Técnicas de Seleção de Características com Aplicações em Reconhecimento de Faces

Teófilo Emídio de Campos

DISSERTAÇÃO APRESENTADA
AO
INSTITUTO DE MATEMÁTICA E ESTATÍSTICA
DA
UNIVERSIDADE DE SÃO PAULO

PARA OBTENÇÃO DO GRAU DE MESTRE
EM
CIÊNCIA DA COMPUTAÇÃO

Área de Concentração : Ciência da Computação
Orientador : Prof. Dr. Roberto Marcondes Cesar Junior

- São Paulo, 25 de maio de 2001 -

Técnicas de Seleção de Características com Aplicações em Reconhecimento de Faces

Este exemplar corresponde à redação
final da dissertação devidamente corrigida
e apresentada por Teófilo Emídio de Campos e aprovada
pela Comissão Julgadora.

São Paulo, 25 de maio de 2001

Banca Examinadora :

- Prof. Dr. Roberto Marcondes Cesar Junior (orientador) - MAC-IME-USP
- Prof. Dr. Junior Barrera - MAC-IME-USP
- Prof. Dr. João Kogler - LSI-POLI-USP

aos meus pais Maria Rita e Benedicto

Agradecimentos

Após esses 27 meses de trabalho aqui no IME-USP, é difícil criar uma lista contendo todas as pessoas que me apoiaram e contribuíram direta e indiretamente para o desenvolvimento dessa dissertação e para o meu crescimento. Por isso eu gostaria de me desculpar por todos os nomes que eu omiti neste espaço.

Inicio meus agradecimentos citando a Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP) pelo apoio financeiro (processos 99/01488-8 e 99/12765-3). Agradeço ao Prof. Roberto Cesar, que fez brotar meu interesse pela pesquisa em visão computacional em 1995 e se tornou mais que um orientador, mas um amigo. Também devo agradecer aos professores Carlos Hitoshi Morimoto e Junior Barrera, pelos cursos extremamente relevantes para o desenvolvimento deste trabalho e pelas discussões nos seminários e, principalmente, na qualificação. Em especial, agradeço ao Prof. Hitoshi por fornecer algumas bases de imagens.

Agradeço à Isabelle Bloch (Ecole Nationale Supérieure des Télécommunications - Paris), pela colaboração na implementação da principal contribuição deste trabalho. De maneira semelhante, agradeço ao Pavel Pudil e ao Petr Somol (Academy of Sciences of the Czech Republic) pelas discussões e por fornecerem o código fonte dos métodos de busca flutuante e flutuante adaptativa para seleção de características. [1]

Há algumas pessoas que inicialmente participavam de meu dia-a-dia, como membro do grupo de pesquisa ou como aluno de mestrado, que sempre me auxiliaram e que hoje eu também considero grandes amigos. Dentre eles estão: Rogério S. Feris, grande companheiro de projetos e de congressos, Franklin C. Flores, cujo apoio e as discussões foram muito importantes no decorrer de todo o período, Sérgio R. Gaspar, que sempre esteve de prontidão para nos auxiliar em algumas tarefas relacionadas com este trabalho, Jorge Bittencourt, pelo apoio fundamental no início do meu mestrado e pela ajuda com o inglês,

[1] I would like to thank Isabelle Bloch (Ecole Nationale Supérieure des Télécommunications - Paris) for the collaboration that lead to the main contribution of this M.Sc. Thesis. I would also like to thank Pavel Pudil and Petr Somol (Academy of Sciences of the Czech Republic) for useful discussions and for providing the source code of the sequencial floating search methods and of the adaptive versions of these methods for feature selection.

Marcel Brun, pelas dicas e ferramentas do MatLab, ao Roberto Hirata, pela manutenção da rede, e a todos os outros colegas do Laboratório de Processamento de Imagens pela amizade e pelos auxílios com Linux e LaTeX.

Por último, gostaria de registrar meus sinceros agradecimentos à minha namorada Silvana M. Vicente, pelo apoio, carinho, compreensão, paciência e pelo tempo que ela dispensou para revisão deste e de outros textos que foram produzidos no decorrer desse período.

Resumo

O reconhecimento de faces é uma área de pesquisa desafiadora que abre portas para a implementação de aplicações muito promissoras. Embora muitos algoritmos eficientes e robustos já tenham sido propostos, ainda restam vários desafios. Dentre os principais obstáculos a serem superados, está a obtenção de uma representação robusta e compacta de faces que possibilite distinguir os indivíduos rapidamente.

Visando abordar esse problema, foi realizado um estudo de técnicas de reconhecimento estatístico de padrões, principalmente na área de redução de dimensionalidade dos dados, além de uma revisão de métodos de reconhecimento de faces. Foi proposto (em colaboração com a pesquisadora Isabelle Bloch) um método de seleção de características que une um algoritmo de busca eficiente (métodos de busca seqüencial flutuante) com uma medida de distância entre conjuntos nebulosos (distância nebulosa baseada em tolerância). Essa medida de distância possui diversas vantagens, sendo possível considerar as diferentes tipicalidades de cada padrão dos conjuntos de modo a permitir a obtenção de bons resultados mesmo com conjuntos com sobreposição. Os resultados preliminares com dados sintéticos mostraram o caráter promissor dessa abordagem.

Com o objetivo de verificar a eficiência de tal técnica com dados reais, foram efetuados testes com reconhecimento de pessoas usando imagens da região dos olhos. Nesse caso, em se tratando de um problema com mais de duas classes, nós propusemos uma nova função critério inspirada na distância supracitada. Além disso foi proposto (juntamente com o estudante de mestrado Rogério S. Feris) um esquema de reconhecimento a partir de seqüências de vídeo. Esse esquema inclui a utilização de um método eficiente de rastreamento de características faciais (Gabor Wavelet Networks) e o método proposto anteriormente para seleção de características. Dentro desse contexto, o trabalho desenvolvido nesta dissertação implementa uma parte dos módulos desse esquema. Detalhes sobre os trabalhos correlatos e outras informações podem ser encontradas em $http://www.vision.ime.usp.br/\tilde{}creativision$.

Abstract

Face recognition is an instigating research field that may lead to the development of many promising applications. Although many efficient and robust algorithms have been developed in this area, there are still many challenges to be overcome. In particular, a robust and compact face representation is still to be found, which would allow for quick classification of different individuals.

In order to address this problem, we first studied pattern recognition techniques, especially regarding dimensionality reduction, followed by the main face recognition methods. We introduced a new feature selection approach in collaboration with the researcher Isabelle Bloch (TSI-ENST-Paris), that associates an efficient searching algorithm (sequential floating search methods), with a tolerance-based fuzzy distance. This distance measure presents some nice features for dealing with the tipicalities of each pattern in the sets, so that good results can be attained even when the sets are overlapping. Preliminary results with synthetic data have demonstrated that this method is quite promising.

In order to verify the efficiency of this technique with real data, we applied it for improving the performance of a person recognition system based on eye images. Since this problem involves more than two classes, we also developed a new criterion function based on the above-mentioned distance. Moreover, we proposed (together with Rogério S. Feris) a system for person recognition based on video sequences. This mechanism includes the development of an efficient method for facial features tracking, in addition to our method for feature selection. In this context, the work presented here constitutes part of the proposed system. Related work and other information can be found at $http://www.vision.ime.usp.br/\tilde{}creativision$.

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Métodos de identificação de pessoas sempre foram muito importantes para toda a sociedade. No mundo moderno, as pessoas normalmente precisam carregar documentos para quaisquer lugares que forem, pois essa é a única forma de provarem suas identidades. Assumindo-se que não existem pessoas completamente idênticas, a necessidade da utilização de tais documentos extingue-se quando se dispõe de métodos capazes de diferenciar cada indivíduo sem confundi-lo com seus semelhantes. Provavelmente esse é o principal objetivo da pesquisa em Biometria. Um sistema biométrico é um sistema de reconhecimento de padrões que estabelece a autenticidade de uma característica fisiológica ou comportamental possuída por um usuário [Pankanti et al., 2000, Ratha et al., 2001].

Dentre as técnicas de reconhecimento biométrico de pessoas que são utilizadas atualmente, as mais precisas são aquelas baseadas em imagens do fundo da retina e as baseadas em imagens de íris [Pankanti et al., 2000, Ratha et al., 2001]. A confiabilidade de sistemas de reconhecimento de íris é tão grande que já existem bancos os adotando para identificar seus usuários. Porém, essas abordagens têm o problema de serem um tanto invasivas, pois, para o funcionamento dos sistemas atuais, é necessário impor certas condições ao usuário. No caso dos sistemas de reconhecimento por imagem de íris, o usuário deve permanecer parado em uma posição definida e com os olhos abertos enquanto uma fonte de luz ilumina os olhos e um *scanner* de íris ou uma câmera captura a imagem. O caráter invasivo acentua-se em sistemas que utilizam imagens de fundo de retina, uma vez que atualmente é preciso utilizar um colírio para dilatar a pupila do usuário antes de efetuar a aquisição da imagem. Nesse ponto está a mais sobressalente vantagem de um sistema de reconhecimento baseado em imagens de faces.

A pesquisa em reconhecimento de faces vem se desenvolvendo no sentido da criação de sistemas capazes de identificar pessoas mesmo quando essas não percebam que estão sendo observadas. Dessa forma, é possível que, no futuro, uma criança desaparecida seja localizada através de imagens de câmeras localizadas em pontos estratégicos de uma cidade, como estações de metrô e cruzamentos de avenidas.

Além dessas, várias outras aplicações motivantes para a pesquisa nessa área foram analisadas em [Chellappa et al., 1995], como:

- identificação pessoal para banco, passaporte, fichas criminais;
- sistemas de segurança e controle de acesso;
- monitoramento de multidões em estações, *shopping centers* etc.;
- criação de retrato falado;
- busca em fichas criminais;
- envelhecimento computadorizado para auxiliar a busca por desaparecidos, e
- interfaces perceptuais homem-máquina com reconhecimento de expressões faciais.

Devido à sua importância prática e aos interesses dos cientistas cognitivos, a pesquisa em reconhecimento de faces é tão antiga quanto a própria visão computacional [Pentland, 2000]. Em [Chellappa et al., 1995], há uma análise de trinta anos de pesquisa em reconhecimento de faces humano e por máquina o qual cita 221 trabalhos. Outra evidência do crescimento dessa área de pesquisa é a existência de conferências específicas de reconhecimento de face e gestos [Bichsel, 1995, Essa, 1996, Yachida, 1998, Crowley, 2000], bem como a existência de revistas com seções temáticas nessa área (por exemplo [Kasturi, 1997]). Além disso, recentemente foi lançado um livro sobre visão dinâmica voltado ao problema de reconhecimento de faces [Gong et al., 2000].

O reconhecimento óptico automático (computacional) de faces é uma sub-área de pesquisa da visão computacional. A área de visão computacional é altamente multidisciplinar. Seu principal objetivo é a investigação de métodos automáticos de extração de informações contidas em imagens [Gong et al., 2000]. Em geral, são utilizados elementos de processamento de imagens e de reconhecimento de padrões para extrair e interpretar tais informações. Em reconhecimento de faces, o objetivo é identificar pessoas que aparecem em imagens.

Para melhorar a possibilidade de êxito em um sistema de reconhecimento de faces, primeiramente é preciso segmentá-las para que somente essas sejam tratadas. Isso permite que não sejam considerados os objetos que estiverem atrás do sujeito a ser reconhecido (*background*), os quais podem influenciar na tomada de decisão do classificador. Para

isso, é utilizado um método de detecção de faces, o qual tenta determinar a localização de faces em uma imagem para que essas sejam posteriormente segmentadas.

No caso de seqüências de vídeo, a segmentação deve ser feita em todas as imagens da seqüência. O problema é que geralmente imagens de cenas dinâmicas (cenas apresentando variações com o tempo, ou seja, movimento) apresentam menor qualidade devido a *borramentos* proporcionados pelo próprio movimento dos objetos e do observador (câmera). Além disso, imagens em movimento atualmente são representadas por seqüências de imagens capturadas em pequenos intervalos de tempo (usualmente até 30 quadros por segundo). Como resultado, tais representações ocupam muito espaço na memória de um computador. Geralmente esse problema é amenizado adotando-se imagens com menor resolução em seqüências de vídeo, o que compromete ainda mais a qualidade das imagens [Chellappa et al., 1995].

Várias aplicações associadas a reconhecimento de faces a partir de seqüências de vídeo requerem que os processos sejam muito eficientes, principalmente aquelas em tempo real. Por isso, em geral adota-se um método de detecção de faces somente no primeiro quadro da seqüência em que a pessoa aparece, sendo subseqüentemente aplicado um procedimento de perseguição (ou rastreamento - *tracking*), que, por sua vez, é mais rápido, pois considera informações obtidas no quadro anterior para segmentar faces, de forma a evitar a realização de buscas por toda a imagem. Como exemplos de métodos rápidos de detecção e perseguição de faces em seqüências de imagens, podem-se citar: [Campos et al., 2000c], [Feris and Cesar-Jr, 2001], [Feris et al., 2000], [Krüger and Sommer, 2000], [Kondo and Yan, 1999], [Rowley et al., 1998], [Wu et al., 1999], [Sung and Poggio, 1998], [Silva et al., 1995], [Cascia and Sclaroff, 1999], [Yang et al., 1997] e [Krüger and Sommer, 1999].

Após a segmentação da face, é necessário normalizá-la em relação a translação, a rotação e a intensidade dos tons de cinza. Essas normalizações são necessárias para reduzir as variações existentes em diferentes imagens de uma mesma pessoa. Tais variações dificultam o processo de reconhecimento.

Para que o processo de reconhecimento seja rápido, devem-se utilizar bons algoritmos de redução da dimensionalidade dos dados. Esses algoritmos têm por objetivo extrair somente as informações essenciais das imagens para possibilitar que seja efetuado reconhecimento (classificação) de forma eficiente. O estudo desses métodos é o principal objetivo deste trabalho. A figura 1.1 mostra a organização desses elementos básicos que compõem um sistema de reconhecimento de faces a partir de seqüências de vídeo.

1.1 Objetivos

O objetivo original deste trabalho é o estudo de métodos de extração de características e de classificação estatística para aplicação em reconhecimento de faces. Mais especifica-

mente, concentramos nosso trabalho no estudo de técnicas de redução de dimensionalidade utilizando principalmente seleção de características com vistas à criação de um método de classificação de faces[1] através de seqüências de vídeo.

As técnicas de detecção e rastreamento de faces em seqüências de vídeo não fazem parte do escopo deste projeto. Este trabalho visa à realização de testes com o emprego de seqüências de imagens com a face já segmentada e normalizada com relação à escala e orientação. A obtenção dessas imagens de forma automática foi realizada por outro estudante deste departamento em seu trabalho de mestrado [Feris, 2001].

Estudamos métodos de classificação de padrões em geral, focalizando na aplicação em reconhecimento de faces. Percebemos a importância dos métodos de seleção de características nessa área de pesquisa, principalmente porque não existem pesquisas bem conhecidas utilizando tais técnicas no projeto de um sistema de reconhecimento de faces. Por isso, concentramo-nos na implementação e testes relacionados a esse problema.

1.2 Contribuições

Além de uma revisão bibliográfica de alguns métodos de reconhecimento de padrões e de faces, na busca por novos métodos eficientes de reconhecimento de faces, estudamos vários algoritmos e implementamos e testamos alguns métodos. Também criamos alguns algoritmos novos. Como resultado, apresentamos as seguintes contribuições:

- Realizamos testes de reconhecimento de pessoas utilizando imagens dos olhos e imagens englobando toda a face. Comparamos os resultados e concluimos que, quando o conjunto de treinamento é pequeno, a taxa de acerto do classificador é maior com a utilização de imagens englobando somente a região dos olhos (vide seção 5.1 e o artigo [Campos et al., 2000d]).

- Visando a avaliar dois algoritmos de busca para seleção de características, realizamos testes utilizando uma base de dados criada com a finalidade de treinar um discriminador de faces e não faces. Comparamos o algoritmo de busca com métodos tradicionais de seleção de características: seleção das m primeiras características e seleção das m maiores características ($m < N$, sendo N o conjunto de todas as características disponíveis). Detalhes sobre esse trabalho estão na seção 5.2.1 e no artigo [Campos et al., 2000c].

- Visando a realizar seleção de características considerando conjuntos com distribuição desconhecida e com fronteiras imprecisas, propusemos uma nova função critério. Tal função avalia conjuntos de características utilizando uma medida de distância entre

[1] É importante ressalvar que há várias tarefas relacionadas com a identificação de faces (vide seção 4.1). Este trabalho se restringe à tarefa de classificação.

conjuntos nebulosos. Foram realizados testes associando-se essa função a um algoritmo de busca para seleção de características em dados sintéticos. Essa abordagem e sua avaliação estão descritas na seção 3.4 e no artigo [Campos et al., 2001].

- A função critério citada no item anterior foi definida para problemas contendo apenas duas classes. Nos propusemos uma versão dessa função para c classes ($c > 2$). Realizamos testes exaustivos com essa função critério em comparação com outras funções para seleção de características. O problema abordado foi o reconhecimento de pessoas a partir de imagens da região dos olhos. Esses experimentos encontram-se relatados na seção 5.3 e parte dos resultados serão publicados em [Campos and Cesar-Jr, 2001].

- Propusemos um novo esquema para reconhecimento de faces a partir de seqüências de vídeo. Trata-se da associação de um método eficiente de detecção e rastreamento de pontos faciais característicos com métodos de normalização, extração de características, combinação de padrões com seleção de características e multiclassificação. Maiores detalhes encontram-se na seção 5.4 e no artigo [Campos et al., 2000b].

Outras contribuições que não se relacionam diretamente com o tema de pesquisa desta dissertação também foram obtidas e encontram-se em anexo.

1.3 Organização do Texto

Como o estudo realizado neste trabalho focaliza a aplicação de técnicas de reconhecimento de padrões (principalmente redução de dimensionalidade) ao reconhecimento de faces, esta dissertação divide-se em duas partes: reconhecimento de padrões (parte I) e aplicações ao reconhecimento de faces (parte II).

Na parte I, há uma revisão de métodos de reconhecimento de padrões (capítulo 2) dando maior atenção aos métodos estatísticos (seção 2.1), pois esse é o foco desta dissertação. Na seção 2.3, são abordados os problemas de generalização em reconhecimento de padrões. Tais problemas clamam pela utilização de métodos de redução de dimensionalidade, os quais são abordados no capítulo 3. Nesse capítulo, uma maior ênfase é dada aos métodos de seleção de características (seção 3.3), pois as principais contribuições dessa dissertação encontram-se nessa área de pesquisa. Também no capítulo 3, são descritos alguns testes realizados com um novo método de seleção de características por nós proposta (seção 3.4).

A parte II inicia-se com uma descrição das tarefas relacionadas com identificação de faces (seção 4.1) e prossegue com uma revisão das principais abordagens de extração de informações de imagens de faces para reconhecimento (seções 4.2 e 4.3). No capítulo 5, são descritos os projetos implementados (bem como seus resultados obtidos) visando a avaliar e criar métodos para proceder com reconhecimento de faces de forma eficiente.

Este texto finalizar-se-á com as conclusões e a descrição de possíveis trabalhos futuros que poderão ser implementados como continuação dessa pesquisa (capítulo 6).

O apêndice A contém a descrição dos símbolos e de algumas abreviações utilizadas no decorrer deste texto. Em anexo estão todas as publicações relacionadas com esta dissertação que foram realizadas no decorrer deste mestrado.

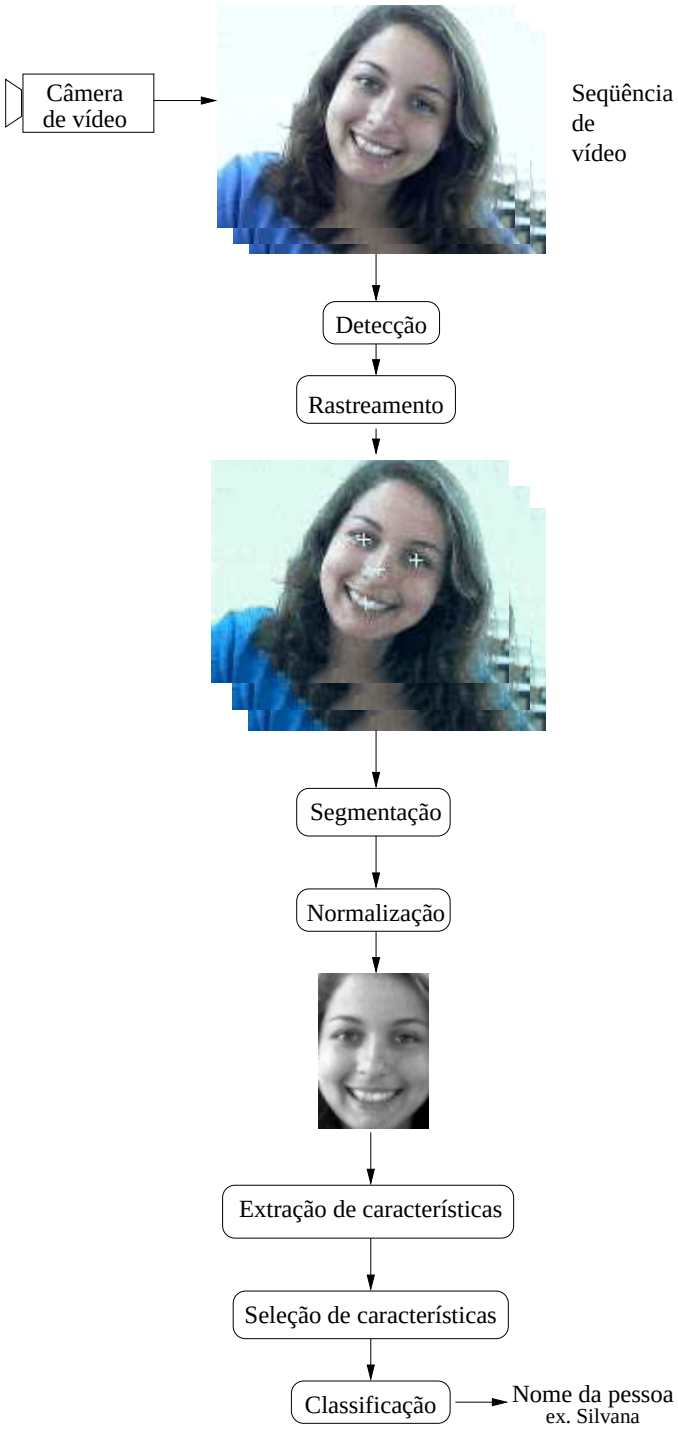

Figura 1.1: Esquema básico de um sistema de reconhecimento de faces a partir de seqüências de vídeo.

Parte I

Reconhecimento de Padrões

Capítulo 2

Conceitos Básicos de Reconhecimento de Padrões

Basicamente, o reconhecimento de padrões é a área de pesquisa que tem por objetivo a classificação de objetos (padrões) em um número de categorias ou classes (vide [Theodoridis and Koutroumbas, 1999]). Assim, dado um conjunto de c classes, $\omega_1, \omega_2, \cdots, \omega_c$, e um padrão desconhecido $\mathbf{x}$, um reconhecedor de padrões é um sistema que, auxiliado por pré-processamentos, extração e seleção de características, associa $\mathbf{x}$ ao rótulo i de uma classe ω_i. No caso de classificação de faces, uma imagem de face é o objeto (ou padrão $\mathbf{x}$) e as classes são seus nomes ou identificações (ω_i).

Segundo [Jain et al., 2000], nos últimos 50 anos de pesquisa, foram obtidos avanços que possibilitaram a evolução da pesquisa em aplicações altamente complexas. Um exemplo é o reconhecimento de faces, o qual consiste em um problema de visão computacional que requer técnicas robustas a translação, rotação, alteração na escala e a deformações do objeto. Além de reconhecimento de faces, os autores de [Jain et al., 2000] destacam os seguintes exemplos de aplicações atuais que requerem técnicas eficientes e robustas de reconhecimento de padrões:

- Bio-informática: análise de seqüências do genoma; aplicações e tecnologia de *micro-arrays*;
- Mineração de dados (*data mining*): a busca por padrões significativos em espaços multi-dimensionais, normalmente obtidos de grandes bases de dados e "data warehouses";

- Classificação de documentos da Internet;
- Análise de imagens de documentos para reconhecimento de caracteres (*Optical Character Rocognition - OCR*);
- Inspeção visual para automação industrial;
- Busca e classificação em base de dados multimídia;
- Reconhecimento biométrico, incluindo faces, íris ou impressões digitais;
- Sensoriamento remoto por imagens multiespectrais;
- Reconhecimento de fala.

Um ponto em comum a essas aplicações é que usualmente as características disponíveis nos padrões de entrada, tipicamente milhares, não são diretamente utilizadas. Normalmente utilizam-se características extraídas dos padrões de entrada otimizadas através de procedimentos guiados pelos dados, como PCA (vide seção 3.2.2).

Uma característica importante de reconhecimento de faces, assim como várias outras aplicações atuais, é que nenhuma abordagem individual é ótima, de modo que métodos e abordagens múltiplas devem ser utilizados combinando-se várias modalidades de sensores, pré-processamentos e métodos de classificação [Jain et al., 2000]. Assim, o projeto de sistemas de reconhecimento de padrões essencialmente envolve três aspectos: aquisição de dados e pré-processamento, representação dos dados e tomada de decisões. Geralmente o desafio encontra-se na escolha de técnicas para efetuar esses três aspectos.

Um problema de reconhecimento de padrões bem definido e restrito permite uma representação compacta dos padrões e uma estratégia de decisão simples. Seja $d_w(\omega_i)$ uma medida de separabilidade global entre os padrões pertencentes a uma classe ω_i (por exemplo, a média das variâncias em todas as características dos padrões de ω_i). Seja $d_b(\Omega)$ uma medida de separabilidade global entre as classes do conjunto de classes Ω (por exemplo, a média das distâncias entre as médias de todas as classes de Ω e a média global). Um problema de reconhecimento de padrões bem definido e restrito é aquele que, em seu espaço de características, possui distribuições de padrões com pequena variações intra-classe e grande variação inter-classes, ou seja, pequenos valores de $d_w(\omega_i)$ e um grande valor de $d_b(\Omega)$ [Theodoridis and Koutroumbas, 1999].

A questão é que, em dados reais, geralmente os padrões a serem reconhecidos não possuem essas peculiaridades. Nesse fato reside a importância de algoritmos de extração e seleção de características, pois eles reduzem a dimensionalidade dando prioridade para uma base do espaço de características que não perde o poder de discriminação dos padrões.

A seguir serão traçados detalhes a respeito dos métodos de reconhecimento estatísticos de padrão.

2.1 Abordagem estatística

2.1.1 Panorama de Reconhecimento de Padrões

Há várias abordagens diferentes para se efetuar reconhecimento de padrões. Dentre elas, podemos destacar:

- casamento (*template matching*) [Gonzalez and Woods, 1992], [Feris et al., 2000], [Theodoridis and Koutroumbas, 1999],
- abordagem sintática (por exemplo: *Hiddem Markov Models*) [Theodoridis and Koutroumbas, 1999], [Morimoto et al., 1996],
- redes neurais [Theodoridis and Koutroumbas, 1999];
- lógica nebulosa [Bloch, 1999, Dubois et al., 1997, Bonventi-Jr. and Costa, 2000];
- morfologia matemática com aprendizado computacional [Barrera et al., 2000];
- estatística.

É importante ressalvar que essa separação entre as abordagens, baseada no artigo [Jain et al., 2000], possui apenas fins didáticos, pois, apesar de possuírem aparentemente princípios diferentes, a maioria dos modelos de redes neurais populares são implicitamente equivalentes ou similares a métodos clássicos de reconhecimento estatístico de padrões. Entretanto, algumas redes neurais podem oferecer certas vantagens, como abordagens unificadas para extração de características, seleção de características e classificação, e procedimentos flexíveis para encontrar boas soluções não lineares [Jain et al., 2000].

A abordagem de morfologia matemática com aprendizado computacional é uma abordagem estatística. Porém, nessa abordagem, o espaço de características utilizado é discreto e não linear. Além disso, o método de classificação se baseia em busca em uma tabela. Com essas características, seu paradigma é bastante divergente de todas as abordagens relacionadas com este trabalho.

Este trabalho concentra-se em métodos estatísticos de reconhecimento de padrões, que é uma das abordagens mais populares e bem conhecidas. Por isso as outras abordagens não serão detalhadas. Porém, na revisão bibliográfica sobre métodos de reconhecimento de faces (capítulo 4), serão citados trabalhos relacionados com as outras abordagens.

2.1.2 Introdução ao Reconhecimento Estatístico

Basicamente, um sistema de reconhecimento estatístico de padrões pode ser composto pelas seguintes partes [Duda and Hart, 1973, Jain et al., 2000] (vide figura 2.1): um sis-

tema de aquisição de dados (por exemplo: sensores ou câmeras); um sistema de pré-processamento, para eliminar ruídos ou distorções; um extrator de características (ou atributos), que cria um vetor de características com dados extraídos dos objetos adquiridos, reduzindo os dados a atributos, propriedades ou características; um seletor de características, que analisa o conjunto de características e elimina as mais redundantes; e um classificador, que analisa um padrão obtido e toma uma certa decisão.

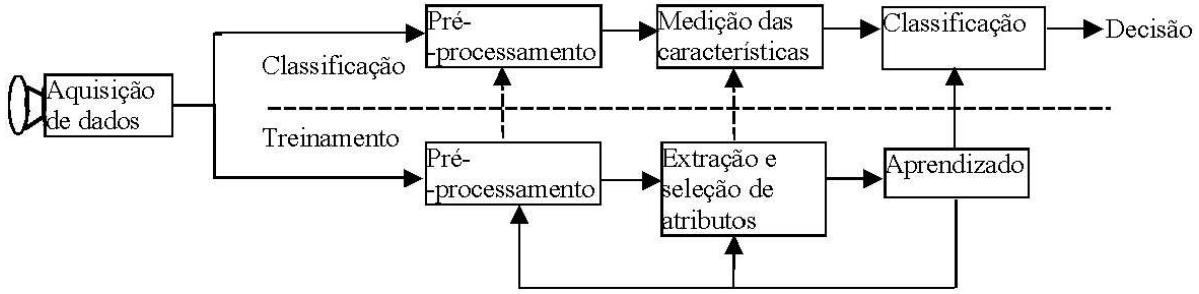

Figura 2.1: Um sistema genérico de reconhecimento de padrões (baseado em [Duda and Hart, 1973] e [Jain et al., 2000]).

O classificador toma decisões baseando-se no aprendizado realizado a partir de um conjunto de treinamento, o qual contém exemplos de padrões de todas as classes existentes no sistema. Conforme será detalhado posteriormente, em reconhecimento estatístico de padrões, a classificação é realizada utilizando estimativas de distribuições probabilísticas, por isso o nome dessa abordagem. O reconhecedor de padrões é avaliado através de um conjunto de testes, preferencialmente composto por padrões de todas as classes, mas que não estejam no conjunto de treinamento. Além do classificador, o pré-processamento, o extrator e o seletor de características podem ser dependentes dos dados de treinamento.

No caso de sistemas estatísticos, quando o problema abordado for muito complexo, torna-se essencial o uso de extração e seleção de características. Exemplos de problemas complexos são aqueles em que há muitas classes ou quando a dimensão dos padrões no formato em que são adquiridos for muito alta.

Na abordagem estatística, cada *padrão* é representado em termos de N *características* (*features*) ou *atributos*. Um padrão é representado por um vetor de características $\mathbf{x} = [x_1, x_2, \cdots, x_N]^t$, modelado como um vetor aleatório, em que cada x_j $(1 \leq j \leq N)$ é uma característica [Theodoridis and Koutroumbas, 1999]. Cada padrão medido $\mathbf{x}_i$ é uma instância de $\mathbf{x}$. O espaço formado pelos vetores de características é chamado de espaco de características, o qual possui dimensão N.

Uma classe ω_i (a i-ésima classe de um conjunto de classes Ω, de c classes) é um conjunto que contém padrões os quais possuem alguma relação ou peculiaridade em comum. Em um exemplo simples de biometria, podemos ter um espaço de características $\mathbf{x}$ em que x_1 representa altura (em cm), x_2 representa peso (em Kg) e x_3 representa o tamanho dos pés (em cm). Nesse mesmo espaço de três dimensões, cada instância de $\mathbf{x}$ representa as

medições tomadas de uma pessoa em um determinado instante. Cada classe representa uma família de pessoas, por exemplo: 'Simpson', 'Jetson', 'Kennedy' e 'Brun'. Nesse caso, o problema de classificação define-se por: dada uma pessoa desconhecida, extrair suas características para obter seu vetor de características $\mathbf{x}$ e determinar a qual família provavelmente essa pessoa pertence.

Os padrões são tratados como vetores aleatórios pois um padrão desconhecido pode ser o representante de uma classe conhecida que sofreu alterações aleatórias proporcionadas por ruídos oriundos do método de aquisição (sensores), da influência de outros fatores externos ou mesmo dos mecanismos de extração de características intrínsecos ao sistema de reconhecimento.

2.2 Métodos de Classificação

2.2.1 Visão Geral

Dado um padrão desconhecido $\mathbf{x}$, pertencente ao conjunto padrões de teste X em um espaço de características, e o conjunto Ω de todas as classes existentes, um classificador é uma função $\Upsilon : X \rightarrow \Omega$, tal que $\Upsilon(\mathbf{x}) = \omega_i$, em que ω_i é uma a i-ésima classe de Ω. Assim, um classificador é uma função que possui como entrada padrões desconhecidos e, como saída, rótulos que identificam a que classe tais padrões provavelmente pertencem (essa definição é válida para todos os classificadores, não só para os estatísticos). Portanto, classificadores são os elementos os quais, de fato, realizam o reconhecimento de padrões. Todos os classificadores devem ser treinados utilizando um conjunto de amostras.

Esse treinamento é utilizado pelo algoritmo do classificador para determinar as *fronteiras de decisão* do espaço de características. *Fronteiras de decisão* são superfícies multidimensionais no espaço de características F que particionam F em c regiões para um problema com c classes, cada região correspondendo a uma classe. Se as regiões S_i e S_j são contíguas, são separadas por uma superfície de decisão. Assim, tem-se $F = \bigcup_{i=1}^{c} S_i$. A regra de decisão faz com que um padrão desconhecido que se encontra na região S_i do espaço de características seja rotulado como um padrão da classe ω_i, ou seja, $\Upsilon(\mathbf{x}) = \omega_i$.

Dessa forma, essencialmente, o que difere um classificador de outro é a forma como esse cria as fronteiras de decisão a partir dos exemplos de treinamento. Os exemplos de treinamento de cada classe podem ser pré-especificados (aprendizado supervisionado) ou aprendidos com base nos exemplos (aprendizado não-supervisionado). No caso de sistemas de reconhecimento de faces, normalmente é realizado aprendizado supervisionado [Chellappa et al., 1995], isto é, as imagens de treinamento possuem um rótulo que identifica de quem é a fotografia. Por esse motivo, não serão descritos métodos não-supervisionados de aprendizado[1].

[1]Detalhes a respeito desse assunto podem ser encontrados em [Jain et al., 1999].

Apesar da existência de vários algoritmos diferentes para determinar fronteiras de decisão (métodos de classificação), pode-se dizer que todos têm em comum os seguintes objetivos:

1. minimizar o erro de classificação;

2. permitir que a classificação seja eficiente computacionalmente.

Porém, a importância de cada um desses objetivos varia de classificador para classificador. Obviamente, o ideal é que um classificador seja rápido e apurado, mas, em problemas complexos, em geral a velocidade do classificador é inversamente proporcional à qualidade dos resultados que ele pode oferecer.

A seguir, há detalhes sobre os métodos de classificação que foram utilizados neste projeto de pesquisa. Detalhes mais específicos sobre outros métodos de classificação se encontram em [Watanabe, 1985, Theodoridis and Koutroumbas, 1999, Duda and Hart, 1973, Backer, 1995].

2.2.2 Classificador Bayesiano

A fim de possibilitar a formalização dos classificadores utilizados neste projeto (K vizinhos mais próximos e mínima distância ao protótipo), inicialmente serão descritos alguns pontos da teoria de decisão e de um classificador Bayesiano. Antes de descrever um classificador Bayesiano, é necessário definir os conceitos a seguir.

Probabilidade a priori de uma classe

Um dado vetor $\mathbf{x}$ pode provir de (ou ser associado a) uma classe i de c classes $\omega_1, \omega_2, \cdots, \omega_c$ com uma probabilidade P_i, chamada de probabilidade a priori da classe i, com $\sum_{i=1}^{c} P_i = 1$.

Função densidade de probabilidade de um padrão

Seja $p(\mathbf{x}|\omega_i)$ a função densidade de probabilidade multivariada de $\mathbf{x}$ quando se sabe que $\mathbf{x}$ pertence à classe ω_i $(1 \leq i \leq c)$, a função densidade de probabilidade local de $\mathbf{x}$ é definida por:

$$p(\mathbf{x}) = \sum_{i=1}^{c} P_i \cdot p(\mathbf{x}|\omega_i) \tag{2.1}$$

Probabilidade a posteriori

Dado um padrão $\mathbf{x}$ com classificação desconhecida, a probabilidade de $\mathbf{x}$ ser da classe ω_j é $P(\omega_j|\mathbf{x})$, que é a probabilidade a posteriori da classe ω_j. Pela regra de Bayes, temos:

$$P(\omega_j|\mathbf{x}) = \frac{p(\mathbf{x}|\omega_j) \cdot P_j}{p(\mathbf{x})}, \tag{2.2}$$

com

$$p(\mathbf{x}) = \sum_{j=1}^{c} p(\mathbf{x}|\omega_j) \cdot P_j \tag{2.3}$$

Taxa de probabilidade de erro

A probabilidade de erro de classificação ao se associar um dado vetor de atributos $\mathbf{x}$ à classe ω_i é definida por:

$$e_i(\mathbf{x}) = 1 - P(\omega_i|\mathbf{x}), i = 1, \cdots, c \tag{2.4}$$

Essa é uma definição geral, sendo válida para regra de decisão arbitrária. O valor esperado dessa probabilidade sobre todos os vetores $\mathbf{x}$ pertencentes à região S_i de decisão para a classe ω_i é a probabilidade de classificação errada em ω_i, denotada ξ_i. Essa é a probabilidade de cometer-se um erro ao atribuir um vetor $\mathbf{x}$ à classe ω_i:

$$\xi_i = \int_{S_i} e_i(\mathbf{x}) \cdot p(\mathbf{x}) d\mathbf{x} = \int_{S_i} [1 - P(\omega_i|\mathbf{x})] \cdot p(\mathbf{x}) d\mathbf{x}, \tag{2.5}$$

em que S_i é a região de aceitação associada à classe ω_i. Como a classificação de um vetor $\mathbf{x}$ só pode ocorrer nas classes mutuamente exclusivas $\omega_1, \cdots, \omega_c$, segue que a probabilidade global de erro, ou taxa de erro, é a soma das probabilidades de erro ξ_i em cada classe:

$$\xi = \sum_{i=1}^{c} \xi_i = \sum_{i=1}^{c} \int_{S_i} [1 - P(\omega_i|\mathbf{x})] \cdot p(\mathbf{x}) d\mathbf{x} \tag{2.6}$$

A expressão entre colchetes é a probabilidade condicional de erro $e_i(\mathbf{x})$; ξ_i é a média dessa probabilidade para todo $\mathbf{x} \in S_i$ e, portanto, é a probabilidade de classificação errada em ω_i.

Infelizmente, na maioria dos casos, o cálculo da probabilidade de erro é extremamente difícil e raramente consegue-se chegar a uma expressão explícita. Na prática, a taxa de erro é geralmente estimada a partir de um conjunto de teste (conjunto de amostras de vetores com classificação conhecida).

Classificador para mínima taxa de erro

A partir da formalização da taxa ou probabilidade de erro, pode-se descrever um classificador que minimiza esse quantificador de desempenho. Inicialmente, é necessário mostrar definições duais às das equações 2.4, 2.5 e 2.6. A probabilidade de acerto ao se classificar um dado $\mathbf{x}$ em ω_i é

$$a_i(\mathbf{x}) = P(\omega_i|\mathbf{x}), i = 1, \cdots, c \tag{2.7}$$

A probabilidade de acerto ao se atribuir um vetor à classe ω_1 é

$$A_i = \int_{S_i} a_i(\mathbf{x}) \cdot p(\mathbf{x}) d\mathbf{x} = \int_{S_i} P(\omega_i|\mathbf{x}) \cdot p(\mathbf{x}) d\mathbf{x} \tag{2.8}$$

A probabilidade de classificação correta ou probabilidade de acerto ou taxa de acerto é

$$A = \sum_{i=1}^{c} \int_{S_i} P(\omega_i|\mathbf{x}) \cdot p(\mathbf{x}) d\mathbf{x} \tag{2.9}$$

Obviamente, a mínima taxa de erro é obtida quando a taxa de acerto é máxima

$$\min \xi \Leftrightarrow \max_{S_i} \sum_{i=1}^{c} \int_{S_i} P(\omega_i|\mathbf{x}) \cdot p(\mathbf{x}) d\mathbf{x} \tag{2.10}$$

A máxima taxa de acerto é obtida quando cada S_i é escolhido como o domínio onde $P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x}), \forall j$.

Assim, o *classificador Bayesiano de mínima taxa de erro* pode ser definido como:

$$\Upsilon(\mathbf{x}) = \omega_i \ \ se \ \ \mathbf{x} \in S_i, \tag{2.11}$$

$$com \ \ S_i = \{\forall \mathbf{x} \in F \ \ tal \ \ que \ \ P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x}), \ \ j = 1, \cdots, c\} \tag{2.12}$$

ou, simplesmente,

$$\Upsilon(\mathbf{x}) = \omega_i \ \ se \ \ P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x}), \ \ j = 1, \cdots, c \tag{2.13}$$

Após essa descrição do classificador de Bayes de mínima taxa de erro, a seguinte questão ingênua pode surgir: se o classificador Bayesiano é um classificador ótimo, então por que outros classificadores são utilizados? O motivo é que o classificador de Bayes só pode ser executado se a probabilidade a priori P_i e a função densidade de probabilidade $p(\mathbf{x}|\omega_i)$ forem conhecidas, o que geralmente não ocorre. Em problemas práticos, na fase de treinamento são utilizados métodos de estimação dessas probabilidades. Entretanto, quando a distribuição das classes possui formas "complicadas" e descontínuas, o preço computacional desses métodos torna-se muito alto quando se deseja obter uma representação precisa dessas probabilidades.

