consideradas manchas naturais. Em ambos os casos, esses dois tipos de alarmes falsos resultam numa perda de confiabilidade do sistema de monitoramento de manchas de óleo.

Tabela 5.2: Matriz de confusão obtida com o melhor desempenho de classificação das manchas.

Predição da Classe	Classe Real	
	Óleo	Similar
Óleo	0,4493	0,0507
Similar	0,0381	0,4619

De modo geral, a utilização de MLPs e Máquinas de Vetor de Suporte (SVMs, do inglês *Support Vector Machines*) não contribuiu para melhorar o desempenho de classificação e diminuir o número de alarmes falsos. No caso do MLP, isto pode ser explicado pela pequena quantidade de amostras de treinamento disponíveis. Aumentar o número de épocas de treinamento ou aumentar o número de neurônios, simplesmente piora o desempenho do MLP por causa do sobreajustamento. Em teoria, a SVM deveria obter desempenho melhor do que o MLP, já que suas características são mais favoráveis para este pequeno conjunto de treinamento. Entretanto, não foi possível obter o desempenho esperado da SVM em nenhum dos três conjuntos de características.

5.4 Resultados Finais

Para melhoria do desempenho e da confiabilidade da detecção de manchas de óleo, duas soluções foram propostas: a combinação de características e a combinação de classificadores. Os resultados que seguem apontam para uma solução complementar utilizando o melhor de cada abordagem.

5.4.1 Combinação de Características

O efeito das características combinadas nos classificadores individuais foi investigado, utilizando os conjuntos C_{g+te} , C_{g+ts} e $C_{g+te+ts}$. Em geral, o uso das características combinadas não mostrou ser mais eficaz do que o uso de um único conjunto de características. Como é evidente na Figura 5.3, em alguns casos o

aumento da dimensionalidade contribuiu para uma deterioração na capacidade de discriminação das duas classes de manchas. Além disso, houve um aumento na complexidade dos classificadores individuais e do custo computacional na fase de treinamento, de modo geral.

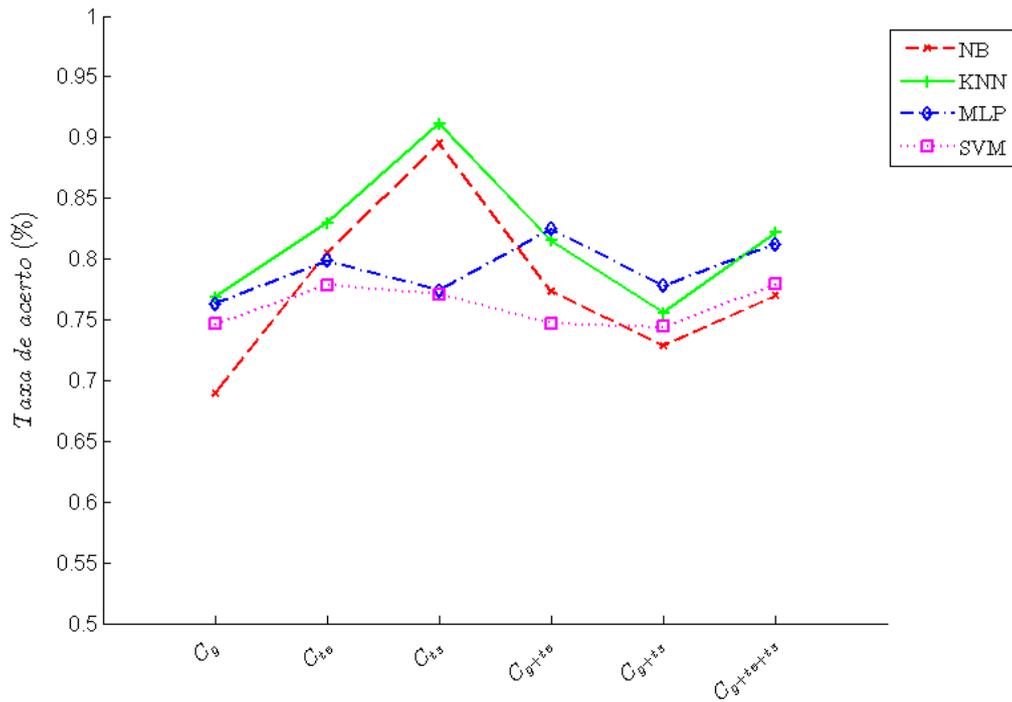


Figura 5.3: Comparação de desempenho dos classificadores individuais sobre os conjuntos de características combinadas.

Pode-se concluir que os descritores de textura têm influência positiva sobre o desempenho dos classificadores, mas o aumento da dimensão dos vetores de treinamento degradou a generalização dos classificadores individuais, em parte por causa da limitada quantidade de amostras de treinamento.

A redução de dimensionalidade dos conjuntos combinados, obtida através da transformação PCA, somente apresentou resultado positivo para o MLP sobre o conjunto $C_{g+te+ts}^*$, conforme mostra a Figura 5.4. Os conjuntos resultantes dessa redução de espaço são diferenciados dos conjuntos originais por um asterisco (C_{g+te}^* , C_{g+ts}^* e $C_{g+te+ts}^*$). Embora a transformação PCA preserve pelo menos 99% da dispersão dos dados, para o conjunto de dados utilizado essa redução de espaço contribui para a degradação no desempenho dos classificadores *Naïve Bayes*, KNN e SVM. Ocorre que uma transformação linear desse tipo pode, no melhor caso, ter

nenhuma influência na separabilidade das classes, quando estas não são linearmente separáveis.

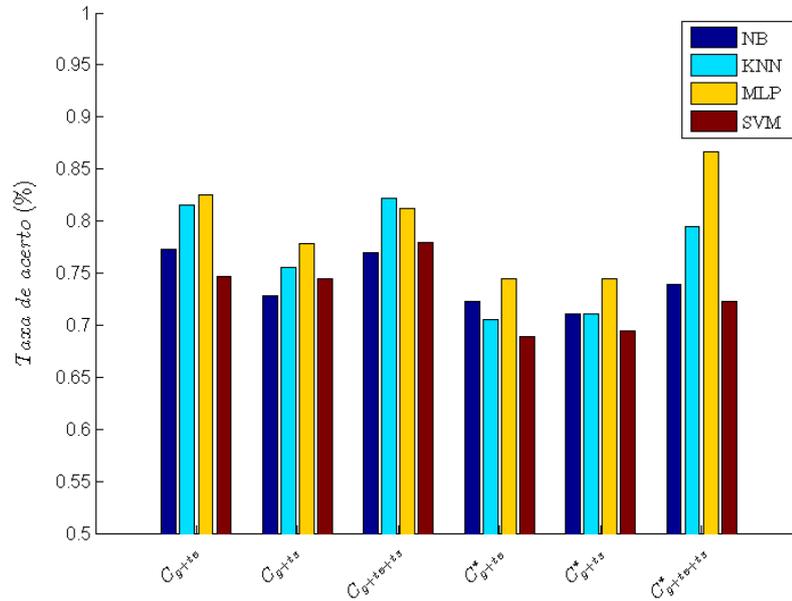


Figura 5.4: Comparação do desempenho de classificação após a redução de espaço utilizando PCA.

Essa degradação de desempenho pode ser observada numericamente quando se compara os resultados de classificação mostrados na Tabela 5.3 com os da Tabela 5.4. A dimensão dos dados após a transformação aparece entre parênteses, ao lado da dimensão original.

Tabela 5.3: Taxas de classificação e variâncias dos conjuntos transformados por PCA.

Conjunto:	C^*_{g+te}		C^*_{g+ts}		$C^*_{g+te+ts}$	
Dimensão:	23(8)		12(7)		27(10)	
Classificador	%	var	%	var	%	var
Naïve Bayes	0,7222	0,0313	0,7111	0,0229	0,7389	0,0186
KNN	0,7056	0,0109	0,7111	0,0248	0,7944	0,0166
MLP	0,7444	0,0263	0,7444	0,0186	0,8667	0,0199
SVM	0,6889	0,0282	0,6944	0,0155	0,7222	0,0160

5.4.2 Combinação de Classificadores

Os métodos de combinação de classificadores apresentam propriedades que demonstraram ser úteis ao problema de detecção de manchas através da classificação de características. O aumento da taxa de acerto, o foco nos vetores da fronteira entre as classes e a redução de variância da estimativa de erro, são algumas dessas propriedades. Além disso, ao contrário dos classificadores individuais, os métodos de combinação, em geral, apresentaram menor sensibilidade ao sobreajustamento.

A Figura 5.5 exibe as taxas de acerto para comparação de desempenho entre classificadores individuais e os métodos *bagging* e *boosting*. Observa-se melhora significativa no desempenho de classificação nos métodos de comitê. Também é possível perceber que, onde foram utilizadas características espectrais de textura, houve uma significativa melhora no desempenho de classificação dos comitê. De fato, o método de extração por ondaletas contribui para gerar melhores discriminantes de textura das manchas, já que as imagens possuem diferentes resoluções. O resultado da classificação de manchas utilizando o método de combinação de classificadores e características, *Variant AdaBoost* (VB), é apresentado na Figura 5.6.

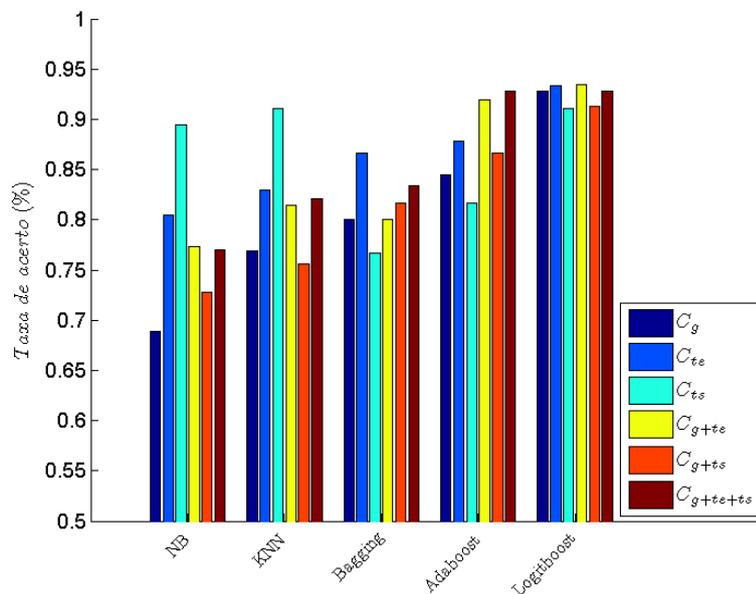


Figura 5.5: Comparação de desempenho dos classificadores sobre os conjuntos de características combinados.