Uma abordagem para se resolver esse problema é assumir um modelo para $p(\mathbf{x}|\omega_i)$. A estimativa de distribuição mais bem conhecida e, provavelmente, uma das mais simples, é a de distribuição normal. Nesse caso, assume-se que:

$$p(\mathbf{x}|\omega_i) = \frac{1}{(2\pi)^{N/2} \cdot \sqrt{det(\Sigma_i)}} exp(-\frac{1}{2}(\mathbf{x} - \mu_i)^t \cdot \Sigma_i^{-1} \cdot (\mathbf{x} - \mu_i)), i = 1, \cdots, c \quad (2.14)$$

em que $\mu_i = E[\mathbf{x}]$ é o valor esperado (tomado pela média) da classe ω_i, e Σ_i é a matriz de covariância $N \times N$ definida por:

$$\Sigma_i = E[(\mathbf{x} - \mu_i) \cdot (\mathbf{x} - \mu_i)^t] \quad (2.15)$$

$det(\Sigma_i)$ denota o determinante de Σ_i e $E[\cdot]$ a média (ou esperança) de uma variável aleatória. É comum o uso do símbolo $\mathcal{N}(\mu, \Sigma)$ para denotar a função de densidade propabilística Gaussiana.

A partir dessas definições e das anteriores, contrói-se o classificador Bayesiano para distribuições normais.

2.2.3 Regra dos K vizinhos mais próximos

A regra de classificação dos K vizinhos mais próximos é um método de classificação que não possui processamento na fase de treinamento, pois não é necessário estimar as distribuições de probabilidades das classes. Entretanto, é necessário um grande número de padrões de treinamento (padrões cuja classe é conhecida a priori), pois pode-se dizer que as tarefas de estimativa e de classificação são fundidas em uma única tarefa. O classificador dos K vizinhos mais próximos (KNN) é um classificador sub-ótimo que cria fronteiras de decisão complexas.

Dado um padrão de teste (desconhecido) $\mathbf{x}$, sua classificação é realizada da seguinte maneira:

- Inicialmente, calcula-se a distância entre $\mathbf{x}$ e todos os padrões de treinamento;
- Verifica-se a quais classes pertencem os K padrões mais próximos;
- A classificação é feita associando-se o padrão de teste à classe que for mais freqüente entre os K padrões mais próximos de $\mathbf{x}$.

Há duas distâncias que normalmente são adotadas para implementar esse classificador:

distância Euclidiana

A distância Euclidiana entre dois vetores ($\mathbf{x}_i$ e $\mathbf{x}_j$) é definida por:

$$d_{\mathcal{E}}(\mathbf{x}_i, \mathbf{x}_j) = ||\mathbf{x}_i - \mathbf{x}_j|| = \sqrt{(\mathbf{x}_i - \mathbf{x}_j)^t \cdot (\mathbf{x}_i - \mathbf{x}_j)} \quad (2.16)$$

distância de Mahalanobis

A distância de Mahalanobis entre um padrão $\mathbf{x}$ e o protótipo σ de uma classe é definida por:

$$d_{\mathcal{M}}(\mathbf{x}, \sigma) = \sqrt{(\mathbf{x} - \sigma)^t \cdot \Sigma^{-1} \cdot (\mathbf{x} - \sigma)}, \tag{2.17}$$

em que Σ é a matriz de covariância dos padrões da classe de σ. [2]

Tomando-se $K = 1$ no classificador de K vizinhos mais próximos, obtém-se o classificador de vizinho mais próximo (1NN). Esse classificador é muito comum em aplicações de reconhecimento de faces após a extração de características usando PCA. Normalmente, a regra de classificação por vizinho mais próximo acarreta numa taxa de erro maior do que a da regra de decisão de Bayes. Porém existe um teorema que diz que, supondo-se que haja infinitos de padrões de treinamento, a taxa de erro com esse classificador não ultrapassa (sendo em geral menor que) o dobro da taxa de erro com o classificador de Bayes (ver demonstração [Kohn, 1998] e [Theodoridis and Koutroumbas, 1999]).

O classificador KNN pode ser descrito formalmente utilizando o classificador de Bayes com mínima taxa de erro. A desigualdade contida na equação 2.13 equivale a $P(\omega_i|\mathbf{x}) \geq P(\omega_j|\mathbf{x})$, contando que $p(\mathbf{x}) \neq 0$. Para estimar P_i a partir dos dados, basta tomar $|T_i|/|T|$, em que $|T|$ é o número total de amostras e $|T_i|$ é o número de amostras na classe ω_i. Para se estimar $p(\mathbf{x}|\omega_i)$, pode-se tomar um volume $B_{\mathbf{x}}$, centrado em $\mathbf{x}$ e contar-se quantas amostras há em seu interior. Dessa forma, a regra de decisão de Bayes fica:

$$decidir \quad \omega_i \quad se \quad \frac{|T_i|}{|T|} \cdot \frac{K_i}{|T_i| \cdot B_{\mathbf{x}}} \geq \frac{|T_j|}{|T|} \cdot \frac{K_j}{|T_j| \cdot B_{\mathbf{x}}} \quad j = 1, \cdots, c \tag{2.18}$$

em que se supõe que volume $B_{\mathbf{x}}$ abarca exatamente K amostras indistintamente das classes envolvidas, com $K = \sum_{i=1}^{c} K_i$. Simplificando,

$$decidir \quad \omega_i \quad se \quad \frac{K_i}{|T| \cdot B_{\mathbf{x}}} \geq \frac{K_j}{|T| \cdot B_{\mathbf{x}}} \quad j = 1, \cdots, c \tag{2.19}$$

A principal vantagem desse método é que ele cria uma superfície de decisão que se adapta à forma de distribuição dos dados de treinamento de maneira detalhada, possibilitando a obtenção de boas taxas de acerto quando o conjunto de treinamento é grande ou representativo. O objetivo de se utilizar $K > 1$ é reduzir a ocorrência de erros causados por ruídos nos padrões de treinamento. Por exemplo, um padrão de treinamento $\mathbf{x}_r$ da classe ω_i que se encontra em uma região do espaço de características povoada por padrões de treinamento da classe ω_j devido à ação de ruídos não prejudicará o desempenho do classificador, pois a verificação de seus vizinhos fará com que um padrão de teste que se localize próximo a $\mathbf{x}_r$ seja classificado como um padrão da classe ω_j. Porém, o uso de valores grandes em K pode reduzir a qualidade dos resultados de classificação quando a distribuição das classes possui muitas sobreposições.

[2] Na página 35 há um exemplo de matriz de covariância.

Assim, deve-se ter preferência ao classificador KNN sobre o 1NN quando se dispõe de um conjunto de treinamento T com muitos exemplos e quando esse conjunto contiver amostras com classificação errada.

Por essas razões, a escolha do número de vizinhos a serem utilizados (K) torna-se um ponto crítico do classificador KNN. Não há uma estratégia definitiva para realizar essa escolha para um caso prático, sendo recomendada a estratégia de tentativa e erro. Porém, pesquisas recentes [Theodoridis and Koutroumbas, 1999] sugerem que, para $K \rightarrow \infty$, quando $|T| \rightarrow \infty$, o desempenho do classificador KNN tende a ser ótimo. Entretanto, para conjunto de treinamento numerosos, é esperado que o classificador 3NN (KNN para K=3) permita a obtenção de um desempenho muito próximo do classificador Bayesiano. Um fato óbvio é que a escolha de $K > 1$ (principalmente $1 < K \leq c$, sendo c o número de classes) pode causar problemas de indecisão quando ocorrem empates, ou seja, quando o número de vizinhos mais próximos pertencente a classes diferentes é igual.

A principal desvantagem dos classificadores K-NN está em sua complexidade na fase de testes. Isso deve-se ao fato de que, caso seja feita uma busca em "força-bruta" (sem ordenação) pelos vizinhos mais próximos, para cada padrão de teste é necessário realizar $K \cdot |T|$ medições de distância, ou seja, a quantidade de operações necessárias é da ordem de $K \cdot O(|T|)$, sendo que $O(n)$ denota a ordem de n cálculos [Theodoridis and Koutroumbas, 1999, Cormen et al., 1990].

2.2.4 Mínima Distância ao(s) Protótipo(s)

O classificador de distância ao protótipo é bastante simples em termos de esforço computacional, tanto na fase de treinamento quanto na de teste. Essa característica deve-se à simplicidade de seu algoritmo.

A fase de treinamento consiste na determinação dos protótipos, no mínimo um para cada classe. Os protótipos são vetores no espaço de característica que usualmente são criados a partir de informações obtidas do conjunto de treinamento ou da distribuição probabilística das classes. Um exemplo um tanto comum de protótipo utilizado é a média (baricentro) do conjunto de treinamento das classes.

Na fase de teste, cada padrão é classificado de acordo com o protótipo mais próximo. Normalmente utiliza-se a distância Euclidiana para calcular a proximidade entre os padrões e os protótipos. Nota-se que a regra de decisão é bastante simples. Se os protótipos forem vistos como padrões de treinamento, é praticamente trivial mostrar que essa regra se equivale à do classificador KNN, para $K = 1$.

Também é fácil notar que há um caso em que o classificador de distância ao protótipo se eqüivale a um classificador Bayesiano. Isso ocorre quando é utilizado apenas um protótipo por classe, sendo cada protótipo definido pelo baricentro do conjunto de treinamento de sua classe (μ_i, onde i identifica a classe). Nesse caso, esse classificador é equivalente

ao classificador Bayesiano para distribuições normais $\mathcal{N}(\mu, \Sigma)$, caso seja assumido que todas as classes possuem distribuições probabilísticas com a mesma matriz de covariância Σ, sendo Σ uma matriz diagonal. Em mais detalhes, essa equivalência ocorre quando a distribuição probabilística das classes é tal que o desvio padrão σ é uniforme para todas as direções do espaço de característica, de forma que $\Sigma = \sigma^2 I$. Graficamente, pode-se ilustrar como distribuições circulares, sendo que esses "círculos" são centrados no baricentro da distribuição de cada classe, e todos os círculos possuem o mesmo raio.

Portanto, nesses casos, mesmo sendo muito simples, esse classificador comporta-se como um classificador ótimo. É importante ressalvar que, quando for usada a distância de Mahalanobis (equação 2.17), não existem restrições quanto à mariz de covariância das classes para que o classificador de mínima distância ao protótipo seja equivalente ao de Bayes para distribuições normais.

Uma fronteira de decisão construída por esse classificador (adotando-se a distância Euclidiana, com um protótipo por classe) é um hiperplano perpendicular ao segmento de reta que une dois protótipos. Esse hiperplano intercepta a mediatriz desse segmento, definindo o lugar geométrico dos pontos eqüidistantes a esses dois protótipos. Dessa forma, pode-se mostrar que o conjunto de todas as fronteiras de decisão gerado pela regra de decisão de mínima distância ao protótipo eqüivale a um diagrama de Voronoi na dimensão N com os sítios na posição dos protótipos [Theodoridis and Koutroumbas, 1999] (detalhes sobre esses diagramas podem ser encontrados em [de Berg et al., 2000]). Como é de se esperar, o mesmo pode ser clamado a respeito do classificador 1NN, com a diferença que, quando dois padrões da mesma classe são vizinhos, não existe uma fronteira de decisão entre eles.

Com relação ao custo computacional desse classificador, para cada padrão de teste, é necessário realizar apenas $c - 1$ comparações (O(c) cálculos para cada padrão), sendo c o número de classes existentes, o que é o principal ponto positivo dessa abordagem. A desvantagem dessa abordagem é a qualidade dos resultados em casos práticos, pois os protótipos freqüentemente não contêm informações suficientes sobre a forma da distribuição das classes, já que os casos semelhantes ao descrito anteriormente não são freqüentes.

2.3 Problemas de generalização

Nesta seção, são discutidos os problemas de generalização de classificadores. Tais problemas são muito relevantes no projeto de sistemas de reconhecimento estatístico de padrões, que também podem ser comuns a sistemas não estatísticos, como redes neurais.

Não importando qual o classificador utilizado, em problemas práticos, ele deve ser treinado usando exemplos de treinamento para estimar a distribuição das classes. Como resultado, o desempenho do classificador depende tanto do número de exemplos de treina-

mento como dos valores específicos das instâncias, ou seja, da qualidade desses exemplos. Ao mesmo tempo, o objetivo do projeto de um sistema de reconhecimento é classificar futuros exemplos de teste mesmo que esses não sejam os mesmos que os de treinamento.

Porém, a otimização de um classificador para maximizar seu desempenho no conjunto de treinamento nem sempre produz um bom resultado para o conjunto de testes. A habilidade de **generalização** de classificadores refere-se a seu desempenho ao classificar padrões de teste que não foram utilizados durante o treinamento.

Os problemas de generalização ocorrem quando um classificador se especializa demais em seus padrões de treinamento, ou quando utiliza mais informações (características) que as necessárias. Basicamente, há três problemas oriundos da redução na capacidade de generalização de um classificador [Jain et al., 2000]:

- sobre-ajuste (*overfitting*), relacionado com o número de parâmetros livres do classificador;
- sobre-treinamento (*overtraining*), relacionado com o número de iterações de treinamento;
- problema da dimensionalidade (*curse of dimensionality*), relacionado com a dimensão do espaço de características.

Assim, o desempenho de um classificador depende da relação entre sua complexidade, a qualidade do conjunto de treinamento (o quanto ele representa a distribuição dos dados) e o número de características utilizadas. A taxa de erro dos classificadores apresentam um comportamento de curva em U com a variação de dos fatores relacionados com esses problemas. A seguir encontram-se mais detalhes sobre o problema da dimensionalidade, pois esse afeta todos os sistemas de reconhecimento de padrão estatístico e também por causa da sua relação com seleção de características.

O Problema da Dimensionalidade

O problema da dimensionalidade, também conhecido como *curse of dimensionality* e como comportamento de curva em U, é um fator muito relevante para decidir-se a dimensionalidade ideal a ser adotada em um problema de reconhecimento de padrões. Trata-se do seguinte fenômeno: o número de elementos de treinamento requeridos para que um classificador tenha um bom desempenho é uma função monotonicamente crescente da dimensão do espaço de características. Em alguns casos (mas não necessariamente em todos), pode-se mostrar que essa função é exponencial, ou seja, $|T| \Leftrightarrow O(e^N)$ [Jain et al., 2000]. Um exemplo é o da técnica de particionamento do espaço de características para classificação baseada em árvores de decisão. Nessa técnica, cada reta suporte dos vetores da base do espaço de características é segmentada em intervalos regulares. A interseção entre

esses intervalos forma células no espaço. O reconhecimento de padrões é feito através da associação de uma classe a cada célula, de acordo com a classe majoritária nas células. Esse é um exemplo de sistema de classificação em que é bastante intuitivo verificar que, para que não hajam células com classificação indefinida, é necessário que o número de elementos de treinamento seja uma função exponencial da dimensão do espaço de características. Isso ocorre devido ao fato de que, em reconhecimento estatístico de padrões, o volume do espaço de característica cresce exponencialmente com a dimensionalidade [Perlovsky, 1998]. Esse fenômeno é bem conhecido pela comunidade de reconhecimento de padrões (ver também [Jain et al., 2000] para um exemplo mais formal).

Quando é utilizado um classificador Bayesiano, nos casos em que o número de elementos de treinamento é arbitrariamente grande ou a função densidade de probabilidade das classes $(p(\mathbf{x}|\omega_i), i = 1, \cdots, c)$ for completamente conhecida, a probabilidade de erro de classificação de uma regra de decisão não aumenta com o número de características consideradas. Porém, nos problemas práticos, para um conjunto de treinamento finito, observa-se que a adição de características pode prejudicar o desempenho de um classificador (se não forem adicionados exemplos de treinamento). Isso ocorre quando o número de exemplos de treinamento não é grande o suficiente em relação ao número de características. Esse fenômeno, chamado fenômeno do pico (*peaking phenomena*), é uma conseqüência do problema da dimensionalidade, tendo também sido amplamente estudado (por exemplo, [Campos et al., 2000d, Belhumeur et al., 1997]). Todos os classificadores comumente utilizados podem sofrer de problema da dimensionalidade.

Apesar de ser teoricamente clara a relação entre a dimensionalidade e o tamanho do conjunto de treinamento ($|T| \Leftrightarrow e^N$), há outros fatores que, quando considerados, ofuscam a exatidão dessa relação, tais como a complexidade do classificador e o número de classes. Segundo [Jain et al., 2000], resultados empíricos fazem com que, geralmente, seja aceita a seguinte relação: $|T_i| \Leftrightarrow 10 \cdot N$, $i = 1, \cdots, c$, sendo $|T_i|$ o número de exemplos de treinamento da classe i. Ou seja, no mínimo deve-se utilizar um número de exemplos de treinamento por classe dez vezes maior que a dimensionalidade.

Para mostrar que o problema da dimensionalidade não depende exclusivamente do número de padrões utilizados no processo de treinamento, criamos a figura 2.2. Nesta figura há um problema de classificação com duas classes cujas distribuições estão mostradas através das formas que circundam as letras que identificam tais classes. Esse espaço de características possui dimensão 2 e os vetores de sua base estão indicados por F_1 e F_2. Supomos que a distribuição dessas classes faz com que o protótipo de cada classe fique nas posições indicadas por p_A e p_B. Assim, caso seja utilizado um classificador de mínima distância ao protótipo, a fronteira de decisão criada divide o espaço de características no lugar geométrico indicado pela linha tracejada. Podemos notar que a taxa de erro desse classificador não será pequena. Por outro lado, se for utilizada somente a característica F_1, podemos notar que a projeção dos padrões e do protótipo nessa característica fará com que a taxa de erro seja praticamente nula, pois a fronteira de decisão será o ponto 0. Esse problema ocorre mesmo que sejam utilizados conjuntos de treinamento grandes, pois ele

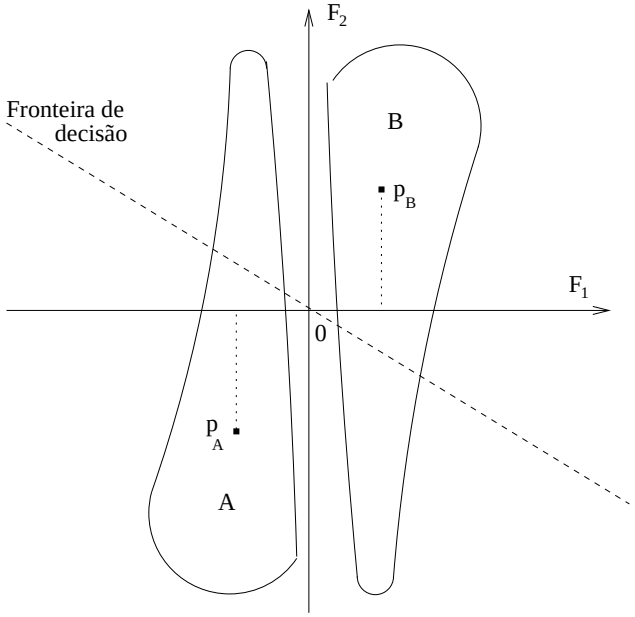

Figura 2.2: Exemplo de problema em que o uso de uma dimensão é melhor que o uso de duas.

decorre de uma deficiência do classificador, e não do número de padrões de treinamento. Essa deficiência decorre do fato de que o classificador de mínima distância ao protótipo com distância Euclidiana não estima a fronteira de decisão com precisão quando a distribuição das classes não é circular.

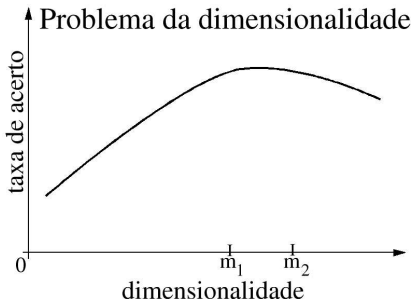

Figura 2.3: Efeito do problema da dimensionalidade.

A curva apresentada a figura 2.3, a qual ilustra o problema da dimensionalidade, apresenta três regiões no eixo da dimensionalidade com significados diferentes:

1. Na primeira região, compreendida entre 0 e m_1, ocorre o comportamento mais esperado intuitivamente, pois a adição de características promove redução na taxa de erro. Isso deve-se ao fato de espaços com dimensões muito pequenas não possuírem informações suficientes para distinguir-se as classes de padrões. Com isso, a adição de novas características melhora os resultados de classificação.

2. A segunda região é aquela em que é atingida uma estabilidade na taxa de acerto. Nessa região, a adição ou eliminação de características não altera (ou altera muito sutilmente) essa taxa. Para um problema de classificação, a melhor solução está na adoção da dimensionalidade m_1, pois esse é o menor valor em que a taxa de acerto é máxima. A estabilização na taxa de acerto se deve ao fato de que as características importantes para se distinguir os padrões já foram inseridas na região anterior, e as características extras não são nem ruidosas nem relevantes para a classificação.

3. A última região é a região em que de fato ocorre o *problema da dimensionalidade*. Note que o aumento no número de características provoca aumento na taxa de erro.

Assim, para obter-se o desempenho máximo de um classificador, é necessário investigar qual é a dimensionalidade ideal para um determinado problema de reconhecimento de padrões. Para isso, pode ser aplicada uma estratégia simples de tentativa e erro em relação à dimensionalidade, usando um método de redução de dimensionalidade (incluindo extração e seleção de características) até que o ponto de máximo desempenho de um classificador seja atingido. Nessa estratégia, são realizados testes de redução de dimensionalidade para a obtenção de sub-espaços de características de vários tamanhos diferentes, até que seja obtida a dimensionalidade que minimiza o erro de classificação. O próximo capítulo apresenta mais detalhes sobre métodos de redução de dimensionalidade. Outros detalhes sobre os problemas de generalização podem ser encontrados em [Theodoridis and Koutroumbas, 1999, Jain et al., 2000].

Capítulo 3

Redução de *dimensionalidade*

3.1 Visão Geral

O termo *dimensionalidade* é atribuído ao número de características de uma representação de padrões, ou seja, a dimensão do espaço de características (N). As duas principais razões para que a dimensionalidade seja a menor possível são: custo de medição e precisão do classificador. Quando o espaço de características contém somente as características mais salientes, o classificador será mais rápido e ocupará menos memória [Jain et al., 2000]. Além disso, conforme discutido na seção 2.3, quando o conjunto de exemplos de treinamento não é muito grande, o problema da dimensionalidade pode ser evitado usando-se um espaço de características pequeno. Isso também propicia a obtenção de menores taxas de erro de classificação.

Em visão computacional, a necessidade redução de dimensionalidade é acentuada, pois a dimensionalidade de imagens é muito grande. O espaço de imagens possui características que podem ser eliminadas para efetuar o reconhecimento de objetos. Uma imagem de largura w e altura h (em *pixels*) pode ser vista como um padrão no espaço de imagens, o qual possui dimensionalidade $N = h \times w$ (vide seção 3.2.2). Esse pode ser um valor muitíssimo elevado em imagens obtidas por *scanners* ou câmeras. Além disso, qualquer alteração em translação, rotação, escala, etc. dos objetos contidos nessa imagens fará com que ocorra grandes erros de classificação. Por isso, é necessária a utilização de algoritmos de redução de dimensionalidade que propiciem a obtenção de representações dos padrões (obtidos das imagens) de forma robusta a essas alterações.

Além da necessidade de utilizar a menor dimensionalidade possível, há outro fator analisado pelo teorema do "patinho feito" [Watanabe, 1985], que diz ser possível fazer dois padrões arbitrários ficarem similares se esses forem codificados com um número suficientemente grande de características similares. Isso enfatiza a necessidade de uma escolha cuidadosa de características.

Para efetuar redução de dimensionalidade, existem basicamente duas abordagens: extração de características e seleção de características. Em linhas gerais, os algoritmos de extração criam novas características a partir de transformações ou combinações do conjunto de características original. Já os algoritmos de seleção, como o próprio nome diz, selecionam, segundo determinado critério, o melhor subconjunto do conjunto de características original.

Freqüentemente, a extração de características precede a seleção, de forma que, inicialmente, é feita a extração de características a partir dos dados de entrada, seguido por um algoritmo de seleção de características que elimina os atributos mais irrelevantes segundo um determinado critério, reduzindo a dimensionalidade.

A escolha entre seleção e extração de características depende do domínio de aplicação e do conjunto específico de dados de treinamento disponíveis. Em geral, a seleção de características reduz o custo de medição de dados, e as características selecionadas mantêm sua interpretação física original, mantendo as propriedades que possuíam quando foram criadas. Já as características transformadas geradas por extração podem prover uma habilidade de discriminação melhor que o melhor subconjunto das características originais. Entretanto, as novas características (combinações lineares ou não lineares das características originais) podem não possuir um significado físico.

É importante lembrar que, se a redução de dimensionalidade for excessiva, o classificador pode ter seu poder de discriminação reduzido (vide o problema da dimensionalidade na seção 2.3). Por isso, é importante analisar a variação do comportamento do classificador com o número de características, de forma que seja possível estimar a dimensionalidade ideal para determinado classificador e conjunto de dados. A seguir, encontram-se maiores detalhes sobre a extração e a seleção de atributos.

3.2 Extração de características

Um método de extração de características cria um novo espaço a partir de transformações ou combinações das características do espaço original. Formalmente, dado um espaço de características $\mathcal{I}$ de dimensão N, um método de extração de características $\mathcal{H}$ é uma função $\mathcal{H} : \mathcal{I} \rightarrow F$, em que F possui dimensão m. Assim, dado um padrão $\mathbf{x}$ em um espaço de características $\mathcal{I}$, temos

$$\mathcal{H}(\mathbf{x}) = \mathbf{y}, \tag{3.1}$$

tal que $\mathbf{y}$ ($\mathbf{y} \in F$) é a nova representação do padrão no espaço F.

Normalmente, $m \ll N$, mas nem sempre a redução de dimensionalidade é promovida diretamente pelos métodos de extração de características. Em geral, eles criam um novo espaço de característica em que a determinação dos vetores mais salientes de sua base é muito simples. Por exemplo, conforme será visto posteriormente, a transformada de Karhunen-Loève pode, nos piores casos, criar um espaço de características com $m = N$. Entretanto, geralmente basta selecionar os primeiros vetores da base criada para reduzir a dimensionalidade de forma eficiente.

Há métodos lineares e não lineares de extração de características. Os processos lineares de extração de características podem ser definidos como uma simples mudança de base do espaço vetorial de características da seguinte forma:

$$\mathbf{y} = H^t \cdot \mathbf{x}, \tag{3.2}$$

em que H é uma matriz mudança de base que leva elementos da base $\mathcal{I}$ para a base F ([Callioli et al., 1998]).

Dentre os extratores de características lineares, podemos citar a transformada de Fourier, a análise de componentes principais (PCA), a análise de discriminantes lineares e outras projeções lineares em geral. Em relação aos extratores não lineares, pode-se citar as redes neurais e os heurísticos. A seguir, estão descritos os métodos de extração de características que foram utilizados no desenvolvimento desta pesquisa.

3.2.1 Transformada de Fourier

A transformada de Fourier é uma ferramenta muito importante em processamento de imagens. Dentre as principais aplicações da transformada de Fourier, encontram-se análise, filtragem, reconstrução e compressão de imagens, bem como reconhecimento de padrões e de objetos. Nesta seção, será focalizada a aplicação ao reconhecimento de padrões como método de redução de dimensionalidade. Detalhes sobre outras aplicações da transformada de Fourier podem ser encontrados nas seguintes referências: [Castleman, 1996, Gonzalez and Woods, 1992, Cesar-Jr, 1997].

Conceitos Básicos

Através da transformada de Fourier, pode-se decompor um sinal em seus componentes de freqüência (senos e cossenos), de forma que um coeficiente de Fourier reflete a importância de determinada freqüência para o sinal. Em sinais discretos, pode ser feita a redução de dimensionalidade (extração de características) através dos descritores de Fourier. Para expor essa técnica, inicialmente serão definidos alguns conceitos básicos.

Dado um sinal contínuo e unidimensional $x(t)$, sua transformada de Fourier é definida

por:

$$y(f) = F(x(t)) = \int_{-\infty}^{\infty} x(t) \cdot e^{-i2\pi ft} dt, \tag{3.3}$$

em que f denota freqüência, ou seja, a variável básica do domínio de Fourier, e t denota tempo. A inversa da transformada de Fourier é definida por:

$$x(t) = F^{-1}(y(f)) = \int_{-\infty}^{\infty} y(f) \cdot e^{i2\pi ft} dt, \tag{3.4}$$

Uma condição suficiente para a existência da transformada de Fourier de um sinal é que ele seja integrável, ou seja,

$$\int_{-\infty}^{\infty} |x(t)| dt < \infty \tag{3.5}$$

A série de Fourier pode ser vista como um caso especial da transformada de Fourier. Dessa forma, uma função periódica $x(t)$, de período $\mathcal{T}_0$, pode ser expressa pela seguinte série de Fourier:

$$x(t) = \sum_{s=-\infty}^{\infty} \alpha_s e^{i2\pi s f_0 t}, \tag{3.6}$$

em que α_s são os coeficientes (complexos) da série e $f_0 = 1/\mathcal{T}_0$ é a freqüência fundamental. Esses coeficientes podem ser definidos como:

$$\alpha_s = \frac{1}{\mathcal{T}_0} \int_{-\mathcal{T}_0/2}^{\mathcal{T}_0/2} x(t) e^{-i2\pi s f_0 t} dt, s = 0, \pm 1, \pm 2, \cdots \tag{3.7}$$

Por isso, pode-se associar a série de Fourier à transformada de Fourier através de uma discretização do domínio da freqüência, em função da periodicidade do sinal $x(\cdot)$ [Cesar-Jr, 1997].

Dessa forma, a partir da transformada contínua de Fourier, pode-se definir a sua versão discreta. Essa transformada determina os descritores de Fourier. Seja $x(n)$ um sinal discreto definido por uma cadeia de tamanho N ($n = 0, 1, \cdots, N-1$), assumindo-se que x é um sinal periódico e que a cadeia $x(n)$ contém um período desse sinal, a transformada discreta de Fourier desse sinal se dá por:

$$y(s) = \sum_{n=0}^{N-1} x(n) e^{-i2\pi ns/N}, s = 0, 1, \cdots, N-1 \tag{3.8}$$

Os coeficientes de $y(s)$ são os descritores de Fourier de $x(n)$. Com esses coeficientes, pode-se obter uma reconstrução perfeita do sinal $x(n)$ utilizando a tranformada inversa de Fourier discreta:

$$x(n) = \frac{1}{N} \sum_{s=0}^{N-1} y(s) e^{i2\pi ns/N}, n = 0, 1, \cdots, N-1 \tag{3.9}$$

Devido ao fato de imagens serem padrões originariamente descritos por matrizes, é importante mencionar que a transformada de Fourier pode ser generalizada de forma a poder ser aplicada em sinais bidimensionais. Detalhes a respeito desse assunto podem ser encontrados em [Gonzalez and Woods, 1992].

Transformada Rápida de Fourier

Pelas equações 3.8 e 3.9, pode-se notar que a transformada discreta de Fourier, bem como sua inversa, são um tanto caras computacionalmente. De fato, a transformada discreta de Fourier possui uma complexidade de tempo quadrática O(N^2), sendo N o tamanho do sinal ou o número total de pixels em uma imagem.

Porém, quando o tamanho do sinal x é uma potência de 2, pode-se aplicar um algoritmo chamado Transformada Rápida de Fourier ("Fast Fourier Transform" - FFT), baseado em um método chamado dobramentos sucessivos [Gonzalez and Woods, 1992]. A ordem de complexidade de execução desse algoritmo é de O($Nlog_2N$), sendo, portanto, altamente eficiente se comparado à transformada de Fourier discreta comum.

Esse algoritmo torna possível a aplicação da transformada de Fourier para imagens ou padrões de alta dimensionalidade. Para se fazer uma comparação prática, foi realizado um teste utilizando o *software* MatLab, que possui a FFT implementada. Foram criados dois sinais aleatórios (ou seja, contendo apenas ruídos) x_1 e x_2. O sinal x_1 possui tamanho 2^{23} e x_2 possui uma dimensão a menos que x_1, ou seja, o tamanho de x_2 é $2^{23} - 1$. O tempo levado para obter-se a transformada de Fourier de x_1 foi de 10,8198 segundos. Já o tempo levado para realizar a mesma tarefa em x_2 (o qual é menor que x_1, mas cujo tamanho não é uma potência de 2) foi de 128,8102 segundos.

Propriedades

A transformada de Fourier possui várias propriedades interessantes para o reconhecimento de padrões e para processamento de imagens. Dentre as propriedades mais importantes da transformada unidimensional, pode-se citar as seguintes:

- Linearidade: $a \cdot x_1(t) + b \cdot x_2(t) \Leftrightarrow a \cdot y_1(f) + b \cdot y_2(f)$, sendo a e b constantes, x_1 e x_2 dois sinais, e y_1 e y_2 suas transformadas de Fourier;
- Teorema da similaridade: $x(a \cdot t) \Leftrightarrow \frac{1}{|a|} \cdot y \cdot (\frac{f}{a})$;
- Teorema da Translação: $x(t - a) \Leftrightarrow e^{-i2\pi af} y(f)$;
- Teorema da Convolução: $x_1(t) * x_2(t) \Leftrightarrow y_1(f) \cdot y_2(f)$.
- Diferenciação: $\frac{d}{dt}x(t) \Leftrightarrow i2\pi f \cdot y(f)$

Redução de Dimensionalidade usando a Transformada de Fourier Discreta

Conhecendo-se os conceitos básicos da transformada discreta de Fourier, pode-se ilustrar como é realizada a redução de dimensionalidade através dela. Suponha que, na equação 3.9, ao invés de se utilizar todos os $y(s)$ coeficientes, sejam utilizados apenas m coeficientes para reconstruir-se o sinal. Isso é equivalente a fazer $y(s) = 0$ para todo $s > m-1$ naquela equação, resultando na seguinte aproximação para $x(n)$:

$$\overline{x}(n) = \frac{1}{N} \sum_{s=0}^{m-1} y(s) e^{i2\pi ns/N}, n = 0, 1, \cdots, N-1 \tag{3.10}$$

Apesar de serem usados apenas m descritores para obter cada componente de $\overline{x}(n)$, n ainda varia de 0 a $N-1$. Isto é, a aproximação do sinal possui o mesmo tamanho que o sinal original. Os primeiros coeficientes de Fourier referem-se às freqüências mais baixas do sinal, que geralmente contêm informações mais globais dos padões comumente encontrados em problemas de visão. Já os últimos referem-se às freqüências mais altas do sinal, as quais são geralmente associadas a informações mais detalhadas ou finas dos padrões ou são causadas por ruídos [Gonzalez and Woods, 1992].

Por isso, pode-se reduzir a dimensionalidade desses padrões (imagens) utilizando apenas seus m primeiros descritores de Fourier. Assim, as imagens reconstruídas a partir desses descritores apresentam borramentos e redução dos detalhes das bordas, mas as informações mais importantes para caracterizar os objetos contidos nas imagens não são perdidas. Portanto, pode-se efetuar reconhecimento de objetos em imagens utilizando-se padrões m-dimensionais, constituidos pelos m primeiros descritores de Fourier das imagens. Dessa forma, para efetuar classificação utilizando padrões com dimensionalidade menor, pode-se representá-los por $\overline{y}$, tal que:

$$\overline{y}(f) = y(f), f = 0, 1, 2, \cdots, m-1 \quad para \quad m < N \tag{3.11}$$

tomando-se o cuidado de definir $\overline{y}(f) = 0$ para todo $f \geq m$ caso seja realizada a reconstrução do padrão.

Na figura 3.1, há um exemplo que ilustra os efeitos da redução da dimensionalidade na reconstrução de um sinal. Foram criados dois sinais aleatórios discretos de tamanho 50 (x_1 e x_2). Posteriormente, foi calculada a transformada de Fourier desses sinais e, com apenas os 25 primeiros coeficientes, foi realizada a reconstrução desses sinais. Pode-se notar que os sinais reconstruídos são uma versão "suavizada" dos sinais originais. Também é possível verificar que, apesar de terem sido utilizados, no processo de reconstrução, metade dos descritores de Fourier disponíveis, os sinais reconstruídos preservaram informações importantes dos originais. Dessa forma, com 25 coeficientes é visualmente possível distingüir qual reconstrução se refere ao sinal x_1 e qual se refere ao sinal x_2. Com isso, fica ilustrado como é possível efetuar uma classificação de padrões de dimensionalidade reduzida através da transformada de Fourier.

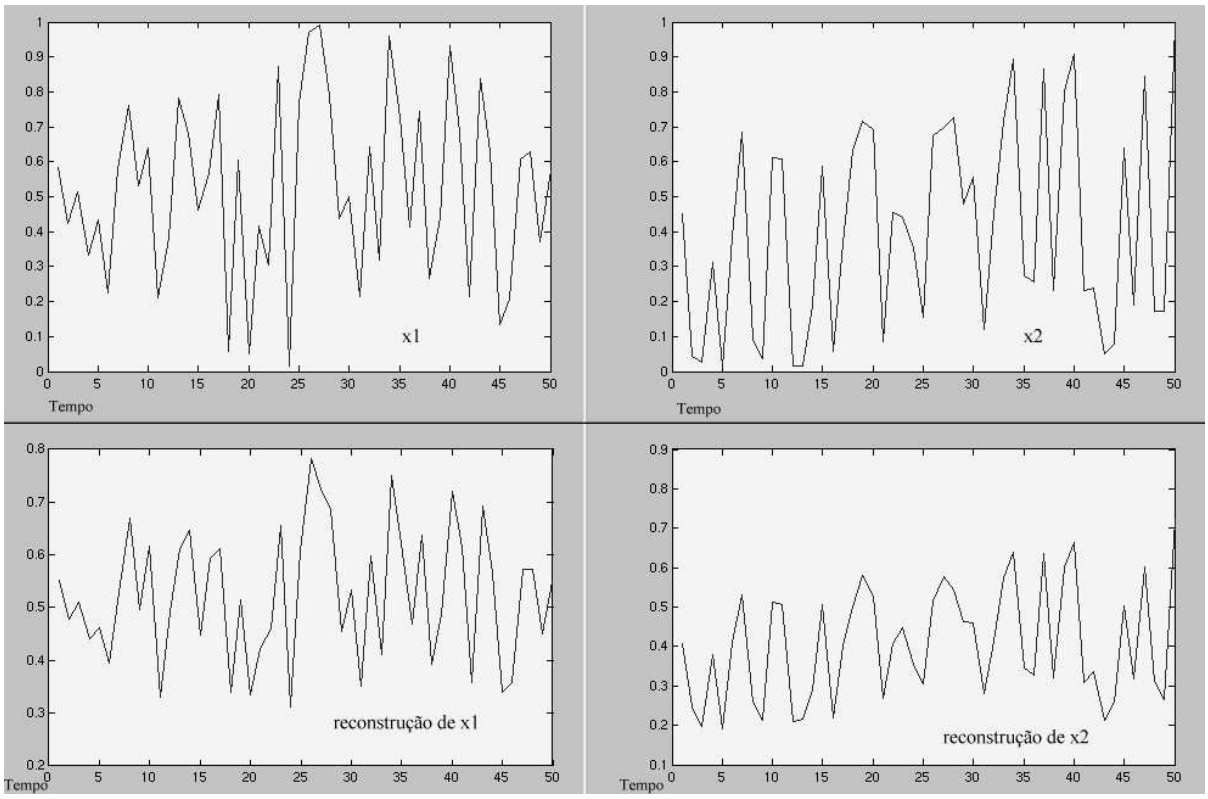

Figura 3.1: Dois exemplos de sinais de tamanho 50 (x_1 e x_2, acima) e suas respectivas reconstruções a partir de 25 descritores de Fourier (abaixo).

Pelas propriedades da transformada de Fourier, podemos notar que a utilização de descritores de Fourier é uma abordagem bastante eficiente de reconhecimento de padrões e visão computacional. Além disso, essa abordagem proporciona redução de dimensionalidade de forma eficiente e sem perda de informações relevantes em visão computacional.

3.2.2 Análise de Componentes Principais (PCA)

Segundo Jain et al. [Jain et al., 2000], o melhor extrator de características linear conhecido é o de análise de componentes principais (PCA). Essa transformada, também conhecida como transformada de Hotelling e por expansão de Karhunen-Loève, é amplamente utilizada pela comunidade de reconhecimento de padrões e de reconhecimento de faces [Kirby and Sirovich, 1990, Turk and Pentland, 1991, Chellappa et al., 1995, Romdhani, 1996, Pentland, 2000].

Visando a tratar imagens como padrões em um espaço linear para efetuar reconhecimento estatístico, essas devem ser representadas de acordo com o conceito de padrão descrito na seção 2.1. Sendo h o número de linhas de uma imagem e w o número de

colunas, pode-se dizer que uma imagem é um padrão de $h \times w$ características ou um vetor no espaço $(h \times w)$-dimensional, o qual chamaremos de "espaço de imagens", representado por $\mathcal{I}$.

Assim, dada uma imagem representada como uma matriz $h \times w$, pode-se construir sua representação como um vetor através de uma leitura coluna a coluna da imagem, colocando o valor de cada *pixel* da imagem em um vetor coluna $\mathbf{x}$. Ou seja, dada uma matriz Z de A linhas e L colunas representando uma imagem,

$$\mathbf{x}^l = Z^{j,k}, \tag{3.12}$$

para $j = 1, 2, 3, \cdots, h$, $k = 1, 2, 3, \cdots, w$ e $l = j + (k-1) \cdot h$. Assim, a dimensionalidade do espaço de imagens N é dada por $N = h \times w$.

A figura 3.2 ilustra didaticamente o processo de criação de um padrão $\mathbf{x}$ a partir de uma imagem de face[1].

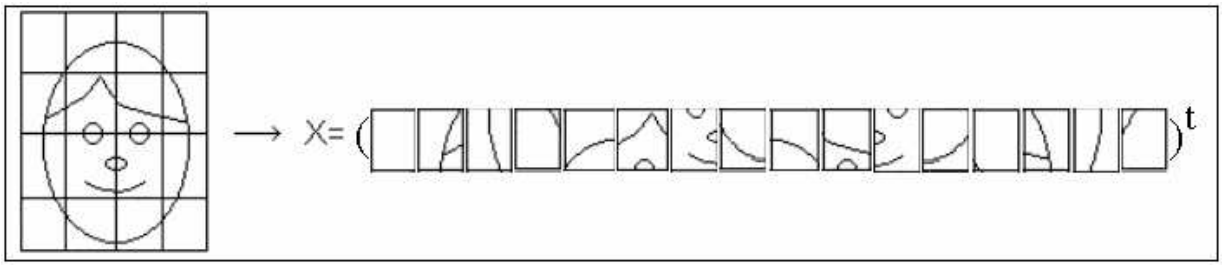

Figura 3.2: Processo de criação de um padrão $\mathbf{x}$ a partir de uma imagem (adaptada de [Romdhani, 1996]).

Dessa forma, a base canônica do espaço de faces pode ser ilustrada de acordo com a figura 3.3.

Em reconhecimento de padrões, é sempre desejável dispor de uma representação compacta e de um bom poder de discriminação de classes de padrões. Para isso, é importante que não haja redundância entre as diferentes características dos padrões, ou seja, que não haja covariância entre os vetores da base do espaço de características. Mas, obviamente, pode-se notar que o espaço de imagens é altamente redundante quando usado para descrever faces, pois cada *pixel* é muito correlacionado com outros *pixels*, já que todas as faces possuem olhos, nariz, boca, bochecha, testa etc, o que faz com que os vetores que representam faces sejam altamente correlacionados.

Para verificar se há covariância entre as características (ou variáveis), utiliza-se a matriz de covariância Σ da matriz dos padrões (de acordo com [Kennedy and Neville, 1986] e [Duda and Hart, 1973]). Dados $|T|$ padrões de treinamento, $\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_{|T|}$, a matriz de

[1]É importante notar que essa figura é apenas uma representação da imagem, pois em imagens reais, os *pixels* não possuem contornos.

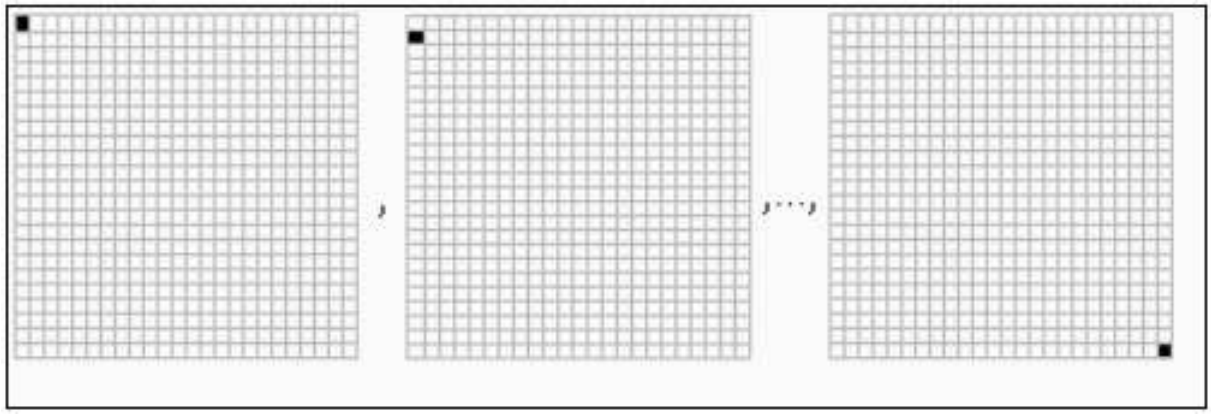

Figura 3.3: Base canônica do espaço de faces (adaptada de [Romdhani, 1996]).

covariância desses padrões é calculada a partir da matriz dos padrões de treinamento X. Ela é definida como uma matriz em que cada coluna possui um padrão de treinamento:

$$X = [\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_{|T|}] \tag{3.13}$$

Dada uma matriz de padrões X, a matriz Σ_X de covariância de X pode ser obtida a partir da seguinte aproximação (vide equação 2.15):

$$\Sigma_X = (X - \mu) \cdot (X - \mu)^t, \tag{3.14}$$

em que μ é a matriz $N \times |T|$ (mesma dimensão que X), e todas suas colunas contêm o valor esperado dos padrões de X, ou seja:

$$\mu_{l,i} = \frac{1}{|T|} \cdot \sum_{j=1}^{|T|} X_{l,j}, \tag{3.15}$$

para $l = 1, 2, 3, \cdots, N$ e $i = 1, 2, 3, \cdots, |T|$.

É importante notar que $\Sigma_X^{l,l}$ é a variância da característica l. Ou seja, os elementos da diagonal da matriz de covariância referem-se à variância das características. Já os elementos fora da diagonal, isto é, $\Sigma_{l,o}$ para $l \neq o$, representam a covariância entre a característica l e o. Se duas características, l e o, são estatisticamente independentes, a covariância é nula ($\Sigma_{l,o} = 0$).

Conforme dito anteriormente, é desejável que os padrões sejam representados em um espaço em que não haja covariância entre características diferentes. Um espaço vetorial com essa propriedade possui uma base cuja matriz de covariância de seus vetores é diagonal. Partindo-se de um conjunto de exemplos de padrões de treinamento para obter uma base com tal propriedade, basta utilizar uma transformada que diagonalize a matriz de covariância da base atual do espaço. Com a diagonalização da matriz de covariância,

a variância das variáveis (características) será maximizada e a covariância entre uma variável e outra será nula.

De acordo com [Theodoridis and Koutroumbas, 1999], [Duda and Hart, 1973] e [Callioli et al., 1998], devido ao processo de criação da matriz de covariância, pode-se mostrar que ela é diagonalizável. Para diagonalizar-se a matriz de covariância dos padrões de treinamento Σ_X, deve-se obter uma representação desses padrões em uma outra base do espaço de características. Em outras palavras, deve-se efetuar uma mudança de base. A matriz mudança de base que possui essa propriedade é definida da seguinte maneira:

$$H = [e_1, e_2, e_3, \cdots, e_m], \tag{3.16}$$

em que e_i é obtido a partir da seguinte decomposição:

$$\lambda_i e_i = \Sigma_X e_i, \tag{3.17}$$

ou seja, e_i é o i-ésimo auto-vetor de Σ_X [Callioli et al., 1998], m é o número total de auto-vetores de Σ_X, e λ_i é o i-ésimo auto-valor de Σ_X. Nos trabalhos em que o PCA é utilizado para reconhecimento de faces, ou seja, quando os padrões de treinamento são imagens de faces, esses autovetores são chamados de *eigenfaces* (vide capítulo 4). Isso deve-se ao fato de que esses auto-vetores, quando visualizados como imagens, possuem a aparência de faces. O mesmo ocorre para imagens regiões características da face, como olhos (*eigeneyes*), nariz (*eigennoses*) e boca (*eigenmouth*).

Assim, as variáveis dos padrões representados em termos dessa nova base do espaço de características não possuem correlação entre si. Essa mudança de base é efetuada através da seguinte operação[2]:

$$\mathbf{y}_i = H^t \cdot \mathbf{x}_i, \tag{3.18}$$

para $i = 1, 2, 3, \cdots, |T|$, em que $\mathbf{y}_i$ é a representação do padrão $\mathbf{x}_i$ nesse novo espaço de características. Para ilustrar o efeito dessa mudança de base, pode ser criada uma matriz Y contendo todos os padrões $\mathbf{y}_i$ (da mesma forma que é feita na criação da matriz X - equação 3.13). Dessa maneira, será verificado que a matriz de covariância de Y, Σ_Y, será diagonal.

É importante lembrar que os auto-valores refletem a importância dos auto-vetores. No caso de PCA, os auto-valores da matriz de covariância são iguais à variância das características transformadas [Theodoridis and Koutroumbas, 1999]. Assim, se um auto-vetor possui auto-valor grande, significa que esse fica em uma direção em que há uma grande variância dos padrões. A importância disso está no fato de que, em geral, é mais fácil distinguir padrões usando uma base em que seus vetores apontam para a direção da maior variância dos dados, além de não serem correlacionados entre si.

Através das figuras 3.4, 3.5 e 3.6, pode-se visualizar o efeito da transformada PCA para o caso bidimensional. Pode-se notar que é realizada uma rotação da base do espaço

[2]Note que essa é a mesma operação mostrada na equação 3.2.

vetorial de forma que o primeiro vetor da nova base fique na direção em que há maior variância dos dados e o segundo fique perpendicular ao primeiro, na direção da segunda maior variação.

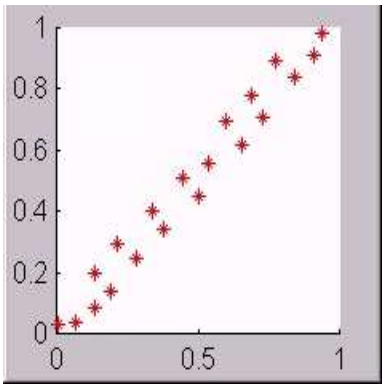

Figura 3.4: Dados artificiais bidimensionais.

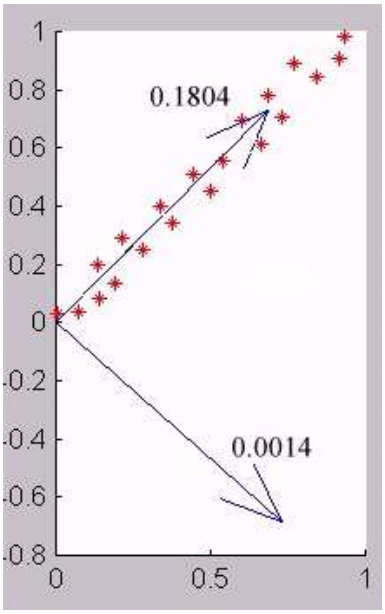

Figura 3.5: Dados de teste com os auto-vetores da matriz de covariância e seus respectivos auto-valores.

O número de auto-vetores obtido é, no máximo, igual ao número de *pixels* da imagem (ou variáveis dos padrões de entrada), ou seja, N. Porém, conforme dito anteriormente, se a matriz H for construída de forma que sejam escolhidos somente os auto-vetores contendo os maiores auto-valores, a variância total dos padrões de entrada não sofre grandes alterações. Em [Romdhani, 1996], o autor discute o conceito de **erro residual**, o qual é calculado atravéz da diferença entre a reconstrução dos padrões com o uso de

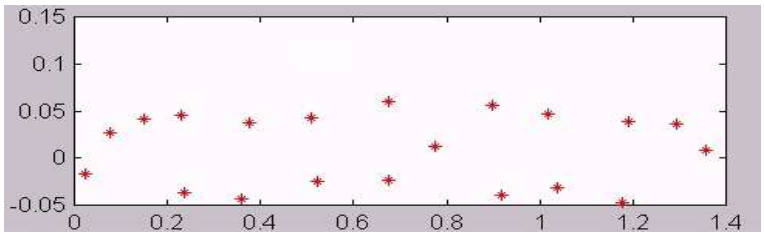

Figura 3.6: Dados no espaço criado.

todos os autovetores e a reconstrução utilizando alguns autovetores (com redução de dimensionalidade). A representação dos padrões no espaço de características formado pelos auto-vetores com os maiores auto-vetores possui erro residual pequeno. Assim, é possível realizar redução de dimensionalidade utilizando-se, na construção de H, somente os m primeiros auto-vetores. Com isso, a dimensionalidade dos vetores $\mathbf{y}_i$ torna-se m, o que significa uma redução de dimensionalidade de $N - m$ dimensões.

Embora essa transformada PCA seja relativamente simples conceitualmente, o processo de treinamento é complexo, visto que, dentre outras operações, é necessário efetuar $|T| \times |T| \times N$ multiplicações para criar a matriz de covariância Σ_X [Campos et al., 2000d]. Porém, sua aplicação é muito rápida e, em geral, produz bons resultados para reconhecimento de faces.

Segundo [Duda and Hart, 1973], PCA é uma técnica de extração de características não supervisionada propícia para dados com distribuição Gaussiana, mas não se tem certeza de que as faces possuam tal distribuição. Através das figuras 3.7, 3.8 e 3.9, pode-se observar um caso simples bidimensional ilustrando um problema que pode ocorrer com a redução de dimensionalidade através de PCA. Nesse caso, será muito mais difícil distinguir os padrões das duas classes utilizando somente o primeiro auto-vetor. Já o segundo auto-vetor possui a direção que melhor discrimina as duas classes.

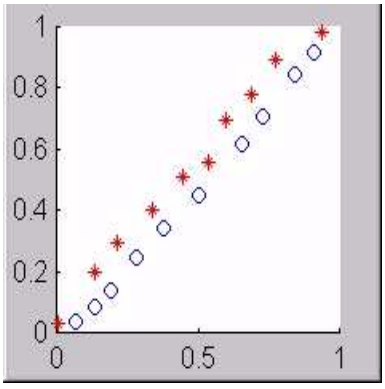

Figura 3.7: Dados artificiais de teste: duas classes em um espaço bidimensional.

Conforme será descrito posteriormente, uma forma de eliminar esse problema con-

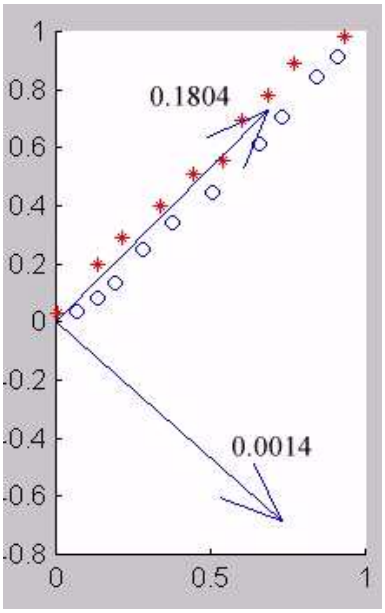

Figura 3.8: Dados de teste de duas classes com os auto-vetores da matriz de covariância e seus respectivos auto-valores.

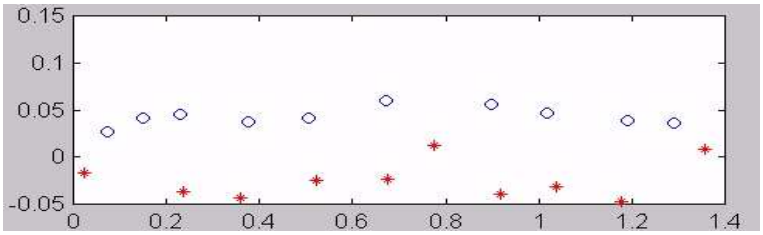

Figura 3.9: Dados no espaço criado: note que o primeiro auto-vetor não possui poder de discriminação.

siste na aplicação de um algoritmo de seleção de características (vide seção 3.3). Com a aplicação de um método de seleção automática de características, os autovetores da base do espaço de características são escolhidos de forma ótima.

3.2.3 Discriminantes Lineares (LDA)

A análise de discriminantes lineares (LDA), também conhecidos como discriminantes lineares de Fisher, é uma técnica que se tornou muito comum para reconhecimento de faces, principalmente a partir de 1997, com a publicação do artigo [Belhumeur et al., 1997]. Nesse artigo, os autores comparam PCA com LDA e mostram que o espaço de características criado pela transformação LDA proporcionou resultados de classificação muito melhores que o espaço criado pela transformada PCA para o reconhecimento de pessoas em imagens com grandes variações de iluminação.

Como pode-se observar na seção 3.2.2, a transformada de PCA é um método linear não supervisionado de extração de características que maximiza o espalhamento dos padrões no espaço de características, independentemente da classe em que esses pertencem [Jain et al., 2000]. Essas características possibilitam a ocorrência de problemas como aquele ilustrado nas figuras 3.7, 3.8 e 3.9. Para evitar tais problemas, podem ser aplicados algoritmos de seleção de características ou utilizar extratores de características que se baseiam em informações da distribuição das classes no espaço original.

Através de LDA, esses problemas podem ser evitados, pois trata-se de um método que utiliza informações das categorias associadas a cada padrão para extrair linearmente as características mais discriminantes. Em LDA, a separação inter-classes é enfatizada através da substituição da matriz de covariância total do PCA por uma medida de separabilidade como o critério Fisher.

Matematicamente, para todos os exemplos de todas as classes, define-se duas medidas:

1. matriz de espalhamento intra-classes, dada por

$$S_w = \sum_{j=1}^{c} \sum_{i=1}^{|T_j|} (\mathbf{x}_i^j - \mu_j) \cdot (\mathbf{x}_i^j - \mu_j)^t, \tag{3.19}$$

   em que $\mathbf{x}_i^j$ é o i-ésimo exemplo da classe j, μ_j é a média da classe j, c é o número de classes, e $|T_j|$ o número de exemplos na classe j;

2. matriz de espalhamento inter-classes, dada por:

$$S_b = \sum_{j=1}^{c} (\mu_j - \mu) \cdot (\mu_j - \mu)^t, \tag{3.20}$$

   em que μ representa a média de todas as classes.

O objetivo é maximizar a medida inter-classes e minimizar a medida intra-classes. Uma maneira de fazer-se isso é maximizar a taxa $\frac{det(S_b)}{det(S_w}$. A vantagem de se usar essa taxa é que foi provado [Fisher, 1938] que, se S_w é uma matriz não singular (com determinante não

nulo), então essa taxa é maximizada quando os vetores colunas da matriz de transformação H são os autovetores de $S_w^{-1} \cdot S_b$.

Pode ser provado que: (1) há no máximo $c-1$ autovetores e, então, o limite superior de m é $c-1$, e (2) são requeridos no mínimo $N+c$ exemplos de treinamento para garantir que S_w não se torne singular (o que geralmente é impossível em aplicações práticas). Para resolver isso, [Belhumeur et al., 1997] propuseram a utilização de um espaço intermediário, o qual pode ser o espaço criado pela transformada PCA. Então, o espaço N-dimensional original é projetado em um espaço g-dimensional intermediário usando PCA e, posteriormente, em um espaço m-dimensional, usando LDA.

Em geral, essa abordagem possibilita a obtenção de resultados melhores que o PCA para redução de dimensionalidade. A figura 3.10 mostra o caso de um espaço de características bidimensional com duas classes. Nesse espaço, caso seja realizada a redução para uma dimensão, a projeção no primeiro componente principal (PCA) acarreta um espaço de característica que proporciona uma alta taxa de erro. Já a projeção no primeiro discriminante linear (LDA) proporcionará a taxa de acerto de 100%. Nesse exemplo, supõe-se a utilização do classificador de vizinho mais próximo.

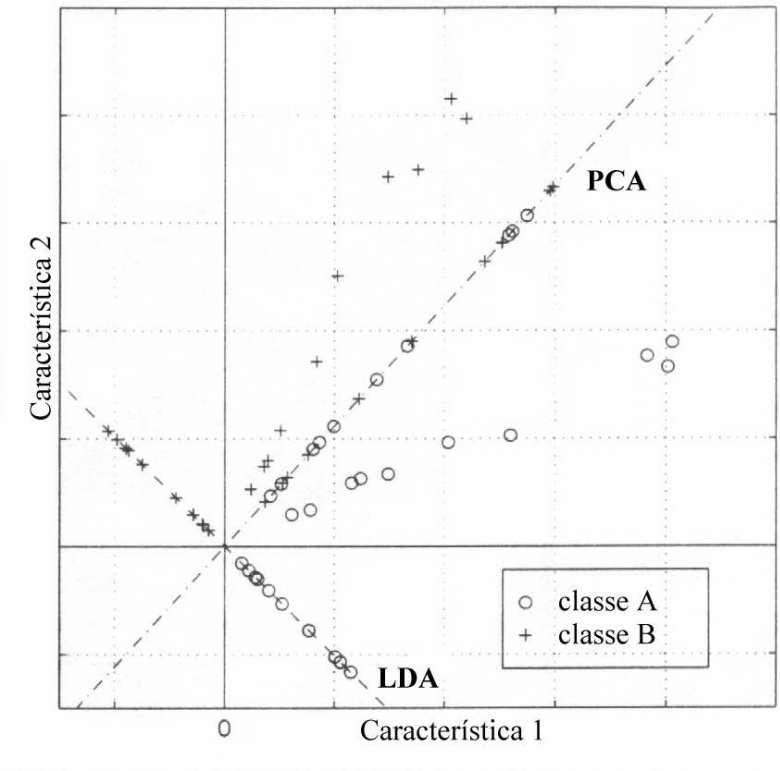

Figura 3.10: Exemplo em que a redução de dimensionalidade com LDA proporciona melhores resultados de classificação que PCA. Há duas classes em um espaço de características bidimensional (adaptada de [Belhumeur et al., 1997]).

Além desse exemplo, no caso ilustrado na figura 3.4, o discriminante linear de Fisher iria determinar, como primeiro vetor da base, exatamente aquele que foi determinado pelo segundo auto-vetor no caso de PCA, ou seja, o vetor cujo auto-valor é 0.0014 na figura 3.8.

Porém, [Martinez and Kak, 2001] mostraram recentemente que o desempenho de PCA pode ser superior ao de LDA quando o tamanho do conjunto de treinamento $|T|$ é pequeno. Esses resultados foram obtidos a partir de testes para reconhecimento de faces em uma base de imagens de 126 pessoas, sendo 26 imagens por pessoa, com problemas de oclusão e variações em expressões faciais. Foram realizadas duas baterias de testes, a primeira com poucas imagens de treinamento por pessoa (somente 2) e a segunda com várias imagens de treinamento (13). Na maioria dos experimentos com conjunto de treinamento pequeno, o desempenho do PCA foi superior ao do LDA. Por outro lado, em todos os testes com conjunto de treinamento grande, o desempenho do LDA foi superior ao do PCA.

A figura 3.11 ilustra um caso em que o desempenho de PCA é superior ao de LDA. Trata-se de um exemplo com duas classes, cujos padrões são representados por '$\times$' para a classe A e 'o' para a classe B. A distribuição dessas classes está ilustrada pelas elipses pontilhadas. Usando-se os dois exemplos de treinamento por classe mostrados na figura, o primeiro vetor do espaço PCA obtido está indicado por 'PCA', e a fronteira de decisão proporcionada por esse método está indicada por 'D_{PCA}'. Já o primeiro vetor do espaço LDA está indicado por 'LDA', e sua respectiva fronteira de decisão, por 'D_{LDA}'. Nota-se claramente que, caso seja reduzida a dimensionalidade para 1, pela distribuição das classes, a fronteira de decisão criada pelo PCA é superior à do LDA[3].

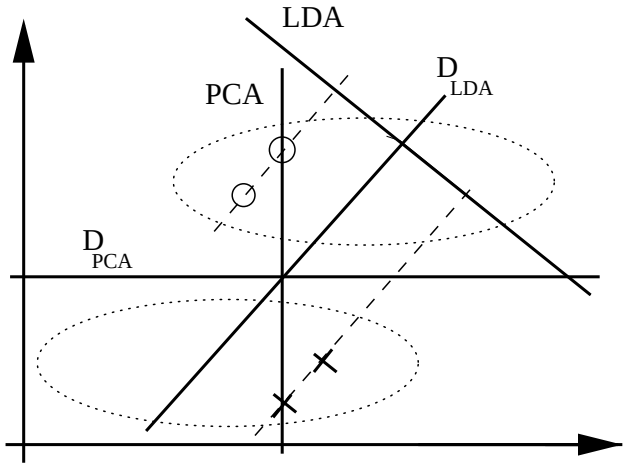

Figura 3.11: Efeito de PCA e LDA no espaço de características com poucas amostras de treinamento. Adaptada de [Martinez and Kak, 2001].

Além de requerer um conjunto de treinamento grande, outro problema dessa abor-

[3]Supõe-se que o classificador utilizado é o de vizinho mais próximo.

dagem é sua incapacidade de obter bons resultados se aplicada a classes com distribuição côncava e com interseção com outras classes, como no caso de dados com distribuição similar aos da figura 3.12 (em todas as dimensões). Nesse caso, a transformada vai tentar minimizar a variação intra-classe e maximizar a variação inter-classes, o que pode resultar em uma representação dos dados pior do que a original para classificadores como os K-vizinhos mais próximos. Isso reforça a necessidade da utilização de algoritmos de seleção de características.

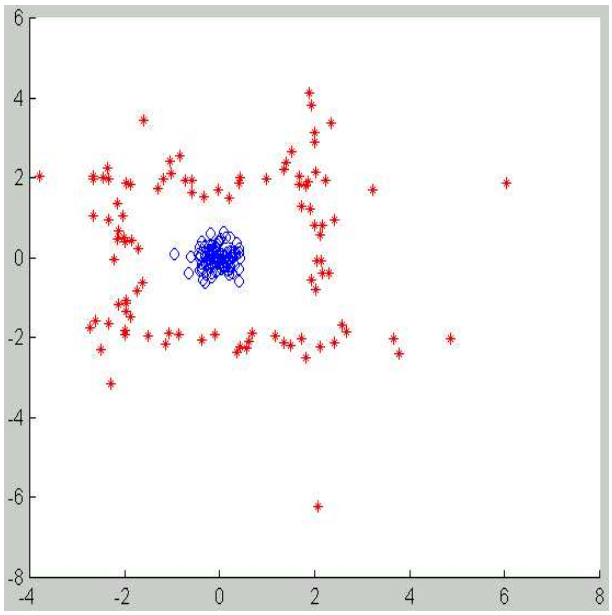

Figura 3.12: Exemplo de distribuição que pode falhar com um discriminante linear.

Maiores detalhes a respeito de discriminantes lineares podem ser obtidos através das referências [Theodoridis and Koutroumbas, 1999] e [Fisher, 1938].

3.3 Seleção de Características

Métodos automáticos de seleção de características são importantes em muitas situações em que se tem disponível um conjunto grande de características e deseja-se selecionar um subconjunto adequado. Além de ser uma forma de redução de dimensionalidade, uma aplicação importante é a fusão de dados procedentes de múltiplas modalidades de sensores ou de múltiplos modelos de dados. A importância de redução de dimensionalidade está explícita no capítulo 3.

A seleção automática de características é uma técnica de otimização que, dado um conjunto de N características, tenta selecionar um subconjunto de tamanho m ($m < N$) que maximiza uma função critério.

Formalmente, dado um conjunto $\mathcal{Y}$ de N características, o algoritmo de seleção de características deve encontrar um subconjunto $\mathcal{X} \subseteq \mathcal{Y}$ tal que $|\mathcal{X}| = m$, em que $|\mathcal{X}|$ denota a cardinalidade de $\mathcal{X}$, e

$$J(\mathcal{X}) = \max_{\mathcal{Z} \subseteq \mathcal{Y}, |\mathcal{Z}|=m} J(\mathcal{Z}), \tag{3.21}$$

em que $J(.)$ é a função critério. Um exemplo simples é definir-se $J(\mathcal{X}) = 1 - E$, sendo E é a taxa ou probabilidade de erro de um classificador. É desejável que a função critério seja maior quanto menor for a redundância entre as características e quanto maior a facilidade de discriminar padrões de classes diferentes.

Dessa forma, o algoritmo de seleção de características poderá reduzir a dimensionalidade de forma que ocorra a menor queda possível no poder de distinção das classes por um classificador no espaço de características. Uma conseqüência da aplicação de um bom algoritmo de seleção de atributos é a redução do número necessário de amostras de treinamento para obter-se bons resultados com um classificador, ou seja, a redução do problema da dimensionalidade (vide seção 2.3).

Além da escolha da função critério, também é importante determinar a dimensionalidade apropriada do espaço de características reduzido. Uma forma simples de resolver esse problema é efetuar a seleção de características para vários valores de m. Conforme foi mencionado na seção 2.3, em [Jain et al., 2000], os autores defendem que, em problemas práticos, sendo $|T|$ o tamanho do conjunto de treinamento, é seguro não ocorrer o problema da dimensionalidade se forem usadas menos que $|T|/10$ características.

Apesar da importância de seleção de atributos, não há regras ou procedimentos definitivos para essa tarefa em cada aplicação particular [Castleman, 1996], principalmente quando o número de características disponíveis for grande. Por esse motivo, um grande conjunto de algoritmos de seleção de atributos tem sido proposto. Em [Jain and Zongker, 1997] foi proposta uma taxonomia sobre este tópico. A seguir serão descritos separadamente alguns algoritmos de seleção de características e algumas funções critério.

3.3.1 Algoritmos de seleção

Há vários métodos diferentes de seleção de características. Baseando-se na taxonomia proposta em [Jain and Zongker, 1997], tais abordagens podem ser agrupadas em categorias conforme descrito na taxonomia exibida na figura 3.13.

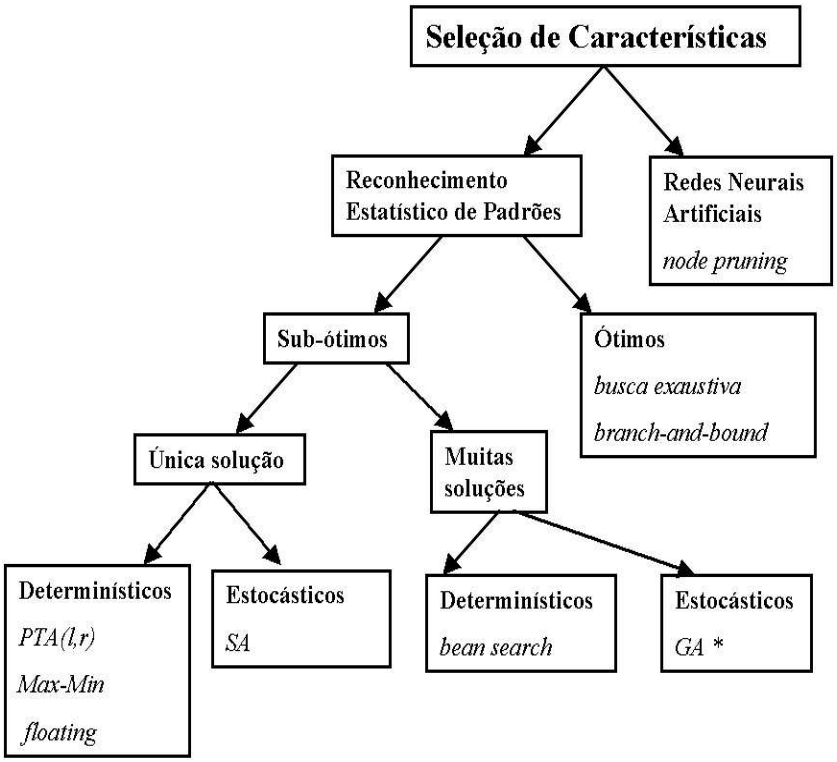

Figura 3.13: Taxonomia dos métodos de seleção de características. Adaptada da figura 1 contida em [Jain and Zongker, 1997].

A seguir, há uma breve descrição de cada uma dessas abordagens. Neste trabalho, foram focalizados os métodos flutuantes (*floating*). Por isso será dedicada uma seção aos métodos determinísticos de solução única (seção 3.3.2). É importante mencionar os métodos citados na figura 3.13 são utilizados nos casos em que não é realizada uma estimativa da função densidade de probabilidade das classes de padrões. O leitor interessado em métodos de seleção para espaços com distribuições probabilísticas previamente estimadas ou conhecidas é referido aos trabalhos [Kittler et al., 2001], que possui uma revisão de tais métodos.

Redes Neurais

Um método de seleção de características bem conhecido que utiliza uma rede neural é chamado **Node Pruning** [Mao et al., 1994] ou "corte de nós". Basicamente, o algoritmo

funciona através de uma rede neural multi-camadas com retro-alimentação, utilizando um algoritmo de aprendizado baseado em retro-propagação (*backpropagation*). É definida uma medida de "saliência de nós" e utilizado um algoritmo que elimina os nós menos salientes. Dessa forma, a complexidade da rede pode ser reduzida após seu treinamento. A eliminação dos nós de entrada significa a eliminação de características do conjunto de características. A saliência de um nó é definida pela soma do aumento no erro sobre todos os padrões de treinamento, como um resultado da remoção daquele nó (vide equação 3.25).

Inicialmente, a rede neural é treinada, sendo posteriormente realizada a eliminação de nós seguida de um re-treinamento da rede, repetindo-se o processo até que seja alcançada a dimensão desejada. A vantagem do método *node-pruning* é que ele simultaneamente determina o melhor subconjunto de características e o classificador ótimo.

Métodos ótimos

Em termos da qualidade do conjunto de características obtido, o único método realmente ótimo é o da **busca exaustiva**. Nesse método, todos os $\binom{N}{m}$ subconjuntos possíveis de tamanho m são avaliados. Essa abordagem é muito cara computacionalmente, mesmo para conjuntos não muito grandes, pois sua complexidade é exponencial.

Algumas funções critério possuem uma propriedade chamada **monotonicidade**. Uma função é monotônica se $J(\mathcal{X} \bigcup \mathcal{Z}) \geq J(\mathcal{X})$, para todo $\mathcal{X}, \mathcal{Z} \subseteq \mathcal{Y}$. Ou seja, o valor da função critério é sempre maior para conjuntos de características maiores. Para este caso, há o algoritmo de busca em árvores chamado **branch-and-bound**, proposto em [Narendra and Fukunaga, 1977]. Esse algoritmo pode retornar a solução ideal sem verificar todas as possibilidades, mas sabemos que, devido ao problema da dimensionalidade (vide seção 2.3), para situações em que o conjunto de treinamento não é grande o suficiente, normalmente a função critério não é monotônica. Por esse fato, o algoritmo *branch-and-bound* não pode ser aplicado em quaisquer situações.

Outra desvantagem desse método é que, no pior caso, todas as configurações são consultadas, o que faz com que o algoritmo tenha complexidade exponencial no pior caso, tornando impraticável para conjuntos de características grandes. Por essas razões existem os métodos sub-ótimos, os quais não garantem que o conjunto de características obtido seja o melhor possível, mas são eficientes em termos de tempo de execução, pois eles não consultam todas as possibilidades para determinar a(s) solução(ões). A seguir serão comentados alguns dos métodos sub-ótimos.

Métodos estocásticos com múltiplas soluções

Os métodos estocásticos com múltiplas soluções são aqueles que, após serem executados, fornecem vários conjuntos de características que obtiveram bons resultados quando avaliados pela função critério. Além disso, uma característica importante desses métodos é

que, a cada vez que eles são executados, eles podem fornecer um conjunto de soluções diferente do anterior.

Essa classe de métodos engloba o uso de algoritmos genéticos para seleção de características [Siedleki and Sklansky, 1989]. Nessa abordagem, o conjunto de características é representado como uma cadeia binária de caracteres de tamanho N em que 0 ou 1 na posição i indica a ausência ou presença da característica i. Essa cadeia é chamada "cromossomo".

Inicialmente, uma população aleatória de cromossomos é criada. Cada cromossomo é avaliado, através da função critério, para determinar sua aptidão (*fitness*), a qual informa se o cromossomo irá "sobreviver" à próxima geração ou "morrer". A partir de mutações ou cruzamentos dos cromossomos atuais, são criados novos cromossomos.

Após várias iterações, a aptidão geral da população será melhorada e sempre haverá várias soluções. Porém, conforme mencionado anteriormente, como os resultados são obtidos a partir de processos aleatórios (portanto não-determinísticos), normalmente são obtidos sub-conjuntos diferentes quando o algoritmo é aplicado ao mesmo conjunto em outro momento. Em [Bruno et al., 1998], essa técnica foi aplicada para efetuar a classificação de formas biológicas.

Métodos determinísticos de múltiplas soluções

Ao contrário dos métodos estocásticos de múltiplas soluções, os métodos determinísticos de múltiplas soluções apresentam sempre os mesmos conjuntos de características.

Dentre esses métodos, alguns tratam o sub-espaço de características como um grafo, chamado "reticulado de seleção de características", em que cada nó representa um sub-conjunto e uma aresta representa a relação de sub-conjunto. Para selecionar os melhores conjuntos, aplica-se um algoritmo padrão de busca em grafos. Como exemplos de métodos dessa categoria, encontram-se o "**best-first search**" e uma versão restrita chamada "**beam search**", os quais foram utilizados em [Siedleki and Sklansky, 1989] para seleção de características.

3.3.2 Métodos Determinísticos com Solução Única

Há vários métodos de seleção de características determinísticos de solução única. A seguir, serão descritos alguns desses métodos que são baseados em técnicas de busca.

Preliminares

A maioria dos métodos determinísticos de solução única são baseados em buscas. Dentre eles, a maioria possui duas abordagens: **para frente** (*botton-up*) e **para trás** (*top-down*). Na abordagem *para frente*, inicia-se com um conjunto de avaliação (temporário) vazio e, conforme o algoritmo é executado, são inseridas características nesse conjunto, até que esse fique com tamanho m. Já na abordagem *para trás*, inicia-se com um conjunto de avaliação contendo todas as características disponíveis e, nas iterações do algoritmo, são excluídas características até que esse conjunto fique com o tamanho m. Em geral, podem-se dizer que os métodos *para frente* são mais rápidos que seus equivalentes *para trás*, pois o custo de medição da função critério em conjuntos de características grandes é maior que o custo em conjuntos pequenos [Jain and Zongker, 1997]. Porém, quando o valor de m é próximo de N, deve-se dar preferência à utilização dos métodos *para trás*.

Abaixo apresentamos as definições utilizadas nos trabalhos de [Pudil et al., 1994] e [Somol et al., 1999] na descrição dos métodos de busca seqüenciais.

Seja $\mathcal{X}_k = \{x_i : 1 \leq i \leq k, x_i \in \mathcal{Y}\}$ um subconjunto de k características do conjunto $\mathcal{Y} = \{y_i : 1 \leq i \leq N\}$ das N características disponíveis, o valor $J(y_i)$ da função critério de seleção de características, quando somente a i-ésima característica y_i, $i = 1, 2, \cdots, N$ for utilizada, é chamado de **significância individual** $\mathcal{S}_0(y_i)$ da característica.

A **significância** $\mathcal{S}_{k-1}(x_j)$ **da característica** x_j, $j = 1, 2, \cdots, k$ no conjunto $\mathcal{X}_k$ é definida por

$$\mathcal{S}_{k-1}(x_j) = J(\mathcal{X}_k) - J(\mathcal{X}_k - x_j) \tag{3.22}$$

A **significância** $\mathcal{S}_{k+1}(f_j)$ **da característica** f_j do conjunto $\mathcal{Y} - \mathcal{X}_k$, tal que $\mathcal{Y} - \mathcal{X}_k = \{f_i : i = k+1, k+2, \cdots, N, f_i \in \mathcal{Y}, f_i \neq x_l, \forall x_l \in \mathcal{X}_k\}$, **em relação ao conjunto** $\mathcal{X}_k$, é definida por

$$\mathcal{S}_{k+1}(f_i) = J(\mathcal{X}_k + f_j) - J(\mathcal{X}_k). \tag{3.23}$$

Nota: para $k = 0$, o termo significância de uma característica no conjunto coincide com o termo significância individual.

Dizemos que a característica x_j do conjunto $\mathcal{X}_k$ é:

1. **a característica mais significante** (melhor) do conjunto $\mathcal{X}_k$ se

$$\mathcal{S}_{k-1}(x_j) = \max_{1 \leq i \leq k} \mathcal{S}_{k-1}(x_i) \Rightarrow J(\mathcal{X}_k - x_j) = \min_{1 \leq i \leq k} J(\mathcal{X}_k - x_i), \tag{3.24}$$

2. **a característica menos significante** (pior) do conjunto $\mathcal{X}_k$ se

$$\mathcal{S}_{k-1}(x_j) = \min_{1 \leq i \leq k} \mathcal{S}_{k-1}(x_i) \Rightarrow J(\mathcal{X}_k - x_j) = \max_{1 \leq i \leq k} J(\mathcal{X}_k - x_i). \tag{3.25}$$

Dizemos que a característica f_j do conjunto $\mathcal{Y} - \mathcal{X}_k$ é:

1. **a característica mais significante** (melhor) em relação ao conjunto $\mathcal{X}_k$ se

$$\mathcal{S}_{k+1}(f_j) = \max_{k+1 \leq i \leq N} \mathcal{S}_{k+1}(f_i) \Rightarrow J(\mathcal{X}_k + f_j) = \max_{k+1 \leq i \leq N} J(\mathcal{X}_k + f_i), \tag{3.26}$$

2. **a característica menos significante** (pior) em relação ao conjunto $\mathcal{X}_k$ se

$$\mathcal{S}_{k+1}(f_j) = \min_{k+1 \leq i \leq N} \mathcal{S}_{k+1}(f_i) \Rightarrow J(\mathcal{X}_k - f_j) = \min_{k+1 \leq i \leq N} J(\mathcal{X}_k + x_i). \tag{3.27}$$

Seja $\mathcal{T}_o$ genericamente uma tupla de o características, o valor da função critério $J(\mathcal{T}_o)$, quando somente as características $t_i, i = 1, 2, \cdots, o, t_i \in \mathcal{T}_o$ forem utilizadas, será chamado **significância individual $\mathcal{S}_0(T_o)$ da o-tupla de características**.

A **significância $\mathcal{S}_{k-o}(\mathcal{T}_o)$ da o-tupla de características** $\mathcal{T}_o = \{t_i : 1 \leq i \leq o, t_i \in \mathcal{X}_k\}$ **no conjunto** $\mathcal{X}_k$ é definida por

$$\mathcal{S}_{k-o}(\mathcal{T}_o) = J(\mathcal{X}_k) - J(\mathcal{X}_k - \mathcal{T}_o). \tag{3.28}$$

A **significância $\mathcal{S}_{k+o}(\mathcal{U}_o)$ da o-tupla de características** $\mathcal{U}_o = \{u_i : 1 \leq i \leq o,$ $u_i \in \mathcal{Y} - \mathcal{X}_k\}$ **no conjunto $\mathcal{Y} - \mathcal{X}_k$ em relação ao conjunto** $\mathcal{X}_k$ é definida por

$$\mathcal{S}_{k+o}(\mathcal{U}_o) = J(\mathcal{X}_k \bigcup \mathcal{U}_o) - J(\mathcal{X}_k). \tag{3.29}$$

Denotamos por $\mathcal{T}_o^i$ a i-ésima tupla contida no conjunto de todas as $\Theta = \binom{k}{o}$ o-tuplas possíveis de $\mathcal{X}_k, 1 \leq i \leq \Theta$. Pode-se dizer que a o-tupla de características $\mathcal{T}_o^i$ do conjunto $\mathcal{X}_k$ é:

1. **a o-tupla de características mais significante (melhor) do conjunto** $\mathcal{X}_k$ se

$$\mathcal{S}_{k-o}(\mathcal{T}_o^n) = \max_{1 \leq i \leq \Theta} \mathcal{S}_{k-o}(\mathcal{T}_o^i) \Rightarrow J(\mathcal{X}_k - \mathcal{T}_o^n) = \min_{1 \leq i \leq \phi} J(\mathcal{X}_k - \mathcal{T}_o^i); \tag{3.30}$$

2. **a o-tupla de características menos significante (pior) do conjunto** $\mathcal{X}_k$ se

$$\mathcal{S}_{k-o}(\mathcal{T}_o^n) = \min_{1 \leq i \leq \Theta} \mathcal{S}_{k-o}(\mathcal{T}_o^i) \Rightarrow J(\mathcal{X}_k - \mathcal{T}_o^n) = \max_{1 \leq i \leq \Theta} J(\mathcal{X}_k - \mathcal{T}_o^i). \tag{3.31}$$

Dizemos que a o-tupla de características $\mathcal{U}_o$ do conjunto $\mathcal{Y} - \mathcal{X}_k$ é:

1. **a o-tupla de características mais significante (melhor) em relação ao conjunto** $\mathcal{X}_k$ se

$$\mathcal{S}_{k+o}(\mathcal{U}_o^r) = \max_{1 \leq i \leq \Psi} \mathcal{S}_{k+o}(\mathcal{U}_o^i) \Rightarrow J(\mathcal{X}_k \bigcup \mathcal{U}_o^r) = \max_{1 \leq i \leq \Psi} J(\mathcal{X}_k \bigcup \mathcal{U}_o^i), \tag{3.32}$$

em que $\Psi = \binom{N-k}{o}$ é o número de todas as o-tuplas possíveis de $\mathcal{Y} - \mathcal{X}_k$;

2. **a o-tupla de características menos significante (pior) em relação ao conjunto $\mathcal{X}_k$** se

$$\mathcal{S}_{k+o}(\mathcal{U}_o^r) = \min_{1\leq i\leq \Psi} \mathcal{S}_{k+o}(\mathcal{U}_o^i) \Rightarrow J(\mathcal{X}_k \bigcup \mathcal{U}_o^r) = \min_{1\leq i\leq \Psi} J(\mathcal{X}_k \bigcup \mathcal{U}_o^i). \tag{3.33}$$

Nota: para $o = 1$, todos os termos relacionados com o significado de o-tuplas de características coincidem com os termos relacionados com a **significância individual** de uma característica.