Na Tabela 5.4 são exibidos os resultados numéricos dos classificadores individuais e dos algoritmos de comitê. As colunas desta tabelas apresentam a taxa de acerto e

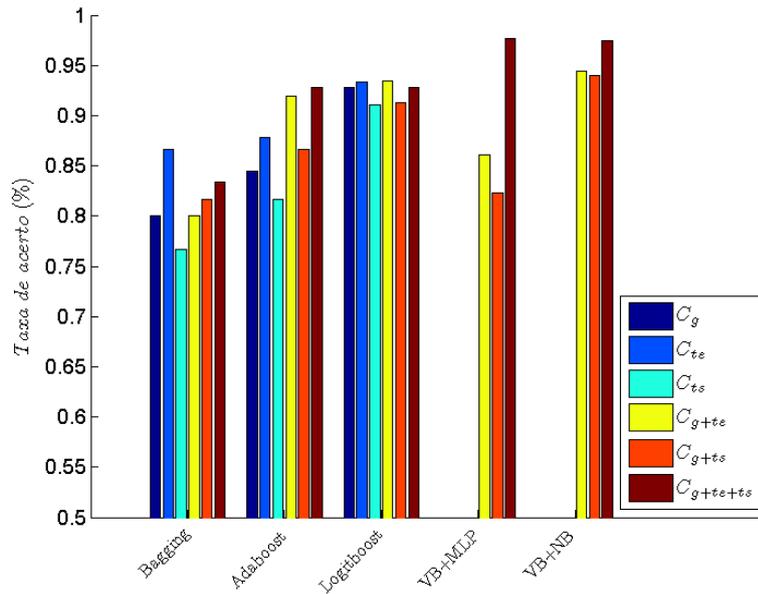


Figura 5.6: Comparação de desempenho entre os métodos de combinação de classificadores sobre os conjuntos de características combinados.

variância do erro para diferentes conjuntos de características. Em destaque aparecem os valores máximos da taxa de classificação e os valores mínimos da variância em cada coluna. Percebe-se que o classificador LogitBoost obteve os melhores desempenhos para os conjuntos de características originais, não combinados. O mesmo ocorreu para o conjunto C_{ts} , onde a variância foi menor, embora a taxa de acerto tenha sido similar à do KNN.

Tabela 5.4: Comparação entre as taxas de acerto e variâncias do erro de classificação.

Conjunto: Dimensão:	C_g 8		C_{te} 15		C_{ts} 7		C_{g+te} 23		C_{g+ts} 15		$C_{g+te+ts}$ 30	
Classificador	%	var	%	var	%	var	%	var	%	var	%	var
NB	0,6890	0,0209	0,8050	0,0245	0,8949	0,0140	0,7729	0,0206	0,7282	0,0243	0,7696	0,0182
KNN	0,7691	0,0199	0,8295	0,0214	0,9112	0,0146	0,8148	0,0184	0,7554	0,0182	0,8214	0,0212
MLP	0,7631	0,0223	0,7985	0,0245	0,7740	0,0319	0,8246	0,0185	0,7778	0,0171	0,8121	0,0189
SVM	0,7462	0,0243	0,7783	0,0269	0,7707	0,0293	0,7467	0,0190	0,7440	0,0184	0,7789	0,0183
Bagging	0,8000	0,0235	0,8667	0,0173	0,7667	0,0136	0,8000	0,0481	0,8167	0,0090	0,8333	0,0123
AdaBoost	0,8444	0,0554	0,8778	0,0535	0,8167	0,0811	0,9194	0,0206	0,8660	0,0497	0,9276	0,0105
Logitboost	0,9278	0,0109	0,9333	0,0107	0,9111	0,0129	0,9341	0,0088	0,9134	0,0104	0,9281	0,0099
VB+MLP	-x-	-x-	-x-	-x-	-x-	-x-	0,8611	0,0401	0,8230	0,0577	0,9766	0,0034
VB+NB	-x-	-x-	-x-	-x-	-x-	-x-	0,9439	0,0071	0,9395	0,0081	0,9749	0,0037

Quando são utilizados os conjuntos de características combinadas, o *Variant AdaBoost* leva vantagem. A capacidade discriminante de cada conjunto de características em separado é, aparentemente, mais eficiente no *Variant AdaBoost* do que a capacidade discriminante das características combinadas nos outros métodos de *boosting*. De fato, esse resultado era esperado, já que, segundo os autores Yin, Liu e Han (2005), o método realiza uma seleção particular de características. Em outras palavras, esta seleção, realizada na fase intermediária do *Variant AdaBoost*, corresponde à combinação de especialistas construídos sobre conjuntos de características codificados por diferentes métodos de extração.

Através dos métodos de *boosting* foi possível obter uma significativa diminuição no número de alarmes falsos. Conforme mostra a Tabela 5.5, essa redução foi especialmente significativa com o *Variant AdaBoost*. O classificador *Variant AdaBoost* obteve notadamente o melhor desempenho sobre os conjuntos de maior dimensionalidade, se comparado com os melhores resultados dos algoritmos de combinação. Isso está associado ao fato do algoritmo encontrar separadamente o melhor conjunto de características para cada classificador do comitê. A Tabela 5.5 mostra a matriz de confusão obtida com o *Variant AdaBoost*. O erro total de classificação do *Variant AdaBoost* é de aproximadamente 2% contra os quase 9% do melhor desempenho do KNN (ver Tabela 5.2). O número total de alarmes falsos é apenas a quarta parte daquele obtido com o KNN. Também observa-se um número insignificante de falsos-negativos e uma maior precisão representada pela pequena variância.

Tabela 5.5: Matriz de confusão do *Variant AdaBoost*.

Predição da Classe	Classe Real	
	Óleo	Similar
Óleo	0,4771	0,0005
Similar	0,0229	0,4995

A escolha do classificador base nos métodos de combinação tem grande importância. Percebe-se que a utilização de algoritmos mais simples como o *Naïve Bayes* podem resultar em taxas de classificação altas com um custo computacional muito menor do que quando são utilizados MLPs. Isto pode ser observado ao se comparar os resultados do *Variant AdaBoost* + MLP (VB+MLP) com o do *Variant AdaBoost* + NB (VB+NB).

5.5 Conclusão

Os experimentos mostraram que as características de textura (especialmente as espectrais) contribuem para um aumento significativo do desempenho dos classificadores, tomando por base o desempenho obtido com as características geométricas. Essa observação é válida também para os conjuntos combinados. Entretanto, o projeto do classificador torna-se mais complexo e o desempenho final é menor do que aquele obtido com os conjuntos individuais de textura.

Para todos os métodos de combinação de classificadores foi observada uma diminuição do erro de generalização e na variância. Além disso, nos métodos de *boosting* percebe-se que não há perda de desempenho com o aumento do número de descritores como ocorre com os classificadores individuais. O melhor desempenho geral foi obtido com o *Variant AdaBoost*, o qual demonstrou excelente capacidade de generalização e precisão.

Apesar do custo computacional dos métodos de combinação ser maior do que nos métodos clássicos, algumas considerações importantes devem ser feitas quanto ao fato de que:

- i. a melhoria de desempenho e a redução de alarmes falsos favorecem a confiabilidade de um sistema de detecção de manchas construído com essa metodologia, já que a tomada de decisão será menos comprometida por erros de avaliação;
- ii. em grande parte, o custo computacional está relacionado à fase de treinamento, sendo que o custo necessário para calcular a predição do classificador é equivalente ao dos outros classificadores;
- iii. assume-se que o método utilizado para a segmentação da mancha contribui minimamente para a introdução de ruído na extração de características, especialmente na construção do conjunto de características geométricas que é bastante sensível à geometria da mancha segmentada.

Em geral, o desempenho dos algoritmos de combinação de classificadores aponta para a comprovação da sua maior eficiência em minimizar o número de alarmes falsos.

Conclusões e Trabalhos Futuros

Através da pesquisa realizada no desenvolvimento dessa dissertação foram investigados diversos métodos para obter a melhoria de desempenho de classificação de manchas de óleo através da combinação de características e da combinação de classificadores. Os dados utilizados nos experimentos foram extraídos de imagens SAR obtidas de diferentes sistemas de imageamento SAR com diferentes parâmetros como número de visadas, estatística do ruído e resolução espacial. A utilização de técnicas de combinação de classificadores proposta possibilitou gerar um sistema de detecção robusto a esta diversidade de parâmetros não controláveis. A metodologia aplicada tornou possível obter uma melhoria significativa de desempenho quando comparada aos classificadores individuais.

Um das grandes dificuldades neste trabalho consiste em lidar com um pequeno número de amostras de manchas disponíveis e a grande dimensionalidade dos conjuntos de características necessária para descrever e classificar corretamente uma mancha. Para atingir um nível satisfatório de desempenho de classificação, foram utilizados diferentes conjuntos de características. Estes conjuntos foram agrupados entre si buscando melhorar o poder discriminatório dos dados. Os experimentos mostram que os conjuntos de características de textura contribuem para o melhor desempenho de classificação. Contudo, não foi possível diminuir o erro de generalização para um nível desejado, visando uma maior confiabilidade do sistema de detecção de manchas.

A solução encontrada nesta pesquisa para melhorar o desempenho de classificação e diminuir o número de alarmes falsos foi a aplicação de métodos de combinação

de classificadores, notadamente os métodos LogitBoost e *Variant AdaBoost*. Um problema complexo tal qual a identificação do tipo de mancha de óleo através de imagens ruidosas como as geradas pelos radares SAR, pode ser solucionado ao dividir a complexidade dessa tarefa entre diversos classificadores simples e combinar as suas saídas. Além disso, nenhuma resposta desses "especialistas" é descartada, pois o método aproveita todo o esforço computacional aplicado. Através destes métodos de combinação foi possível alcançar taxas de erro de classificação menores do que os classificadores individuais e ainda reduzir significativamente a variância. Em outras palavras, essa redução da variância do erro de generalização representa uma maior precisão do classificador.