A seguir apresentamos a descrição dos principais métodos de seleção de características determinísticos de solução única.

Melhores Características Individuais

O método de seleção de características pelas melhores caracteríticas individuais consiste na avaliação de todas as características tomadas individualmente e seleção das m melhores. O algoritmo abaixo detalha esse método. Note que, para fim de facilitar a exposição, o parâmetro k dos conjuntos X foi omitido nesse e nos próximos algoritmos, pois o valor de k varia conforme a execução dos algoritmos e os algoritmos podem ser chamados com conjuntos de diferentes tamanhos.

BF($\mathcal{Y}, m$)
$\mathcal{X} \leftarrow \emptyset$
ENQUANTO $|\mathcal{X}| < m$ FAÇA
  $\mathcal{X} \leftarrow \mathcal{X} \bigcup \{\max_{1\leq i\leq N} \mathcal{S}_0(y_i), \forall y_i \notin \mathcal{X}\}$
RETORNE $\mathcal{X}$

Como as características são avaliadas individualmente, esse método não é classificado nem como *para frente*, nem como *para trás*. Trata-se de um método bastante intuitivo e computacionalmente simples, mas que não garante que o melhor subconjunto seja determinado, pois algumas características podem ser boas tomadas individualmente, mas podem formar um conjunto ruim quando associadas entre si. Outros detalhes sobre esse método encontram-se em [Jain and Zongker, 1997, Theodoridis and Koutroumbas, 1999]

Busca Seqüencial para Frente (SFS)

O método de busca seqüencial para frente, como o próprio nome diz, é um método *botton-up*. Dado um conjunto de características já selecionadas (inicialmente nulo), a cada iteração é seleciona a característica que, unida ao conjunto determinado pela iteração anterior, produz o melhor resultado da função critério. Essa característica é adicionada ao conjunto de características anterior e uma nova iteração é realizada. São realizadas m iterações. O algoritmo a seguir detalha esse processo, devem-se assumir que inicialmente

$\mathcal{X} \leftarrow \emptyset$.

SFS($\mathcal{Y}, \mathcal{X}, m$)
ENQUANTO $|\mathcal{X}| < m$ FAÇA
  $\mathcal{X} \leftarrow \mathcal{X} \bigcup \{ \max_{1 \leq j \leq N} \mathcal{S}_{k+1}(f_j), \forall f_j \notin \mathcal{X} \}$
RETORNE $\mathcal{X}$

Observa-se que a instrução $\mathcal{X} \leftarrow \emptyset$ não foi incluída no algoritmo da função SFS($\cdot$), pois essa função será utilizada posteriormente para conjuntos não vazios. Isso repetir-se-á na função SBS($\cdot$) a seguir.

A desvantagem desse método é que, uma vez que uma característica tenha sido selecionada, ela não pode ser descartada do subconjunto ótimo, o que pode proporcionar o chamado efeito *nesting*. O efeito *nesting* ocorre quando o subconjunto ótimo não contém elementos do conjunto já selecionado, o que impossibilita que seja obtido o conjunto de características ótimo.

A principal vantagem da busca seqüencial para frente é o custo computacional quando se deseja obter conjuntos pequenos em relação ao total de caracteríscias. Outros detalhes a respeito desses métodos podem ser encontrados em [Jain and Zongker, 1997, Theodoridis and Koutroumbas, 1999].

Busca Seqüencial para Trás (SBS)

O algoritmo de busca seqüencial para trás é uma versão *top-down* do algoritmo anterior. A diferença entre SBS e SFS é que o SBS é iniciado com o conjunto de características completo (contendo todas as N características) e vai eliminando as menos importantes, ou seja, as que menos alteram a função critério quando são eliminadas. O algoritmo a seguir detalha esse processo, devem-se assumir que inicialmente $\mathcal{X} \leftarrow \mathcal{Y}$.

SBS($\mathcal{X}, m$)
ENQUANTO $|\mathcal{X}| > m$ FAÇA
  $\mathcal{X} \leftarrow \mathcal{X} - \{ \min_{1 \leq j \leq k} \mathcal{S}_{k-1}(x_j), \forall f_j \notin \mathcal{X}_k \}$
RETORNE $\mathcal{X}$

Assim como o método de busca seqüencial para frente, a desvantagem desse método é que, uma vez eliminada uma característica, ela não retornará ao subconjunto ótimo novamente. Como conseqüência, também pode ocorrer o efeito *nesting* caso o melhor subconjunto contenha alguma das características que foram eliminadas.

A principal vantagem desse método é o custo computacional, quando se deseja obter conjuntos grandes em relação ao total de características. Outros detalhes sobre esse método encontram-se em [Jain and Zongker, 1997, Theodoridis and Koutroumbas, 1999].

Mais l - Menos r (PTA) [Somol et al., 1999, Theodoridis and Koutroumbas, 1999]

O método *mais l - menos r*, cujo nome original é "*Plus l - Take Away r*" (PTA), foi criado visando a evitar o efeito *nesting*. Basicamente, em cada iteração, primeiro o algoritmo adiciona l elementos ao conjunto de características usando o método de seleção para frente (SFS) e, posteriormente, elimina r características usando a busca seqüencial para trás (SBS). Os valores de l e r devem ser determinados pelo usuário. Na versão *botton-up*, l deve ser maior que r. Já na versão *top-down*, $l < r$. Segue o algoritmo que detalha esse processo:

PTA($\mathcal{Y}, m, l, r$)
SE $l > r$ ENTÃO
  $\mathcal{X} \leftarrow \emptyset$
  ENQUANTO $|\mathcal{X}| < m$ FAÇA
    $\mathcal{X} \leftarrow \text{SFS}(\mathcal{Y}, \mathcal{X}, |\mathcal{X}| + l)$
    $\mathcal{X} \leftarrow \text{SBS}(\mathcal{X}, |\mathcal{X}| - r)$
SENÃO
  SE $l < r$ ENTÃO
    $\mathcal{X} \leftarrow \mathcal{Y}$
    ENQUANTO $|\mathcal{X}| > m$ FAÇA
      $\mathcal{X} \leftarrow \text{SBS}(\mathcal{X}, |\mathcal{X}| - r)$
      $\mathcal{X} \leftarrow \text{SFS}(\mathcal{Y}, \mathcal{X}, |\mathcal{X}| + l)$
  SENÃO
    RETORNE ERRO!
RETORNE $\mathcal{X}$

Conforme mencionado, esse método de busca evita o problema de *nesting*, mas com ele surge um novo problema: a determinação dos valores de l e r. Se forem tomados valores muito pequenos, é possível que o problema *nesting* não seja evitado. Por outro lado, se os valores de l e r forem muito grandes, o algoritmo torna-se muito lento.

Algoritmos de Busca Seqüencial Generalizada (GSFS e GSBS) [Somol et al., 1999, Theodoridis and Koutroumbas, 1999]

Os algoritmos de busca seqüencial generalizada inserem (no caso do GSFS) ou removem (no caso do GSBS) tuplas (subconjuntos) de características ao invés de o fazerem com apenas uma característica por iteração. Para possibilitar o funcionamento dos algoritmos generalizados, devem-se utilizar funções que determinam a significância de tuplas.

Os dois algoritmos de busca generalizada mais conhecidos são os seguintes:

1. **GSFS**: essa é a versão generalizada do algoritmo SFS. Devem-se assumir que inicialmente $\mathcal{X} \leftarrow \emptyset$

   GSFS($\mathcal{Y}, \mathcal{X}, m, o$)
   ENQUANTO $|\mathcal{X}| < m$ FAÇA
   $\mathcal{X} \leftarrow \mathcal{X} \bigcup \{ \max_{1 \leq i \leq \Psi} \mathcal{S}_{k+o}(\mathcal{U}_o^i)$
   RETORNE $\mathcal{X}$

2. **GSBS**: essa é a versão generalizada do algoritmo SBS. Devem-se assumir que inicialmente $\mathcal{X} \leftarrow \mathcal{Y}$

   GSBS($\mathcal{X}, m, o$)
   ENQUANTO $|\mathcal{X}| > m$ FAÇA
   $\mathcal{X} \leftarrow \mathcal{X} - \{ \min_{1 \leq i \leq \Theta} \mathcal{S}_{k-o}(\mathcal{T}_o^i)$
   RETORNE $\mathcal{X}$

Além desses algoritmos, há também uma versão generalizada do algoritmo PTA, em que, para cada passo, ao invés de serem inseridas ou excluídas características individuais, são avaliadas tuplas de tamanho definido pelo usuário (para frente e para trás). Esse algoritmo proporciona resultados muito próximos do resultado ótimo, mas seu custo computacional pode torná-lo proibitivo em conjuntos de características grandes [Pudil et al., 1994].

Como esses algoritmos inserem ou removem tuplas de características ao invés de características individuais, a probabilidade de ocorrer o efeito *nesting* é reduzida. Porém, o problema da escolha do tamanho dessas tuplas (o) é fundamental para a obtenção do equilíbrio entre tempo de execução e qualidade dos resultados. Quando o tamanho das tuplas for muito grande, o algoritmo torna-se muito lento. Por outro lado, quando esse valor for pequeno, os resultados se aproximam das versões não generalizadas desses algoritmos.

Métodos de Busca Seqüencial Flutuante (SFSM)

Os métodos de busca seqüencial flutuante para frente e para trás, propostos em [Pudil et al., 1994] podem ser vistos como generalizações do método *mais l - menos r*, em que os valores de l e r são determinados e atualizados dinamicamente. Como os próprios nomes dizem, o método de busca para frente (SFFS) é a versão *botton-up*, enquanto o de busca para trás (SFBS), *top-down*.

O fluxograma da figura 3.14 resume o funcionamento da versão *para frente* desse algoritmo. A seguir, apresentamos o algoritmo em sua forma completa. Para tornar mais clara a exposição, é suposto que k características já foram selecionadas do conjunto completo de características $\mathcal{Y} = \{y_j | j = 1, 2, \cdots, N\}$ para formar o conjunto $\mathcal{X}_k$ com a correspondente função critério $J(\mathcal{X}_k)$. Porém, esse algoritmo deve iniciar-se com $k = 0$ e $\mathcal{X} = \emptyset$. Adicionalmente, os valores de $J(\mathcal{X}_i)$ de todos os subconjuntos precedentes de

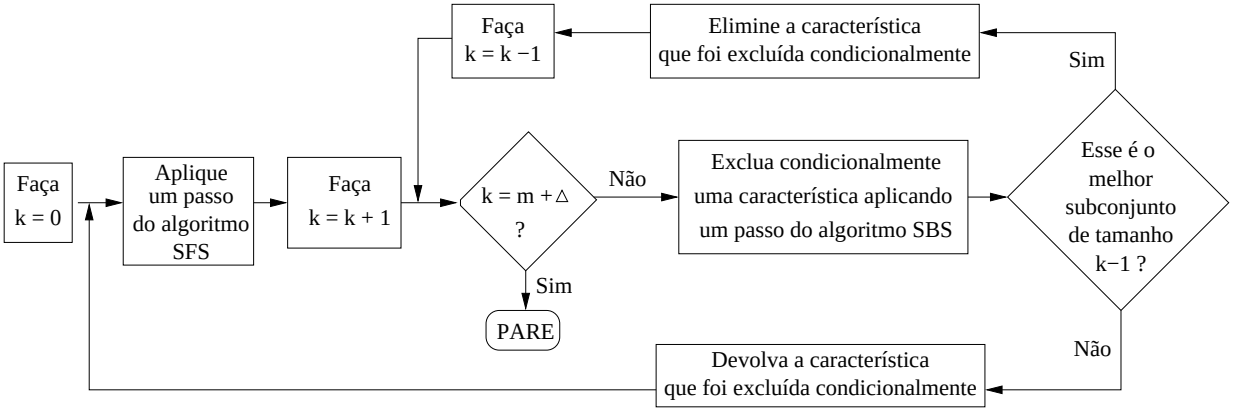

Figura 3.14: Fluxograma simplificado do algoritmo SFFS. Adaptada de [Jain and Zongker, 1997].

tamanho $i = 1, 2, \cdots, k-1$, são conhecidos e foram armazenados.

SFFS$(\mathcal{Y}, \mathcal{X}, m)$

1:

$\mathcal{X}_{k+1} \leftarrow \text{SFS}(\mathcal{Y}, \mathcal{X}_k, k+1)$

SE $k = m + \delta$ ENTÃO

RETORNE $\mathcal{X}_m$

2:

SE $J(\mathcal{X}_k) \geq J(\mathcal{X}_{k+1} - x_j), \forall j = 1, 2, \cdots, k$ ENTÃO

$k \leftarrow k + 1$

VÁ AO PASSO 1

SE $\exists x_r, 1 \leq r \leq k : J(\mathcal{X}_{k+1} - x_r) > J(\mathcal{X}_k)$ ENTÃO

$\mathcal{X}'_k \leftarrow \mathcal{X}_{k+1} - x_r$

Note que, neste ponto, $J(\mathcal{X}'_k) > J(\mathcal{X}_k)$

SE $k = 2$ ENTÃO

$\mathcal{X}_k \leftarrow \mathcal{X}'_k$

$J(\mathcal{X}_k) \leftarrow J(\mathcal{X}'_k)$

RETORNE AO PASSO 1

3:

$\mathcal{X}'_{k-1} \leftarrow \text{SBS}(\mathcal{X}'_k, k-1)$

SE $J(\mathcal{X}'_{k-1}) \leq J(\mathcal{X}_{k-1})$ ENTÃO

$\mathcal{X}_k \leftarrow \mathcal{X}'_k$

$J(\mathcal{X}_x) \leftarrow J(\mathcal{X}'_k)$

VÁ AO PASSO 1

SE $J(\mathcal{X}'_{k-1}) > J(\mathcal{X}_{k-1})$ ENTÃO

$k \leftarrow K - 1$

SE $k = 2$ ENTÃO
$\mathcal{X}_k \leftarrow \mathcal{X}'_k$
$J(\mathcal{X}_k) \leftarrow J(\mathcal{X}'_k)$
VÁ AO PASSO 1
SENÃO
REPITA O PASSO 3

Pode-se notar que a condição de parada é que $|\mathcal{X}_k| = m + \delta$, em que δ é um valor de tolerância que é utilizado para que o algoritmo não pare na primeira vez em que o conjunto $\mathcal{X}_k$ tenha tamanho m, pois o problema de *nesting* só pode ser evitado se forem realizados cálculos com $\mathcal{X}_{k+1}$. Normalmente utiliza-se um valor pequeno para δ (por exemplo, $\delta \leq 3$).

A versão *top-down* desse algoritmo (SFBS) é bastante análoga a esse, diferenciando-se somente na ordem em que os algoritmos SFS e SBS são executados e em alguns critérios de avaliação dos conjuntos. Obviamente, no SFBS, inicia-se com $k = N$.

Esses métodos proporcionam soluções muito próximas da solução ótima com um pequeno custo computacional. Segundo Jain et al. [Jain and Zongker, 1997, Jain et al., 2000], esses são os métodos que melhor combinam tempo de execução com qualidade dos resultados.

Métodos Adaptativos de Busca seqüencial flutuante [Somol et al., 1999] (ASFSM)

Os métodos adaptativos de busca seqüencial flutuante para frente e para trás (ASFFS e ASFBS) foram construídos como uma evolução dos métodos de busca seqüencial flutuante (SFSM) de forma a tornar o algoritmo generalizado, adicionando-se ou removendo-se tuplas de características, ao invés de características individuais.

Tomando-se o algoritmo SFFS como exemplo, podem-se notar que somente os passos *para trás* são condicionais e somente esses permitem que o conjunto de características de um determinado tamanho seja melhorado. Por outro lado, os passos *para frente* não podem ser condicionais, pois se eles fossem, o algoritmo poderia teoricamente cair em um ciclo infinito (repetindo a adição condicional e remoção condicional de características). Por não serem condicionais, os passos *para frente* podem encontrar um subconjunto que é pior que o melhor de uma certa dimensão encontrado em iterações anteriores.

Para eliminar esse problema, se o passo *para frente* encontrar um subconjunto que é pior que o melhor de todos encontrado em um passo anterior, deve-se descartar o subconjunto atual e considerar o melhor subconjunto como o conjunto atual. Essa troca *violenta* entre o conjunto atual e o melhor conjunto encontrado não proporciona um ciclo infinito, pois esse caso só ocorre quando o melhor conjunto de características foi encontrado em um passo *para trás*.

Os métodos ASFSM (adaptativos seqüenciais flutuantes) não são simples general-

izações dos métodos SFSM, pois, além de inserirem ou excluírem tuplas de características em seus passos, o tamanho dessas tuplas também é determinado dinamicamente. São realizados testes com tuplas de vários tamanhos para determinar-se a solução, mas, para limitar o tempo de execução do algoritmo, o usuário deve definir o tamanho máximo absoluto das tuplas, r_{max}. Para tornar o algoritmo mais eficiente, há um mecanismo que faz com que o tamanho das tuplas seja inversamente proporcional à distância entre o tamanho do conjunto sendo avaliado no passo atual (conjunto atual) e o tamanho final m. Assim, quando os conjuntos sendo avaliados são muito menores ou muito maiores que m, o ASFM é mais rápido, pois são inseridas ou excluídas tuplas menores de características. Com isso, o algoritmo chega mais rápido a um conjunto atual de tamanho próximo de m e vai aumentando a precisão da busca. Um outro parâmetro que deve ser definido pelo usuário é b, o qual é usado para determinar a relação entre o tamanho do conjunto atual e o tamanho máximo das tuplas. Assim, os parâmetros b, r_{max} e m são utilizados para determinar o tamanho máximo das tuplas para a busca no conjunto atual, sendo r o tamanho atual da tupla. O algoritmo a seguir descreve como r é calculado durante a execução do ASFSM:

```
SE |k − m| < b ENTÃO
   r ← r_max
SENÃO
   SE |k − m| < b + r_max ENTÃO
      r ← r_max + b − |k − m|
   SENÃO
      r ← 1
```

A determinação dos valores de b e r_{max} não é automática. Porém esses parâmetros não são tão críticos em relação à execução do método e de seus resultados quando comparados com os parâmetros o (tamanho das tuplas, no caso dos algoritmos generalizados tradicionais), l e r (no caso do método PTA). Uma característica importante desse método é que, se $r_{max} = 1$, ele é executado exatamente da mesma maneira que os métodos SFSM, o que faz com que a desigualdade a seguir seja sempre válida:

$$J(\mathcal{X}_m^{ASFM}) \geq J(\mathcal{X}_m^{SFSM}), \tag{3.34}$$

em que $\mathcal{X}_m^{ASFM}$ e $\mathcal{X}_m^{SFSM}$ são, respectivamente, o subconjunto obtido com o método ASFM e o subconjunto obtido com o método SFSM. Por outro lado, o limite inferior do tempo de execução do ASFM é igual ao tempo de execução do método SFSM. Quando o valor de b e r_{max} são grandes e quando N é grande e m possui um valor próximo de $N/2$, o tempo de execução do ASFSM pode ser muito grande se comparado com SFSM . Caso contrário, o tempo de execução é menor. Maiores detalhes sobre esse método podem ser encontrados em [Somol et al., 1999].

Na seção 5.2.1 e no artigo [Campos et al., 2000c], mostramos os testes e resultados obtidos da comparação desses dois métodos para um problema de seleção de características

com dados reais.

Recentemente, o grupo de pesquisa de Pudil (Academy of Sciences of the Czech Republic), criador dos métodos SFSM e ASFSM, propôs novos algoritmos de busca para seleção de características [Kittler et al., 2001]. Dentre eles, os principais métodos são os seguintes:

- **Busca oscilatória** [Somol and Pudil, 2000]: Esse método faz a busca sem que seja necessário definir um sentido (para frente ou para trás). A inicialização é feita com um conjunto de características de tamanho m, que é o tamanho do conjunto desejado. São executadas inserções e remoções de características para maximizar a função critério. Para isso, são utilizados outros métodos de busca, como os métodos seqüenciais, os flutuantes ou o método exaustivo. A escolha desses métodos depende da relação entre qualidade dos resultados e tempo de execução desejados. A busca pode ser restringida por um limite de tempo, caso o método seja aplicado a sistemas de tempo real. Os autores mostraram que, na maioria dos casos, os métodos de busca oscilatória proporcionaram resultados melhores que os outros métodos sub-ótimos existentes [Kittler et al., 2001].
- ***Fast Branch and Bound*** [Somol et al., 2000, Somol et al., 2001]: O algoritmo rápido de *branch and bound* baseia-se em um mecanismo de predição o qual permite que os mesmos resultados que o *branch and bound* sejam obtidos com um número menor de computações da função critério em nós internos da árvore. Informações sobre a contribuição individual das características são computadas durante a execução do algoritmo. A predição opera individualmente dependente de características particulares e do contexto da busca na árvore. Os experimentos dos autores [Kittler et al., 2001] mostraram que o tempo de execução desse algoritmo é menor que o de todas as outras versões do método *Branch and Bound* existentes.

Como esses métodos são muito recentes, eles não foram incorporados no conjunto de experimentos de seleção de características realizados no decorrer deste trabalho de mestrado.

3.3.3 Funções critério

Conforme mencionado anteriormente, uma das partes mais importantes na redução da dimencionalidade é a escolha de uma função critério. Em seleção de características, o objetivo das funções critério é minimizar o erro de classificação. Dessa forma, dado um conjunto de características $\mathcal{X}$, um exemplo de função critério é: $J(\mathcal{X}) = 1 - E(\mathcal{X})$, sendo E a probabilidade de erro de um classificador usando $\mathcal{X}$ como conjunto de características. Essa probabilidade de erro pode ser determinada através da taxa de acerto de um classificador ou da distância entre as classes de padrões de treinamento no espaço de características. A seguir, serão descritas suscintamente algumas funções critério conhecidas.

Desempenho de um Classificador

Um critério amplamente utilizado é o de erro de classificação com a utilização de um subconjunto de características. Basicamente, quando não se dispõe de informações a respeito da distribuição dos dados, utilizam-se os padrões de treinamento e de teste no espaço determinado pelo conjunto de características para avaliar um classificador. A taxa de acerto é utilizada como função critério, de forma que, quanto maior a taxa de reconhecimento, melhor é o conjunto de características.

Segundo [Kohn, 1998], deve-se tomar o cuidado de não empregar o conjunto de treinamento e de testes utilizado no processo de seleção de características (ou *projeto do classificador*) para estimar a probabilidade de erro do classificador após a seleção de características. Caso isso seja feito, o classificador estará ajustado especificamente para o conjunto padrões utilizado em seu projeto, e a estimativa da probabilidade de erro será muito otimista.

Outro ponto do qual se deve tomar cuidado é evitar o problema da dimensionalidade. Assim, é necessário que seja utilizado um conjunto de treinamento grande o suficiente para que a qualidade da estimativa da taxa de erro seja boa.

Basicamente, essa abordagem possui dois problemas. O primeiro é que o erro de classificação, por si só, não pode ser confiavelmente estimado quando a razão entre o tamanho do conjunto de exemplos e o do conjunto de características for pequena (vide seção 2.3). O segundo e principal problema dessa abordagem é que a escolha de um classificador é um problema por si só, e o subconjunto selecionado ao final claramente depende do classificador [Jain et al., 2000].

Nas seções 5.2.1 (publicada em [Campos et al., 2000c]) e 5.3.1 (com parte dos resultados publicados em [Campos and Cesar-Jr, 2001]), estão descritos experimentos de seleção de características utilizando funções critério baseadas em desempenho de classificadores.

Distâncias entre Classes

Visando a otimizar o conjunto de características para minimizar a probabilidade de erro independentemente de classificadores específicos, deve-se maximizar a distância entre padrões de classes diferentes no espaço de características.

Quando se dispõe de um conjunto de amostras treinamento para cada classe, pode-se supor que tal conjunto possui uma boa representação das mesmas e estimar a distância entre as classes. Considerando um espaço métrico Ω, uma **distância** ou métrica é uma função $d : \Omega \times \Omega \rightarrow \mathbb{R}^+$ que deve obedecer as seguintes condições [Lima, 1970]:

1. (a) $\forall \omega \in \Omega : d(\omega, \omega) = 0$;
   (b) $\forall \omega_i, \omega_j \in \Omega : d(\omega_i, \omega_j) = 0 \Rightarrow \omega_i = \omega_j$

2. $\forall \omega_i, \omega_j \in \Omega : d(\omega_i, \omega_j) = d(\omega_j, \omega_i)$;

3. $\forall \omega_i, \omega_j, \omega_l \in \Omega : d(\omega_i, \omega_j) \leq d(\omega_i, \omega_l) + d(\omega_j, \omega_l)$;

Há várias formas de medir-se a distância entre conjuntos de classes diferentes no espaço de características. Dentre elas, pode-se citar [Theodoridis and Koutroumbas, 1999, Kohn, 1998]:

- **Distância entre os centróides das classes**: Para calcular essa medida, basta determinar os centróides das classes e medir a distância entre eles.

- **Distância entre vizinhos mais próximos, mais distantes e média**: No cálculo dessas distâncias, devemos considerar, respectivamente, o mínimo, o máximo ou a média das distâncias entre os padrões de treinamento de duas classes diferentes;

- **Distâncias baseadas em matrizes de espalhamento**: Essas distâncias utilizam medidas de separabilidade baseadas em análise de discriminantes. Na seção 3.2.3 (equações 3.19 e 3.20), há uma breve descrição de matrizes de espalhamento.

- **Distância de Mahalanobis**: A distância de Mahalanobis (equação 2.17) pode ser utilizada para medir a distância entre classes de padrões. Isso pode ser feito através da soma ou da média da distância entre todos os padrões de duas classes diferentes.

- **Distância de Bhattacharyya e divergência**. Essas são distâncias baseadas nas funções densidade de probabilidade das classes, de forma que a distância espacial entre os conjuntos não é considerada, mas sim a diferença entre a forma deles.

- **Distâncias nebulosas**. As distâncias nebulosas são medidas que utilizam informações obtidas a partir da *fuzzyficação* dos conjuntos, como os suportes dos conjuntos e os coeficientes de pertinência dos padrões. Em [Bloch, 1999], há uma revisão bastante completa de distâncias nebulosas aplicadas a processamento de imagens. Em [Campos et al., 2001], foi utilizada uma distância nebulosa como função critério de um algoritmo de seleção de características. Os resultados obtidos com essa abordagem estão descritos na seção 3.4.

É importante lembrar que uma distância (ou métrica) é definida somente para entre dois elementos, ou seja, não se pode medir a distância entre três ou mais classes. Porém, na maioria dos problemas de reconhecimento de padrões reais, têm-se mais de duas classes. Por isso, ao efetuar seleção de características com base em alguma distância, é necessário definir uma função critério que possa avaliar a separabilidade entre todas as classes de uma maneira global. Para a maioria das distâncias citadas acima, isso pode ser feito através de operações simples como a soma, a média ou o ínfimo dos resultados obtidos para todos os pares de conjuntos (classes) existentes. Na seção 5.3 está descrita uma função critério para várias classes inspirada na distância descrita na seção 3.4. Maiores detalhes e informações sobre outras medidas de distância (ou métricas) podem ser encontrados em

[Kohn, 1998, Theodoridis and Koutroumbas, 1999, Duda and Hart, 1973, Bloch, 1999].

3.4 Método Proposto para Seleção de Características

Nesta seção, apresentamos uma das principais contribuições desta dissertação de mestrado. Trata-se de um trabalho que foi desenvolvido em cooperação com a pesquisadora Isabelle Bloch (Ecole Nationale Supérieure des Télécommunications - Paris) e publicado em [Campos et al., 2001]. Inicialmente será descrito o problema e introduzidos os conjuntos nebulosos, pois nossa abordagem se baseia em uma distância nebulosa. Posteriormente, nosso método de seleção, suas propriedades e os experimentos realizados com esse serão descritos.

3.4.1 Descrição do Problema

As medidas de distância entre agrupamentos ou classes geralmente utilizadas como função critério para seleção de característica são mais adequadas a conjuntos convexos, tendendo a privilegiar conjuntos linearmente separáveis (por exemplo, a distância de Mahalanobis). O problema é que, com esses critérios de distância, não é possível detectar bons agrupamentos côncavos ou com médias próximas, como o exemplo da figura 3.12, em que a distribuição dos padrões de uma classe se encontra no interior da de outra classe, embora as distribuições das classes não se interceptem. Nesse caso, mesmo que os dois agrupamentos estejam bem definidos, possibilitando a obtenção de boas taxas de reconhecimento com um classificador de K vizinhos mais próximos, dificilmente uma função critério comum baseada em distância identificaria o potencial desses agrupamentos.

Visando a evitar esse problema, criamos uma função critério baseada em uma medida de distância que, juntamente com o algoritmo de seleção de características, maximiza a distância entre padrões que pertencem a classes diferentes e minimiza a distância entre elementos que pertencem à mesma classe. Isso é feito independentemente da forma da distribuição dos padrões no espaço de características.

Após um estudo de diversas métricas entre conjuntos nebulosos, com base no artigo [Bloch, 1999], constatamos que uma medida que possui as propriedades desejadas é a distância[4] nebulosa baseada em tolerância, proposta em [Lowen and Peeters, 1998]. Nessa medida, a distância é determinada através de uma vizinhança em torno de cada padrão de treinamento.

[4]Conforme será explicado posteriormente, na realidade essa medida não é uma distância, é uma semi-pseudo-métrica.

3.4.2 Conjuntos Nebulosos

A lógica nebulosa foi criada com inspiração no comportamento humano, que se baseia na interpretação do mundo sem precisão e na descrição desse por atributos lingüísticos. Dessa forma, a relação de pertinência entre elementos e um conjunto nebuloso não é binária (pertence/não-pertence), mas assume um valor real.

Formalmente, seja F um espaço Cartesiano representando um espaço de características ou o espaço de imagens (usualmente $\mathbb{Z}^N$ ou $\mathbb{R}^N$); seja $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \cdots, \mathbf{x}_n$ variáveis espaciais, ou padrões no espaço F; um objeto *crisp* (não nebuloso) é, usualmente, um subconjunto de F. Um objeto nebuloso é definido bi-univocamente pela função de pertinência de um objeto *crisp*, denotada por ν. Uma função de pertinência que caracteriza um objeto nebuloso é portanto uma função $\nu : F \rightarrow [0, 1]$. Para cada $\mathbf{x}$ em F, $\nu_{\omega_i}(\mathbf{x})$ é um valor em $[0, 1]$ que representa o grau de pertinência de $\mathbf{x}$ ao conjunto nebuloso ω_i. Denotamos por C o conjunto de todos os conjuntos nebulosos definidos em F [Bloch, 1999].

As funções de pertinência podem ser criadas com base na relação entre cada elemento e o(s) suporte(s) dos conjuntos. Um suporte $p_i^{\omega_j}$ da classe ω_i é um ponto em F tal que $\nu_{\omega_i}(p_i^{\omega_j}) = 1$. Assim, os suportes de uma classe são os pontos mais típicos dessa. Um suporte pode ser determinado, por exemplo, pelo ponto médio da distribuição dos padrões de uma classe (caso haja somente um suporte por classe). Os suportes definem as regiões de uma classes que possuem maior tipicalidade, ou seja, regiões que são mais representativas dessa classe. Por isso, geralmente as funções de pertinência retornam valores maiores quanto maior a proximidade entre os padrões e os suportes de uma classe.

Maiores detalhes a respeito de conjuntos nebulosos, classificação com lógica *fuzzy* (nebulosa) e aplicações podem ser encontrados no livro [Dubois et al., 1997] e na tese [Bonventi-Jr. and Costa, 2000]. No presente trabalho, o uso de conjuntos nebulosos aplica-se à função critério utilizada em um algoritmo de seleção de características.

3.4.3 *Fuzzyficação*

Para se utilizar essa distância como função critério deve-se, inicialmente, transformar os conjuntos de treinamento em conjuntos nebulosos (*fuzzy*). Esse processo é chamado *fuzzyficação*. Seja um padrão $\mathbf{x}$ e uma classe de padrões ω, a função de *fuzzyficação* utilizada é definida como:

$$\nu_\omega(\mathbf{x}) = \begin{cases} \frac{1}{1+d(\mathbf{x},p_j^\omega)}, \mathbf{x} \in \omega, \\ 0, \mathbf{x} \notin \omega, \end{cases} \tag{3.35}$$

para $j = 1, 2, \cdots, \mathcal{P}$, em que $\mathbf{x}$ é um padrão, $\nu_\omega(\mathbf{x})$ é a função de pertinência desse padrão ao conjunto ω, p_j^ω representa o j-ésimo suporte da classe ω e $d(\cdot)$ é a distância Euclidiana, sendo $\mathcal{P}$ o número de suportes disponíveis para cada classe. Em nossos testes, foi utilizado

somente um suporte por classe, o qual foi definido pelo baricentro do conjunto ω_i. Dessa forma, a função de pertinência é inversamente proporcional à distância do padrão ao centróide de cada classe.

3.4.4 Semi-pseudo-métrica baseada em Tolerância

Para definir a distância fuzzy baseada em tolerância, inicialmente define-se uma distância local:

$$d_{\mathbf{x}}^{\tau}(\nu_{\omega_i}, \nu_{\omega_j}) = \inf_{\mathbf{y},\mathbf{z} \in B(\mathbf{x},\tau)} |\nu_{\omega_i}(\mathbf{y}) - \nu_{\omega_j}(\mathbf{z})|, \tag{3.36}$$

em que $B(\mathbf{x}, \tau)$ denota uma hiperesfera de dimenção N, com raio τ centrada em $\mathbf{x}$. Essa hiperesfera é chamada "bola". O parâmetro τ é chamado *tolerância* dessa distância. Assim, define-se a distância fuzzy baseada em tolerância por [Lowen and Peeters, 1998]:

$$d_p^{\tau}(\nu_{\omega_i}, \nu_{\omega_j}) = [\int_{\mathcal{F}} [d_{\mathbf{x}}^{\tau}(\nu_{\omega_i}, \nu_{\omega_j})]^p d\mathbf{x}]^{1/p}, \tag{3.37}$$

em que $\mathcal{F}$ representa todo o espaço de características.

Uma medida que não satisfaz a condição 1(b) mensionada na definição de distância (página 58) é chamada de pseudo-métrica [Lima, 1970]. Os criadores da medida descrita anteriormente chamam-na de *semi-psedo-métrica baseada em tolerância*, pois as condições 1(b) e 3, especificadas na definição de métrica podem falhar (ver [Lowen and Peeters, 1997] para maiores detalhes). Essa medida de distância, juntamente com o processo de fuzzificação descrito anteriormente, foram utilizados como uma função critério para efetuar seleção de características. Em nossos experimentos, utilizamos $p = 2$.

3.4.5 Algoritmo e complexidade

Para efetuar o cálculo dessa medida de distância, propusemos o seguinte algoritmo:

DistânciaFuzzy $(p, \tau, \nu_{\omega_m}, \nu_{\omega_n})$
$S \leftarrow 0$
$T \leftarrow \omega_m + \omega_n$
1:
para i de 1 até $|T|$ faça
Compute todos os padrões que pertencem a $B(\mathbf{x}_i, \tau)$ na estrutura de dados B_E
2:
para i de 1 até $|T|$ faça
$S \leftarrow S + [\text{DiferençaLocal}(\mathbf{x}_i, \tau, \nu_{\omega_m}, \nu_{\omega_n}, B_E)]^p$

RETORNE $S^{1/p}$

Sendo que a diferença local é calculada através do seguinte algoritmo:

DIFERENÇALOCAL($\mathbf{x}_i, \tau, \nu_{\omega_m}, \nu_{\omega_n}, B_E$)
$D_{min} \leftarrow$ MAIOR NÚMERO INTEIRO POSSÍVEL
$b \leftarrow$ NÚMERO DE PADRÕES NA BOLA $B(\mathbf{x}_i, \tau)$
1:
PARA i de 2 até b FAÇA
2:
PARA j de 1 até i FAÇA
$D \leftarrow |\nu_{\omega_m}(\mathbf{x}_i) - \nu_{\omega_n}(\mathbf{x}_j)|$

SE $D_{min} > D$ ENTÃO
$D_{min} \leftarrow D$

RETORNE D_{min}

Pode-se mostrar que a complexide da instrução 1 do algoritmo DISTÂNCIAFUZZY é de $O(|T|^2)$ e a complexidade da instrução 2 é de $O(|T|) \cdot O(\text{DIFERENÇALOCAL})$. Em relação ao algoritmo DIFERENÇALOCAL, a complexidade do laço 1 e 2 é de $O(b^2)$. Assim, supondo que $\forall \mathbf{x} \in T$ o número de padrões nas bolas $B(\mathbf{x}, \tau)$ é b e a complexidade do algoritmo DISTÂNCIAFUZZY é de $O(|T|^2) + O(|T|) \cdot O(b^2)$.

Assim, no **melhor caso** (em termos de tempo de execução), se τ for tão pequeno que $B(\mathbf{x}, \tau)$ contenha apenas $\mathbf{x}$, $\forall \mathbf{x} \in T$, a complexidade desse algoritmo será $O(|T|^2) + O(|T|) = O(|T|^2)$. No **pior caso**, se τ for tão grande que $B(\mathbf{x}, \tau)$ contenha todos os padrões de $|T|$, a complexidade desse algoritmo será $O(|T|^2) + O(|T|) \cdot O(|T|^2) = O(|T|^3)$.

3.4.6 Considerações Sobre o Comportamento da Função Critério

Nesta seção, serão discutidas as principais propriedades dessa abordagem, as quais nos motivaram a utiliza-la em seleção de características. Tais propriedades se relacionam com a distância entre os suportes (protótipos) das classes diferentes e com o quão os conjuntos são compactos (compacidade). Cada parâmetro das equações 3.36 e 3.37 será discutido isoladamente, sendo posteriormente analisados os resultados da integração desses parâmetros nessas equações. Para facilitar a ilustração dos casos, os resultados a serem mencionados em relação a compacidade são válidos para conjuntos (classes de padrões) com **distribuições aproximadamente isotrópicas**. As considerações a respeito da distância entre os protótipos também são válidas para conjuntos de padrões com **dis-**

tribuições normais. Posteriormente há uma discussão considerando casos genéricos.

1. **Compacidade.** Fixando-se a distância entre os protótipos de classes diferentes e o raio da bola τ, quando a distribuição de uma classe ω_i for compacta (possuir compacidade grande), para a maioria dos padrões $\mathbf{x}_i \in \omega_i$, os valores de $\nu_{\omega_i}(\mathbf{x}_i)$ serão grandes, pois o grau de pertinência de um padrão a sua classe é inversamente proporcional à distancia entre esse e o protótipo dessa classe. Caso contrário (quando a compacidade da classe for grande), os valores de $\nu_{\omega_i}(\mathbf{x}_i)$ serão pequenos para a maioria dos padrões $\mathbf{x}_i : \mathbf{x}_i \in \omega_i$.

2. **Distância entre os protótipos.** Seja ω_i e ω_j duas classes e $\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$ padrões com $\mathbf{y} \in \omega_i$ e $\mathbf{z} \in \omega_j$, $\mathbf{x} \in \omega_i \bigcup \omega_j$, fixando-se a compacidade da distribuição das classes de padrões e o raio da bola τ, quando a distância entre os protótipos de classes diferentes for grande, será mais provável que um dado padrão $\mathbf{x}$ esteja próximo do protótipo de uma classe e distante de outra. Sendo p^{ω_i} o protótipo que se encontra mais próximo do padrão $\mathbf{x}$ e p^{ω_j} o protótipo que se encontra mais distante do padrão $\mathbf{x}$, o valor de $\nu_{\omega_i}(\mathbf{y})$ será grande, e o valor de $\nu_{\omega_j}(\mathbf{z})$ será pequeno (para $\mathbf{y} \in \omega_i$ e $\mathbf{z} \in \omega_j$). Com isso, a diferença $|\nu_{\omega_i}(\mathbf{y}) - \nu_{\omega_j}(\mathbf{z})|$ será grande. Se isso ocorrer na maioria dos padrões dentro da bola $B(\mathbf{x}, \tau)$, o valor de $d^{\tau}_{\mathbf{x}}$ será grande. Como isso provavelmente ocorrerá para a maioria dos padrões, o valor total da distância $d^{\tau}_{p}(\nu_{\omega_i}, \nu_{\omega_j})$ será grande. Caso a distância entre os protótipos de classes diferentes seja pequena, seguindo o mesmo racionício, conclui-se que o valor de $d^{\tau}_{p}(\nu_{\omega_i}, \nu_{\omega_j})$ será pequeno.

3. **Tamanho da bola.** Fixando-se a distância entre os protótipos e a compacidade, devemos considerar dois casos:

   - Quando for utilizada uma bola muito pequena, para todos os padrões $\mathbf{x}$, a bola $B(\mathbf{x}, \tau)$ irá conter somente os padrões da classe de $\mathbf{x}$. Nesse caso, a seguinte igualdade será válida: $d^{\tau}_{\mathbf{x}} = \nu_{\omega_l}(\mathbf{x})$ (para $\mathbf{x} \in \omega_l$, ω_l podendo ser ω_i ou ω_j). Com isso,

   $$d^{\tau}_{p}(\nu_{\omega_i}, \nu_{\omega_j}) = [\int_{\mathcal{F}} [\nu_{\omega_l}(\mathbf{x})]^p d\mathbf{x}]^{1/p}, \tag{3.38}$$

   o que significa que o valor de $d^{\tau}_{p}(\nu_{\omega_i}, \nu_{\omega_j})$ será exclusivamente dependente da compacidade das classes.

   - Quando for utilizada uma bola muito grande, para qualquer padrão $\mathbf{x}$, $B(\mathbf{x}, \tau)$ conterá todos os padrões de treinamento do espaço de características. Com isso, pode-se mostrar que a seguinte igualdade se torna válida:

   $$d^{\tau}_{p}(\nu_{\omega_i}, \nu_{\omega_j}) = [|T| \cdot [\inf_{\mathbf{y},\mathbf{z} \in F} |\nu_{\omega_i}(\mathbf{y}) - \nu_{\omega_j}(\mathbf{z})|]^p]^{1/p} \tag{3.39}$$

   Como resultado, a importância da compacidade e da distância entre os protótipos é reduzida, pois o valor da métrica dependerá exclusivamente da mínima

diferença global entre o grau de pertinência de dois padrões de classes diferentes. Assim, não importando a distribuição dos padrões no espaço de características, se existirem dois padrões $\mathbf{y}$ e $\mathbf{z}$ tais que $\nu_{\omega_i}(\mathbf{y}) = \nu_{\omega_j}(\mathbf{z})$, então teremos em $d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = 0$.

Por isso, a determinação do valor de τ é muito importante na utilização da distância de [Lowen and Peeters, 1998] como função critério. Para determinar o melhor valor de τ para um dado conjunto de padrões de treinamento, uma estratégia possível é a de tentativa e erro com vários valores diferentes de τ, sendo que o valor máximo deve ser menor que $\sup_{\mathbf{y},\mathbf{z}\in F} d_E(\mathbf{y}, \mathbf{z})$. Na seção 5.3, estão descritos experimentos de seleção de características com variação no tamanho da bola.

Considerando a utilização de uma bola cujo tamanho seja ideal para avaliar um determinado conjunto de características com um certo conjunto de treinamento de duas classes, podemos construir uma lista de possibilidades, denotando por $d_{1a}, d_{1b}, d_{2a}, d_{2b}, d_{3a}, d_{3b}$ seus prováveis resultados. A relação entre os resultados será comentada posteriormente.

1. Ambas as classes são compactas e...
   (a) a distância entre os protótipos é pequena $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{1a}$
   (b) a distância entre os protótipos é grande $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{1b}$
2. Ambas as classes são esparsas e...
   (a) a distância entre os protótipos é pequena $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{2a}$
   (b) a distância entre os protótipos é grande $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{2b}$
3. Uma classe possui compacidade grande e a outra possui compacidade pequena e...
   (a) a distância entre os protótipos é pequena $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{3a}$
   (b) a distância entre os protótipos é grande $\Rightarrow d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j}) = d_{3b}$

A figura 3.15 ilustra esses casos. Considerando que as duas classes possuem distribuições aproximadamente isotrópicas e que a bola $B(\mathbf{x}, \tau)$ possui tamanho ideal. Podemos afirmar que, intuitivamente, é mais provável que a distância d_{1b} será maior que todas as outras. Da mesma forma, podemos dizer que as distâncias d_{2a} e d_{1a} provavelmente serão as maiores distâncias e que a distância d_{3a} provavelmente será menor que d_{2b} que, por sua vez, provavelmente será menor que d_{3b} .

Essas estimativas resultam da análise dos casos considerando as propriedades citadas anteriormente.

Caso as distribuições dos conjuntos (classes de padrões) com **distribuições não normais**, convexas ou com formas mais "complicadas", torna-se mais difícil realizar uma

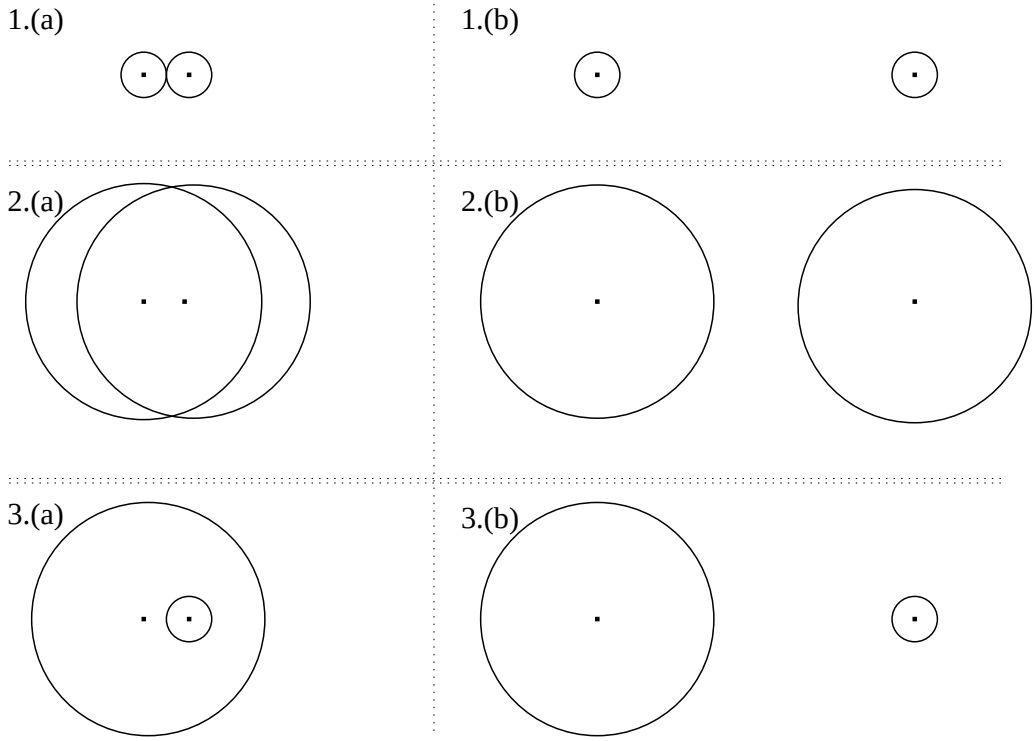

Figura 3.15: Exemplos de distribuições de duas classes em um espaço de características com dimensão 2. Cada círculo representa a compacidade de uma classe e os pontos representam protótipos.

estimativa dos resultados dessa função critério. Porém pode-se dizer que a influência do número de padrões de classes diferentes que a bola $B(\mathbf{x}, \tau)$ engloba, para diferentes $\mathbf{x}$, tem mais importância que a distância entre os protótipos. A bola $B(\mathbf{x}, \tau)$ serve como uma medida de sobreposição das distribuições das classes no espaço de características. Se duas classes estiverem muito sobrepostas, o valor da função critério será pequeno. A seguir, mostramos resultados que ilustram esse fato.

3.4.7 Experimentos de Seleção de Características com Dados Artificiais

Para avaliar o desempenho dessa função critério para seleção de características, realizamos testes com os métodos de busca SFSM [Pudil et al., 1994] em dados artificiais. O método SFSM foi escolhido devido a sua velocidade, visto que realizamos testes com o método ASFSM e constatamos que a diferença entre qualidade dos conjuntos de características obtidos após a seleção com os métodos adaptativos (ASFSM) e não-adaptativos (SFSM) não compensa a diferença de tempo de execução entre esses dois métodos (vide seção 5.2.1 e o artigo [Campos et al., 2000c]).

Comparamos a função critério que utiliza a distância nebulosa baseada em tolerância com o desempenho de um classificador de mínima distância ao protótipo. Para avaliar o desempenho dos conjuntos de dados, utilizamos dois classificadores: k-vizinhos mais próximos e o de mínima distância ao protótipo.

Esse algoritmo foi testado 100 vezes (cem experimentos de seleção) em um conjunto de dados artificiais de seis dimensões, duas classes, com 100 exemplos por classe. Segue a descrição da distribuição das duas classes nesse espaço de características:

- **Características 1 e 2.** Nessas características, os padrões possuem distribuições Gaussianas com médias diferentes (vide figura 3.16). Note que, nessas dimensões, ambos os conjuntos possuem compacidade grande e distância entre os protótipos gande.

- **Características 3 e 4.** Nessas características há distribuições ruidosas (vide figura 3.17). Pode-se dizer que nessas dimensões ambas as classes possuem compacidade grande e a distância entre os protótipos é pequena ou nula.

- **Características 5 e 6.** Nessas características, a classe ω_j possui distribuição Gaussiana "dentro" da classe ω_i, a qual é gerada como uma mistura de 4 Gaussianas formando um anel (vide figura 3.18). Note que nessas dimensões, a classe ω_j possui compacidade pequena, e a classe ω_i possui compacidade grande, e a distância entre os protótipos é muito pequena (podendo ser nula em algumas realizações dos padrões).

Nas figuras 3.16, 3.17 e 3.18, os padrões da classe ω_i são representados por asteriscos (*), e os da classe ω_j são representados por círculos (o). Para criar tais figuras, foram gerados aleatoriamente 100 padrões por classe seguindo as distribuições descritas. Visando a facilitar a visualização dos resultados de seleção de características, eferuamos a redução para obter um espaço de dimensionalidade 2.

Antes de realizar a seleção de características, normalizamos os dados de forma que todos os padrões do espaço de treinamento ficassem com média 0 e variância unitária em relação a todas as características. Isso é importante para evitar problemas com os classificadores e também com a função critério, pois esses utilizam a distância Euclidiana para efetuar medições [Belhumeur et al., 1997, Theodoridis and Koutroumbas, 1999]. Além disso, com a normalização dos padrões no espaço de características, a tarefa de determinar o tamanho ideal da bola torna-se mais simples. Em nossos experimentos, utilizamos uma bola de raio $\tau = 0.5$. As figuras 3.16, 3.17 e 3.18, mostram exemplos dos dados com essa normalização já realizada.

Conforme mensionado anteriormente, para avaliar os resultados, foram geradas amostragens com 100 padrões para cada classe com as distribuições descritas anteriormente. Essas amostragens foram geradas 100 vezes. Assim, o total de padrões gerados foi 20000, e foi

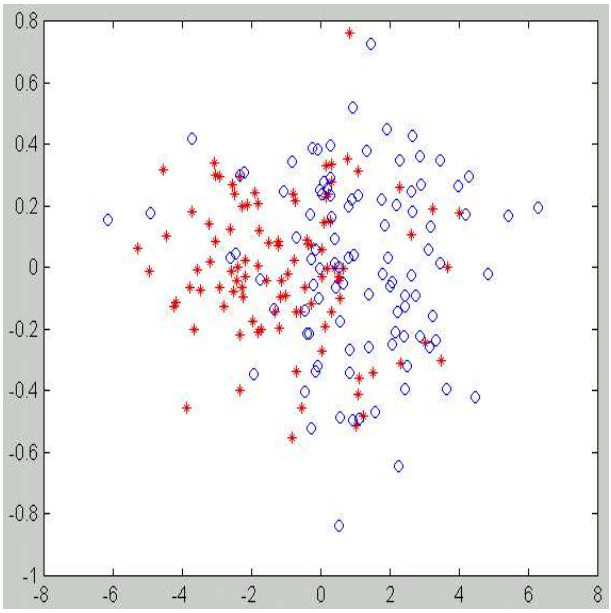

Figura 3.16: Amostragem dos dados artificiais utilizados em [Campos et al., 2001] nas características 1 e 2.

realizado um total de 100 experimentos de seleção de características. A distância foi utilizada com $\tau = 0.5$ e o classificador de k-vizinhos com $k = 3$. As taxas de acerto foram calculadas pela média nos 100 experimentos.

3.4.8 Resultados com os Dados Artificiais

Conforme esperado, o seletor de características baseado em distância nebulosa selecionou as características 5 e 6 em todos os 100 esperimentos. Já o algoritmo de seleção com o desempenho do classificador freqüentemente selecionou as características 1 e 2, mas várias outras combinações de características também foram selecionadas. A tabela 3.1 detalha quantas vezes cada par de características foram selecionados quando foi utilizado o desempenho do classificador como função critério.

Na tabela 3.2 estão as métias da taxa de acerto dos classificadores utilizando os conjuntos de características selecionados. Nos experimentos com os classificadores sem interseção entre o conjunto de treinamento e o de testes, foram utilizados 67 padrões no treinamento e 33 na fase de testes.

Para fornecer informações mais precisas sobre os resultados obtidos, criamos a tabela 3.3. Nessa tabela, é mostrado o desvio padrão dos resultados de classificação obtidos em nossos testes.

Tabela 3.1: Características selecionadas utilizando o desempenho do classificador como função critério.

Características		Freqüência
1	2	13
1	3	5
1	4	4
1	5	6
2	5	6
3	5	10
4	5	6
5	6	50

Tabela 3.2: Porcentagem de classificação correta dos dois classificadores usando o conjunto de características selecionado com os dois critérios após 100 experimentos de seleção de características.

	DP^1	DP^2	Knn^1	Knn^2
CR	63.15 %	83.71 %	95.56 %	89.47 %
FD	63.43 %	81.26 %	100.00 %	95.07 %

A notação utilizada se encontra na tabela 3.4.

Tabela 3.3: Desvio padrão dos resultados mostrados na tabela 3.2.

	DP^1	DP^2	Knn^1	Knn^2
CR	8.40 %	8.69 %	6.67 %	11.25 %
FD	7.46 %	10.47 %	0.05 %	3.14 %

A notação utilizada se encontra na tabela 3.4.

Tabela 3.4: Notação utilizada nas tabelas 3.2 e 3.3.

DP: classificador de distância ao protótipo
Knn: classificador dos K vizinhos mais próximos
[1]: $\alpha = \beta$
[2]: $\alpha \bigcap \beta = \emptyset$, $|\alpha| = 2|\beta|$, para:
• α: conjunto de treinamento
• β: conjunto de testes
CR: função critério baseada na taxa de classificações corretas
FD: função critério baseada na distância nebulosa de [Lowen and Peeters, 1997].

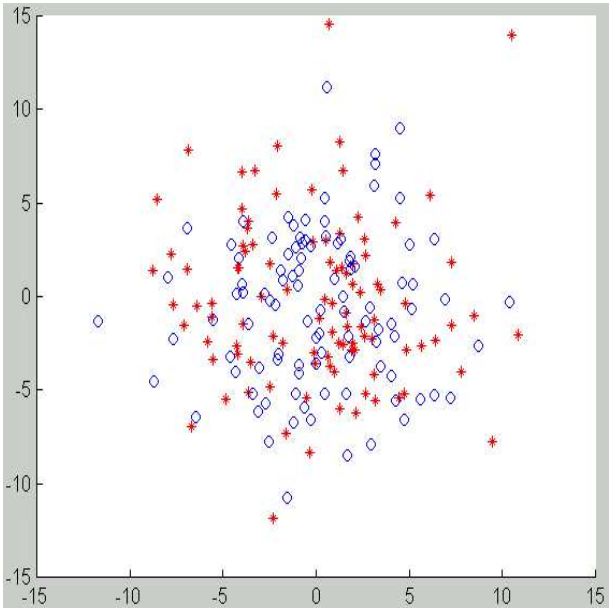

Figura 3.17: Amostragem dos dados artificiais utilizados em [Campos et al., 2001] nas características 3 e 4.

3.4.9 Discussão

Analisando-se a tabela 3.1, notamos que ao utilizar-se esse classificador como função critério, freqüentemente foram selecionadas as características 5 e 6. Isso parece um fato inesperado, já que nesses características as duas classes possuem a mesma média. Porém, há dois fatores que contribuem para isso: o fato da classe ω_j ser muito compacta e o fato de que esse classificador ter sido treinado com 2/3 dos padrões e testado com os 1/3 restantes. Como os padrões foram gerados aleatoriamente, muitas vezes o protótipo das duas classes não coincidem, com isso, o classificador cria uma fronteira de decisão que acaba propiciando um bom resultado, já que a grande maioria dos padrões da classe ω_j fica concentrada a um dos lados da fronteira de decisão. De qualuer forma, os resultados obtidos por esse classificador ao utilizar as características 5 e 6 são um tanto aleatórios.

Os resultados de classificação obtidos (tabelas 3.2, 3.3 e 3.4) mostram que a distância nebulosa baseada em tolerância permite a obtenção de um bom desempenho para conjuntos côncavos ou com conjuntos apresentando sobreposição entre classes diferentes. Após uma analise desses resultados, a seguinte questão pode ser levantada:

> *Observa-se que foi utilizado, como suporte de cada classe, um único protótipo. Isso é surpreendente na medida em que, para as características 5 e 6, temos uma classe "dentro" da outra. Intuitivamente seria, neste caso, mais apropriado utilizar um suporte maior para a classe circundante. Como explicar então que, apesar de terse escolhido um suporte pontual para as duas classes, os resultados parecem satis-*

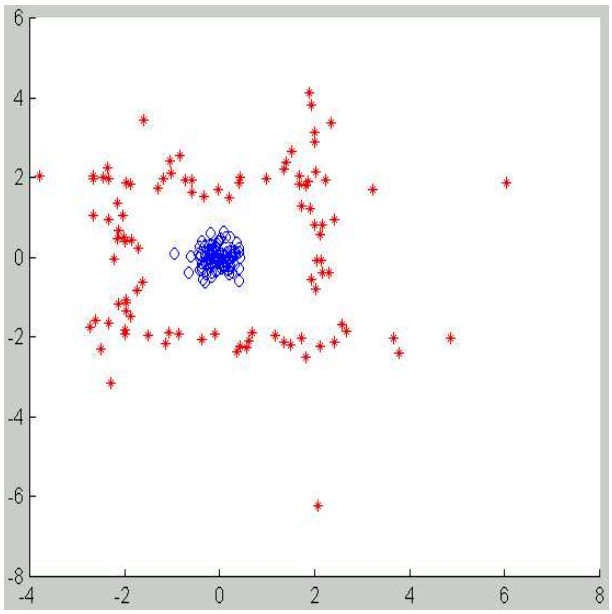

Figura 3.18: Amostragem dos dados artificiais utilizados em [Campos et al., 2001] nas características 3 e 4.

fatórios?[5]

Conforme mencionado anteriormente (e também em [Campos et al., 2001]), a distância nebulosa utilizada é calculada de tal forma que seja considerada uma bola em torno de cada padrão dos conjuntos em que tal distância está sendo medida (vide equações 3.36 e 3.37). É calculado o ínfimo da diferença entre o grau de pertinência de todos os pares de padrões que se encontram nessa vizinhança. O resultado da distância entre dois conjuntos nebulosos é dado pelo somatório dos resultados obtidos para todas as vizinhanças existentes (há uma vizinhança para cada elemento dos conjuntos). A figura 3.19 ilustra a vizinhança nas características 5 e 6 mencionadas em [Campos et al., 2001], sendo que a região clara representa a distribuição da classe ω_i, enquanto a região escura representa a distribuição na classe ω_j.

Por isso, a influência do processo de "fuzzyficação" e do suporte dos conjuntos nebulosos no resultado final da distância não é tão grande quanto a influência das áreas em que há sobreposição entre a distribuição dos padrões de classes diferentes, ou seja, é dada mais importância às áreas de sobreposição do que à forma dos aglomerados. Conforme mencionado anteriormente, dizemos que um conjunto de distribuições de classes possui regiões de "sobreposição" quando existem $\mathbf{x}$ tais que bola $B(\mathbf{x}, \tau)$ engloba padrões de mais de uma classe. Por isso, nas características 5 e 6, o resultado final da métrica entre a classe ω_i e a classe ω_j foi maior que nas características 1 e 2, mesmo com os suportes desses dois

[5]Agradeço ao assessor da FAPESP por levantar essa questão.

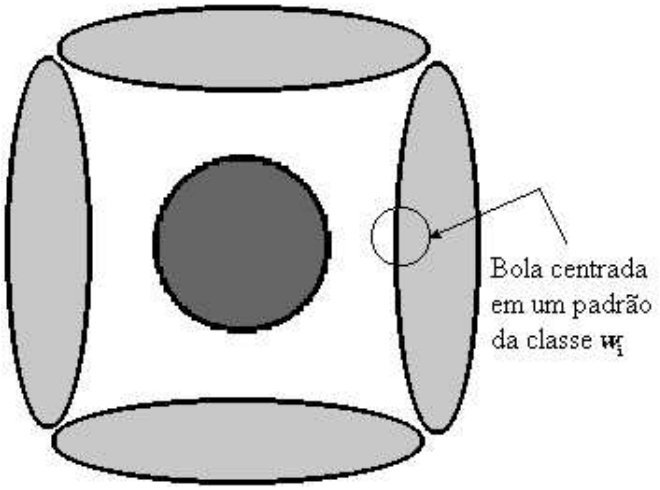

Figura 3.19: Cálculo da diferença local (equação 3.36) em um padrão da classe ω_i nas características 5 e 6.

conjuntos nebulosos encontrando-se tão próximos nas características 5 e 6. A figura 3.20 mostra a região de sobreposição existente entre as duas classes nas características 1 e 2.

Figura 3.20: Região de sobreposição entre as duas classes nas características 1 e 2.

Na seção 5.3, está descrita uma nova função critério inspirada na distância nebulosa baseada em tolerância. Essa função mede a separação entre mais de duas classes de padrões sem precisar associar o resultado de medições entre todos os pares possíveis de classes. Também foram realizados testes com dados reais com o objetivo de aperfeiçoar um método de reconhecimento de faces.

Parte II

Reconhecimento de Faces

Capítulo 4

Revisão de Reconhecimento de Faces

Conforme dito no capítulo 1, devido à idade da pesquisa em reconhecimento de faces e à importância dessas pesquisas, esse problema foi amplamente abordado por vários cientistas, utilizando técnicas muito distintas. Nesta seção, inicialmente serão introduzidas as tarefas básicas de identificação de faces (seção 4.1). Posteriormente (seção 4.2), será feita uma revisão geral dos métodos mais conhecidos de extração automática de características faciais para reconhecimento, principalmente os que se baseiam em imagens frontais.

4.1 Tarefas de Identificação de Faces

Consideremos uma base de dados que consiste em um conjunto de treinamento T, de faces de c pessoas conhecidas, sendo Ω o conjunto de todas as classes (ou pessoas) existentes e $\omega_i, \omega_2, \cdots, \omega_c$ classes de padrões (pessoas). Consideremos também que $\mathbf{x}$ é um padrão originário de uma face cuja classificação é desconhecida. De acordo com [Gong et al., 2000], no mínimo quatro tarefas relacionadas com identificação podem ser visadas:

1. **Classificação**: Consiste na identificação de uma face $\mathbf{x}$ assumindo-se que ela é de uma pessoa do conjunto Ω. Em outras palavras, assumindo-se que $\mathbf{x}$ pode ser classificado como um padrão de alguma classe ω_i, tal que $\omega_i \in \Omega$, a tarefa de classificação consiste em determinar o valor de i.

2. **Conhecido-desconhecido**: Objetiva decidir se a face é ou não um membro de Ω, ou seja, se $\mathbf{x}$ pode ser classificado como um padrão de alguma classe de Ω.

3. **Verificação**: Dado que a identidade ω_i de uma face $\mathbf{x}$ foi determinada através de um outro meio não visual, essa tarefa busca confirmar a identidade dessa pessoa usando imagens de face, ou seja, confirmar se $\mathbf{x}$ é da classe ω_i. Isso equivale à tarefa "conhecido-desconhecido" com $c = 1$.

4. **Reconhecimento Completo**: Visa determinar se uma face é de uma classe de Ω e, em caso positivo, determinar sua identidade ω_i.

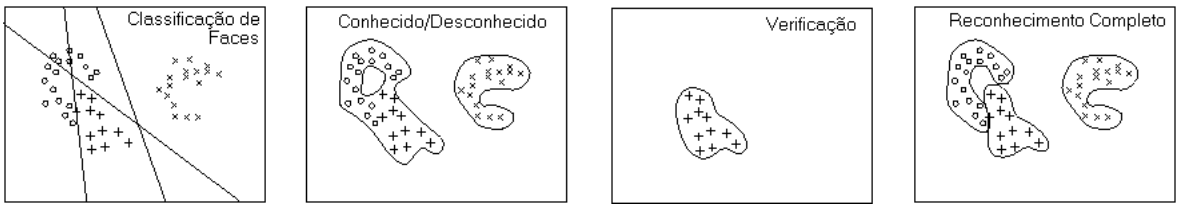

Figura 4.1: Imagens de três faces diferentes mostradas em um espaço de faces hipotético. São mostrados bons exemplos de fronteiras de decisão para cada tarefa de identificação de faces (baseadas em [McKenna et al., 1997]).

A figura 4.1 ilustra as possíveis fronteiras de decisão geradas por classificadores para executar as quatro tarefas de identificação em um espaço hipotético de faces F com três classes (pessoas), onde se assume que F contém todas as possíveis imagens de faces e exclui todas as outras imagens[1]. A separabilidade das identidades em F dependem da técnica utilizadas para modelar F. Na tarefa de classificação, todas as c classes podem ser modeladas. Por outro lado, as outras três tarefas sofrem a necessidade de considerar uma classe adicional, contendo as "faces desconhecidas". Provavelmente, este é o motivo pelo qual a tarefa de classificação de faces seja a mais popular na realização de testes de algoritmos de extração de características faciais para reconhecimento. Maiores detalhes sobre essas tarefas de identificação podem ser encontradas em [McKenna et al., 1997].

Outro problema relacionado é o de categorização de faces, que trata da classificação das pessoas em categorias discriminando, por exemplo, gênero [Valentin et al., 1996], faixa etária e etnia. Nesse caso, as classes representam as categorias a que as pessoa pertencem (e não a identidade de cada indivíduo). Essa tarefa equivale a "classificação", com c representando o número de categorias do problema abordado. Como este trabalho é centrado no problema de reconhecimento, não serão detalhadas técnicas de categorização.

Além dessa tarefa, há também o *reconhecimento de expressões faciais*. Para tal, alguns autores utilizaram métodos que exploram especificamente parâmetros que são influenciados por alterações da forma da boca, dos olhos e do contorno da face relacionadas com expressões faciais, utilizando, por exemplo, fluxo óptico. Vários autores utilizaram métodos muito similares ao de categorização de faces, em que o treinamento e a classi-

[1]Maiores detalhes sobre fronteiras de decisão se encontram no capítulo 2

ficação são efetuados de forma que cada classe (ou categoria) represente um tipo diferente de expressão facial.

Além dessa tarefa, há a de distinguir entre imagens de faces e imagens de outros objetos é um outro problema de duas classes (faces e não-faces) pertencente ao escopo de detecção de faces.

É importante lembrar que este trabalho se restringe a classificação de faces. Portanto, em todos os testes realizados, foi suposto que as imagens de teste eram de pessoas "conhecidas" pelo classificador, ou seja, pessoas que tinham ao menos uma imagem de suas faces no conjunto de treinamento dos classificadores.

4.2 Métodos de Reconhecimento de Faces

Através da revisão de Chellappa et al. [Chellappa et al., 1995], publicada em 1995 e de outros trabalhos propostos após essa publicação, tais como [Pentland, 2000], [Yachida, 1998], [Essa, 1996], [Crowley, 2000], [Bichsel, 1995], [Turk, 1998] e do livro [Gong et al., 2000], pode-se agrupar os métodos de reconhecimento de face nas seguintes categorias: por atributos (locais), holísticas, baseadas na transformada de Gabor, tridimensionais e de seqüências de vídeo. A seguir, essas abordagens serão comentadas. O que mais difere entre elas são os métodos de extração de características das imagens de faces, e não os métodos de classificação que, em sua maioria, são redes neurais ou métodos estatísticos. Por isso, pouco será dito a respeito dos métodos de classificação empregados pelos autores.

Por atributos

A abordagem de reconhecimento de faces por atributos é bastante intuitiva. Alguns dos métodos dessa categoria baseiam-se na construção de um vetor de características a partir de medidas de distância e ângulos entre pontos característicos da face, como cantos dos lábios, centro da boca, nariz, narinas, pupilas, pontos extremos dos olhos, sobrancelhas, orelhas, pontos do contorno do queixo, etc. e combinações dessas medidas [Cox et al., 1995].

Esses métodos também são aplicados a imagens de faces de perfil, em que os pontos característicos são, por exemplo, a ponta do nariz, o ponto entre os lábios, a sobrancelha, a testa, o queixo e o pescoço (vide figura 4.2). A vantagem do reconhecimento a partir de perfis está no uso de informações que não ficam disponíveis em imagens frontais bidimensionais, como o tamanho do nariz e do papo, além do fato de não ser difícil extrair esses pontos quando o fundo é uniforme.

O reconhecimento de faces por atributos, em geral, é feito através do uso de um sistema de reconhecimento de padrões, podendo ser, por exemplo, estatístico ou por redes neurais.

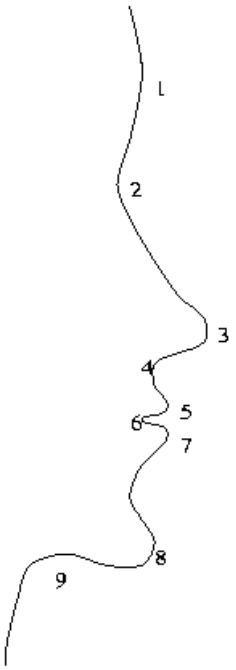

Figura 4.2: Exemplos de pontos importantes para o reconhecimento a partir de imagens de perfil.

Os vetores obtidos são usados para formar o espaço de características. Com a ausência de detectores automáticos de pontos característicos da face, muitos autores utilizaram operadores humanos nos trabalhos primordiais.

Alguns dos métodos automáticos de extração desses pontos baseiam-se em contornos. Também podem ser considerados como abordagens locais os sistemas de reconhecimento que utilizam informações como projeções horizontais dos mapas de bordas binários (verticais e horizontais) de imagens de face ou de partes da face, como nariz e boca [Brunelli and Poggio, 1993]. É muito comum utilizar o reconhecimento por atributos para imagens de pessoas de perfil. O uso de energia de curvatura do contorno do perfil como forma de extração do vetor de características também é uma abordagem baseada em atributos.

A vantagem de abordagens locais é a invariância à translação, à escala e à rotação no plano (caso sejam efetuadas normalizações), e a desvantagem está nos problemas ocasionados com alterações devido a expressões faciais e a rotações em profundidade.

Métodos holísticos

Os métodos holísticos consideram todos os *pixels* da imagem ou de regiões características da face. Nessa abordagem, a dimensionalidade dos dados é igual ao número de pixels das imagens consideradas. Para evitar o problema da dimensionalidade, podem ser uti-

lizados métodos estatísticos de redução de dimensionalidade, como, por exemplo, Análise dos Componentes Principais (PCA) [Turk and Pentland, 1991], Discriminantes Lineares [Belhumeur et al., 1997] e Redes Neurais [Lawrence et al., 1996, Romdhani, 1996]. O método de reconhecimento com PCA é o mais popular, tendo sido freqüentemente utilizado em associação com pré-processamentos de normalização de imagens para melhorar o desempenho. A classificação pode ser feita de diversas maneiras, geralmente através de redes neurais ou sistemas estatísticos. Métodos baseados em pirâmides (como *Wavelets* [Castleman, 1996]) para reconhecimento de faces considerando toda a imagem (sem modular por regiões como é feito nos métodos descritos na próxima seção) também podem ser classificados como holísticos.

Em [Brunelli and Poggio, 1993], é feito um estudo comparando o desempenho de métodos por atributos (locais) com um método holístico. O método holístico testado foi o de *template matching* (casamento), testando o desempenho de imagens de olhos, nariz, boca e também de toda a face (vide figura 4.3). Os métodos holísticos proporcionaram resultados melhores que os locais. A vantagem de abordagens globais está no fato de que pequenas variações locais não prejudicam muito o reconhecimento. A principal desvantagem está nos problemas de variação de iluminação e, em alguns casos, no custo computacional.

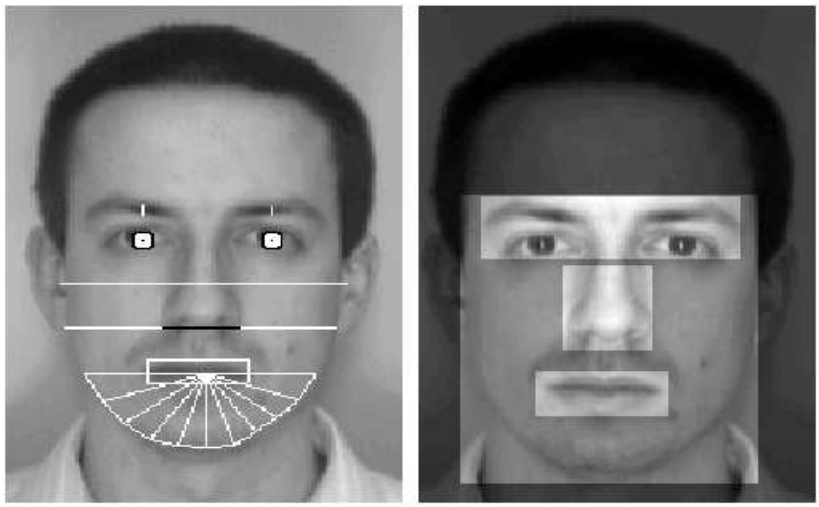

Figura 4.3: Atributos utilizados para extração de características locais e *templates* testados (abordagem local) baseada em [Brunelli and Poggio, 1993].

Técnicas baseadas na Transformada de Gabor

Neste item serão comentadas abordagens baseadas em extratores de características que utilizam transformada de Gabor em regiões da imagem, e não na imagem toda, o que diferencia essa abordagem das abordagens holísticas. Esses métodos são um pouco mais

recentes e muito promissores. A mais conhecida é uma abordagem local que utiliza *jets*. Um *jet* é um vetor em que cada posição é determinada através de uma transformada de Gabor bidimensional com a janela Gaussiana (modulada por uma exponencial complexa) em um determinado local da imagem, como, por exemplo, o centro do nariz, as pupilas, os cantos da boca etc.. Cada variável de um vetor *jet* é determinada pelo cálculo da transformada com uma janela de escala (variância da Gaussiana), orientação e/ou freqüência diferente.

Na fase de treinamento, cada *jet* é calculado em uma posição diferente de uma imagem (modelo) de treinamento. Na abordagem de *Elastic Graph Matching* [Wiskott et al., 1997, Wiskott et al., 1995, Lades et al., 1993], os *jets* são tratados como nós e as ligações entre os nós são arestas, formando um grafo. No primeiro quadro da figura 4.4, é mostrada a posição inicial dos *jets* numa imagem, bem como a disposição do grafo. Nos outros quadros, é mostrada a topologia dos grafos após o casamento com a imagem da face com variações de expressão facial, escala e rotação.

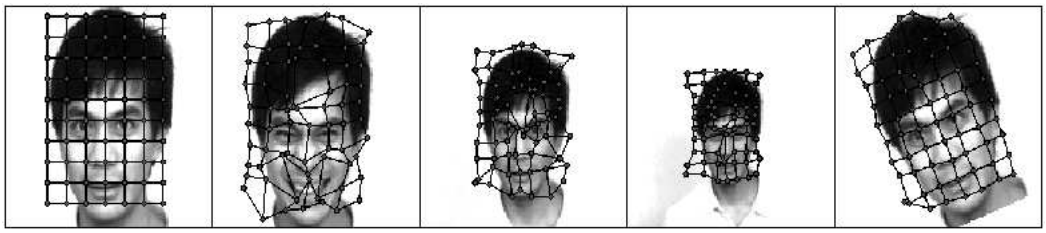

Figura 4.4: Elastic Graph Matching.

Antes de efetuar a classificação, é necessário realizar um processo de combinação com imagens, que se refere ao posicionamento dos *jets* em determinados pontos da imagem. Para isso, um procedimento tenta localizar a posição para cada nó do grafo que, ao mesmo tempo, maximiza a similaridade das características e minimiza o custo da topografia. Esse processo é feito para as imagens de treinamento e de testes, de forma que os pontos importantes da face são localizados e os *jets* são calculados nessas posições. As imagens de teste são classificadas de acordo com a similaridade com as imagens de treinamento, com base na combinação dos grafos e os *jets* obtidos. Essa abordagem pode apresentar problemas com variações na iluminação e com a imagem do fundo.

Uma outra abordagem que também utiliza a transformada de Gabor é a de *Gabor Wavelet Networks* [Krüger and Sommer, 2000], que consiste em uma técnica que representa um modelo discreto da face como a combinação linear de funções da wavelet bidimensional de Gabor. Essa abordagem é bastante recente e mostra-se muito eficiente e precisa, principalmente no rastreamento de faces e de pontos característicos da face em movimento [Feris, 2001]. Mas as *Gabor Wavelet Networks* também podem ser utilizadas para detecção e reconhecimento de faces. Ao centro da figura 4.5 é ilustrada a representaçao da face da esquerda no espaço de GWN, cujas Wavelets otimizadas se encontram nas posições demarcadas pelos pontos pretos na imagem da direita.

Figura 4.5: Gabor Wavelet Networks (obtida de [Feris, 2001]).

Métodos tridimensionais

Há vários métodos de reconhecimento baseados em informações tridimensionais, como os que utilizam *range images* (figura 4.6) e os baseados em visão estéreo. A vantagem desses métodos está na possibilidade de obtenção de informações relevantes da cena real que não podem ser obtidas através de imagens bidimencionais, pois combinam informações da profundidade das superfícies e também da textura. A desvantagem está na necessidade de sistemas de aquisição menos usuais, como um *scanner* tridimensional (para *range images*) e de uma câmera extra (para visão estéreo), além do fato de o custo computacional para tratar tais informações ser maior. A pesquisa em métodos tridimensionais para reconhecimento de faces não avançou muito nos últimos anos pois, o uso de sistemas bidimensionais baseados em treinamento com múltiplos pontos de vista (ou várias orientações da face) [Moghaddam and Pentland, 1994], bem como sistemas de rastreamento e determinação da orientação tridimensional a partir de imagens bidimensionais [Cascia and Sclaroff, 1999], possibilitam a obtenção de bons resultados dispensando a utilização de tais sistemas.

Vídeo

Até 1995, a pesquisa em sistemas de reconhecimento de pessoas baseada em seqüências de vídeo estava começando a se desenvolver. Haviam sido realizados alguns testes com seqüências de vídeo com a finalidade exclusiva de mostrar vários resultados de classificação no decorrer do tempo. Em [Yacoob et al., 1995], para cada quadro, foi feito um teste de classificação individual visando comparar o método baseado em *eigenfaces* com o de combinação de grafos (*elastic graph matching*) em seqüências com variação de expressão facial. Além disso, alguns pesquisadores empregavam seqüências temporais para integrar as informações estáticas da câmera com sons de sílabas. Esses sistemas possuíam dois módulos: o de vídeo, o qual capturava apenas um quadro para efetuar o reconhecimento empregando algum método comum; e o de áudio, que normalmente utiliza informações da freqüência da voz do indivíduo para efetuar o reconhecimento. Os resultados de classificação eram combinados utilizando um método de superclassificação, proporcionando uma taxa de reconhecimento melhor que a dos sistemas isola-

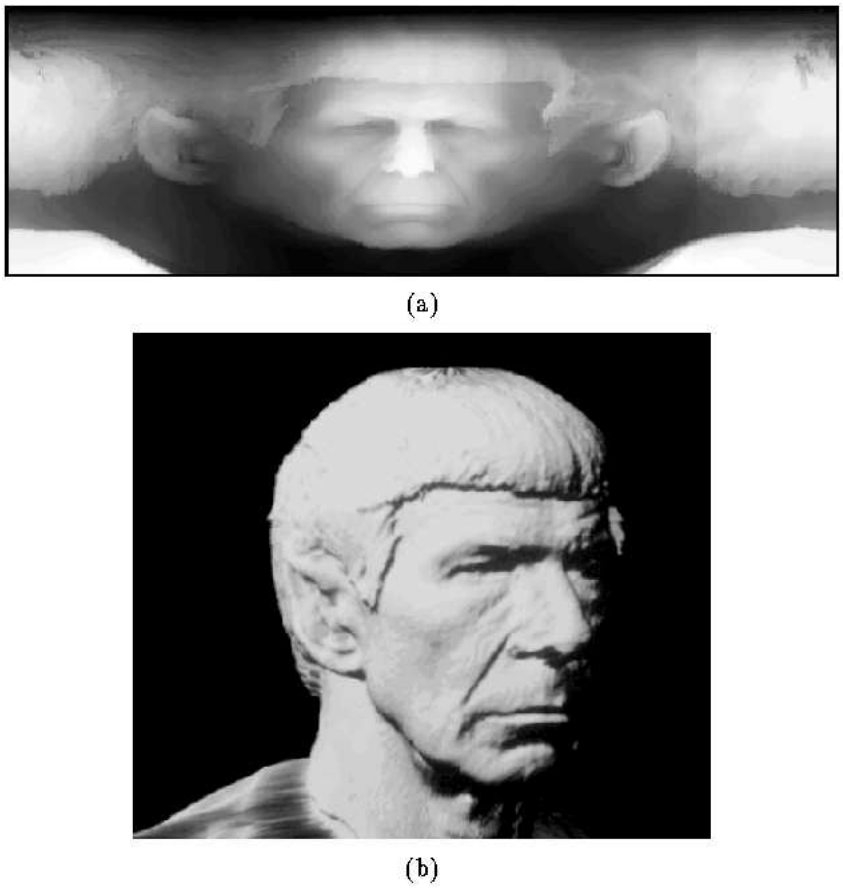

Figura 4.6: *Range image* (a) e sua reconstrução tridimensional (b) (de [Chellappa et al., 1995]).

dos. Como exemplos de métodos dessa abordagem, têm-se os que estão descritos em [Brunelli et al., 1995, Brunelli and Falavigna, 1995].

Com o surgimento de métodos eficientes de detecção e rastreamento de pessoas, juntamente com o uso de treinamento com faces em diversas orientações, foram criados sistemas melhores de reconhecimento de faces em seqüências de vídeo com procedimentos de escolha de "bons" quadros das seqüências [McKenna et al., 1997]. Porém, até final da década de 90, não haviam sistemas conhecidos de identificação em seqüências de vídeo que realmente aproveitassem o movimento para extrair informações extras, como o modo em que as pessoas se movimentam (*gait*).

Em [Burton et al., 1999], há um estudo psicofísico em que foi comparada a taxa de acerto para reconhecimento de pessoas utilizando: (1) somente a imagem da face; (2) somente a imagem do corpo; e (3) a imagem completa da pessoa em movimento. Como é de se esperar, a ordem decrescente da taxa de acerto foi (3), (1) e (2), mas a taxa de acerto obtida usando imagens contendo somente o corpo das pessoas foi muito superior à taxa de

acerto por sorteio[2]. Esse resultado mostra que informações a respeito da maneira como as pessoas andam também podem ser importantes para efetuar-se identificação automática de faces.

A pesquisa em reconhecimento de gestos (de cabeça [Morimoto et al., 1996] e de mão), bem como a de reconhecimento de movimentos do corpo, como danças e interpretações dramáticas, vem desenvolvendo-se muito rapidamente e resultados bastante promissores estão emergindo [Pentland, 2000]. Isso permitiu o surgimento dos primeiros grupos de pesquisa que exploram informações do movimento para efetuar o reconhecimento. Um deles é o da Universidade de Londres, responsável pelo artigo [Li et al., 2000] e pelo livro [Gong et al., 2000], que possui capítulo específico sobre reconhecimento de faces em seqüências de vídeo. O método de extração de características que eles utilizaram foi o de análise de discriminantes lineares.

4.3 Considerações Sobre o Estado-da-Arte

Segundo [Pentland, 2000], o primeiro sistema conhecido de reconhecimento automático de faces provavelmente é o de Kohonen, proposto em 1989. Kohonen demonstrou que uma simples rede neural pode desempenhar reconhecimento de faces usando imagens de faces registradas (normalizadas e alinhadas). Foi empregada uma rede que computa a descrição das faces através da aproximação dos auto-vetores da matriz de auto-correlação das imagens de face. Como sabemos, esses auto-vetores ficaram posteriormente conhecidos como *eigenfaces*. O sistema de Kohonen não foi um sucesso prático, pois ele depende de alinhamento e normalização das faces.