Apesar das suas vantagens, os algoritmos de combinação de classificadores podem aumentar significativamente o custo computacional na fase de treinamento e a complexidade do projeto do classificador. O algoritmo LogitBoost é menos complexo e possui menor custo computacional do que o *Variant AdaBoost*. Entretanto, embora o LogitBoost possibilite obter altas taxas de acerto com um menor custo computacional, com o *Variant AdaBoost* foi possível reduzir ao mínimo o número de alarmes falsos. O maior custo computacional dos métodos de *boosting* é perfeitamente compensado pela diminuição da taxa de erro, o aumento da habilidade de generalização e a facilidade de parametrização.

É importante registrar que o melhor desempenho dos métodos de *boosting* não significa necessariamente que os outros classificadores utilizados sejam pouco eficientes. A escolha do classificador e o seu desempenho de classificação estão intimamente relacionados às características do conjunto de dados e suas idiossincrasias. A presença de ruído, a grande dimensão, a limitada quantidade de amostras e até mesmo o tipo de normalização dos dados podem influenciar negativamente no desempenho de um classificador. Cabe ao pesquisador entender as vantagens e limitações de cada método de classificação utilizado para extrair deles o melhor desempenho possível em sua aplicação de interesse.

6.1 Contribuições Finais

Os resultados apresentados neste trabalho permitem concluir que a metodologia proposta é eficiente em produzir classificadores de bom desempenho mesmo quando treinados com poucas amostras de grande dimensionalidade.

Uma das contribuições desta pesquisa é a geração de um novo conjunto

de descritores espectrais de textura baseados em ondaletas. Também foram desenvolvidos *scripts* em MATLAB® para automatização dos testes de classificação, parametrização e validação de diversos classificadores, dentre eles: *Naïve Bayes*, KNN, MLP, SVM, AdaBoost, LogitBoost e *Variant AdaBoost*.

Ao longo da pesquisa, foram produzidos diversos trabalhos científicos, apresentados em eventos relevantes da área como as conferências internacionais *Image Analysis and Recognition* (ICIAR), *Pattern Recognition* (ICPR) e *Statistical Techniques in Pattern Recognition / Structural and Syntactic Pattern Recognition* (SSR/SSPR). Dois desses trabalhos foram publicados em periódicos internacionais classificados pela CAPES como QUALIS A. Os três trabalhos mais relevantes da pesquisa são listados a seguir:

- ▶ Lecture Notes in Computer Science/SSR/SSPR - Combinação de características para a melhoria de classificação de manchas de óleo utilizando imagens SAR (LOPES et al., 2006);
- ▶ ICPR - Utilização do *boosting* na melhoria de detecção de manchas em imagens SAR (RAMALHO; MEDEIROS, 2006);
- ▶ Lecture Notes in Computer Science/ICIAR - Melhoria de confiabilidade de sistemas de detecção de manchas utilizando *boosting* na seleção de características (RAMALHO; MEDEIROS, 2007).

6.2 Perspectiva de Trabalhos Futuros

No decorrer da pesquisa percebeu-se a necessidade de incorporar novas imagens SAR ao conjunto de amostras, possibilitando enriquecer o projeto e a avaliação de desempenho dos classificadores utilizados. A partir da disponibilidade de um conjunto maior de amostras, sugere-se:

- ▶ investigar regras de votação diferentes no passo intermediário do *Variant AdaBoost*;
- ▶ testar a construção de comitês utilizando SVMs ou RBFs;
- ▶ investigar o uso dos Mapas Auto-Organizáveis (SOM, do inglês *Self-Organizing Maps*) na segmentação e na classificação das manchas com base no conhecimento do ruído e na análise multiescala de textura;

- ▶ desenvolver uma interface amigável para o sistema de detecção de manchas;
- ▶ estender a metodologia para problemas multi-classe, possivelmente permitindo separar os tipos de manchas de óleo em classes distintas (manchas recentes × degradadas, derramamento criminoso × acidental);
- ▶ investigar a sensibilidade do desempenho dos classificadores quanto aos diferentes descritores de textura e realizar a seleção de características;
- ▶ avaliar com melhor precisão o desempenho dos classificadores através da análise da curva Característica Operador Recebedor (ROC, do inglês *Receiver Operating Characteristic*) obtida das estimativas de falsos-positivos e falsos-negativos durante os testes de classificação.

No *Variant AdaBoost*, é necessário definir uma estratégia de seleção do conjunto de características com melhor resultado na classificação intermediária e armazenar essa informação para ser utilizada nas fases seguintes. A estratégia definida por Yin, Liu e Han (2005) é uma votação ponderada dos classificadores intermediários. Porém, outros tipos de combinação intermediária podem ser investigados como a votação de maioria ou a combinação das saídas por produto ou soma.

Como sugere o método de combinação de classificadores *boosting*, qualquer tipo de classificador base pode ser utilizado. É importante investigar o desempenho do *boosting* com a SVM e a RBF para efeito de comparação da sensibilidade do algoritmo ao tipo de classificador base utilizado. Utilizando o algoritmo AdaBoost, Li, Wang e Sung (2007) construíram um comitê de SVMs e compararam seu desempenho com árvores de decisão e RNAs. Os resultados experimentais demonstram uma melhoria de desempenho de generalização do comitê sobre a SVM em problemas de classificação binária não-balanceada, no qual o número de amostras de uma classe é muito maior que o de outra.

Em seu trabalho, Hammouche, Diaf e Postaire (2006) descrevem uma metodologia que utiliza o conceito de agrupamento de dados para, através da análise de características de textura, classificar pixels em uma imagem. A rede SOM é um método de agrupamento que tem a propriedade da preservação da topologia dos dados, o que pode ser útil na segmentação baseada no conhecimento do modelo do ruído das imagens SAR e também na análise de textura da imagem. Utilizando este método, pode-se investigar o desempenho de um algoritmo de segmentação com base

no agrupamento de pixels similares de áreas homogêneas das imagens que, em geral, representam as manchas de óleo.

Além de classificar uma mancha como sendo proveniente de um derramamento de óleo ou de um fenômeno natural, é possível agregar mais informação e discriminar as manchas de óleo em classes diferentes. Com um número significativo de imagens de teste, pode-se incluir características externas como a ação dos ventos na região imageada, classificando assim as manchas quanto ao grau de degradação e, possivelmente, quanto ao tipo de causa do derramamento.

Recentemente, Frate e Salvatori (2004) publicaram um estudo sobre a sensibilidade do classificador com relação às características. Diversos experimentos de classificação foram realizados retirando-se alguns elementos do conjunto original de características. Chegou-se à conclusão que algumas das características geométricas contribuem para o aumento do erro de classificação. Em trabalho posterior pretende-se realizar um teste de sensibilidade das características de textura a fim de determinar se alguma delas possui influência negativa no desempenho dos classificadores de tal forma que possa ser descartada.

Embora os resultados obtidos com este trabalho sejam claros ao apontar a combinação de classificadores como uma solução prática do problema de classificação de manchas em um conjunto pequeno de imagens SAR, falta informação adicional para afirmar qual dos métodos de combinação utilizados produz, em média, o melhor resultado. Em outras palavras, baseando-se apenas na estimativa de taxa de erro e matrizes de confusão, é difícil selecionar o método de combinação melhor aplicável porque os desempenhos são muito próximos entre si. Esta dificuldade está relacionada, em parte, à imprecisão das estimativas de falsos-positivos e falsos-negativos, devido à pequena quantidade de dados amostrais. A partir da aquisição de mais imagens de manchas e conseqüente aumento do número de amostras, será possível realizar testes de classificação com conjuntos de validação significativamente maiores. Um maior número de informações possibilitará a realização de estimativas mais precisas e, também, a aplicação de outras ferramentas de comparação de classificadores como a curva ROC. Esta ferramenta permite avaliar estatisticamente os classificadores quanto à sua sensibilidade e especificidade sobre os dados de teste utilizando diversos limiares de verossimilhança. A análise de desempenho de classificadores através das curvas ROC também oferece uma interpretação gráfica que possibilita selecionar com precisão o classificador com

o melhor desempenho dentro de parâmetros previamente definidos. No caso em particular de classificação de manchas, o cálculo da área sob a curva ROC permitirá selecionar aquele método de combinação que minimiza as taxas de falsos-positivos e de falsos-negativos.

Imageamento SAR

As imagens SAR são geradas por radares imageadores instalados em satélites ou em aviões. Atualmente os sistemas embarcados nos satélites ERS e ENVISAT da ESA (Agência Espacial Européia) e RADARSAT da CSA (Agência Espacial Canadense) são os principais fornecedores de imagens para a aplicações de monitoramento de derramamento de óleo (BREKKE; SOLBERG, 2005). Lu e Weng (2007) definem a escala, a resolução espacial e as necessidades do usuário como sendo os fatores mais importantes para a seleção do sistema de imageamento. Outro fator a ser levado em consideração é o tempo de disponibilização da imagem de uma região de interesse. Em geral, a órbita dos satélites é polar e a área de cobertura depende da latitude, decrescendo em direção ao equador. Por exemplo, no Mar Mediterrâneo o satélite ERS revisita uma mesma área a cada 25 dias, enquanto que o RADARSAT-1 e o ENVISAT demoram poucas horas (BREKKE; SOLBERG, 2005). A Tabela A.1, adaptada de Brekke e Solberg (2005), lista alguns desses sistemas atualmente em operação, e suas principais características¹. Algumas dessas características estão ilustradas na Figura A.1 adaptada de Richards e Jia (2006).

Os radares SAR operam na faixa de microondas subdividida em bandas de frequência que vão de 1 a 40 GHz. A banda C (4 a 8 GHz) é a de maior penetração na atmosfera e, portanto, é bastante adequada para a utilização em sistemas de detecção de manchas no mar. A resolução do radar está intimamente relacionada ao tamanho dos objetos imageados. Para o registro de manchas de óleo, que em geral são extensas, não é essencial o uso de imagens de precisão (PRI).

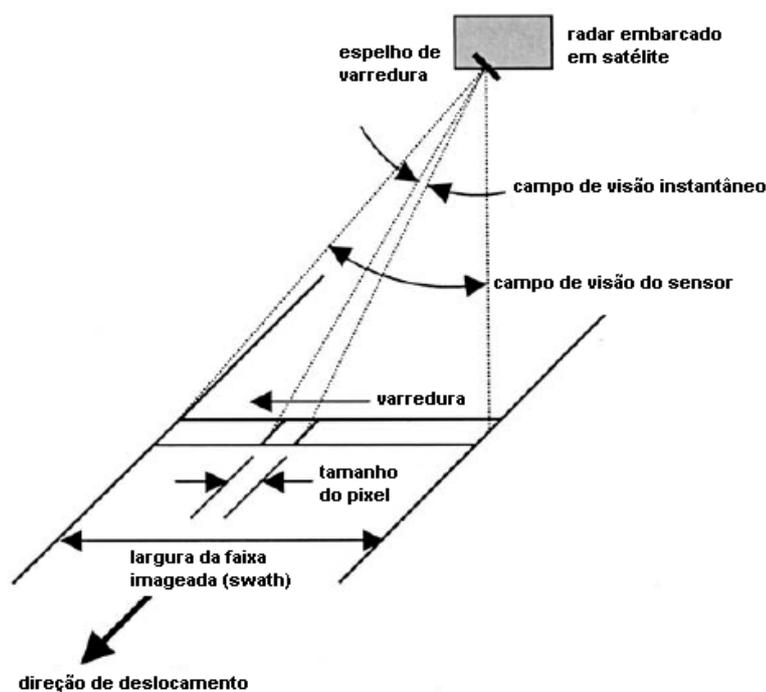
¹Informações adicionais sobre as características de diversos sistemas de imageamento SAR estão disponíveis em <http://www.sat.cnpm.embrapa.br>.

Tabela A.1: Satélites equipados com sistema de imageamento SAR.

Satélite	Agência	Características			
		Polarização	Resolução (m)	Swath ^a (km)	Modo ^b
ERS-2	ESA	VV	30 × 26.3	100	PRI
RADARSAT-1	CSA	HH	50 × 50 e 100 × 100	300 a 500	SCN e SCW
ENVISAT	ESA	HH	150 × 150	400	WSM

^aLargura da Faixa de Imageamento.

^bPRI = Imagem de precisão; WSM = Modo Swath Largo; SCN = ScanSAR Estreito; SCW = ScanSAR Largo.

**Figura A.1:** Modelo do sistema de imageamento SAR.

O sinal emitido pelo radar reflete na superfície terrestre de diferentes formas. Uma parte dessa energia é retroespalhada (*backscattering*) de volta para o sensor, onde é medida para gerar uma imagem. O sinal do radar possui polarização e, dependendo do campo elétrico utilizado, essa polarização pode ser horizontal (H) ou vertical (V). Quando o sinal interage com o alvo, ele sofre uma rotação na polarização. O sinal de retorno é registrado em antenas distintas. Quando a transmissão e recepção possuem a mesma polarização (HH ou VV), o sinal recebido

é de maior intensidade.

Na reflexão especular, que ocorre quando o sinal atinge uma superfície homogênea, pouca energia do sinal original é refletida de volta para o sensor. Numa superfície rugosa, o sinal é refletido em todas as direções e boa parte da sua energia retorna ao sensor (RICHARDS; JIA, 2006). Durante a formação da imagem SAR, os pixels recebem intensidades de tons de cinza proporcionalmente à intensidade do sinal refletido. Assim, áreas homogêneas como as manchas de óleo no mar, aparecerão escuras na imagem SAR.

O sistema SAR é caracterizado por gerar imagens contaminadas com o ruído *speckle*. Uma forma de diminuir a intensidade do ruído é obter uma imagem gerada a partir da média de vários imageamentos consecutivos ou visadas (*looks*). A relação sinal/ruído definida por $SNR = 0,5227/\sqrt{N}$, em que N representa o número de visadas, é diretamente afetada pelo valor de N . Uma região homogênea de uma imagem SAR de amplitude com 1 visada possui $\beta = \frac{\sigma^2}{\mu} = 0,5227$, em que σ^2 é a variância e μ a média das intensidades dos pixels segundo uma distribuição Rayleigh (MEDEIROS; MASCARENHAS; COSTA, 1998). Numa imagem de apenas 3 visadas, o valor dessa relação numa região homogênea diminui para $\beta = 0,2941$, indicando que o ruído foi significativamente reduzido, dada a redução na variância das intensidades dos pixels.

Apesar da redução do ruído facilitar o processo de segmentação da imagem, infelizmente também acarreta perda de informação, principalmente na atenuação da textura. Para o método discutido neste trabalho, imagens com apenas 1 visada são desejadas. Algoritmos de suavização do ruído como o proposto por (ARAÚJO, 2004) podem ser aplicados com sucesso na fase de segmentação para que não haja a necessidade de utilizar imagens com múltiplas visadas, reduzindo o tempo desde a aquisição até o processamento final das imagens.

O efeito do ruído *speckle* no imageamento SAR e da filtragem desse ruído é mostrado Figura A.2 adaptada de Medeiros, Mascarenhas e Costa (1998). A Figura A.2 (b) exibe uma imagem SAR após o processo de filtragem não-linear adaptativa do ruído *speckle*, baseada na abordagem Máximo *a posteriori* (MAP, do inglês *Maximum a Posteriori*). Na Figura A.2 (d) percebe-se que o histograma evidencia duas classes predominantes de pixels na imagem que antes não eram distinguíveis e a redução da variância dos tons de cinza presentes.

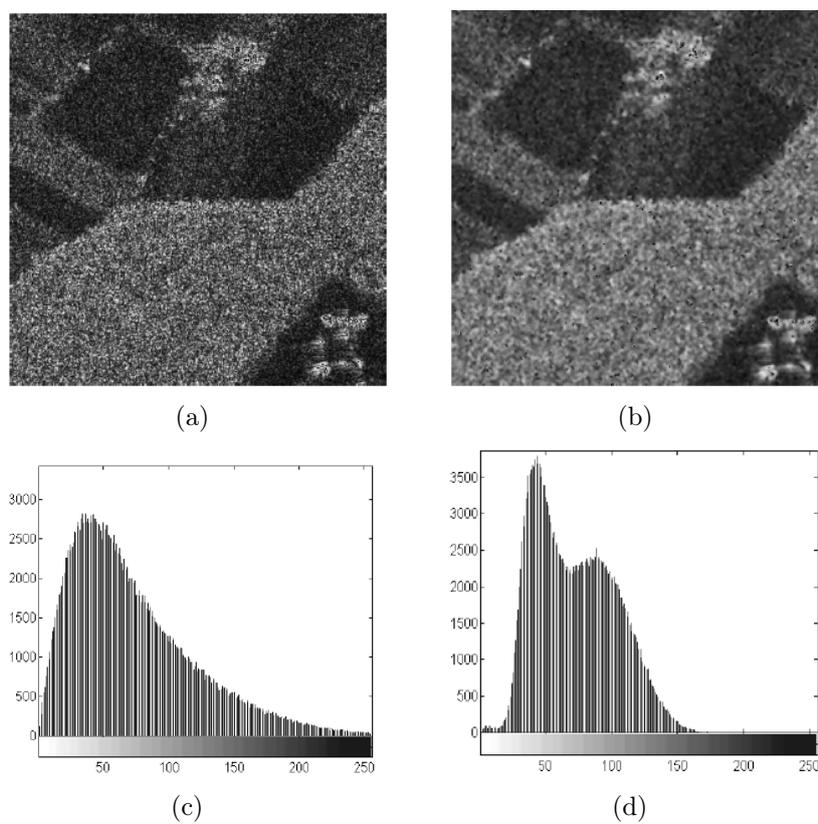


Figura A.2: Histograma de intensidade de pixels de uma imagem SAR original (a,c) e após a filtragem do ruído (b,d) por um filtro não-linear adaptativo.

Apêndice **B**

Imagens Utilizadas nos Experimentos

Ao contrário das imagens produzidas através de dispositivos óticos, as imagens SAR são geradas por sensores que trabalham na faixa do espectro de microondas. Esse tipo de imageamento não sofre influência direta das condições climáticas (nuvens, por exemplo) e são independentes da luz solar. Além disso, os sistemas de imageamento SAR embarcados em satélites ou em aviões permitem cobrir uma extensa área em poucos minutos. Essas características fazem com que este seja o tipo ideal de imageamento para sistemas de vigilância ambiental.

A quantidade de imagens SAR contendo manchas disponíveis para os testes neste trabalho esteve limitada por dois fatores:

- i. A ocorrência do evento é rara se considerado o tamanho das áreas imageadas;
- ii. As imagens SAR têm um custo elevado e são difíceis de ser adquiridas, especialmente aquelas que envolvem acidentes;

As imagens utilizadas para a avaliação da metodologia deste trabalho são mostradas na Figura B.1. Estas imagens são recortes de imagens maiores como mostrado na Figura B.2. Esta figura exhibe dois retângulos delimitando a área em torno de duas manchas. Ao todo foram utilizadas 20 imagens provenientes de sistemas instalados em satélites, como o ERS e RADARSAT, devidamente creditadas em Araújo (2004), sendo 10 delas contendo manchas de óleo e outras 10 contendo manchas similares. A maioria das imagens são provenientes do sistema RADARSAT e foram geradas no modo ScanSAR Wide. As imagens da Figura

B.1 foram modificadas digitalmente em brilho e contraste para obter um melhor resultado visual na impressão.

Preparação das Imagens para a Extração de Características

Para realizar a extração de características é necessário, em primeiro lugar, segmentar as manchas escuras presentes nas imagens. Os procedimentos utilizados, descritos em detalhes no trabalho de Araújo (2004), consistem da suavização do ruído *speckle*, utilizando uma transformação por ondaleta não-decimada, e da segmentação das manchas utilizando o crescimento de regiões.

A suavização do ruído utilizada por Araújo (2004) é um processo de filtragem através do qual uma imagem é decomposta em vários níveis por uma família de ondaletas *à trous* (MALLAT, 1999). Todos os níveis de decomposição possuem as mesmas dimensões da imagem original. Em cada nível de decomposição é retirado parte do ruído *speckle* presente na imagem. O resultado desse procedimento de filtragem pode ser observado na Figura B.3. No histograma de níveis de cinza da imagem filtrada é possível distinguir duas classes de pixels.

Posteriormente, as imagens são segmentadas através do crescimento de regiões por agregação de pixels (GONZALEZ; WOODS, 1992). A segmentação inicia com um conjunto de pontos sementes cujas regiões em torno deles crescem anexando a cada ponto semente aqueles pixels que possuam propriedades similares como nível de cinza ou textura. No trabalho de Araújo (2004), para definir os pontos sementes é utilizada a decomposição por ondaleta *à trous* do histograma e a sua correlação com o primeiro nível de decomposição (SITA; RAMAKRISHNAN, 2000). Este procedimento localiza picos no histograma, os quais são utilizados como sementes do algoritmo de crescimento de regiões. Um critério de parada define quando o algoritmo de crescimento deve ser finalizado e o resultado final é uma imagem binária.