Nos anos seguintes, muitos pesquisadores tentaram esquemas de reconhecimento de faces baseados em atributos locais (bordas, distâncias entre pontos característicos e outras abordagens) com o emprego de redes neurais. Enquanto muito sucesso foi obtido em bases de imagens pequenas com faces alinhadas, nenhum trabalho obteve sucesso em problemas mais realísticos de grandes bases de dados e com localização, orientação e escala da face desconhecidos [Pentland, 2000].

O método de reconhecimento de faces utilizando a transformada de Karhunen-Loève foi proposto em [Kirby and Sirovich, 1990] e está descrito com maiores detalhes na seção 3.2.2. Em [Turk and Pentland, 1991], foi demonstrado que o erro residual da codificação usando *eigenfaces* pode ser usada tanto para detectar faces em imagens naturais como para a determinação precisa da localização, escala e orientação de faces na imagem. Também foi mostrado que esse método pode ser usado para obter o reconhecimento de faces confiável em imagens com poucas restrições.

A partir de 1993, surgiram vários outros sistemas de reconhecimento robustos a ima-

[2]Quando o conjunto de treinamento de todas as classes possui o mesmo tamanho, pode-se dizer que a taxa de acerto por sorteio é igual a $1/c$, sendo c o número de classes existentes.

gens não normalizadas. Segundo Pentland [Pentland, 2000], de acordo com os métodos de avaliação FERET (descrito a seguir), os três melhores algoritmos são os que foram propostos em [Moghaddam and Pentland, 1997], [Moghaddam et al., 1998], [Zhao et al., 1999] e [Wiskott et al., 1997]. Desses trabalhos, os três primeiros baseiam-se em PCA e em métodos discriminantes, divergindo no método de classificação. Já [Zhao et al., 1999] é baseado em *Gabor jets*, *flexible templates* e casamento de grafos.

Para avaliar os algoritmos de reconhecimento de faces, foi criado o programa FERET (Face Recognition Technology) [Phillips et al., 1998], que é conhecido por ser o conjunto de testes (com bases de imagens estáticas) mais abrangente proposto até o momento. A base de dados do FERET possui faces com variações de translação, escala e iluminação de modo consistente com as fotografias 3×4 ou as de carteira de habilitação para motoristas americanos. Há imagens de pessoas obtidas de fotos tiradas em datas diferentes (a diferença chega a um ano).

O maior teste do FERET possui imagens de 1196 pessoas diferentes. Nesse teste, os algoritmos citados acima possuem desempenho muito similar. Com imagens frontais adquiridas no mesmo dia, o desempenho daqueles algoritmos foi de mais de 95% de acerto. Para imagens obtidas com câmeras e iluminações diferentes, o desempenho foi entre 80 e 90%. Para imagens tomadas um ano depois, a taxa de reconhecimento típica foi de 50%. A diferença entre os algoritmos foi menor que 0.5%.

Para testes com 200 pessoas, os três algoritmos praticamente não erraram. Entretanto, nesse experimento, mesmo um simples método de combinação por correlação pode, algumas vezes, propiciar o mesmo resultado, com a diferença de tratar-se de um método lento. Por isso, Pentland [Pentland, 2000] sugere que, para que um novo algoritmo seja considerado potencialmente competitivo, esse deve ser testado com bases de dados possuindo, no mínimo, 200 indivíduos, devendo resultar em uma taxa de reconhecimento maior que 95%.

Porém, esses resultados são válidos somente para imagens estáticas, e ainda não há um bom método definitivo de testes de algoritmos destinados a reconhecimento de pessoas a partir de seqüências de vídeo. A maioria das bases de seqüências de imagens de faces disponível foi criada para testar métodos de rastreamento e de determinação da orientação tridimensional. Por isso, em geral, elas possuem poucas pessoas diferentes. Nos experimentos descritos em [Li et al., 2000] foram realizados testes com uma base de seqüências de imagens de 20 sujeitos, sendo que o treinamento foi realizado com apenas 10 deles, pois os autores também fizeram testes de identificação de "conhecido/desconhecido". Melhor taxa de acerto obtida foi de 94,31

Capítulo 5

Métodos Propostos e Resultados

Conforme mencionado anteriormente, o principal objetivo deste trabalho é o estudo de algoritmos de redução de dimensionalidade com a finalidade de possibilitar a implementação de um sistema de classificação de faces que seja rápido, eficiente e robusto. Com isso, uma possível aplicação será a criação de um sistema de reconhecimento de faces a partir de seqüências de vídeo com poucas restrições em relação à iluminação e aos movimentos das pessoas. Um sistema desse tipo, com o objetivo de efetuar reconhecimento em tempo real, possui as seguintes características:

- Necessidade de um sistema de extração de características e classificação rápido e barato, para que se respeite as restrições de tempo como, por exemplo, aquelas determinadas pela taxa de aquisição de quadros por segundo (caso cada quadro seja classificado) e pelo tempo exigido para que seja dada uma resposta ao usuário [Farines et al., 2000].

- Possibilidade de haver muitas imagens por pessoa tanto para treinamento quanto para testes [Chellappa et al., 1995]. Porém, dependendo do classificador utilizado, é desejável utilizar poucos exemplos de treinamento para permitir a obtenção de resultados em tempo real. Já a disponibilidade de muitas imagens por pessoa na fase de testes permite o uso de métodos de super-classificação[1] para obter-se melhores resultados.

[1]Na página 114 está descrido como um superclassificador pode ser empregado para reconhecimento a partir de seqüências de vídeo.

- Flexibilidade com relação a iluminação, escala e orientação da face (em conformidade com [McKenna et al., 1997]).

Visando a obter um sistema rápido e que não tenha um grande custo em relação a memória, é desejável que a dimensionalidade dos dados não seja grande. Isso deve-se ao fato de que a extração de medidas para realizar a classificação dos padrões de teste fica mais barata computacionalmente quando a dimensionalidade é pequena. Além disso, há outras vantagens em efetuar redução de dimensionalidade as quais foram comentadas no capítulo 2.

Basicamente, foram estudadas duas abordagens para redução da complexidade dos dados de um reconhecedor de faces. A primeira, e mais óbvia, é a de reduzir a dimensionalidade dos dados observados através da simples utilização de uma janela na imagem. A segunda abordagem testada consiste na aplicação de um algoritmo de seleção de características, selecionando somente os atributos com maior poder de discriminação das classes. Tais testes estão descritos a seguir.

5.1 Uso de regiões menores da imagem

5.1.1 Introdução e Motivação

Os métodos baseados em Análise dos Componentes Principais (PCA) estão entre os que possibilitam a obtenção dos melhores resultados em termos de reconhecimento de faces frontais. Apesar da qualidade dos resultados obtidos, essa técnica tem a desvantagem de ser um tanto cara computacionalmente, pois todos os pixels da imagem são utilizados para obter-se sua representação em função da covariância entre essa imagem e todas as outras imagens da base de dados (vide seção 3.2.2).

Alguns pesquisadores utilizaram *eigenfaces* e *eigenfeatures* para efetuar o reconhecimento. Os termos *eigenfeatures*, *eigeneyes*, *eigennose* e *eigenmouth* foram criados em [Moghaddam and Pentland, 1994]. *Eigenfeature* refere-se aos componentes principais obtidos com imagens de regiões restritas da face, como boca (*eigenmouth*), nariz (*eigennose*) e olhos (*eigeneyes*). Segundo [Moghaddam and Pentland, 1994], estudos de movimentos dos olhos indicam que essas regiões particulares das faces representam marcas importantes para reconhecimento, especialmente em uma tarefa de tentativa de discriminação para identificação de pessoas.

Em [Brunelli and Poggio, 1993], os resultados alcançados através da utilização de um quadro (*template*) abrangendo somente a região dos olhos surpreendentemente foram melhores que os resultados com um quadro que cobria toda a face. De maneira similar, no artigo [Moghaddam and Pentland, 1994], os resultados obtidos com *eigenfeatures*, que incluíram olhos, nariz e boca, foram melhores que o de *eigenfaces*.

Além desses fatores, [Moghaddam and Pentland, 1994] discutem uma vantagem potencial do uso de regiões características, também chamados de módulos, das faces. Trata-se da eliminação da possibilidade de ocorrência de erros provocados pelo uso ou não de barba, bigode, chapéu, variações no comprimento do cabelo, presença de feridas e cicatrizes na face, etc. Esses elementos podem prejudicar o desempenho quando utiliza-se a imagem de toda a face, mas não quando forem utilizadas somente as regiões importantes. A figura 5.1 ilustra três casos em que o uso de toda a imagem da face causou erro de classificação, e o uso de módulos resultou na classificação correta.

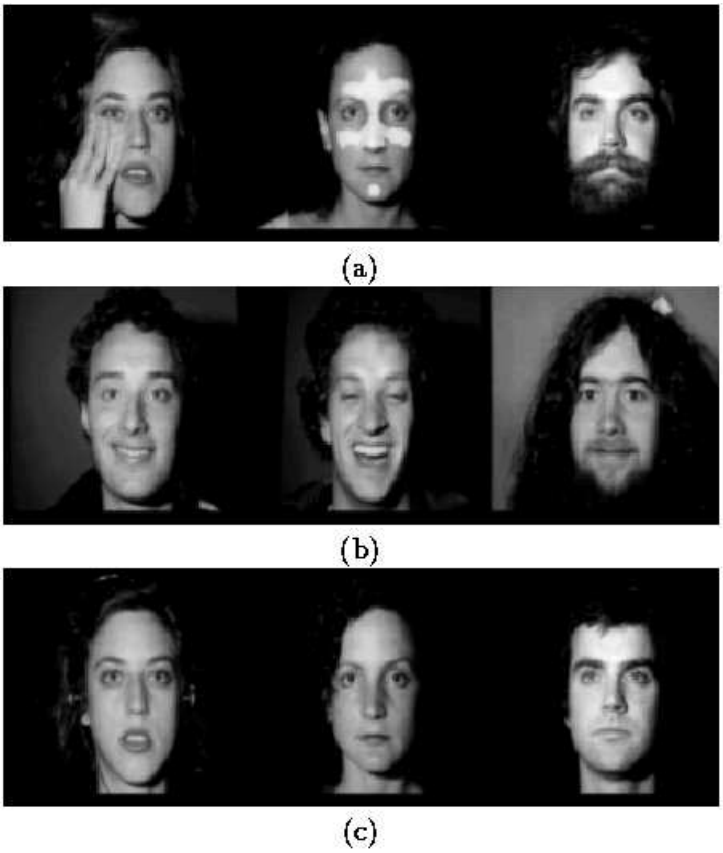

Figura 5.1: Reconhecimento por regiões características: (a) imagens de teste; (b) resultados de classificação incorreta devido ao uso da imagem de toda a face; (c) resultado de classificação correta devido ao uso de módulos (figura baseada em [Moghaddam and Pentland, 1994]).

Neste trabalho de mestrado, realizamos testes visando verificar os resultados de Brunelli em um sistema de reconhecimento baseado em PCA, mas comparando apenas o desempenho do classificador com imagens de faces versus com imagens contendo os olhos. Também verificamos a relação existente entre esses resultados e o número de autovetores utilizados (dimensionalidade). Este trabalho, publicado em [Campos et al., 2000d], iniciou-se como parte das tarefas exigidas na disciplina de Tópicos em Inteligência Ar-

tificial: Reconhecimento de Faces, ministrada pelo Prof. Carlos Hitoshi Morimoto, no primeiro semestre de 1999.

5.1.2 Base de Imagens

Foi utilizada uma base de imagens pública a qual foi criada e disponibilisada pelo MIT (Massashusetts Institute of Technology). Essa base é composta por imagens de dezesseis adultos, seis imagens por pessoa. Várias imagens continham pessoas usando óculos, bigode ou barba e com diferentes comprimentos de cabelo. Além disso, as imagens possuem grandes variações na iluminação, fundo (*background*) irregular e diferentes expressões faciais. Porém, as imagens consideradas não possuem problemas de auto-oclusão dos olhos. Há duas características importantes dessa base de imagens de faces:

- A primeira refere-se à orientação das faces. Há 3 posições diferentes, sendo a primeira em posição normal (*upright*), a segunda com a cabeça inclinada (rotação no plano da imagem) para a esquerda e a terceira com a cabeça inclinada para a direita.
- Outra característica importante é que as imagens foram adquiridas a duas distâncias diferentes entre a câmera e a pessoa (escalas).

A combinação desses dois parâmetros resulta em 6 imagens por pessoa, como ilustra a figura 5.2

Figura 5.2: Exemplo de imagens de um indivíduo da base utilizada.

5.1.3 Pré-processamento

As imagens usadas para construir as *eigenfaces* foram criadas a partir de recortes da base original para que os cabelos e o fundo da imagem não influenciassem no reconhecimento,

pois esses podem apresentar muitas variações. Tais recortes englobavam a região entre a testa e o queixo dos indivíduos. Já para a construção dos *eigeneyes*, foram utilizados recortes que englobam somente a região dos dois olhos, incluindo parte das sobrancelhas. O tamanho desses recortes foi determinado de acordo com uma proporção baseada na distância entre os olhos. A figura 5.3 mostra um exemplo desses recortes. [2]

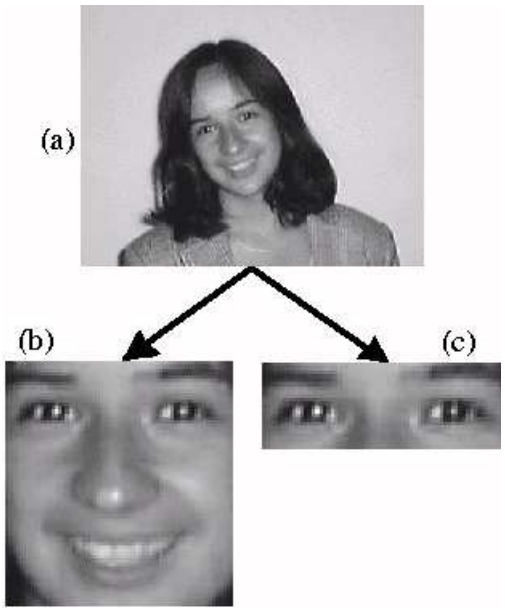

Figura 5.3: Processo de obtenção das imagens de face e de olhos: (a) imagem original, de 128×120 *pixels*; (b) recorte de face; (c) recorte de olhos.

Nesses experimentos, a determinação da posição dos olhos foi feita por um operador humano, pois, conforme mencionado anteriormente, segmentação automática não faz parte do escopo deste trabalho. Como as imagens apresentavam grandes variações na orientação da cabeça e na escala, antes de efetuar os recortes das faces e dos olhos, foi realizada a rotação das imagens de forma que os olhos ficassem na mesma linha horizontal. Após realizar os recortes, para viabilizar o uso de PCA, foi necessário redimensionar as imagens para que todas ficassem com a mesma resolução. Foi efetuado o redimensionamento utilizando o método de "vizinho mais próximo" [Gonzalez and Woods, 1992] para interpolar os *pixels* da imagem de saída.

A resolução escolhida foi de 64×64 pixels, pois essa engloba faces mesmo nas imagens em que a pessoa está mais afastada da câmera. Além disso, essa é uma resolução que equilibra custo computacional com qualidade das imagens, já que é desejável utilizar as menores imagens possíveis, mas sem perder muitos detalhes.

[2]Essa imagem é apresentada apenas para ilustrar o processo de formação dos dados, não tendo sido utilizada no experimento.

5.1.4 Testes e Resultados

O pré-processamento descrito acima foi realizado em todas as imagens da base, sendo criado assim, um conjunto de imagens de olhos e outro de faces. Posteriormente, as imagens de treinamento da base de faces são utilizadas para treinar uma transformada PCA, obtendo-se, dessa forma, os *eigenfaces*. O mesmo foi feito para as imagens de olhos possibilitando a obtenção dos *eigeneyes*. Alguns *eigeneyes* e *eigenfaces* obtidos a partir de uma base treinada com 5 imagens por pessoa são mostrados na figura 5.4. A seção 3.2.2 contém maiores detalhes a respeito da transformada PCA.

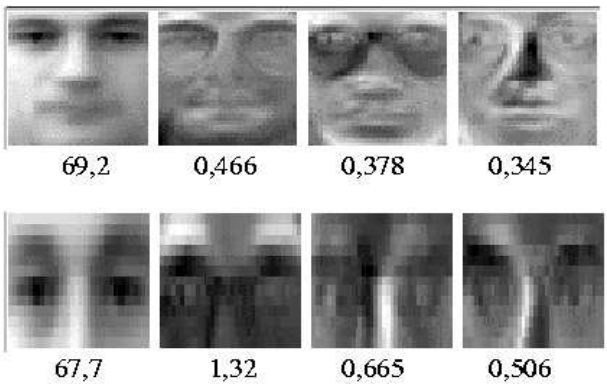

Figura 5.4: Os quatro primeiros auto-vetores mostrados como imagens e seus respectivos auto-valores, obtidos através da base de faces (acima) e da base de olhos (abaixo)

Dois experimentos foram realizados: no primeiro utilizando três imagens por pessoa para treinar o sistema e, no segundo, cinco. Em ambos os experimentos foi utilizada apenas uma imagem de teste por pessoa, a qual não foi utilizada no treinamento. A classificação foi efetuada utilizando a técnica do vizinho mais próximo.

Os resultados obtidos são bastante satisfatórios como meios de comparação entre faces e olhos para reconhecimento de pessoas. Obviamente, se forem realizados testes utilizando imagens pertencentes ao conjunto de treinamento, a taxa de acerto será de 100%, já que foi utilizado o classificador de vizinho mais próximo (vide seção 2.2.3). Os resultados dos testes realizados com imagens que não pertenciam ao conjunto de treinamento estão ilustrados nas tabelas 5.1 (com treinamento usando 3 imagens por pessoa) e 5.2 (com treinamento usando 5 imagens por pessoa).

Através dessas tabelas, é possível notar que, em geral, o reconhecimento com olhos foi melhor que com faces. Esse fato é intuitivamente inesperado, já que as imagens de faces contém mais informações que as de olhos. Mas, devido ao problema da dimensionalidade, sabe-se que o aumento na dimensionalidade dos dados deve ser compensado por um aumento do número de exemplos de treinamento para que a taxa de acerto permaneça estável. Isso justifica o fato de que as taxas de reconhecimento aumentam significativamente para ambos os sistemas de classificação quando se aumenta o tamanho do conjunto

Tabela 5.1: Desempenho do classificador para reconhecimento de olhos e de faces quando treinado com 3 imagens por pessoa.

# Auto-vetores	Olhos %	Faces %
3	25,00	31,25
4	25,00	37,50
5	50,00	37,50
10	56,25	43,75
13	62,50	43,75
15	62,50	43,75
24	62,50	43,75
48	62,50	43,75

Tabela 5.2: Desempenho do classificador para reconhecimento de olhos e de faces quando treinado com 5 imagens por pessoa.

# Auto-vetores	Olhos %	Faces %
3	40,00	46,67
15	73,33	66,67

de treinamento. Além disso, pode-se notar que, quando treinado com 3 imagens por pessoa, o desempenho do sistema não melhora se forem utilizados mais que 13 autovetores. Isso ocorre pois 13 é a dimensionalidade ideal para esse problema, o que indica que esse número de autovetores é suficiente para discriminar esses padrões. Portanto, quando são utilizados mais autovetores, esses não adicionam informações relevantes para a classificação. Maiores detalhes sobre o problema da dimensionalidade estão na seção 2.3.

Além desse problema genérico de reconhecimento de padrões, há um fator relativo às seguintes propriedades específicas da face que corroboram com esses resultados [Gong et al., 2000]:

- a boca (e também o queixo) não é um objeto tão rígido quanto os olhos (considerando-se que em todas as imagens os olhos estavam abertos), sofrendo grandes variações com expressões faciais, com a fala ou mesmo com movimentos da cabeça;
- a projeção do nariz em um plano bidimensional (plano da imagem) faz com que sua imagem sofra grandes alterações com variações na orientação da cabeça.

Por isso, as imagens de faces são mais distorcidas, esse fato causa uma maior dificuldade em obter boas taxas de reconhecimento usando tais imagens com um conjunto de treinamento pequeno. Isso requer que o classificador tenha um poder de generalização maior, já que essas partes da face são características que podem ser muito correlacionadas e ruidosas.

Assim, para possibilitar a obtenção de boas taxas de acerto utilizando imagens de toda a face, dever-se-ia aumentar o tamanho do conjunto de treinamento [Campos et al., 2000d].

5.2 Testes com Algoritmos de Busca para Seleção de Características

Foram estudados e testados alguns métodos de seleção de características sob dois aspectos: o algoritmo de seleção e a função critério. Segue a descrição de testes com algoritmos de seleção. Os testes realizados com diferentes funções critérios estão descritos nas seções 3.4 (dados sintéticos) e 5.3 (dados reais de faces).

5.2.1 Descrição do Problema

Em [Campos et al., 2000c], fizemos um estudo comparando o desempenho de quatro estratégias de seleção de características, das quais duas são baseados em busca automática. O problema abordado foi a discriminação entre classes de padrões obtidos a partir de descritores de Fourier. Esses descritores foram obtidos a partir de um método proposto em [Campos et al., 2000a] para discriminação de imagens contendo faces "versus" imagens não contendo faces de uma forma rápida. Tal método de discriminação é constituído pelos seguintes passos:

- obtenção do mapa de bordas horizontais binário da imagem através do Laplaciano da Gaussiana unidimensional vertical [Gonzalez and Woods, 1992];
- formação de um sinal unidimensional a partir de uma "varredura" vertical do mapa de bordas obtido, semelhante à formação dos espaços de características da abordagem de PCA para reconhecimento de faces, descrita na seção 3.2.2.
- obtenção de 30 descritores de Fourier [Gonzalez and Woods, 1992, Cesar-Jr, 1997] desse sinal unidimensional.

O principal objetivo desse sistema de discriminação faces $\times$ não faces é possibilitar a criação de um método de detecção de faces através de sua aplicação em janelas que varrem a imagem. A arquitetura desse processo está ilustrada na figura 5.5. Esse processo foi realizado em 219 imagens de faces e em 219 imagens de outros objetos (não-faces). Desse total, 2/3 foi utilizado para treinar um classificador de mínima distância ao protótipo, e o restante para testá-lo. Como o objetivo de nossa pesquisa não engloba detecção de faces, detalharemos apenas as partes referentes à seleção de características e classificação envolvidas nesse projeto. O leitor interessado no método de extração de características proposto pode consultar [Campos et al., 2000a, Campos et al., 2000c] (em anexo).

5.2.2 Métodos de Seleção Avaliados

Foram realizados testes com diferentes estratégias de seleção de características. Os resultados obtidos foram avaliados para determinar o melhor método de busca para selecionar características e a melhor dimensionalidade. As quatro técnicas de seleção que foram testadas são as seguintes:

- utilização dos m primeiros coeficientes, que é a abordagem mais comum em se tratando de coeficientes de Fourier em visão computacional (vide seção 3.2.1);
- utilização dos m maiores coeficientes;
- métodos de seleção SFSM (vide seção 3.3);
- métodos de seleção ASFSM (vide seção 3.3).

Os métodos automáticos SFSM e ASFSM (Métodos de Busca Seqüencial Flutuante e suas versões Adaptativas) foram escolhidso pois, até 1999, esses eram indicados como os melhores algoritmos de busca para seleção de característica (vide seção 3.3).

A base de imagens utilizada contém 146 padrões de faces e 146 padrões de não-faces para efetuar o treinamento. Para a realização dos testes de classificação, há outros 73 padrões por classse. Com isso, as duas classes são suficientemente bem representadas, tanto para treinamento quando para testes. Por isso, na realização de seleção automática de características (SFSM e ASFSM), pode-se utilizar os resultados de classificação como função critério da seleção de características (vide seção 3.3.3). Para os métodos adaptativos, foram utilizados os parâmetros $r_{max} = m - 2$ e $b = 1$, sendo m a dimensionalidade desejada. Em ambos os métodos automáticos, foi adotada a seguinte estratégia: se $m < N/2 \Rightarrow$ faça a busca para frente (SFFS ou ASFFS); senão $\Rightarrow$ faça a busca para trás (SFBS ou ASFBS).

Para efetuar a seleção automática de características, os procedimentos de busca efetuam um número muito grande de avaliações dos sub-conjuntos de características (vide seção 3.3). Por isso, é importante que a função critério não consuma muito tempo de execução. Um dos classificadores mais rápidos é o de mínima distância ao protótipo (seção 2.2.4), por isso empregamos esse classificador no cálculo da função critério.

5.2.3 Resultados

Através da figura 5.6, podemos observar os resultados obtidos. Essa figura mostra o resultado da função critério proporcionado pelo sub-conjunto de característica selecionado. Foram feitos experimentos de seleção com várias dimensionalidades m entre 3 e 30. Esses resultados comprovam a superioridade dos métodos de seleção de característica, principalmente o ASFSM. É interessante notar também que os melhores resultados foram obtidos

utilizando dimensionalidade menor que 20. Isso confirma o fato de que um aumento no número de características não garante melhora no desempenho do classificador.

No pior caso, a maior diferença entre o resultados de SFSM e ASFSM foi de 4,35%. Mas, em termos de tempo de execução, no pior caso, o algoritmo SFSM levou 2 segundos, enquanto o ASFS levou 4 horas e 22 minutos para determinar o conjunto de características. Esse é um fator muito relevante na escolha do algoritmo de seleção.

É importante lembrar que o total de características disponíveis é 30 e, por isso, os resultados de todos os métodos "convergem" para o mesmo valor quando a dimensionalidade vai para 30.

Como conclusão, temos que a dimensionalidade ideal para esse problema depende do método de busca para seleção de características. No caso dos métodos ASFSM e SFSM, os melhores resultados já são obtidos com 6 descritores de Fourier. No caso dos outros dois métodos, os melhores resultados foram obtido com 9 descritores de Fourier.

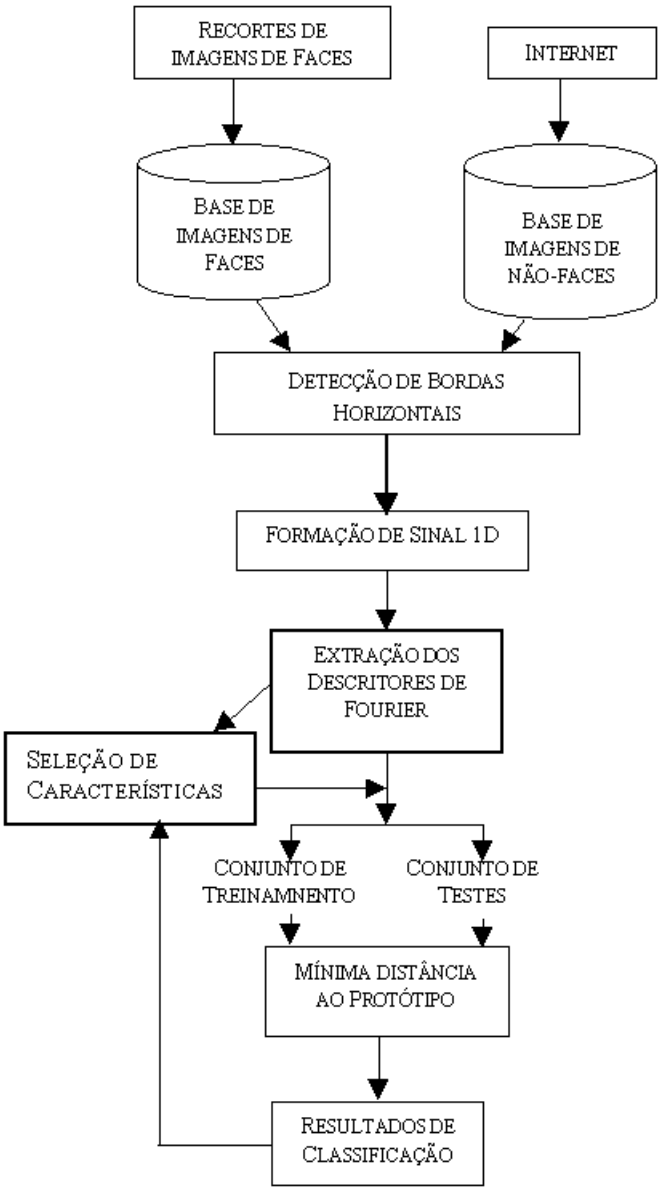

Figura 5.5: Esquema do sistema de discriminação faces × não-faces.

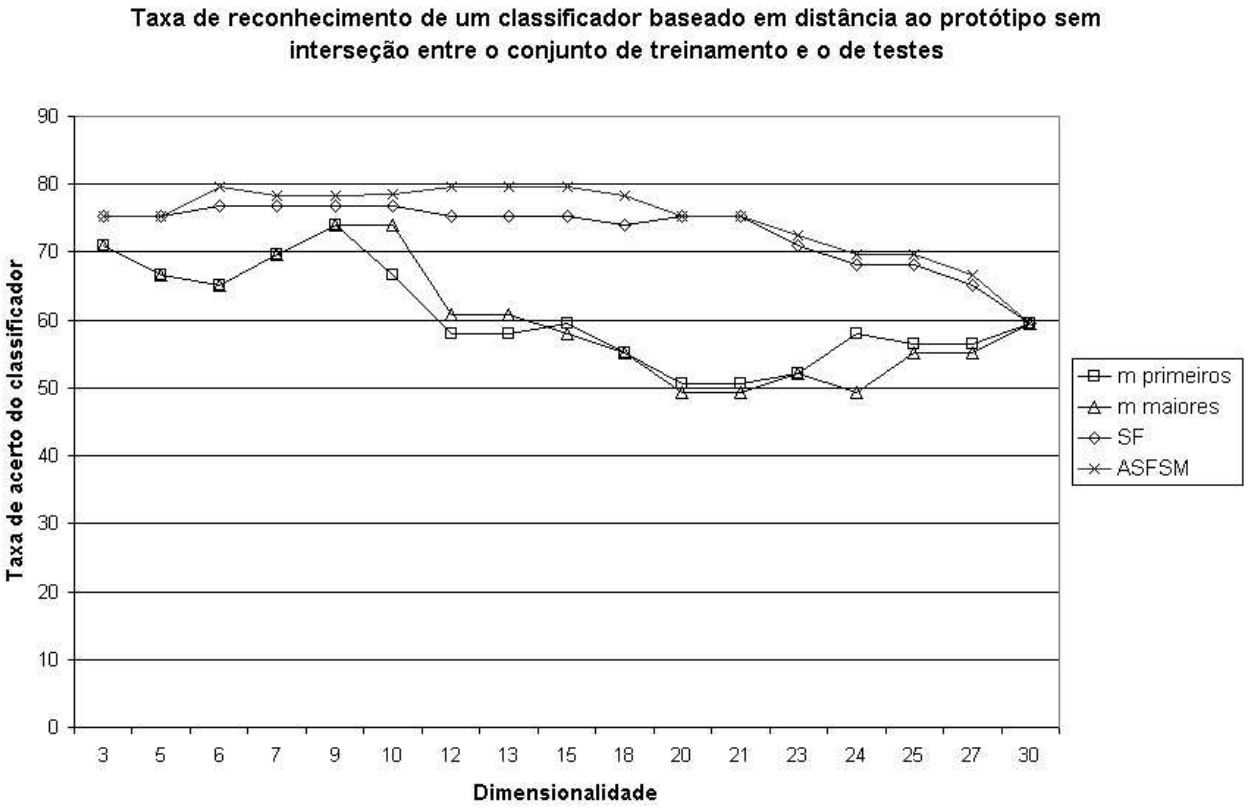

Figura 5.6: Resultados obtidos (em % de taxa de acerto do classificador) pelos conjuntos de características selecionados.

5.3 Função Critério Baseada em Distância Nebulosa para c Classes

A distância nebulosa proposta em [Lowen and Peeters, 1998], foi definida para o cálculo da distância entre dois conjuntos nebulosos. Em [Campos et al., 2001] (trabalho descrito na seção 3.4), propusemos a utilização dessa medida de distância como função critério para realizar seleção de características utilizando o método de busca proposto em [Pudil et al., 1994] (SFSM). Em aplicações práticas, como reconhecimento de pessoas, usualmente há mais de duas classes (mais de duas pessoas a serem reconhecidas). Por isso, é necessário criar uma solução para o fato de que uma distância só pode ser medida entre dois elementos. Conforme mencionado anteriormente, uma solução possível é calcular o ínfimo das distância entre todos os pares de conjuntos, conforme a seguinte equação:

$$g_p^\tau(\nu_1, \nu_2, \cdots, \nu_c) = \inf_{k=2,\cdots,c;l=1,\cdots,m} d_p^\tau(\nu_k, \nu_l) \tag{5.1}$$

De acordo com o que foi discutido na seção 3.4.5, a complexidade de tempo para calcular-se $d_p^\tau(\nu_k, \nu_l)$ é de $O(|T|^2) + O(|T|) \cdot O(b^2)$, sendo, no pior caso, $O(|T|^2)$, e, no melhor caso, $O(|T|^3)$, em que $|T|$ é o número total de padrões no conjunto de treinamento composto por duas classes. Suponhamos que cada classe possua $|T|/c$ padrões de treinamento. Para implementar a equação 5.1, é necessário calcular c^2 vezes a distância $d_p^\tau(\nu_k, \nu_l)$, o que resulta em uma complexidade de

$$O(c^2) \cdot (O(|T|^2/c^2) + O(|T|/c) \cdot O(b^2)) \quad = \tag{5.2}$$

$$O(|T|^2) + O(c) \cdot O(|T|) \cdot O(b^2) \tag{5.3}$$

Isso implica que, no pior caso, o tempo de execução da função critério da equação 5.1 é de $O(c) \cdot O(|T|^3)$ e, no melhor caso é de $O(|T|^2)$.

Como os algoritmos de busca para seleção de características avaliam, através da função critério, muitos conjuntos de características para chegarem ao resultado final, é necessário que essa seja o mais eficiente possível. Para implementar uma função critério eficiente, com as mesmas propriedades que a função proposta na seção 3.4 para problemas com mais de duas classes, propusemos uma função critério baseada na seguinte diferença local:

$$f_{\mathbf{x}}^\tau(\nu_1, \nu_2, \cdots, \nu_c) = \inf_{\mathbf{y},\mathbf{z} \in B(\mathbf{x},\tau); j=2,\cdots,c; i=1,\cdots,j} |\nu_i(\mathbf{y}) - \nu_j(\mathbf{z})| \tag{5.4}$$

Note que essa equação é bastante semelhante à 3.36, com a diferença de que deve ser calculado o ínfimo da diferença entre os graus de pertinência de todos os padrões de todas as classes que estão na bola $B(\mathbf{x}, \tau)$. Assim, a função critério fica:

$$f_p^\tau(\nu_1, \nu_2, \cdots, \nu_c) = [\int_{\mathcal{F}} [f_{\mathbf{x}}^\tau(\nu_1, \nu_2, \cdots, \nu_c)]^p d\mathbf{x}]^{1/p}, \tag{5.5}$$

Conforme o método que utilizamos para efetuar a fuzzificação (vide seção 3.4.3), $\nu_{\omega_i}(\mathbf{x}_j) = 0$ se $\mathbf{x}_j \notin \omega_i$. Por isso, para implementar a diferença local da equação 5.4, pode ser empregado um algoritmo praticamente idêntico ao algoritmo DiferençaLocal (vide seção 3.4.5). A diferença é que o número de padrões que pode ser incluído em uma bola $B(\mathbf{x}, \tau)$ pode ser maior, pois é possível ocorrerem casos em que, em uma mesma bola, haja padrões de mais de duas classes.

Supondo novamente que cada classe possui $|T|/c$ padrões de treinamento, temos que há um total de $|T|$ padrões no espaço de características. Com isso, a complexidade desse algoritmo é da ordem de:

$$O((c \cdot |T|/c)^2) + O(c \cdot |T|/c) \cdot O(b^2) \quad = \tag{5.6}$$
$$O(|T|^2) + O(|T|) \cdot O(b^2) \tag{5.7}$$

o que significa uma vantagem em relação à função da equação 5.1 para problemas com um número de classes c grande.

Assim, no melhor caso, ou seja, quando cada bola contiver apenas elementos de até duas classes diferentes, a complexidade dessa função critério será de $O(|T|^2)$. No entanto, no pior caso, quando todas as bolas utilizadas no cálculo da diferença local englobam padrões de todas as classes existentes no espaço de características, a complexidade da função critério da equação 5.5 será de:

$$O((c \cdot |T|/c)^2) + O(c \cdot |T|/c) \cdot O((c \cdot |T|/c)^2) \quad = \tag{5.8}$$
$$O(|T|^3) \tag{5.9}$$

Portanto, no pior caso esse algoritmo não apresenta, vantagens sobre a função da equação 5.1. Porém, é importante ressalvar que se pode deduzir que o caso médio (com uma bola de tamanho próximo do ideal) da função critério da equação 5.5 certamente será mais rápido que o da função da equação 5.1.

Considerando que todas as classes possuem distribuições aproximadamente isotrópicas, pode-se verificar que a função $f_p^\tau(\nu_{\omega_1}, \nu_{\omega_2}, \cdots, \nu_{\omega_c})$ (com a diferença local da equação 5.4) possui propriedades semelhantes a todas as que foram descritas na seção 3.4.6. Os mesmos efeitos que ocorrem na diferença $d_{\mathbf{x}}^\tau$ e na distância $d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j})$ em relação à compacidade, distância entre os protótipos e tamanho da bola são esperados para a diferença local $f_{\mathbf{x}}^\tau$ para a função $f_p^\tau(\nu_{\omega_1}, \nu_{\omega_2}, \cdots, \nu_{\omega_c})$. Obviamente, deve-se considerar que há vários protótipos e várias classes de padrões (ao invés de 2), e que todas as classes possuem o comportamento mencionado. Pode-se mostrar que todas prováveis relações entre os resultados de $d_p^\tau(\nu_{\omega_i}, \nu_{\omega_j})$ para as possibilidades mostradas na página 65 também são válidas para $f_p^\tau(\nu_{\omega_1}, \nu_{\omega_2}, \cdots, \nu_{\omega_c})$. Obviamente, devemos lembrar que aquelas relações ocorrem com 2 classes. A generalização dessas propriedades para c classes é válida para a função $f_p^\tau(\nu_{\omega_1}, \nu_{\omega_2}, \cdots, \nu_{\omega_c})$. Por exemplo, os casos 3.(a) e 3.(b) ocorrem na função $f_p^\tau(\nu_{\omega_1}, \nu_{\omega_2}, \cdots, \nu_{\omega_c})$ quanto algumas classes possuem compacidade grande e outras possuem compacidade pequena.

5.3.1 Experimentos dessa Função Critério para Seleção de Eigeneyes

Base de imagens

Para avaliar essa função critério associada ao método de seleção de características proposto na seção 3.4 [Campos et al., 2001], foi utilizada uma base de imagens de olhos para reconhecer pessoas. Essa base originou-se de uma base de imagens de faces com 29 classes (pessoas), 6 amostras por classes (para cada pessoa havia 6 imagens de seus olhos), com fundo (*background*) razoavelmente controlado e resolução de 512×342. As imagens possuíam pessoas com grandes variações de pose (orientação da cabeça) e diferentes expressões faciais. Foi realizada a segmentação e a normalização das imagens dos olhos utilizando o mesmo procedimento descrito na seção 5.1 [Campos et al., 2000d], com a diferença de que a resolução das imagens após esse pré-processamento é de 13×36 pixels.

A transformada de Karhunem-Loève (PCA) foi aplicada em todas as imagens disponíveis para obter os vetores da base do "espaço de olhos" com 468 dimensões, chamados *eigeneyes*. Usualmente, para efetuar redução de dimensionalidade utilizando PCA, são simplesmente selecionados os m primeiros componentes ($m \ll N$), sendo o restante descartado. Porém, há evidências de que nem sempre essa seja a melhor estratégia [Theodoridis and Koutroumbas, 1999, Jain et al., 2000, Belhumeur et al., 1997], principalmente quando se trata de imagens de faces com grandes variações de iluminação e expressões faciais, o que ocorre em nossa base de imagens.

Após a obtenção da representação das imagens no espaço de olhos, foi realizada uma normalização do espaço de características em relação à média e ao desvio padrão, da mesma forma que a normalização descrita na seção 3.4.

Testes e Resultados Preliminares

Foram realizados testes considerando variações no tamanho da bola utilizada na distância nebulosa (parâmetro τ, que define a tolerância). Esses testes tiveram como objetivo a determinação do tamanho da bola que propiciasse os melhores resultados de seleção de características para o problema abordado. Para simples fim ilustrativo, os valores obtidos pela função critério com a variação do raio da bola τ são mostrados no gráfico da figura 5.7. Visando obter o melhor classificador de vizinhos mais próximos para esse problema, também foram realizados testes para verificar a variação do desempenho do classificador de K vizinhos mais próximos (KNN) para $K = 1, 2, 3, 4, 5$.

As figuras 5.10 a 5.21) ilustram os resultados para cada valor de K. Nas figuras 5.8 e 5.9, são mostrados os resultados do classificador de distância ao protótipo aplicado no conjunto de características determinado pelo nosso método de seleção. Cada gráfico ilustra

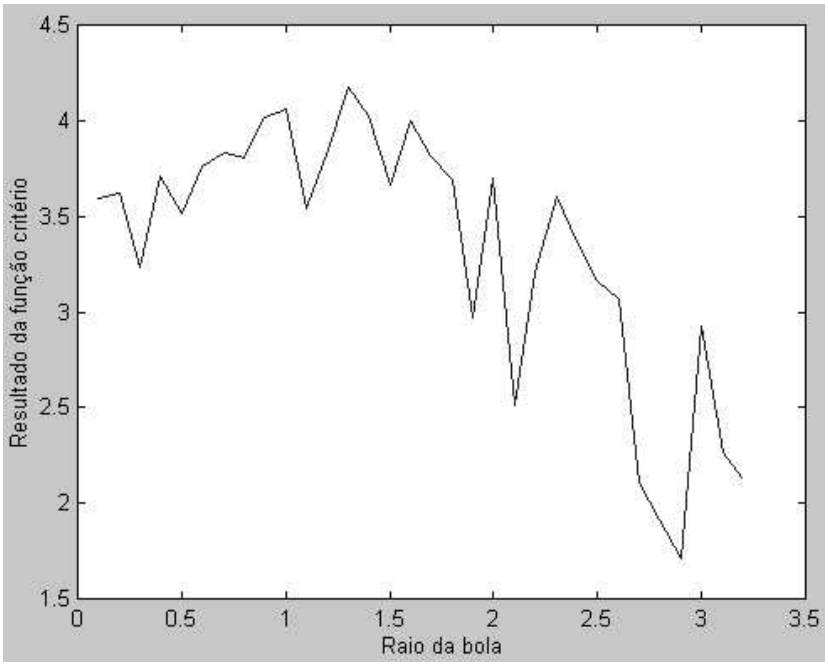

Figura 5.7: Resultado da função critério com a variação de τ.

a variação da taxa de acerto de uma técnica de classificação em função do tamanho da bola utilizada na distância nebulosa. Mais especificamente, no eixo das abscissas, encontra-se o raio da bola τ, enquanto no eixo das ordenadas, encontra-se a taxa de acerto dos classificadores em percentagem. Nos experimentos realizados, foi efetuada uma seleção de características em busca das 15 melhores características (eigeneyes). Os resultados foram comparados com o método mais tradicional de efetuar-se redução de dimensionalidade com PCA, ou seja, selecionando simplesmente os 15 primeiros componentes.