O procedimento de segmentação produz as máscaras binárias exibidas na Figura B.4. A máscara é utilizada para extrair da imagem original apenas os pixels que pertencem à mancha. Isto é feito através de uma multiplicação pixel a pixel das duas imagens. Aos pontos claros da máscara são atribuídos o valor 1 e ao restante o valor 0. A imagem final, resultante dessa multiplicação, preserva somente a intensidade dos pixels pertencentes à mancha, dos quais são extraídas as características utilizadas para classificá-las em grupos diferentes. A Figura B.5 ilustra esse procedimento.

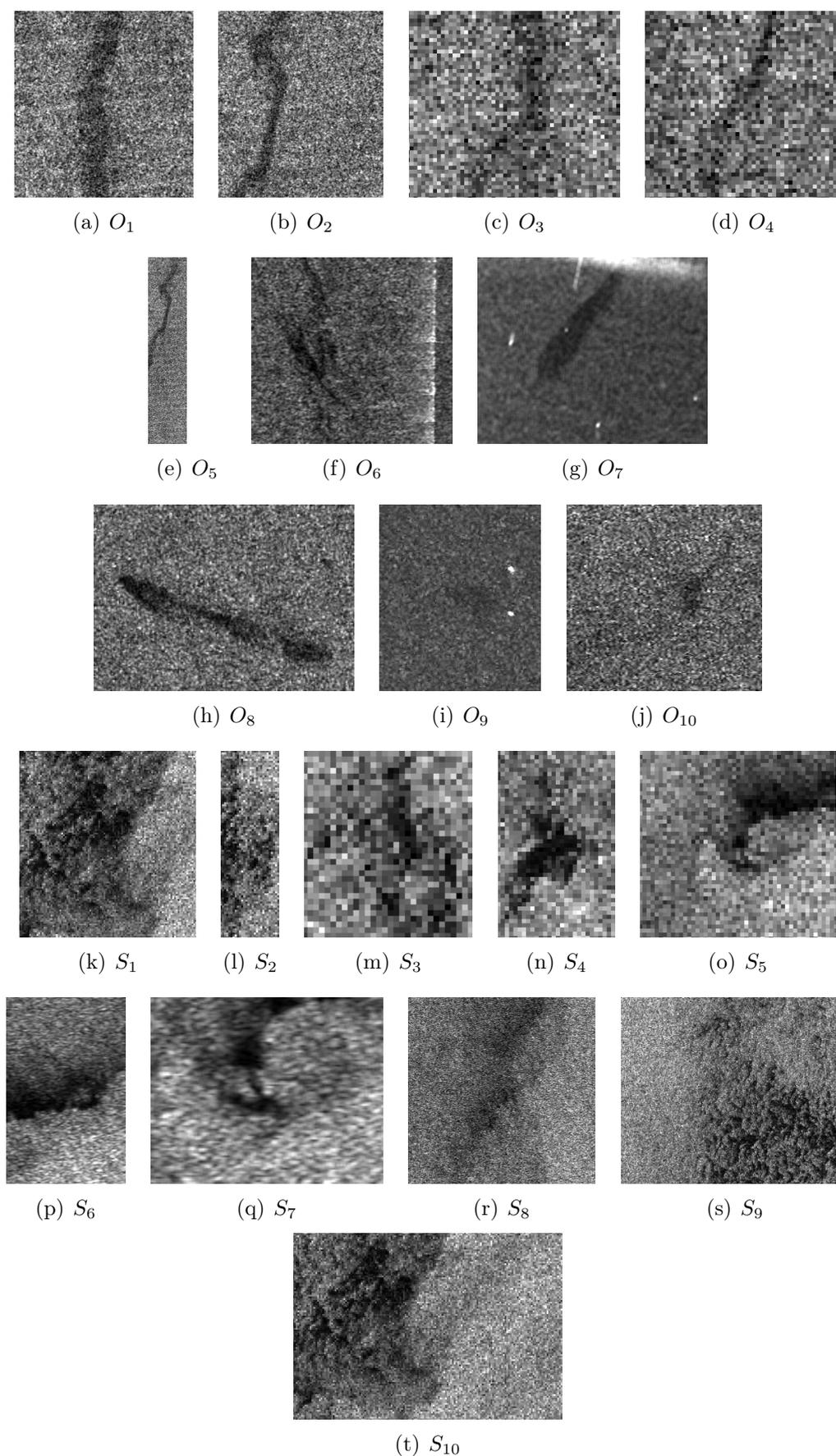


Figura B.1: Exemplo de imagens SAR de diferentes resoluções e números de visadas. As manchas (a) a (j) foram causadas por derramamento de óleo e as restantes (k) a (t) são manchas similares.

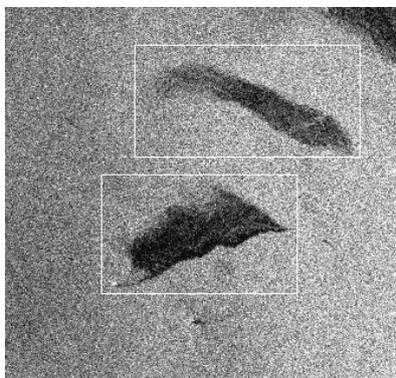


Figura B.2: Exemplo de imagem SAR do sistema ERS-2 contendo manchas delimitadas por ROIs.

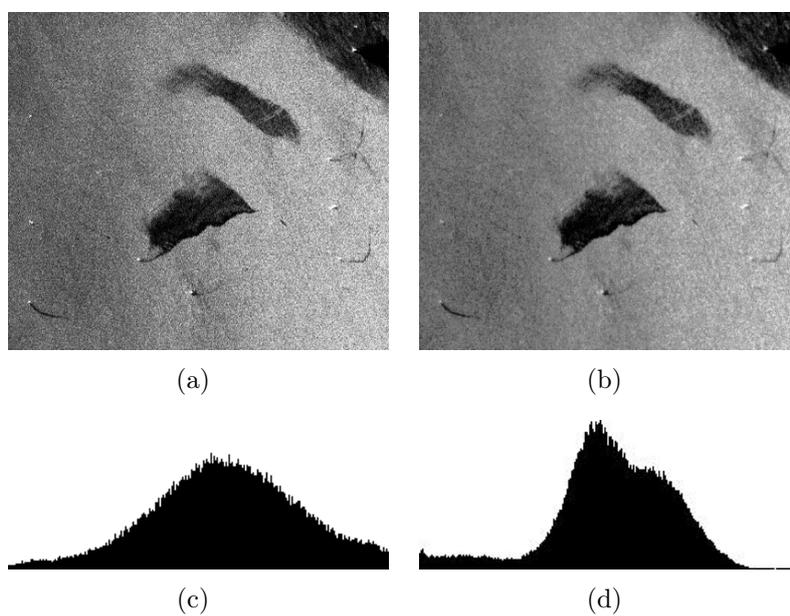


Figura B.3: Resultado da filtragem do ruído *speckle* utilizando a ondaleta *à trous* (ARAÚJO, 2004). (a) Imagem original. (b) Imagem filtrada. (c) e (d) Respectivos histogramas.

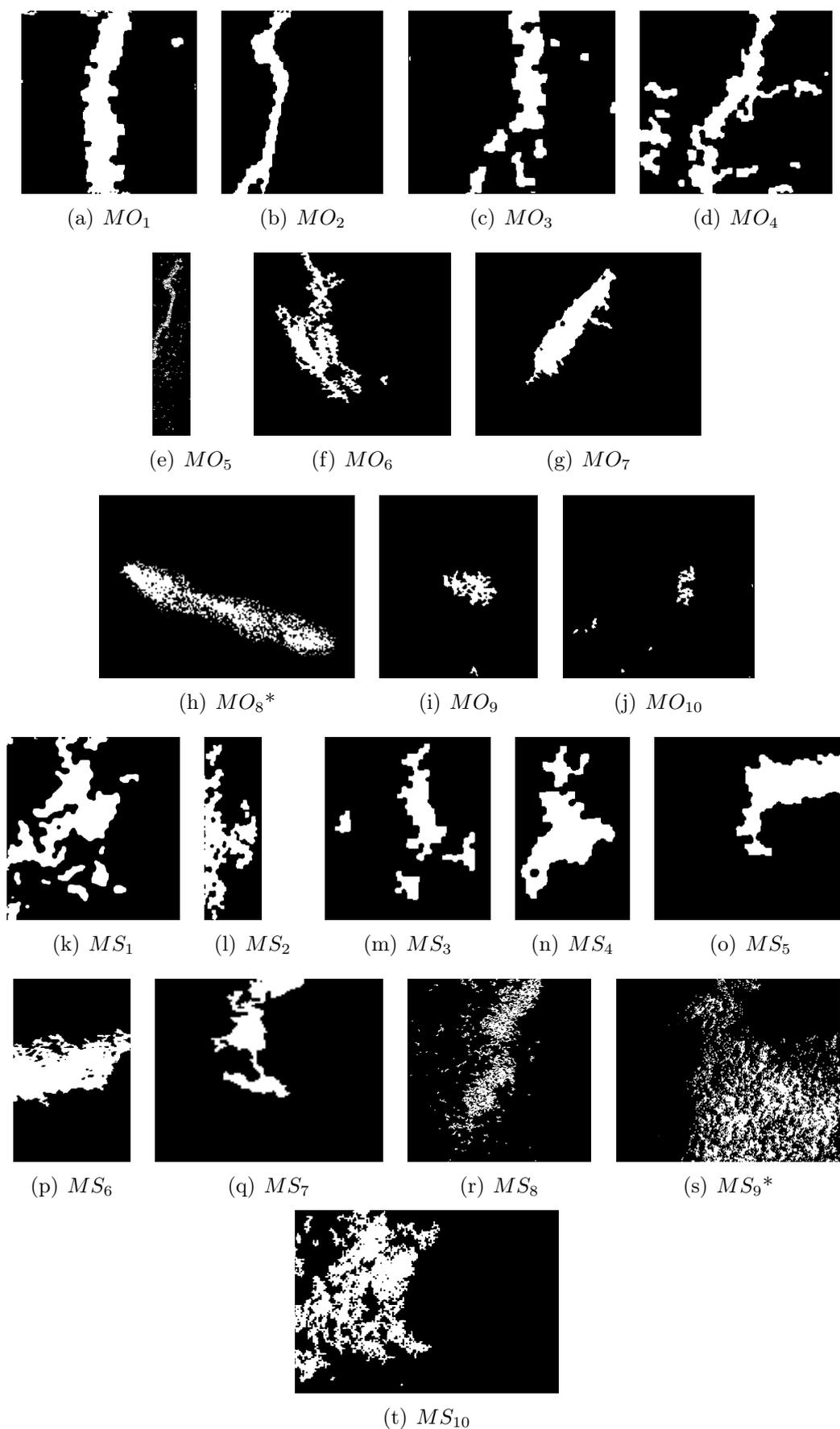


Figura B.4: Máscaras utilizadas na extração de características. As máscaras marcadas com "*" foram criadas manualmente. As demais foram segmentadas utilizando o procedimento de Araújo (2004).

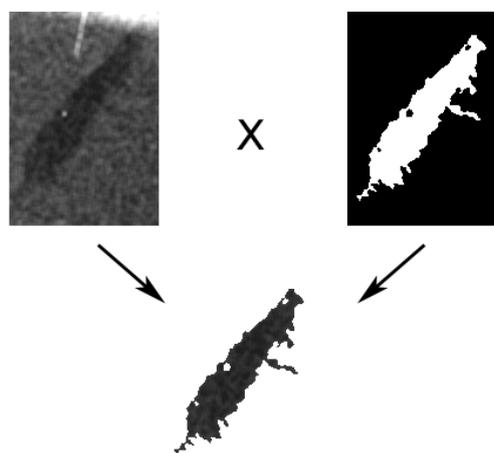


Figura B.5: Resultado da multiplicação da imagem da mancha pela sua máscara.

Métodos de Extração de Características Utilizados

Os sistemas de imageamento, dos quais as imagens SAR utilizadas neste trabalho são provenientes, apresentam número de visadas e resolução diferentes. Daí a necessidade de utilizar diferentes métodos de extração de características capazes de descrever esse conjunto heterogêneo de amostras. Brekke e Solberg (2005) relatam diversas técnicas de extração de características baseadas na geometria e na textura da mancha. Alguns dos métodos relacionados foram adotados neste trabalho. O método de extração de características espectrais de textura foi desenvolvido durante a pesquisa.

Características Geométricas da Mancha

Manchas causadas por derramamento de óleo têm, em geral, uma forma alongada bem definida. Manchas naturais, em geral, não têm uma forma bem definida, mas possuem um certo nível de espalhamento e de complexidade significativamente diferente das manchas de óleo.

No trabalho de Frate et al. (2000), foram utilizados 11 descritores baseados na geometria da mancha que medem a sua forma, espalhamento e complexidade, dentre outros. Estes descritores são apresentados a seguir:

Área (A) área da mancha em km^2 .

Perímetro (P) comprimento da borda da mancha em km .

Complexidade (C) definida como sendo

$$C = \frac{P}{2\sqrt{\pi A}}.$$

Espalhamento (S) derivado dos autovalores λ_1 e λ_2 da transformação PCA e calculada sobre os vetores cujos componentes são as coordenadas dos pixels pertencentes à mancha

$$S = \frac{100\lambda_2}{\lambda_1 + \lambda_2}.$$

Desvio Padrão da Mancha (Osd) desvio padrão calculado em dB da intensidade dos pixels pertencentes à imagem.

Desvio Padrão da Cena (Bsd) desvio padrão calculado em dB da intensidade dos pixels que rodeiam a mancha.

Contraste Máximo (ConMax) diferença em dB entre a média do fundo e o menor valor dentro da área da mancha.

Contraste Médio (ConMe) diferença em dB entre a média do fundo e o valor médio dentro da área da mancha.

Gradiente Máximo (GMax) valor máximo em dB do gradiente da borda.

Gradiente Médio (GMe) valor médio em dB do gradiente da borda.

Desvio Padrão do Gradiente (GSd) desvio padrão calculado em dB dos valores do gradiente da borda.

Após a ação de ventos e das correntes marinhas ou após algum tempo de degradação, as manchas de óleo perdem seu formato característico, tornando-se parecidas com as manchas naturais. Quando somente a geometria de uma mancha escura não fornece toda a informação necessária para determinar sua natureza, torna-se necessário definir um outro tipo de característica que ajude na interpretação de uma mancha desconhecida.

Características Espaciais de Textura

Sabendo-se que as manchas de óleo contribuem para o aparecimento de áreas homogêneas em imagens SAR e que as manchas naturais não apresentam necessariamente essa mesma propriedade, é possível descrever uma mancha escura desconhecida a partir da análise da sua textura. A análise da textura de um objeto em uma imagem fornece informações tais como suavidade, rugosidade e regularidade (GONZALEZ; WOODS, 1992). Segundo Gonzalez e Woods (1992), a informação de textura de um objeto presente em uma imagem digital pode ser obtida utilizando pelo menos duas abordagens, a estatística e a espectral, as quais foram utilizadas neste trabalho.

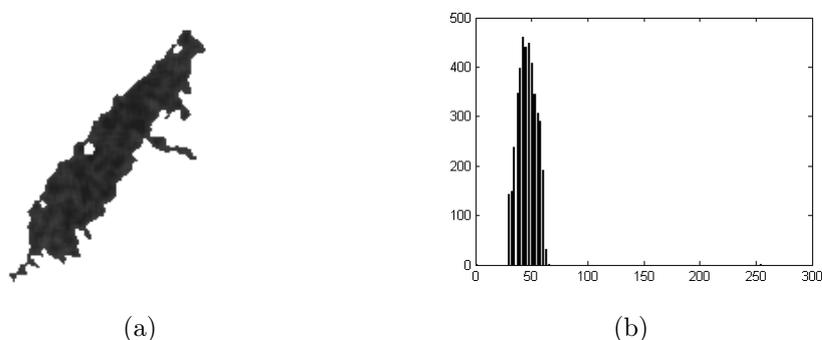


Figura C.1: (a) Mancha de óleo; (b) Histograma da imagem.

Na abordagem estatística, pode-se descrever a textura através dos momentos calculados sobre o histograma de níveis de cinza da imagem, de uma ROI ou de um objeto segmentado (ver Figura C.1). Também podem ser computadas as medidas estatísticas de segunda ordem, derivadas da Matriz de Coocorrência de Níveis de Cinza (GLCM, do inglês *Gray Level Cooccurrence Matrix*) (GONZALEZ; WOODS, 1992), como ilustra a Figura C.2. Os valores locais de textura podem ser obtidos tomando-se por base uma pequena janela centralizada em cada pixel da imagem.

Considere uma imagem com N níveis de cinza z_i , $i = 1, \dots, N$ e $p(z_i)$ o histograma correspondente. Os valores P_{ij} da GLCM representam as frequências obtidas para um nível de cinza relativo a sua vizinhança em uma dada direção. Normalizados, os valores P_{ij} podem ser interpretados como a probabilidade de ocorrência de um determinado nível de cinza.

As características extraídas da GLCM servem como medida para a diferenciação

de texturas locais que não seguem um determinado padrão de repetitividade, fornecendo informações relevantes para a classificação das mesmas, como foi observado por Haralick, Shanmugam e Dinstein (1973), Haralick (1979) e resumido no trabalho de Liew et al. (1995).

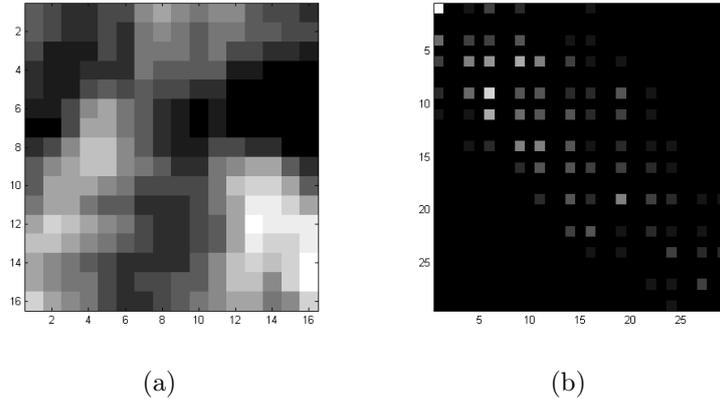


Figura C.2: (a) Detalhe de uma mancha de óleo e (b) sua respectiva GLCM.

Neste trabalho, são utilizadas 15 características definidas em Haralick (1979). A seguir são comentadas as principais características, as quais possuem uma interpretação clara do seu significado como descritores de textura. A partir delas são derivadas as demais.

Energia (E) medida da energia ou uniformidade dos pixels P_{ij} (valores mais altos de energia são atribuídos a distribuições periódicas ou constantes de pixels)

$$E = \sum_{i,j} P_{ij}^2.$$

Contraste (Con) mede a frequência espacial de uma imagem (reflete a quantidade de variações locais de intensidade dos pixels)

$$Con = \sum_{i,j} (i - j)^2 P_{ij}.$$

Entropia (Ent) mede o grau de complexidade da variação de intensidade dos pixels (texturas complexas tendem a uma maior entropia)

$$Ent = - \sum_{i,j} P_{ij} \log P_{ij}.$$

Média (μ) mede o valor médio de intensidade de pixels

$$\mu = \frac{1}{N} \sum_i z_i p(z_i).$$

Assimetria (Asm) mede a assimetria da distribuição intensidade de pixels em torno da média, $\mu_n = \frac{1}{N} \sum (z_i - \mu)^n p(z_i)$, de ordem $n = 3$ e normalizada pelo desvio padrão σ

$$Asm = \mu_3 / \sigma^3.$$

Curtose (Cur) mede o grau de concentração de níveis de intensidade dos pixels para a medida central de ordem $n = 4$

$$Cur = \mu_4 / \sigma^4.$$

Correlação (Cor) mede o grau de correlação entre os pixels considerando a média μ e variância σ nas direções x e y

$$Cor = \frac{\sum_{i,j} (i - \mu_x)(j - \mu_y) P_{ij}}{(\sigma_x \sigma_y)}.$$

Homogeneidade Local (Hom) mede o grau de homogeneidade entre os pixels adjacentes (valores mais altos representam pequenas diferenças de intensidade em pares de pixels)

$$Hom = \sum_{i,j} \frac{P_{ij}}{1 + (i - j)^2}.$$

Probabilidade Máxima (Pmax)

$$Pmax = \max P_{ij}.$$

Características Espectrais de Textura

Em geral, a textura de um objeto numa imagem digital é um padrão periódico que pode ser descrito através de uma análise espectral. A análise de textura pelo espectro de Fourier é útil para descrever a ocorrência de padrões periódicos ou semi-periódicos em uma imagem (GONZALEZ; WOODS, 1992). Entretanto, essa análise fornece apenas uma interpretação global da textura.