Para cada técnica de classificação são mostrados os resultados obtidos com a utilização dos 15 *eigeneyes* selecionados por nossa técnica (mostrados nas linhas contínuas) em comparação com o resultado obtido com a utilização dos 15 primeiros *eigeneyes* (mostrado na linha tracejada).

Foram realizados vários testes de classificação com os dois classificadores utilizados: K vizinhos mais próximos (KNN) e distância ao protótipo. No caso do classificador de distância ao protótipo, os protótipos foram definidos através da média dos padrões de treinamento de cada classe.

Conforme pode ser notado pelas figuras 5.8 e 5.9, foram realizadas duas baterias de testes com o classificador de distância ao protótipo. Na primeira, todos os padrões foram utilizados para treinar e testar o classificador ("treinamento=testes"). Na segunda, foram utilizados 2/3 dos padrões disponíveis para treinar o classificador (determinar os protótipos) e o restante para testar.

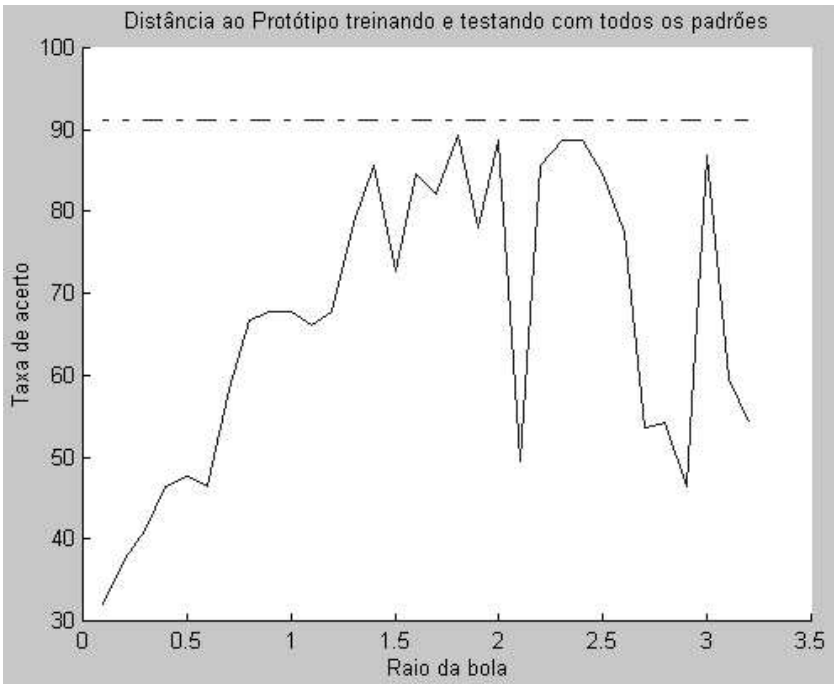

Figura 5.8: Distância ao Protótipo, treinando e testando com todos os padrões disponíveis.

O mesmo foi realizado com o classificador de K vizinhos mais próximos. Além disso, foram realizados experimentos utilizando a estratégia **leave-one-out** (vide figuras 5.10 a 5.21). Nessa estratégia, para cada classe, o conjunto de treinamento inicialmente é composto por todos os padrões, menos o primeiro, o qual é utilizado para testar a classificação. Na segunda iteração de testes, o conjunto de treinamento é composto por todos os padrões menos o segundo, o qual é utilizado para teste. Esse processo repete-se até que todos os padrões de cada classe tenham sido utilizados para testar o classificador (com o restante sendo utilizado para treinar). Ao final, é calculada a taxa de acerto média, a qual é mostrada nos gráficos referidos (juntamente com os outros resultados).

Conforme mostrado nas figuras 5.10 a 5.21, foram realizados experimentos com o classificador de K vizinhos mais próximos variando o valor de K entre 1 e 5. Os resultados com K = 2 não foram mostrados, pois esses são idênticos aos obtidos com K = 1. É importante ressalvar que para evitar problemas de empate, os quais poderiam ocorrer quando o número de vizinhos próximos pertencentes a classes diferentes é igual, foi utilizada uma estratégia simples de desempate que dá prioridade à classe que possui um padrão mais próximo do elemento de teste.

Dentre os pontos mais importantes dos resultados obtidos, nota-se que ao se treinar o classificador com 2/3 dos padrões e testar com o restante, para vários valores de τ, foram obtidos resultados superiores àqueles obtidos com a utilização dos 15 primeiros *eigeneyes*. Também é notável que, para $K = 3$, o mesmo ocorreu ao treinar e testar o

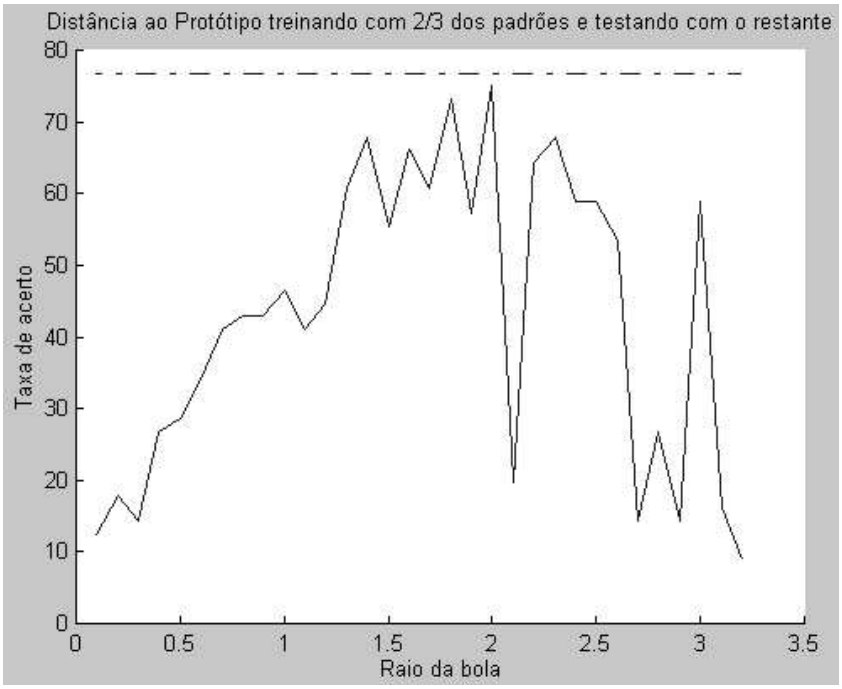

Figura 5.9: Distância ao Protótipo, treinando com 2/3 dos padrões e testando com os 1/3 restantes.

classificador com todos os padrões disponíveis. Os melhores resultados ocorreram algumas vezes quando foi utilizada uma bola de raio τ entre 1.2 e 2.8.

Um resultado notável é o que está ilustrado na figura 5.10, em que, para todos os valores de τ, a taxa de acerto obtida foi de 100%, tanto para o subconjunto obtido pelo nosso método quando com a utilização das 15 primeiras características. Isso se deve ao fato de que, quando $K = 1$, se o conjunto de testes tiver sido usado no treinamento, não há erro ao se utilizar a transformada de PCA com um número razoável de componentes principais.

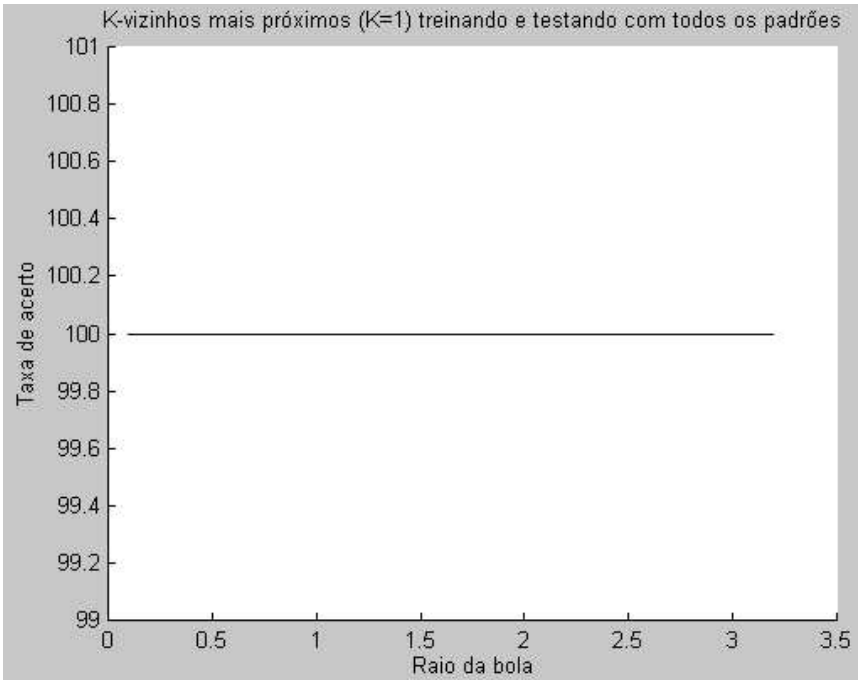

Figura 5.10: K vizinhos mais próximos (K=1), treinando e testando com todos os padrões disponíveis.

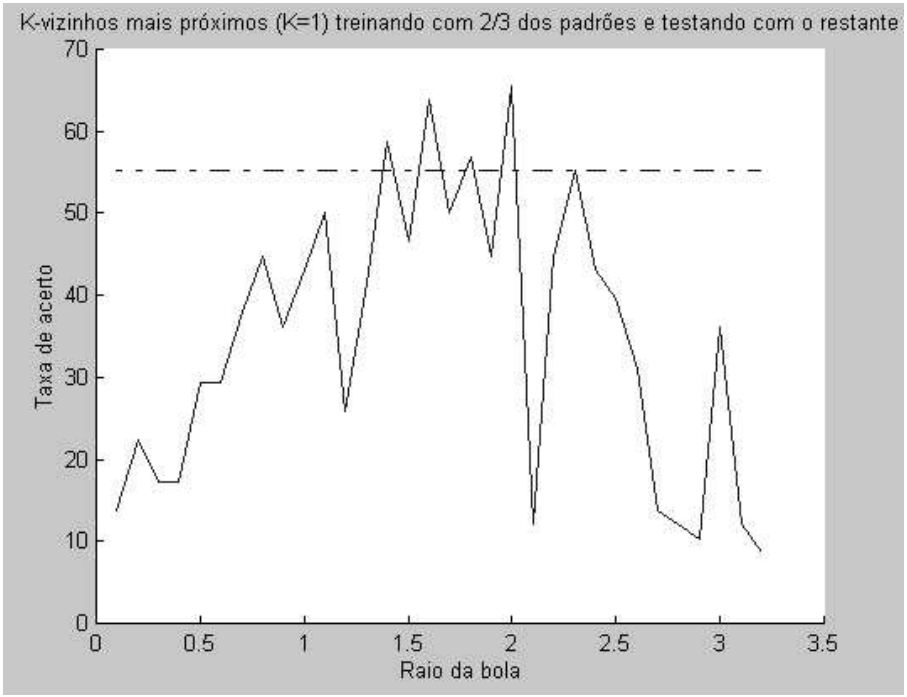

Figura 5.11: K vizinhos mais próximos (K=1), treinando com 2/3 dos padrões e testando com os 1/3 restantes.

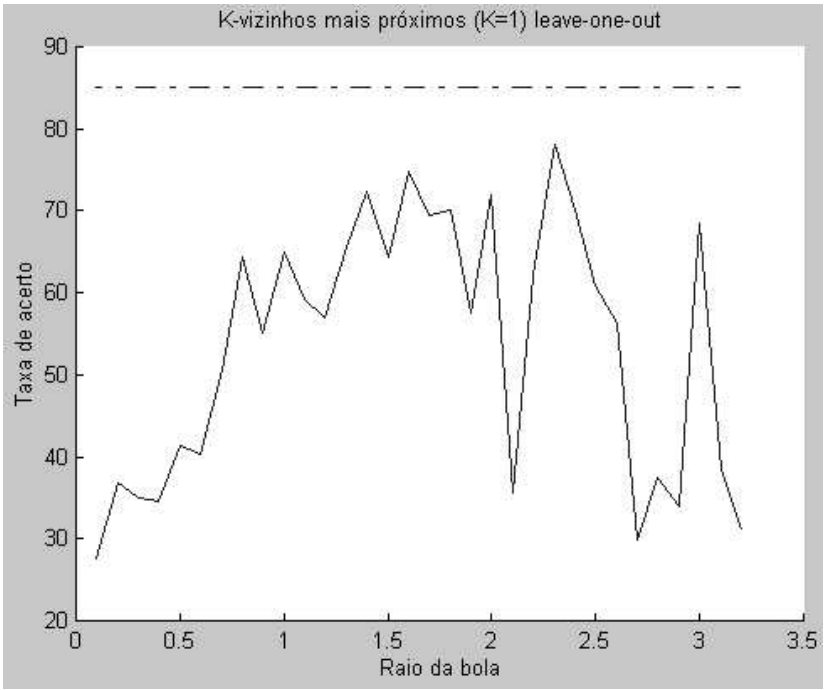

Figura 5.12: K vizinhos mais próximos (K=1), *leave-one-out*.

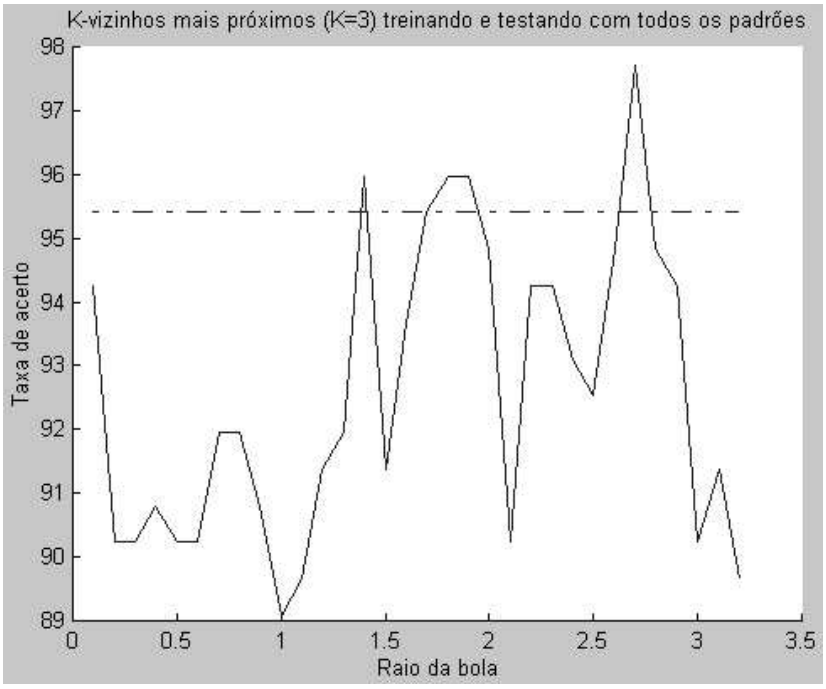

Figura 5.13: K vizinhos mais próximos (K=3), treinando e testando com todos os padrões disponíveis.

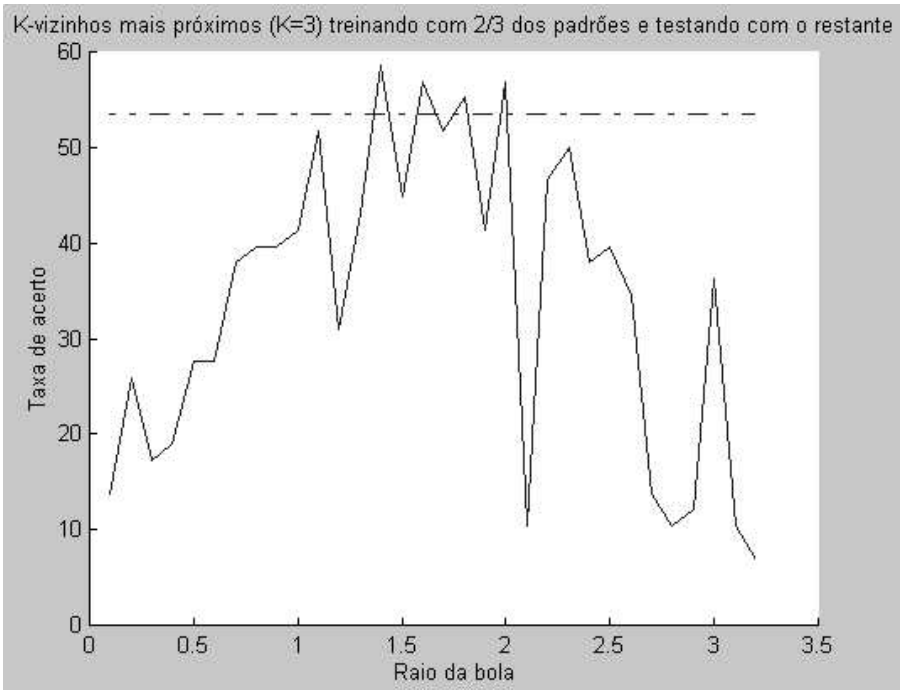

Figura 5.14: K vizinhos mais próximos (K=3), treinando com 2/3 dos padrões e testando com os 1/3 restantes.

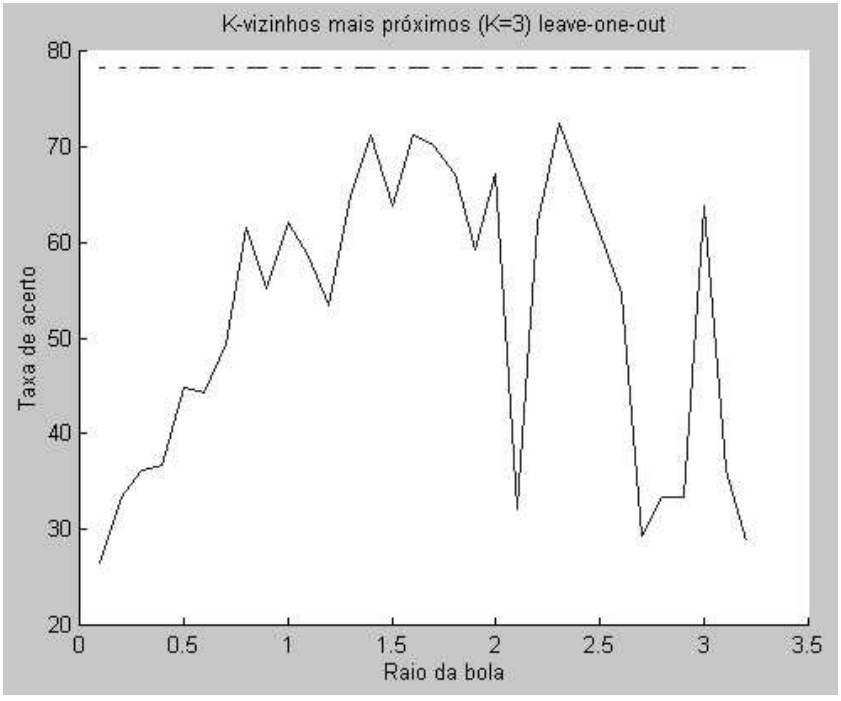

Figura 5.15: K vizinhos mais próximos (K=3), *leave-one-out*.

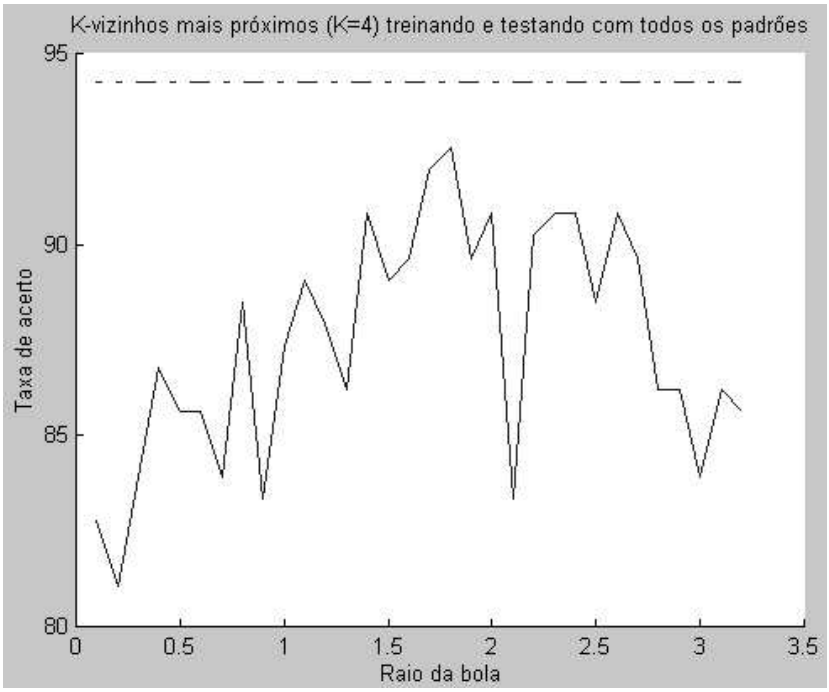

Figura 5.16: K vizinhos mais próximos (K=4), treinando e testando com todos os padrões disponíveis.

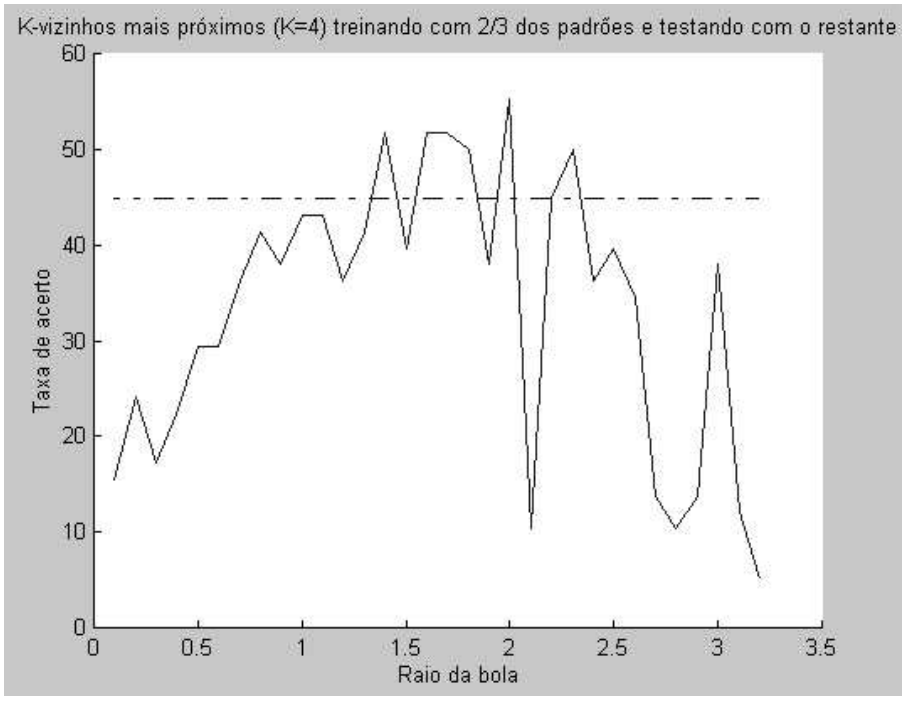

Figura 5.17: K vizinhos mais próximos (K=4), treinando com 2/3 dos padrões e testando com os 1/3 restantes.

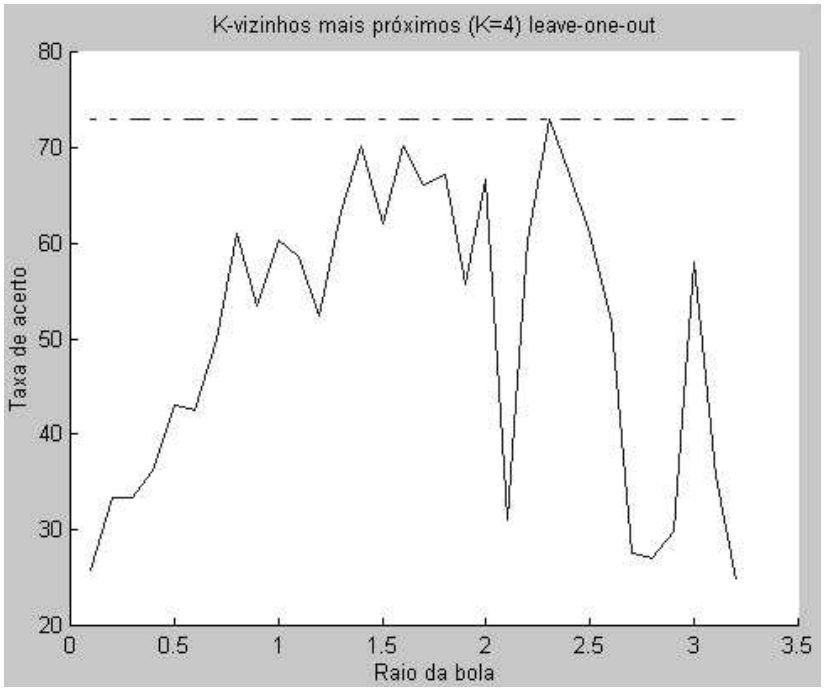

Figura 5.18: K vizinhos mais próximos (K=4), *leave-one-out*.

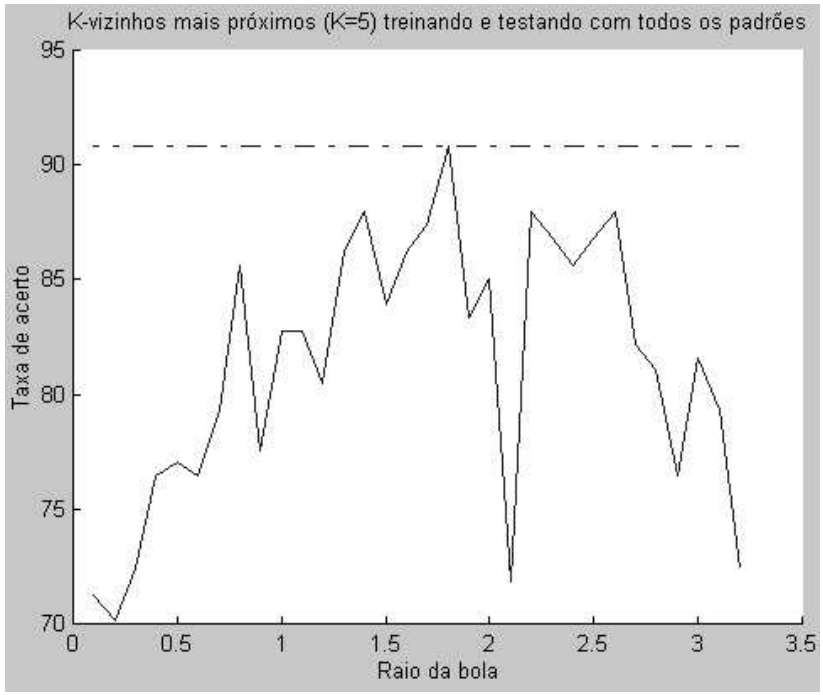

Figura 5.19: K vizinhos mais próximos (K=5), treinando e testando com todos os padrões disponíveis.

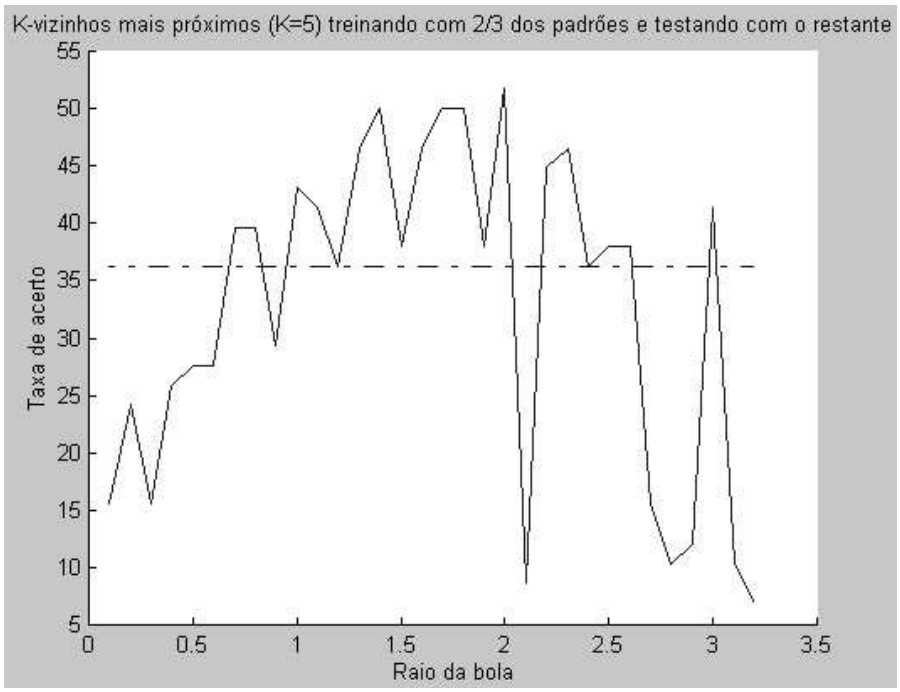

Figura 5.20: K vizinhos mais próximos (K=5), treinando com 2/3 dos padrões e testando com os 1/3 restantes.

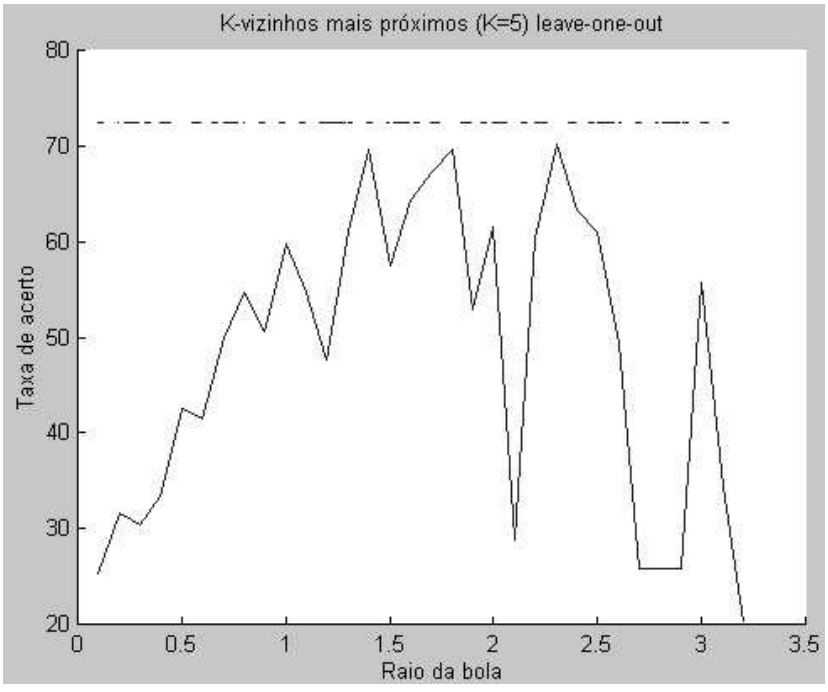

Figura 5.21: K vizinhos mais próximos (K=5), *leave-one-out*.

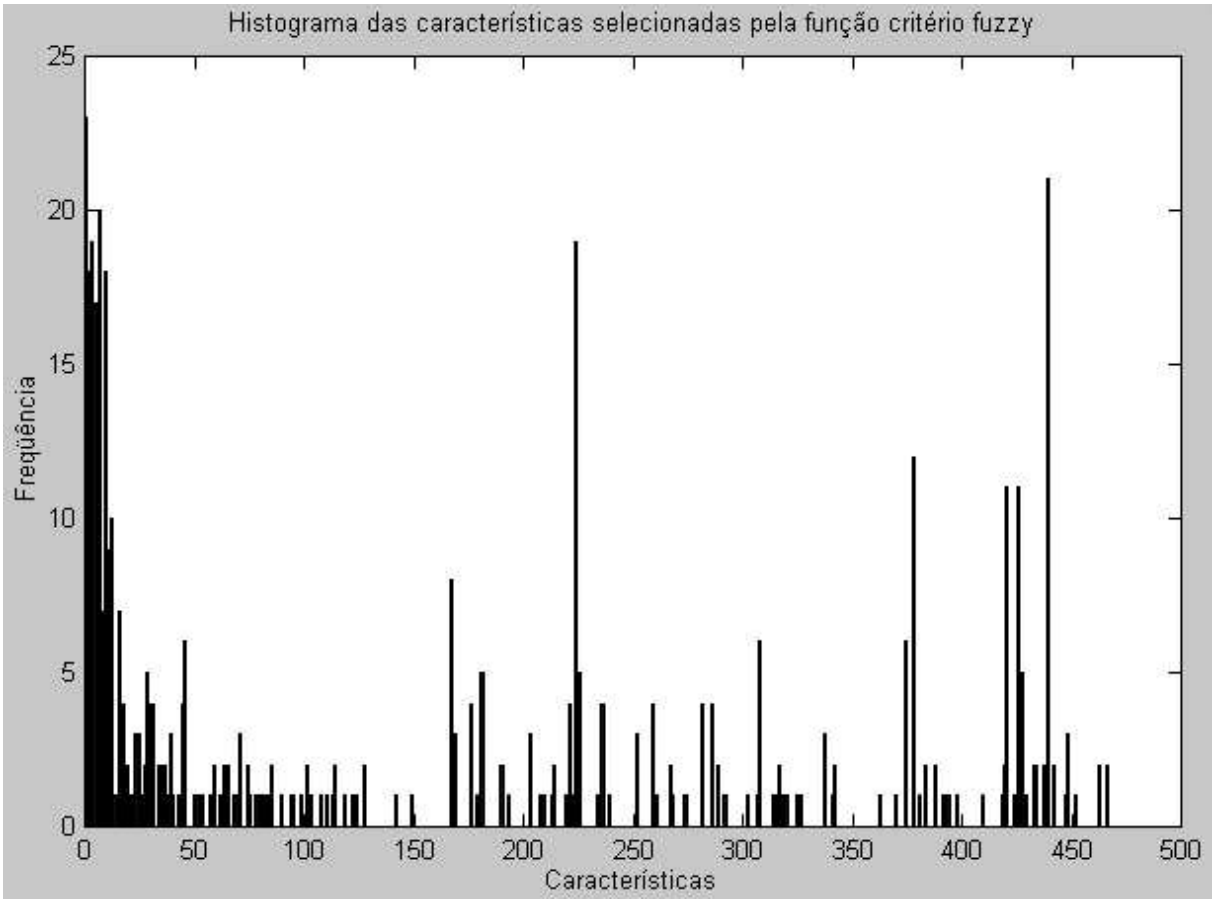

Figura 5.22: Histograma das características selecionadas em todos os experimentos realizados.

O gráfico da figura 5.22 é o histograma dos componentes selecionados após todos os 32 esperimentos. Analisando o histograma, pode-se verificar que, se for criado um sub-conjunto $\mathcal{Y}$ composto pelas características que foram selecionadas mais de 6 vezes, esse sub-conjunto teria as seguintes características:

$$\mathcal{Y} = \{x_1, x_2, x_4, x_5, x_7, x_8, x_{10}, x_{11}, x_{13}, x_{16}, x_{168}, x_{225}, x_{379}, x_{422}, x_{427}, x_{441})\} \qquad (5.10)$$

Isso mostra que, segundo o critério utilizado (máxima função critério nebulosa), um conjunto formado pelas 15 primeiras características não é o melhor conjunto de características.

5.3.2 Resultados utilizando outras funções critério

Para verificar a eficiência do nosso método, também realizamos testes de seleção de características utilizando outros critérios. Foram utilizados como funções critério as taxas de acerto do classificador KNN, com $K = 3$ para as seguintes estratégias:

- treinamento e teste com todo o conjunto;
- treinamento com 2/3 do conjunto e testes com o restante;
- *leave-one-out*.

O gráfico da figura 5.23 mostra os resultados obtidos com essas funções critério em comparação com a nossa função critério e com a seleção dos 15 primeiros eigeneyes. No caso de nossa função critério, os resultados mostrados nesse gráfico são aqueles obtidos com os melhores valores de τ. Cada coluna representa um critério utilizado (vide legenda lateral), sendo que, no eixo das abscissas, estão os métodos utilizados para testar os conjuntos obtidos a partir da seleção, enquanto no eixo das ordenadas estão as taxas de acerto em %. Parte desses resultados serão publicados em [Campos and Cesar-Jr, 2001].

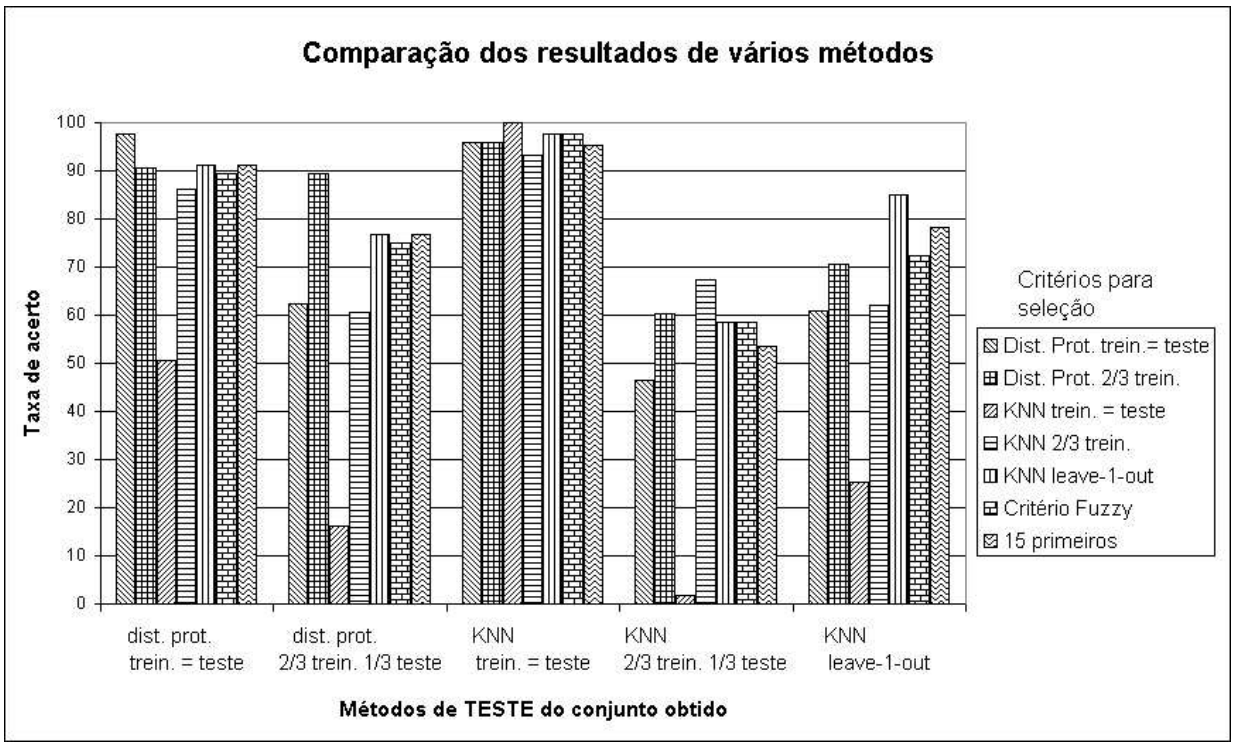

Figura 5.23: Resultados com funções critério baseadas no desempenho de classificadores em comparação com os resultados da função nebulosa e com a seleção dos 15 primeiros autovetores.

Obviamente, um conjunto que foi selecionado utilizando uma determinada estratégia de classificação proporciona resultados muito bons quando a mesma estratégia foi utilizada para avaliar o conjunto resultante. Os experimentos relacionados com o classificador KNN foram realizados utilizando $K = 3$.

A figura 5.24 mostra o histograma dos componentes selecionados utilizando as funções critério baseadas no desempenho de classificadores.

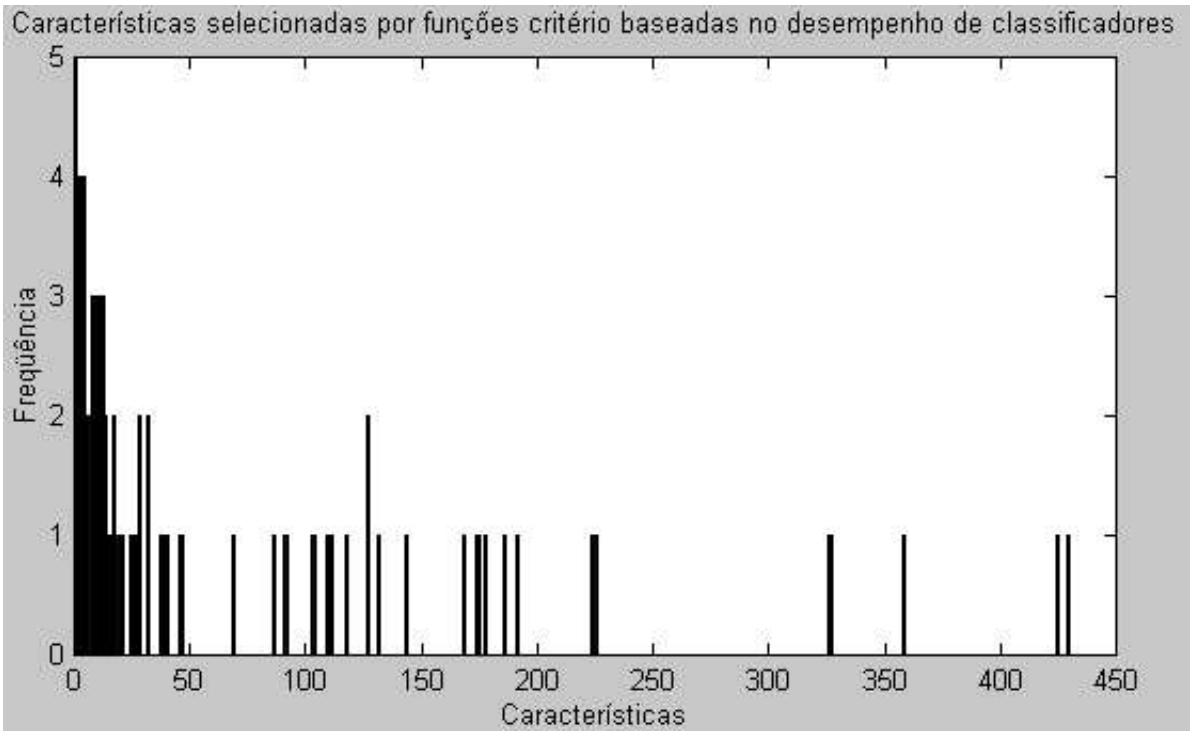

Figura 5.24: Histograma das características selecionadas através de funções critério baseadas no desempenho de classificadores.