Quando a textura da imagem é fortemente influenciada por algum tipo de ruído, como nas imagens SAR, pode não existir um padrão periódico predominante para representá-la. Uma descrição mais detalhada da textura pode ser realizada através da transformada de Fourier janelada, a qual é útil para descrever padrões periódicos locais. Esta transformada, muitas vezes é chamada de transformada de Gabor.

Neste trabalho, porém, as imagens das manchas foram obtidas de diferentes sistemas de imageamento e possuem diferentes resoluções espaciais, o que sugere uma análise de textura multiescala. Devido ao princípio da incerteza, o janelamento da transformada de Gabor não é capaz de simultaneamente descrever as variações de intensidade dos pixels e localizá-las espacialmente. A análise por ondaletas possui propriedades favoráveis para resolver a questão da análise multiescala de textura.

Em essência, uma ondaleta é uma função Ψ de suporte compacto (ou seja, tem amplitude diferente de zero somente num intervalo finito limitado) e que possui média zero $\int_{-\infty}^{+\infty} \Psi(t) dt = 0$. A análise por ondaleta consiste em quebrar o sinal $x(t)$ original em versões da ondaleta mãe transladadas de b e escalonadas de a , conforme descreve a equação (C.1) para uma ondaleta genérica.

$$CWT(a, b) = \frac{1}{\sqrt{a}} \int_{-\infty}^{+\infty} x(t) \Psi \left(\frac{t-b}{a} \right) dt, \quad (C.1)$$

em que $\Psi(t)$ é a ondaleta mãe, $\Psi_{a,b}(t) = \frac{1}{\sqrt{a}} \Psi \left(\frac{t-b}{a} \right)$ representa as ondaletas filhas geradas por sucessivos deslocamentos e dilatações.

A ondaleta Haar possui implementação bastante simples e de baixo custo computacional para aplicação em imagens (STOLLNITZ; DEROSE; SALESIN, 1995). Trata-se de uma ondaleta ortogonal, definida pela equação (C.2) para um espaço de funções V^k (MALLAT, 1999). A Figura C.3 ilustra uma ondaleta Haar.

$$\Psi_{j,k} = \Psi(2^k x - j), \quad (C.2)$$

em que $j = 0, \dots, 2^k - 1$ e

$$\Psi(x) = \begin{cases} 1, & x \in [0, \frac{1}{2}) \\ -1, & x \in [\frac{1}{2}, 1) \\ 0, & x \notin [0, 1) \end{cases} \quad (C.3)$$

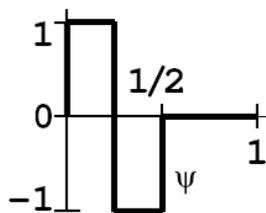


Figura C.3: Função base da ondaleta Haar.

Para a extração de características, uma imagem é decomposta em vários níveis pela ondaleta Haar e são computados os coeficientes de aproximação e de detalhe. Em cada nível de decomposição a dimensão do sinal de entrada é reduzida pela metade. As decomposições subseqüentes são realizadas sobre o coeficiente de aproximação. Mudanças bruscas de intensidade de pixels são evidenciadas pelos coeficientes de alta frequência ou coeficientes de detalhe. A energia destes coeficientes é utilizada para compor o vetor de descritores de textura de uma região da imagem.

Extração das características espectrais de textura utilizando ondaletas

Neste trabalho, o procedimento de extração de características de textura das manchas utilizando ondaletas, inicia com as máscaras ilustradas no Apêndice B. A imagem original da mancha é multiplicada ponto-a-ponto pela sua máscara, gerando uma nova imagem onde aparecem apenas os pixels pertencentes à mancha. Esta imagem é, então, transformada em um vetor para, posteriormente ser aplicada a transformada Haar unidimensional e gerar a primeira decomposição do vetor. As decomposições são realizadas sucessivamente até um nível N escolhido arbitrariamente. Os coeficientes de detalhe D_n formam um conjunto de descritores de textura com a dimensão desejada. O procedimento está ilustrado na Figura C.4.

O vetor de características espectrais de textura é gerado a partir da energia computada sobre 7 coeficientes de detalhe obtidos em 7 níveis decomposições Haar unidimensional. Em uma análise prévia realizada com um número menor de decomposições, observou-se que o vetor de características não oferece boa separabilidade dos dados. Devido à decomposição multiescala, o primeiro coeficiente de detalhe que aparecem na Figura C.4 (d) possui metade do tamanho do vetor original. O mesmo acontece nos níveis subseqüentes. A energia desses cada um desses vetores, tomada como sendo a soma dos quadrados das intensidades dos pixels, é utilizada para formar o conjunto final de características espectrais de textura.

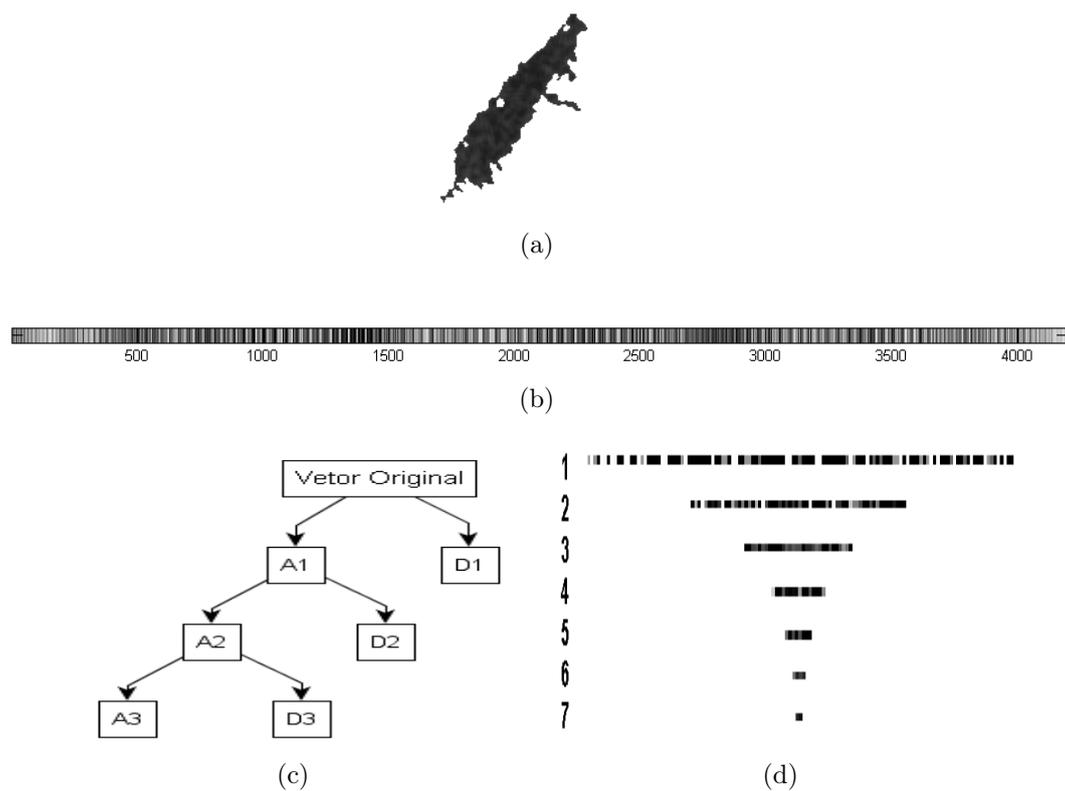


Figura C.4: (a) Mancha de óleo segmentada. (b) Imagem vetorizada da mancha. (c) Esquema de decomposições. (d) Coeficientes de detalhe obtidos em 7 decomposições sucessivas.

Referências Bibliográficas

- ARAÚJO, R. T. S. *Detecção de Manchas de Óleo na Superfície do Mar em Imagens de Radar de Abertura Sintética*. Dissertação (Mestrado em Engenharia Elétrica) — Universidade Federal do Ceará, 2004.
- ASSILZADEH, H.; MANSOR, S. B. Early warning system for oil spill using SAR images. In: *Proc. ACRS 2001 - 22nd Asian Conference on Remote Sensing*. Singapore: [s.n.], 2001. v. 1, p. 460–465.
- BAILLY, J. S.; ARNAUD, M.; PUECH, C. Boosting: a classification method for remote sensing. *International Journal of Remote Sensing*, v. 28, n. 7, p. 1687–1710, 2007.
- BARNI, M.; BETTI, M.; MECOCCI, A. A fuzzy approach to oil spill detection on SAR images. *IEEE International Geoscience and Remote Sensing Symposium, 1995. IGARSS '95.*, v. 1, p. 157–159, jul. 1995.
- BARRON, A. R. Universal approximation bounds for superpositions of a sigmoidal function. *IEEE Transactions on Information Theory*, v. 39, n. 3, p. 930–945, 1993.
- BAUM, E. B.; HAUSSLER, D. What size net gives valid generalization? *Neural Computation*, v. 1, p. 151–160, jan. 1989.
- BITTENCOURT, V. G. *Aplicação de Técnicas de Aprendizado de Máquina no Reconhecimento de Classes Estruturais de Proteínas*. Dissertação (Mestrado em Ciências) — Universidade Federal do Rio Grande do Norte, Natal, nov. 2005.

- BREIMAN, L. Bagging predictors. *Machine Learning*, v. 24, n. 2, p. 123–140, 1996.
- BREKKE, C.; SOLBERG, A. H. S. Oil spill detection by satellite remote sensing. *Remote Sensing of Environment*, v. 95, n. 1, p. 1–13, mar. 2005.
- BREVE, F. A.; PONTI-JUNIOR, M. P.; MASCARENHAS, N. D. A. Multilayer perceptron classifier combination for identification of materials on noisy soil science multispectral images. *Proceedings of the XX Brazilian Symposium on Computer Graphics and Image Processing (SIBGRAPI 2007)*, p. 239–244, 2007.
- BÜHLMANN, P. Boosting methods: Why they can be useful for high-dimensional data. *Proceedings of the 3rd Workshop on Distributed Statistical Computing - DSC 2003*, Viena, mar. 2003.
- CAI, Y.-D.; FENG, K.-Y.; LU, W.-C.; CHOU, K.-C. Using logitboost classifier to predict protein structural classes. *Journal of Theoretical Biology*, v. 238, p. 172–176, 2006.
- DETTLING, M.; BÜHLMANN, P. Boosting for tumor classification with gene expression data. *Bioinformatics*, v. 19, n. 9, p. 1061–1069, jun. 2003.
- DUDA, R. O.; HART, P. E.; STORK, D. G. *Pattern Classification*. 2. ed. Inglaterra: Wiley, 2000. 738s p.
- EFRON, B.; TIBSHIRANI, R. *An Introduction to the Bootstrap*. Nova Iorque: Chapman and Hall, 1993. 436 p.
- FENG, K.-Y.; CAI, Y.-D.; CHOU, K.-C. Boosting classifier for predicting protein domain structural class. *Biochemical and Biophysical Research Communications*, v. 334, p. 213–217, 2005.
- FISCELLA, B.; GIANCASPRO, A.; NIRCHIO, F.; PAVESE, P.; TRIVERO, P. Oil spill detection using marine SAR images. *International Journal of Remote Sensing*, v. 21, n. 18, p. 3561–3566, 2000.
- FRATE, F. D.; PETROCCHI, A.; LICHTENEGGER, J.; CALABRESI, G. Neural networks for oil spill detection using ERS-SAR data. *IEEE Transactions on Geoscience and Remote Sensing*, v. 38, n. 5, p. 2282–2287, 2000.

- FRATE, F. D.; SALVATORI, L. Oil spill detection by means of neural networks algorithms: a sensitivity analysis. *IEEE International Geoscience and Remote Sensing Symposium*, v. 2, p. 1370–1373, 2004.
- FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, p. 119–139, 1997.
- FREUND, Y.; SCHAPIRE, R. E. A short introduction to boosting. *Journal of Japanese Society for Artificial Intelligence*, v. 14, n. 5, p. 771–780, 1999.
- FRIEDMAN, J.; HASTIE, T.; TIBSHIRANI, R. Additive logistic regression: a statistical view of boosting. *The Annals of Statistics*, v. 38, n. 2, p. 337–374, 2000.
- GADE, M.; REDONDO, J. M. Marine pollution in european coastal waters monitored by the ERS-2 SAR: a comprehensive statistical analysis. *OCEANS '99 MTS - IEEE*, v. 3, p. 1239–1243, 1999.
- GONZALEZ, R. C.; WOODS, R. E. *Processamento de Imagens Digitais*. São Paulo: Edgar Blücher, 1992. 509 p.
- GUNN, S. *Support Vector Machine MATLAB Toolbox*. 1998. Acessado em 1/3/2007. Disponível em: <<http://www.isis.ecs.soton.ac.uk/resources/svminfo>>.
- HAMMOUCHE, K.; DIAF, M.; POSTAIRE, J.-G. A clustering method based on multidimensional texture analysis. *Pattern Recognition*, v. 39, n. 7, p. 1265–1277, jul. 2006.
- HARALICK, R. M. Statistical and structural approaches to texture. *Proceedings of the IEEE*, v. 67, n. 5, p. 786–804, 1979.
- HARALICK, R. M.; SHANMUGAM, K.; DINSTEN, I. Textural features for image classification. *IEEE Transactions on Systems, Man, and Cybernetics*, v. 3, n. 6, p. 610–621, 1973.
- HAYKIN, S. *Redes Neurais, princípios e prática*. 2. ed. Porto Alegre: Bookman, 2001.

- ITOPF The International Tanker Owners Pollution Federation Ltd. *Oil Tank Spill Statistics:2006*. 2006. Acessado em 25/11/2007. Disponível em: <<http://www.itopf.com>>.
- JAIN, A. K.; DUIN, R. P.; MAO, J. Statistical pattern recognition: a review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 22, n. 1, p. 4–37, jan. 2000.
- KEARNS, M. J.; VALIANT, L. G. Cryptographic limitations on learning boolean formulae and finite automata. *Proceeding on 21th Annual Association for Computing Machinery Symposium on Theory of Computing*, Nova Iorque, p. 433–444, 1989.
- KOTSIANTIS, S. B.; PANAYIOTIS, P. E. Logitboost of simple bayesian classifier. *Informatica - International Journal of Computing and Informatics (Slovenian Society Informatika)*, v. 29, n. 1, p. 53–59, 2005.
- KUNCHEVA, L. I.; WHITAKER, C. J. Measures of diversity in classifier ensembles and their relationship with the ensemble accuracy. *Machine Learning*, v. 51, n. 2, p. 181–207, 2003.
- LI, X.; WANG, L.; SUNG, E. Adaboost with SVM -based component classifiers. *Engineering Applications of Artificial Intelligence*, 2007. [In press]. Acessado em 27/12/2007. Disponível em: <<http://www.sciencedirect.com>>.
- LIEW, S. C.; LIM, H.; KWOH, L. K.; TAY, G. K. Texture analysis of SAR images. *International Geoscience and Remote Sensing Symposium - IGARS'95*, v. 2, p. 1412–1414, jul. 1995.
- LIU, J. J.; MACGREGOR, J. F. On the extraction of spectral and spatial information from images. *Chemometrics and Intelligent Laboratory Systems*, v. 85, n. 1, p. 119–130, jan. 2007.
- LOPES, D. F. A.; RAMALHO, G. L. B.; MEDEIROS, F. N. S. de; COSTA, R. C. S.; ARAÚJO, R. T. S. Combining features to improve oil spill classification in SAR images. *SSPR 2006. Lecture Notes in Computer Science, Berlin*, Springer-Verlag, v. 4109, p. 928–936, 2006.

- LU, D.; WENG, Q. A survey of image classification methods and techniques for improving classification performance. *International Journal of Remote Sensing*, v. 28, n. 5, p. 823–870, mar. 2007.
- MALLAT, S. *A Wavelet tour of signal processing*. 2a. ed. [S.l.]: Academic Press, 1999.
- MARTINEZ, A.; MORENO, V. An Oil Spill Monitoring System Based on SAR Images. *Spill Science and Technology Bulletin*, v. 3, n. 1/2, p. 65–71, 1996.
- MEDEIROS, F. N.; MASCARENHAS, N. D.; COSTA, L. da F. Adaptive speckle MAP filtering for SAR images using statistical clustering. *International Symposium on Computer Graphics, Image Processing and Vision*, p. 303–310, 1998.
- MERCIER, G.; GIRARD-ARDHUIN, F. Oil slick detection by SAR imagery using support vector machines. *Proceedings on Oceans*, v. 1, p. 90 – 95, jun. 2005.
- MERCIER, G.; GIRARD-ARDHUIN, F. Unsupervised oil slick detection by SAR imagery using kernel expansion. *Geoscience and Remote Sensing Symposium, 2005. IGARSS '05. Proceedings. 2005 IEEE International*, v. 1, p. 25–29, jul. 2005.
- PARK, J. H.; ZHOU, S. K.; SIMOPOULOS, C.; OTSUKI, J.; COMANICIU, D. Automatic cardiac view classification of echocardiogram. *IEEE International Conference on Computer Vision (ICCV2007)*, 2007.
- PONTI-JUNIOR, M. P. *Combinação de Múltiplos Classificadores para Identificação de Materiais em Imagens Ruidosas*. Dissertação (Mestrado em Ciências da Computação) — Universidade Federal de São Carlos, 2004.
- RAMALHO, G. L. B.; MEDEIROS, F. N. S. de. Using boosting to improve oil spill detection in SAR images. *18th International Conference on Pattern Recognition (ICPR 2006)*, IEEE Computer Society, Hong-Kong, v. 2, n. 20-24, p. 1066–1069, 2006.
- RAMALHO, G. L. B.; MEDEIROS, F. N. S. de. Improving reliability of oil spill detection systems using boosting for high-level feature selection. *International Conference on Image Analysis and Recognition 2007. Lecture Notes in Computer Science*, Montreal, n. 4633, p. 1172–1181, 2007.

- RICHARDS, J. A.; JIA, X. *Remote Sensing Digital Image Analysis - An Introduction*. 4a. ed. Heidelberg: Springer-Verlag, 2006. 439 p.
- SÁ, J. P. M. de. *Pattern Recognition - Concepts, Methods And Applications*. 2. ed. Berlim: Springer, 2001. 738 p.
- SCHAPIRE, R. E. The strength of weak learnability. *Machine Learning*, v. 5, n. 2, p. 197–227, 1990.
- SCHAPIRE, R. E.; FREUND, Y.; BARLETT, P.; LEE, W. S. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, v. 26, n. 5, p. 1651–1686, 1998.
- SITA, G.; RAMAKRISHNAN, A. G. Wavelet domain nonlinear filtering for evoked potential signal enhancement. *Computers and Biomedical Research*, v. 33, n. 6, p. 431–446, 2000.
- SOLBERG, A. S.; BREKKE, C.; SOLBERG, R. Algorithms for oil spill detection in radarsat and ENVISAT SAR images. *IEEE International Geoscience and Remote Sensing Symposium, 2004. IGARSS '04. Proceedings*, v. 7, p. 4909–4912, 2004.
- STOLLNITZ, E. J.; DEROSE, T. D.; SALESIN, D. H. Wavelets for computer graphics: a primer - part 1. *IEEE Computer Graphics and Applications*, v. 15, n. 3, p. 76–84, 1995.
- THEODORIDIS, S.; KOUTROUMBAS, K. *Pattern Recognition*. 2. ed. San Diego: Elsevier, 2003. 689 p.
- TOPOUZELIS, K.; KARATHANASSI, V.; PAVLAKIS, P.; ROKOS, D. Oil spill detection using RBF neural networks and SAR data. *Proceeding on 20th Congress of International Society for Photogrammetry and Remote Sensing*, v. 35, p. 724–729, jul. 2004.
- WEBB, A. R. *Statistical Pattern Recognition*. 2. ed. Inglaterra: Wiley, 2002. 496 p.
- YIN, X. C.; LIU, C. P.; HAN, Z. Feature combination using boosting. *Pattern Recognition Letters*, v. 26, n. 14, p. 2195–2205, out. 2005.