Comparando-se os resultados mostrados na figura 5.23, nota-se que, dentre essas funções critério, a que proporcionou melhores resultados quando o conjunto selecionado foi avaliado por outras estratégias de classificação foi KNN *leave-one-out*. Em segundo lugar, ficaram os resultados obtidos com a seleção dos 15 primeiros autovetores. Os resultados obtidos com nossa função critério ficaram em terceiro lugar. Esse é um resultado bastante promissor, já que os resultados obtidos com o uso do desempenho de classificadores como função critério são "viciados" a esses classificadores, proporcionando os melhores resultados. Porém, nota-se que a simples utilização das 15 primeiros características proporcionou melhores resultados que a nossa função critério em mais de metade dos experimentos. Isso sugere que nossa função critério deve ser aprimorada. A seguir há algumas sugestões de medidas para aprimorar nossa função critério.

5.3.3 Sugestões para Aperfeiçoar a Função Critério

Uma forma de aperfeiçoar a função critério que nós propusemos é a utilização de mais suportes por classes. O uso de apenas um suporte por classe (como foi feito) não é uma boa maneira de se obter uma descrição completa das tipicalidades de um aglomeramento no espaço de características. Além disso, a função de fuzzyficação utilizada é muito simples

e não descreve com precisão a distribuição dos padrões dos conjuntos nebulosos, pois somente os padrões coincidentes com os suportes (protótipos) dos conjuntos possuem grau de pertinência máximo (igual a 1). Há métodos de fuzzyficação que fazem com que regiões (não pontuais) dos conjuntos tenham grau de pertinência máximo. Outro ponto que pode ser aprimorado no processo de fuzzyficação refere-se ao grau de pertinência dos padrões às diferentes classes. Neste trabalho, foi considerado que $\nu_{\omega_i}(\mathbf{x}_j) = 0, \forall \mathbf{x}_j \notin \omega_i$. Com isso, a distância nebulosa perde informação a respeito da distância entre os protótipos desses dois conjuntos (ω_i e $\omega_j : \mathbf{x}_j \in \omega_j$) quando a bola não for grande o bastante para englobar elementos das duas classes. Para eliminar esse problema, é necessário implementar uma nova função de fuzzyficação que considere o grau de pertinência de cada padrão a todos os conjuntos existentes no espaço de características.

5.4 Sistema para Reconhecimento a partir de Seqüências de Vídeo

5.4.1 Introdução e Descrição do Método

Esta seção contém a proposta de uma aplicação prática para reconhecimento de pessoas que relaciona os tópicos que foram estudados e desenvolvidos neste trabalho de mestrado. O fluxograma dessa proposta pode ser visto na figura 5.25, tendo sido documentado em [Campos et al., 2000b]. Não foram realizados testes integrando todo o sistema, porém é proposta a utilização dos métodos de redução de dimensionalidade e de classificação de imagens estáticas já implementados, os quais foram descritos nesta dissertação.

Basicamente, esse projeto foi criado a partir da união das duas idéias para redução de dimensionalidade discutidas nesta dissertação: o emprego de imagens menores (seção 5.1) e a utilização de métodos automáticos de seleção de características (seção 5.2.1).

O sistema de reconhecimento proposto deverá utilizar quatro recortes da imagem de entrada: para os dois olhos, o nariz e a boca. As tarefas de detecção e perseguição de pontos característicos da face, bem como a de normalização das imagens de olhos, nariz e boca, não fazem parte do escopo deste projeto, sendo importante ressalvar que essas tarefas foram realizadas através de um método baseado em Gabor Wavelet Networks [Feris and Cesar-Jr, 2001]. Esse método detecta e persegue os pontos característicos, determinando os parâmetros da transformação afim que leva uma imagem frontal a uma determinada escala e posição em que os pontos se encontram. Através desses parâmetros, pode-se realizar a inversa da transformação afim e obter imagens normalizadas. Esse processo de normalização é importante para reduzir as variações dos padrões introduzidas pelos movimentos da face, o que melhora o desempenho do sistema de reconhecimento.

As imagens utilizadas tanto para treinar quanto para testar o sistema de reconhecimento são imagens das regiões características normalizadas com relação à transformação afim. Para efetuar o treinamento, deve ser utilizada uma seqüência de vídeo por pessoa. O reconhecimento deve ser feito utilizando análise de componentes principais (PCA), com uma base para cada região da face. Dessa forma, é criada uma base para olhos esquerdos, outra para olhos direitos, uma para os narizes e outra para as bocas, obtendo-se as *eigenfeatures* (*eigenlefteyes, eigenrighteyes, eigennoses* e *eigenmouth*).

Após a obtenção de todas as *eigenfeatures*, essas deverão ser concatenadas de forma a criar um espaço de características formado por todas as *eigenfeatures*. Para reduzir a dimensionalidade desse espaço, é proposta a aplicação do algoritmo de seleção de características descrito na seção 5.3. A figura 5.26 esquematiza o método de geração do espaço de características descrito.

5.4.2 Motivação

A utilização de um algoritmo de seleção de características é motivada pelo fato de que tais métodos podem ser utilizados para efetuar fusão de multisensores [Somol et al., 2001, Jain and Zongker, 1997]. Considerando-se que as representações de cada região característica da face no espaço PCA podem ser vistas como dados provenientes de sensores diferentes (câmeras), surge a necessidade de reduzir a dimensionalidade de maneira a valorizar os sensores com maior poder de discriminação. Além disso, como podemos concluir da seção 5.2.1, a aplicação de algoritmos de seleção de características pode proporcionar melhora na taxa de acerto de classificadores.

Outro motivo é que, conforme mencionado na seção 3.2.2, a transformada PCA faz uma rotação no espaço de características de forma que o primeiro vetor da base fique na direção em que há mais variação entre os padrões, o segundo vetor na direção em que ocorre a segunda maior variação perpendicular ao primeiro, e assim por diante. Ou seja, a variação específica entre elementos de classe diferente não é otimizada.

Em [Jain et al., 2000], os autores mostram os resultados de uma abordagem de reconhecimento parecida com a abordagem proposta aqui e na seção anterior. Trata-se da aplicação de seleção de características usando a técnica de busca flutuante (SFSM) sobre as características obtidas a partir da transformada PCA sobre imagens de dígitos. O uso dos autovetores selecionados proporcionou resultados superiores ao uso dos primeiros autovetores.

Outro fator motivador para a aplicação de seleção de características sobre PCA está em um dos resultados obtidos em [Belhumeur et al., 1997], em que o desempenho de um sistema de reconhecimento de pessoas baseado em PCA foi melhorado com a eliminação dos três primeiros auto-vetores. Os autores de [Belhumeur et al., 1997] justificam que há algumas evidências de que esses auto-vetores são influenciados pelas mudanças de iluminação e não por variações inter-classes. Provavelmente, esse fato ocorre principalmente porque, em [Belhumeur et al., 1997], foram realizados testes com imagens apresentando grandes variações de iluminação, e os primeiros auto-vetores apontam para o sentido em que há maior variação dos dados. Esse resultado fornece evidências de que é possível obter resultados melhores aplicando um método de seleção de características sobre as *eigenfeatures* ao invés de utilizar simplesmente os primeiros auto-vetores.

Em [Moghaddam and Pentland, 1994] os autores declararam que não estava definida uma estratégia de realizar fusão ótima das informações obtidas das diferentes regiões da face. Tanto em [Moghaddam and Pentland, 1994] como em [Brunelli and Poggio, 1993], foi utilizado um classificador para cada região da face. Para combinar os resultados, foi utilizado um método de super-classificação.

Também conhecidos como "métodos de combinação", os **super-classificadores** são utilizados quanto se dispõe de vários resultados de classificação e deseja-se combinar os resultados para decidir a qual classe os dados pertencem. Tais esquemas podem ser apli-

cados quando são utilizados sistemas de multi-sensores e vários classificadores diferentes para classificar um determinado conjunto de dados, ou quando vários padrões separados formam um conjunto que pode pertencer à mesma classe. Um exemplo desse caso é o de seqüências de vídeo.

Quando são utilizados classificadores que informam qual o grau de certeza de se classificar um padrão a uma classe, como o casamento, podem ser utilizados métodos de classificação baseados em operações sobre os resultados de diversas classificações, como soma, média, mediana e máximo. Também pode ser aplicado um outro classificador que utiliza um vetor de características construído a partir dos resultados dos outros classificadores [Brunelli and Poggio, 1993]. Por outro lado, se os classificadores a serem combinados informam apenas qual a classe em que o padrão provavelmente pertence, deve ser utilizado, por exemplo, o esquema de votação. Maiores detalhes sobre super-classificadores encontram-se em [Jain et al., 2000].

No caso de [Brunelli and Poggio, 1993], a classificação das regiões foi feita usando *template matching* e o método de super-classificação utilizado foi a soma dos resultados (graus de similaridade dos templates de cada pessoa). Já em [Moghaddam and Pentland, 1994], a classificação das regiões foi feita por vizinho mais próximo no *eigenspace* e a super-classificação, através do esquema de votação.

5.4.3 Detalhamento

A estrutura proposta aqui (seleção de *eigenfeatures*) é uma forma de fundir os dados para a utilização de um único classificador para todas as regiões das imagens. Se, ao invés de fundir os dados dessa forma, fosse utilizado um classificador para cada região e um superclassificador para unir os resultados, certamente o processo de reconhecimento seria mais complexo e mais lento. Um super-classificador deve ser utilizado somente para combinar os resultados de classificação de cada quadro da seqüência de vídeo.

Para efetuar o reconhecimento de pessoas em seqüências de vídeo, primeiro os quadros devem ser representados no espaço de características criado a partir de *eigenfeature selection*. Inicialmente o espaço de características deve ser povoado pelos elementos de treinamento obtidos a partir de seqüências de vídeo em que as pessoas variam a pose e a expressão facial. Dessa forma, cada classe pode ter muitos elementos de treinamento. Posteriormente, para cada pessoa, deverá ser utilizada uma outra seqüência de vídeo para testar o sistema. Cada quadro das seqüências de teste é classificado individualmente através de um classificador de mínima distância ao protótipo ou de K-vizinhos mais próximos (descritos na seção 2.2). Conforme dito anteriormente, um super-classificador é utilizado para decidir o resultado da classificação a partir dos resultados obtidos pelos quadros individuais da seqüência. Para efetuar essa tarefa, foi proposta a utilização do esquema de votação.

5.4.4 Outras aplicações

Além do reconhecimento de pessoas, uma outra possível aplicação da análise dos resultados de seleção de características é a determinação da importância de cada região característica da face nos processos de classificação ou reconhecimento de expressões faciais. Em [Brunelli and Poggio, 1993], foram realizadas análises experimentais as quais mostraram que a ordem decrescente da taxa de acerto das regiões características, quando essas são tomadas individualmente para reconhecer pessoas, é a seguinte:

1. olhos;
2. nariz;
3. boca;
4. toda a face.

Os autores mencionaram que esse resultado é consistente com a habilidade humana para reconhecer pessoas.

Com a aplicação do sistema descrito nesta seção, pode ser feita uma análise do número de autovetores selecionados para cada região da face. Essa análise pode fornecer resultados mais completos a respeito da importância de cada região da face para efetuar diferentes tarefas, como reconhecimento de pessoas e reconhecimento de expressões faciais.

5.4.5 Discussão

Conforme mencionado no capítulo 4, o maior problema enfrentado no projeto de sistemas de reconhecimento de faces a partir de seqüências de vídeo é a dificuldade de avaliar-se tais sistemas. O principal motivo é a ausência de bases públicas de seqüências de imagens para que possam ser utilizadas na realização de testes e estabelecimento de uma *benchmark* internacional. Uma base de seqüências ideal deve ser constituída por vídeos de muitas pessoas diferentes, e sem problemas de auto-oclusão e com iluminação razoavelmente controlada.

Como este trabalho volta-se principalmente para seleção de características, a implementação do sistema proposto nesta seção fica como uma proposta de aplicação do nosso método a um problema prático. Com a futura disponibilização ou mesmo criação de bases de seqüências de vídeo, esta proposta poderá ser avaliada.

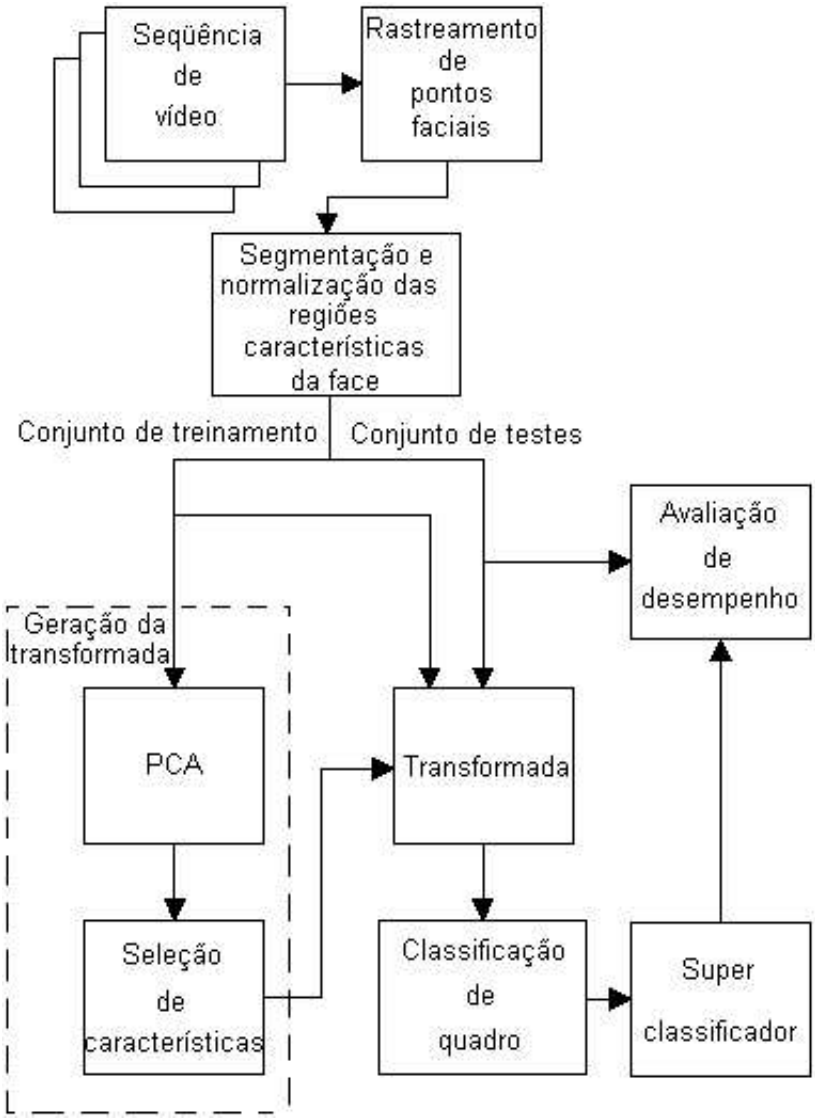

Figura 5.25: Esquema do projeto de reconhecimento a partir de seqüências de vídeo.

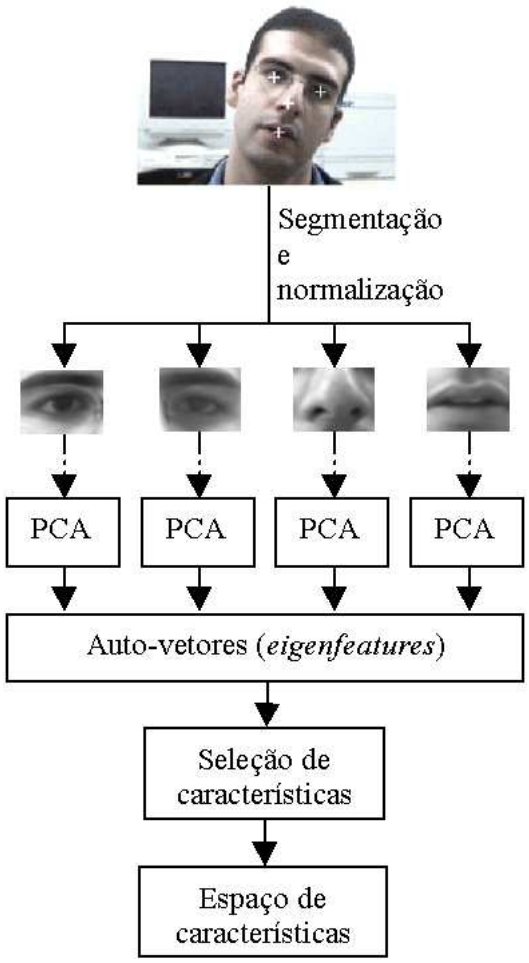

Figura 5.26: Geração do espaço de características.

Capítulo 6

Conclusões

A pesquisa em reconhecimento de faces ainda continua com muitos desafios em aberto. Um deles é o de reconhecimento a partir de seqüências de vídeo com pessoas agindo naturalmente em ambientes sem controle de iluminação. Um dos problemas mais importantes para abordar-se esse desafio é o de redução de dimensionalidade.

Neste trabalho, foram estudadas várias técnicas de reconhecimento de padrões que se associam a reconhecimento de faces. O enfoque foi dado a métodos de redução de dimensionalidade, principalmente em se tratando de seleção de características.

Esses estudos culminaram na realização de testes práticos com algumas técnicas de redução de dimensionalidade, bem como na elaboração de novas estratégias para efetuar seleção de características. Também foi proposto (juntamente com outro estudante do grupo de pesquisa) um esquema de reconhecimento de pessoas a partir de seqüências de vídeo utilizando somente regiões características das faces (olhos nariz e boca). Mas como o cerne desta dissertação é o estudo de técnicas de seleção de características e suas aplicações em reconhecimento de faces, a realização de testes com seqüências de vídeo ficou como trabalho futuro. Porém, a implementação de técnicas eficientes que possibilitam a realização de classificação em seqüências de vídeo já foi efetuada.

Pode-se notar que foram efetuadas várias contribuições pontuais no decorrer do desenvolvimento desse projeto de pesquisa. Deve-se ressaltar que a principal contribuição foi a elaboração de uma nova função critério para seleção de características com um método eficiente de busca. Essa função critério se baseia em uma distância nebulosa que foi proposta recentemente. O método de busca utilizado também é bastante recente. Os resultados experimentais obtidos mostraram que a abordagem proposta tem bom potencial

para algumas aplicações.

Devido à complexidade do problema de reconhecimento de pessoas a partir de seqüências de vídeo perante a atual tecnologia, é de se esperar que ainda haja muito trabalho a ser feito. No desenvolvimento desta dissertação, alguns pequenos passos para a elaboração de métodos de redução de dimensionalidade (principalmente seleção de características) foram dados, mas restaram várias tarefas a serem desenvolvidas futuramente. Dentre elas, de imediato podemos citar as seguintes:

- utilizar mais protótipos no processo de fuzzyficação dos conjuntos para o método de seleção de características utilizando a distância nebulosa baseada em tolerância (seção 3.4);
- para o mesmo problema, realizar testes com outras funções de fuzzyficação;
- localizar (ou criar) uma base de seqüências de imagens de pessoas em movimento para efetuar testes com os algoritmos envolvidos no projeto ilustrado na seção 5.4;
- testar diversos algoritmos superclassificadores nesse mesmo projeto;
- investigar formas de extrair informações obtidas exclusivamente a partir do movimento de faces;
- utilizar os novos métodos de busca para seleção de características propostos pelo grupo responsável pela publicação dos métodos SFSM e ASFSM [Kittler et al., 2001];
- fazer uma comparação de desempenho entre LDA e PCA com seleção de características.

Essas são apenas algumas propostas, mas muitas outras podem surgir após a leitura deste texto. Recomendamos que as novas idéias implementadas sejam comparadas com os outros métodos já existentes utilizando as mesmas bases de dados. Pode-se notar que os métodos propostos aqui não foram comparados com todos seus similares existentes, o que abre mais uma possibilidade de continuação deste trabalho.

Conforme mencionado anteriormente, foi proposto um esquema de reconhecimento de pessoas a partir de seqüências de vídeo. Dentro desse contexto, este trabalho concentra-se na parte de seleção de características. As outras partes daquele esquema podem ser desenvolvidas e integradas futuramente por outros pesquisadores.

Apêndice A

Notação Utilizada

$\mathbf{x}$, $\mathbf{y}$, $\mathbf{z}$	Vetores de observação ou padrões. Geralmente $\mathbf{x}$ e $\mathbf{y}$ foram usados para representar padrões em espaços de características diferentes
x, y, z, f	Características; variáveis aleatórias (na seção 3.3); sinais (na seção 3.2.1)
b	Escalar; amostragem da variável aleatória de um vetor aleatório $\mathbf{x}$
Θ	Conjunto de todas as tuplas de características (seção 3.3)
$\overline{\mathbf{x}}$	Uma aproximação de $\mathbf{x}$
$\overline{\mathbf{y}}$	A representação de um padrão $\mathbf{x}$ após extração de características com redução de dimensionalidade
χ, Y	Conjuntos ou seqüências de observações em diferentes espaços de características
KNN	Regra de classificacão por K vizinhos mais próximos
$\mathcal{I}$	Espaço de imagens; espaço de características

	de dimensionalidade elevada
F	Espaço de faces; espaço de características após extração de características.
$\mathcal{Y}$, $\mathcal{X}$, $\mathcal{Z}$, $\mathcal{T}$, $\mathcal{U}$	Conjuntos (ou tuplas) de características (feature sets) - seção 3.3
N	Dimensionalidade total do espaço de características F
m	Dimensionalidade de um sub-espaço de F ($m \leq N$)
ω	Uma classe de padrões (cluster)
Ω	Conjunto de todas as classes de padrões
c	Número de classes
T	Conjunto de treinamento
$\|T\|$	Número de exemplos no conjunto de treinamento; cardinalidade de T
X	Conjunto de todos os exemplos de treinamento (seção 3.2.2) (vetores de observação), e de testes (seção 2.2)
D	Fronteira de decisão
S_i	Região (partição) do espaço de características correspondente à classe ω_i
S_w	Matriz de espalhamento intra-classes
S_b	Matriz de espalhamento inter-classes
$\mathcal{S}_k(x_j)$	Significância da característica x_j (seção 3.3)
$B_{\mathbf{x}}$	Uma bola no espaço de características centrado no padrão $\mathbf{x}$
$\Upsilon(\cdot)$	Classificador
$\mathcal{H}(\cdot)$	Função de extração de características

H	Matriz mudança de base
i, j, r, s, n, p, q, l, o	Índices
K	O número de vizinhos verificados pelo classificador K-NN
λ	Autovalor
σ	Desvio padrão
μ	Vetor médio
Σ	Matriz de covariância
u	Autovetor
Z^t	Transposta da matriz Z
I	Matriz identidade
$det(Z)$	Determinante da matriz Z
$\mathbb{R}^N$	Espaço N-dimensional de Reais
$f(\cdot)$	Função (por exemplo, função critério na seção 5.3.1)
$d(\cdot)$	Função de distância
$h(\cdot)$	Função de similaridade
$d_{\mathcal{E}}(\cdot)$	Distância Euclidiana
$d_{\mathcal{M}}(\cdot)$	Distância de Mahalanobis
$P(\cdot)$ $L(\cdot)$	Probabilidade
$E(\cdot)$	Esperança
$L(\cdot)$	Semelhança
$R(\cdot)$	Risco

$p(\cdot)$	Função densidade de probabilidade
P_i	Probabilidade a priori da classe ω_i
$P(\omega_j\|\mathbf{x})$	Probabilidade a posteriori da classe ω_i
$a_i(\mathbf{x})$	Probabilidade de acerto ao classificar-se um dado $\mathbf{x}$ em ω_i
A	Taxa de acerto de um classificador
E	Taxa de erro de um classificador
$\mathcal{N}(\mu, \Sigma)$	Função Gaussiana (distribuição normal)
$\xi(\cdot)$	Função de erro
$C(\cdot)$	Função de perda; custo
$e(\mathbf{x})$	Probabilidade de erro de classificação do padrão $\mathbf{x}$
$J(\cdot)$	Função critério
$\|\|\mathbf{x}\|\|$	Norma Euclidiana de $\mathbf{x}$
$\nu_{\omega_i}(\mathbf{x})$	Função de pertinência de $\mathbf{x}$ à classe ω_i
C	Conjunto de todos os conjuntos nebulosos definidos em F
$p_i^{\omega_j}$	O i-ésimo suporte do conjunto nebuloso ω_j
$exp(.)$	Exponencial neperiano ($e^{(\cdot)}$)
t	Variável de tempo
f	Variável de freqüência
$F(\cdot)$	Transformada de Fourier
$\mathcal{T}_0$	Período de uma função $\mathbf{x}(t)$ (na seção 3.2.1), e tupla de características (na seção 3.3)

SFSM	Métodos de busca seqüencial flutuante (Sequential Floating Search Mehtods)
SFFS	Busca seqüencial flutuante para frente (Sequential Floating Forward Search)
SFBS	Busca seqüencial flutuante adaptativa para trás (Sequential Floating Backward Search)
ASFSM	Métodos de busca seqüencial flutuante adaptativa (Adaptive Sequential Floating Search Mehtods)
ASFFS	Busca seqüencial flutuante adaptativa para trás (Sequential Floating Backward Search)
ASFBS	Busca seqüencial flutuante para trás (Sequential Floating Backward Search)
h	Altura de uma imagem (em *pixels*)
w	Largura de uma imagem (em *pixels*)

Referências Bibliográficas

[Backer, 1995] Backer, E. (1995). *Computer-Associated Reasoning in Cluster Analysis*. Prentice Hall.

[Barrera et al., 2000] Barrera, J., Terada, R., Jr, R. H., and Hirata, N. S. T. (2000). Automatic programming of morphological machines by pac learning. *Fundamenta Informaticae*, 41(1-2):229–258.

[Belhumeur et al., 1997] Belhumeur, P. N., Hespanha, J. P., and Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):711–720.

[Bichsel, 1995] Bichsel, M., editor (1995). *1st International Conference on Face and Gesture Recognition*. MultiMedia Laboratory Department of Computer Science university of Zurich, Zurich, Switzerland. Proceedings.

[Bloch, 1999] Bloch, I. (1999). On fuzzy distances and their use in image processing under imprecision. *Pattern Recognition*, 11(32):1873–1895.

[Bonventi-Jr. and Costa, 2000] Bonventi-Jr., W. and Costa, A. H. R. (2000). Comparação entre métodos de definição de conjuntos nebulosos de cores para classificação de pixels. In *1st. Workshop on Artificial Intelligence and Computer Vision (parallel to IBERAMIA'2000-SBIA'2000)*, Atibaia - Brasil. IME - USP.

[Brunelli and Falavigna, 1995] Brunelli, R. and Falavigna, D. (1995). Person identification using multiple cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17:955–966.

[Brunelli et al., 1995] Brunelli, R., Falavigna, D., Poggio, T., and Stringa, L. (1995). Automatic person recognition by acoustic and geometric features. *MVA*, 8:317–325.

[Brunelli and Poggio, 1993] Brunelli, R. and Poggio, T. (1993). Face recognition: Features versus templates. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 15(10):1042–1052.

[Bruno et al., 1998] Bruno, O. M., Cesar-Jr., R. M., Consularo, L. A., and da F. Costa, L. (1998). Automatic feature selection for biological shape classification in synergos. In *11th SIBGRAPI*, pages 363–370, Rio de Janeiro - RJ. IEEE Computer Society Press.

[Burton et al., 1999] Burton, A. M., Wilson, S., and Cowan, M. (1999). Face recognition in poor quality video: Evidence from security surveillance. *Psychological Science*.

[Callioli et al., 1998] Callioli, C. A., Rodriguese, H. H., and Costa, R. C. F. (1998). *Álgebra Linear e Aplicações*. Editora Atual, SP, sexta edition.

[Campos et al., 2001] Campos, T. E., Bloch, I., and Cesar-Jr, R. M. (2001). Feature selection based on fuzzy distances between clusters: First results on simulated data. In *Lecture Notes in Computer Science*, Rio de Janeiro, Brasil. Springer-Verlag Press.

[Campos and Cesar-Jr, 2001] Campos, T. E. and Cesar-Jr, R. M. (2001). Eigeneyes selection using the performance of a classifier for fast face recognition. In *53a. Reunião Anual da SBPC*, Salvador - BA, Brasil.

[Campos et al., 2000a] Campos, T. E., Feris, R. S., and Cesar-Jr, R. M. (2000a). Discriminação de faces × não-faces usando descritores de fourier. In *52a. Reunião Anual da SBPC*, Brasilia - DF, Brasil.

[Campos et al., 2000b] Campos, T. E., Feris, R. S., and Cesar-Jr, R. M. (2000b). A framework for face recognition from video sequences using gwn and eigenfeature selection. In *1st. Workshop on Artificial Intelligence and Computer Vision (parallel to IBERAMIA'2000-SBIA'2000)*, pages 141–145, Atibaia - SP, Brasil. IME - USP.

[Campos et al., 2000c] Campos, T. E., Feris, R. S., and Cesar-Jr, R. M. (2000c). Improved face × non-face discrimination using fourier descriptors through feature selection. In *13th SIBGRAPI*, pages 28–35. IEEE Computer Society Press.

[Campos et al., 2000d] Campos, T. E., Feris, R. S., and Jr., R. M. C. (2000d). Eigenfaces versus eigeneyes: First steps towards performance assessment of representations for face recognition. In *Lecture Notes in Artificial Intelligence*, volume 1973, pages 197–206, Acapulco, Mexico. Springer-Verlag Press.

[Cascia and Sclaroff, 1999] Cascia, M. L. and Sclaroff, S. (1999). Fast, reliable head tracking under varying illumination. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[Castleman, 1996] Castleman, K. R. (1996). *Digital Image Processing*. Englewood Cliffs, NJ.

[Cesar-Jr, 1997] Cesar-Jr, R. M. (1997). *Análise Multi-Escala de Formas Bidimensionais*. PhD thesis, IFSC - USP, São Carlos.

[Chellappa et al., 1995] Chellappa, R., Wilson, C. L., and Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, 83(5):703–740.

[Cormen et al., 1990] Cormen, T. H., Leiserson, C. E., and Rivest, R. L. (1990). *Introduction to Algorithms*. The MIT Press, McGraw-Hill Book Company.

[Cox et al., 1995] Cox, I. J., Ghosn, J., and Yianilos, P. N. (1995). Feature-based recognition using mixture-distance. Technical report, NEC Research Institute.

[Crowley, 2000] Crowley, J. L., editor (2000). *4th IEEE International Conference on Face and Gesture Recognition*. Grenoble, France. Proceedings.

[de Berg et al., 2000] de Berg, M., van Kreveld, M., Overmars, M., and Schwarzkopf, O. (2000). *Computational Geometry, Algorithms and Applications*. Springer Verlag, 2nd edition.

[Dubois et al., 1997] Dubois, D., Prade, H., and Yager, R. R., editors (1997). *Fuzzy Information Engineering*. Wiley Computer Publishing, USA.

[Duda and Hart, 1973] Duda, R. O. and Hart, P. E. (1973). *Pattern Classification and Scene Analysis*. Wiley-Interscience, USA, 1st edition.

[Essa, 1996] Essa, I., editor (1996). *2nd IEEE International Conference on Face and Gesture Recognition*. Killington, USA. Proceedings.

[Farines et al., 2000] Farines, J. M., da S. Fraga, J., and de Oliveira, R. S. (2000). *Sistemas de Tempo Real*. IME/USP, São Paulo-SP, Brasil. XII Escola Nacional de Computação.

[Feris, 2001] Feris, R. S. (2001). Rastreamento eficiente de faces em um subespaço wavelet. Master's thesis, Universidade de São Paulo, Instituto de Matemática e Estatística.

[Feris et al., 2000] Feris, R. S., Campos, T. E., and Cesar-Jr, R. M. (2000). Detection and tracking of facial features in video sequences. In *Lecture Notes in Artificial Intelligence*, volume 1973, pages 129–137, Acapulco, Mexico. Springer-Verlag Press.

[Feris and Cesar-Jr, 2001] Feris, R. S. and Cesar-Jr, R. M. (2001). Detection and tracking of facial landmarks using gabor wavelet networks. In *Lecture Notes in Computer Science*, Rio de Janeiro, Brasil. Springer-Verlag Press.

[Fisher, 1938] Fisher, R. A. (1938). The statistical utilization of multiple measurements. In *Annals of Egenics*, volume 8, pages 376–386.

[Gong et al., 2000] Gong, S., McKenna, S., and Psarrou, A. (2000). *Dynamic Vision: From Images to Face Recognition*. Imperial College Press, UK.

[Gonzalez and Woods, 1992] Gonzalez, R. C. and Woods, R. E. (1992). *Digital Image Processing*. Addison-Wesley Publishing Compuany.

[Jain et al., 2000] Jain, A. K., Duin, R. P. W., and Mao, J. (2000). Statistical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37.

[Jain et al., 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: a review. *ACM Computing Surveys*, 31(3):264–323.

[Jain and Zongker, 1997] Jain, A. K. and Zongker, D. (1997). Feature-selection: Evaluation, application, and small sample performance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(2):152–157.

[Kasturi, 1997] Kasturi, R., editor (1997). *IEEE Transactions on Pattern Analysis and Machine Intelligence*. Theme Section of the Journal - Face and Gesture Recognition.

[Kennedy and Neville, 1986] Kennedy, J. B. and Neville, A. M. (1986). *Basic Statistical Methods for Engineers and Scientists*. Harper and Row, Publishers, third edition.

[Kirby and Sirovich, 1990] Kirby, M. and Sirovich, L. (1990). Application of the karhunen-loève procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):103–108.

[Kittler et al., 2001] Kittler, J., Somol, P., and Pudil, P. (2001). Advances in statistical feature selection. In *Lecture Notes in Computer Science*, Rio de Janeiro, Brasil. Springer-Verlag Press.

[Kohn, 1998] Kohn, A. F. (1998). Reconhecimento de padrões, uma abordagem estatística. Apostila, EP - Universidade de São Paulo.

[Kondo and Yan, 1999] Kondo, T. and Yan, H. (1999). Automatic human face detection and recognition under non-uniform illumination. *Pattern Recognition*, 32(10):1707–1718.

[Krüger and Sommer, 1999] Krüger, V. and Sommer, G. (1999). Affine real-time face tracking using a wavelet network. In *ICCV'99 Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems*, Corfu, Greece.

[Krüger and Sommer, 2000] Krüger, V. and Sommer, G. (2000). Gabor wavelet networks for object representation. In *22. DAGM Symposium*, Kiel, Germany.

[Lades et al., 1993] Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *Transactions on Computers*, 42(3):300–311.

[Lawrence et al., 1996] Lawrence, S., Giles, C. L., Tsoi, A. C., and Back, A. D. (1996). Face recognition: A hybrid neural network approach. Technical report, I. A. C. S., U. of Maryland.

[Li et al., 2000] Li, Y., Gong, S., and Liddell, H. (2000). Exploiting the dynamics of faces in spatial-temporal context. In *6th International Conference on Control, Automation, Robotics and Vision (ICARCV2000)*, Singapore.

[Lima, 1970] Lima, E. L. (1970). *Elementos de Topologia Geral*. Ao Livro Técnico S. A.

[Lowen and Peeters, 1997] Lowen, R. and Peeters, W. (1997). On various classes of semi-pseudometrics used in pattern recognition. In *7th IFSA World Congress*, volume I, pages 232–237, Prague.

[Lowen and Peeters, 1998] Lowen, R. and Peeters, W. (1998). Distances between fuzzy sets representing gray level images. *Fuzzy Sets and Systems*, 99(2):143–153.

[Mao et al., 1994] Mao, J., Mohiuddin, K., and Jain, A. K. (1994). Parsimonious network design and feature selection through node pruning. In *Proc. 12th ICRP*, pages 622–624, Jerusalem.

[Martinez and Kak, 2001] Martinez, A. M. and Kak, A. C. (2001). Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233.

[McKenna et al., 1997] McKenna, S., Gong, S., and Raja, Y. (1997). Face recognition in dynamic scenes. In *British Machine Vision Conference (BMVC)*. Essex.

[Moghaddam and Pentland, 1994] Moghaddam, B. and Pentland, A. (1994). Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the identification and Inspection of Humans*, volume 2277. SPIE.

[Moghaddam and Pentland, 1997] Moghaddam, B. and Pentland, A. P. (1997). Probabilistic visual learning for object representation. *IEEE Transactions on Patterns Analysis and Machine Intelligence*, 19(7):696–710.

[Moghaddam et al., 1998] Moghaddam, B., Wahid, W., and Pentland, A. (1998). Beyond eigenfaces: Probabilistic matching for face recognition. In *3rd IEEE International Conference on Automatic Face and Gesgure Recognition*, Nara, Japan.

[Morimoto et al., 1996] Morimoto, C. H., Yacoob, Y., and Davis, L. (1996). Recognition of head gestures using hidden markov models. In *ICPR*, Vienna, Austria.

[Narendra and Fukunaga, 1977] Narendra, P. M. and Fukunaga, K. (1977). A branch and bound algorithm for feature subset selection. *IEEE Trans. Computers*, 26(9):917–922.

[Pankanti et al., 2000] Pankanti, S., Bolle, R. M., and Jain, A. (2000). Biometrics: The future of identification. *Computer*, pages 46–49.

[Pentland, 2000] Pentland, A. (2000). Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):107–119.

[Perlovsky, 1998] Perlovsky, L. I. (1998). Conundrum of combinatorial complexity. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(6):666–670.

[Phillips et al., 1998] Phillips, P., Wechsler, H., Huang, J., and Rauss, P. (1998). The feret database and evaluation procedure for face recognition algorithms. *Image and Vision Computing*, 16(5):295–306.

[Pudil et al., 1994] Pudil, P., Novovicová, J., and Kittler, J. (1994). Floating search methods in feature selection. *Pattern Recognition Letters*, 15:1119–1125.

[Ratha et al., 2001] Ratha, N. K., Senior, A., and Bolle, R. (2001). Automated biometrics. In *Lecture Notes in Computer Science*, Rio de Janeiro, Brasil. Springer-Verlag Press.

[Romdhani, 1996] Romdhani, S. (1996). Face recognition using principal component analysis. Master's thesis, Department of Electronics and Electrical Engineering, University of Glasgow, UK.

[Rowley et al., 1998] Rowley, H. A., Baluja, S., and Kanade, T. (1998). Neural network-based face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):23–38.

[Siedleki and Sklansky, 1989] Siedleki, W. and Sklansky, J. (1989). A note on genetic algorithms for large-scale feature selection. *Pattern Recognition Letters*, 10:335–347.

[Silva et al., 1995] Silva, L., Aizawa, K., and Hatori, M. (1995). Detection and tracking of facial features. In *SPIE Visual Communications and Image Processing'95 (VCIP'95)*, volume 2501, pages 1161–1172, Taipei, Taiwan.

[Somol and Pudil, 2000] Somol, P. and Pudil, P. (2000). Oscillating search algorithms for feature selection. In *Proceedings of the 15th International Conference on Pattern Recognition*, pages 406–409, Los Alamitos. IEEE Computer Society.

[Somol et al., 2001] Somol, P., Pudil, P., and Grim, J. (2001). Branch and bound algorithm with partial prediction for use with recursive and non-recursive criterion forms. In *Lecture Notes in Computer Science*, Rio de Janeiro, Brasil. Springer-Verlag Press.

[Somol et al., 2000] Somol, P., Pudil, P., Novovicová, J., Ferri, F. J., and Kittler, J. (2000). Fast branch and bound algorithm in feature selection. In *Proceedings of the SCI 2000 Conference*, volume IIV, pages 646–651, Orlando, Florida.

[Somol et al., 1999] Somol, P., Pudil, P., Novovicová, J., and Paclík, P. (1999). Adaptive floating search methods in feature selection. *Pattern Recognition Letters*, 20:1157–1163.

[Sung and Poggio, 1998] Sung, K. K. and Poggio, T. (1998). Example-based learning for view-based human face detection. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(1):39–55.

[Theodoridis and Koutroumbas, 1999] Theodoridis, S. and Koutroumbas, K. (1999). *Pattern Recognition*. Academic Press, USA, 1st edition.

[Turk, 1998] Turk, M., editor (1998). *1998 Workshop on Perceptual User Interfaces*. Microsoft - http://research.microsoft.com/PUIWorkshop, San Francisco, USA. Proceedings.

[Turk and Pentland, 1991] Turk, M. A. and Pentland, A. P. (1991). Face recognition using eigenfaces. In *Proc. of the IEEE Computer Society Conferece*.

[Valentin et al., 1996] Valentin, D., Abdi, H., and O'Toogle, A. J. (1996). Principal component and neural networks analyses of face images: Explorations into the nature of information available for classifying faces by sex. In *Progress in Mathematical Psycology*. Hillsdale: Erlbaum.

[Watanabe, 1985] Watanabe, S. (1985). *Pattern Recognition: Human and Mecanical*. New York: Wiley.

[Wiskott et al., 1995] Wiskott, L., Fellous, J. M., Krüger, N., and von der Malsburg, C. (1995). Face recognition and gender determination. In *International Conference on Automatic Face and Gesture Recognition*, pages 92–97, University of Zurich. MultiMedia Laboratory.

[Wiskott et al., 1997] Wiskott, L., Fellous, J. M., Krüger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):775–779.

[Wu et al., 1999] Wu, H., Chen, Q., and Yachida, M. (1999). Face detection from color images using a fuzzy pattern matching method. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(6):557–563.

[Yachida, 1998] Yachida, M., editor (1998). *3rd IEEE International Conference on Face and Gesture Recognition*. Nara, Japan. Proceedings.

[Yacoob et al., 1995] Yacoob, Y., Lam, H. M., and Davis, L. S. (1995). Recognizing faces showing expressions. In *International Conference on Automatic Face and Gesture Recognition*, Zurich, Swistzerland.

[Yang et al., 1997] Yang, J., Lu, W., and Waibel, A. (1997). Skin color modeling and adaptation. Technical report, CMU-CS-97-146.

[Zhao et al., 1999] Zhao, W. Y., Chellappa, R., and Phillips, P. J. (1999). Subspace linear discriminant analysis for face recognition. *IEEE Transactions on Image Processing*.

Índice Remissivo